Language, Form, and Logic

Language, Form, and Logic

In Pursuit of Natural Logic's Holy Grail

PETER LUDLOW AND SAŠO ŽIVANOVIĆ

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford OX2 6DP
United Kingdom

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and in certain other countries

First Edition published in 2022
Impression: 1

Published in the United States of America by Oxford University Press
198 Madison Avenue, New York, NY 10016, United States of America

British Library Cataloguing in Publication Data
Data available

Library of Congress Control Number: 2021948637

ISBN 978–0–19–959153–4

Printed and bound in Great Britain by
Clays Ltd, Elcograf S.p.A.

In memory of Víctor Sánchez Valencia

The parallax of time helps us to the true position of a conception, as the parallax of space helps us to that of a star.

—T. H. Huxley

Contents

Acknowledgements

We have been working together on this project for quite some time (twenty years or so) and in that time we have been helped by numerous friends and colleagues. Even before our collaboration began, however, we were assisted on these issues by numerous people. These include Jon Barwise, George Boolos, Noam Chomsky, Hartry Field, Patrick Grim, James Higginbotham, David Law, Richard Larson, Gary Mar, Fernando Pereira, Paul Pietroski, Victor Sánchez Valencia, Barry Schein, Suart Shieber, and Michal Starke.

We are also indebted to Johan van Benthem, Josh Dever, Lawrence Moss, Juhani Yli-Vakkuri, and Amanda Saksida for comments on the penultimate draft of this manuscript, and to Peter Momtchiloff for editorial guidance.

Some of the people mentioned above have left us: Jon Barwise, George Boolos, James Higginbotham, and Victor Sánchez. The loss of these thinkers has weighed on us and it often feels like we confront a void when we can no longer turn to them for advice and criticism. This is particularly true in the case of Victor Sánchez, whose papers inspired the core ideas of our work, and who provided us a great deal of personal encouragement for this project. He is this book's *sine qua non,* and it is therefore dedicated to his memory.

List of Figures and Tables

Figures

Tables

1
Introduction

When we use the expression 'Natural Logic', we intend a conception of logic wherein logic (or at least some part of it) and logical relations are expressed in terms of natural language forms or structures. This means that while formal logic gives a theory of inferential relations for artificial languages like the propositional calculus or the predicate calculus, Natural Logic maintains that such inferences (or some of them) can be accounted for in natural language, by appeal to the forms of natural language expressions. In short, Natural Logic articulates relations between natural language forms, rather than the forms of an artificial language.[1]

For the past 120 years, the focus of most work in logic has principally been on the development of logical calculi with little intended concern for the formal structure of natural language. That is, the syntactic forms postulated for purposes of logical reasoning (e.g. the logical forms of the propositional and predicate calculi) were not intended to resemble the syntactic forms postulated for natural language sentences. This apparent mismatch between logical form and grammatical form is ordinarily taken to be an unsurprising consequence of the fact that, since Frege and Russell, one of the central goals of logic has been to help clarify (if not establish) the foundations of mathematics and to develop tools that instantiate norms for human reasoning. These goals have been thought to be orthogonal to investigations into the structure of natural language. Furthermore, it has also been held that natural language forms are rather poor media in which to couch logical reasoning, since natural language is ambiguous, vague, etc. Indeed, it is sometimes held that there can be no theory of natural language forms, given the amorphous nature of natural language.

Yet if we step back a bit and view the development of logic over the last two millennia, we find a rather different picture. Up until the beginning of the 20th century, logic was very much concerned with representing the inferences that are made in natural language. As we know, classical Aristotelian logic attempted to characterize the valid syllogistic arguments, the canonical example being the following.

[1] One might think we could be more ecumenical and regard the semantics of natural language as being a chapter in Natural Logic as well. We don't have a quarrel with that usage, but most writing on this topic stipulates that Natural Logic runs off of the forms of natural language expressions, and even though work in semantics informs the project it is not, in and of itself, Natural Logic. We will follow the standard usage throughout this book.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović,
Oxford University Press. DOI: 10.1093/oso/9780199591534.003.0001

(1) a. All men are animals.
b. All animals are mortal.
c. ∴ All men are mortal.

The strategy was to elucidate the natural language forms that underwrite valid inferences. Thus the argument above was valid because it instantiated the following form.

(2) a. All As are Bs.
b. All Bs are Cs.
c. ∴ All As are Cs.

As we are typically taught, the syllogistic logic of Aristotle, sometimes called term logic, applies only to a small fragment of natural language—basically the four categorical propositions.

(3) a. All As are Bs.
b. No A is a B.
c. Some A is a B.
d. Some A is not a B.

Even for this limited fragment, the inference rules are arguably ad hoc—there are 256 possible forms of the Aristotelian syllogism, and the handful of valid forms share no obvious syntactic properties.

What is less well known is that in the 2000+ years after Aristotle a number of efforts were made to expand the scope of classical logic while at the same time doing so with fewer and more intuitive rules of inference. As we will see in Chapter 2, two millennia of work in natural logic successfully unearthed a number of deep insights about the nature of inferences in natural language, but was hampered by a lack of understanding of the formal structure of natural language, as well as by a lack of proper technical resources for describing such structure.

One key idea in Medieval Logic, and one that we will discuss in some detail, is the idea that all of the Aristotelian syllogisms could be reduced to two rules corresponding to two syntactic environments, which were sometimes dubbed the *dictum de omni* and the *dictum de nullo*. The rules for the environments are intuitively this: If you are in a *de omni* environment, take any predicate and you can substitute a broader predicate; if you are in a *de nullo* environment, take any predicate and you can substitute a narrower predicate.

For example, consider the sentence (4a), where the verb is considered to be in *dictum de omni* environment. We can substitute a wider predicate for 'bark'. So, it entails the sentence (4b). The *dictum de nullo* works in the opposite direction.

Consider the sentence (5a). In this case, we can introduce a verbal predicate that is narrower. So it entails the sentence (5b).

(4) a. Dogs bark.
    b. ∴ Dogs make noises.

(5) a. Cats don't bark.
    b. ∴ Cats don't bark loudly.

One project simmering for centuries among the Medieval logicians was the idea that you could possibly reduce all of the cases of the syllogism to just these two rules, and that armed with these two rules, you could vastly expand the scope of Natural Logic to include cases that escaped the reach of the traditional syllogism, and perhaps extend it to any case that might be considered a logical inference. This working hypothesis is one that we wish to highlight and explore in this book. And while we cannot say that it was the constant concern of Medieval logicians, we do not think it is unfair to call it the *Holy Grail* of Medieval logic. It is in any case our Holy Grail for purposes of this book, and the questions posed by the Holy Grail are these:

- Can one reduce all of Natural Logic to two basic rules?
- If so, how can we do it?
- If we can't do it, is there a nearby project that we can execute instead?
- If we can execute one of these projects, what are the consequences when we execute it?

Natural language is complicated, and when one considers more complex natural language constructions it can be difficult to identify the *de omni* and *de nullo* environments, and the Medieval project arguably ran ashore on precisely this problem: How do we identify the *dictum de omni* and *dictum de nullo* environments in complex natural language constructions? If we can't do that, then we can hardly achieve the Holy Grail of natural logic.

In this book, we are going to argue that recent work in linguistics and formal logic has spawned a research program that preserves the insights of the traditional Natural Logic program as well as helps us to expand it in a number of fruitful ways. Critically, contemporary syntactic and logical tools can help us identify the *de nullo* and *de omni* environments in natural language, and develop proof procedures for natural language structures.

In a sense, the first step in updating the natural logic project took place with the development of model-theoretic semantics for natural languages, principally following the work of Montague (1970, 1973).

That work allowed us to give a *semantic* account of natural language entailment relations. In that work, entailments are not defined on the basis of linguistic *forms*, but rather on the basis of set-theoretic models of meaning for basic linguistic expressions. Two of the most important aspects of this semantic work have been the study of *directional entailingness* in the semantics of natural language and the development of *generalized quantifier* theory.

Let's talk about directional entailingness first. To some extent, directional entailingness can be understood as a semantic reconstruction of the Medieval *dicta de omni et nullo*. A *dictum de omni* environment is simply one that is *upward entailing*—for any predicate P in such an environment, you can substitute a predicate with an extension that is a *superset* of the extension of P. As for *de nullo* environments, they are understood as being *downward entailing*—for any predicate P in such an environment, you can substitute a predicate that has an extension that is a *subset* of the extension of predicate P. We can state the idea thus:

(6) a. A predicate A is in a UPWARD ENTAILING environment in a sentence φ iff whenever all As are Bs, $[_{\varphi} \ldots A \ldots]$ entails $[_{\varphi} \ldots B \ldots]$.

b. A predicate A is in a DOWNWARD ENTAILING environment in a sentence φ iff whenever all Bs are As, $[_{\varphi} \ldots A \ldots]$ entails $[_{\varphi} \ldots B \ldots]$.

Generalized quantifier (GQ) theory also incorporates several ideas that will be of interest to us, even though we will ultimately move beyond the theory. The first key idea is that a generalized quantifier consists of a determiner plus a *restriction*. Here we follow the GQ literature and understand 'determiner' to include a broad class of expressions that might appear before a noun (phrase) in natural language. So, for example, not only 'the' and 'a' and 'every' count as determiners in GQ, but so too do numerical expressions like 'seven' and 'finitely many' (an idea we will abandon in Chapter 5). Accordingly, an expression like 'seven red balls' counts as a generalized quantifier, with 'seven' being the determiner and 'red balls' being the restriction. The way generalized quantifiers work is that they combine with another predicate (typically expressed by a verb phrase) which is called the *scope* of the generalized quantifier. Thus, in 'seven red balls are bouncing', 'seven red balls' is the generalized quantifier, 'seven' is the determiner, 'red balls' is the restriction, and 'are bouncing' is the scope of the generalized quantifier.

The key semantic idea of generalized quantifier theory is that determiners express cardinality relations involving sets. So, to take our example 'seven red balls are bouncing', the determiner expresses a relation between the set of things that are red balls and the set of things that are bouncing. Specifically, the determiner says that the intersection of those two sets has a cardinality of (at least) seven. This idea is extended to the other determiners—they all express relations between sets that can be expressed in terms of cardinality. Here are some additional examples:

(7)
$$
\begin{aligned}
[\![\text{some}]\!](X, Y) &\text{ iff } |X \cap Y| > 0\\
[\![\text{every}]\!](X, Y) &\text{ iff } |X - Y| = 0\\
[\![\text{no}]\!](X, Y) &\text{ iff } |X \cap Y| = 0\\
[\![\text{(at least) two}]\!](X, Y) &\text{ iff } |X \cap Y| \geq 2\\
[\![\text{(exactly) two}]\!](X, Y) &\text{ iff } |X \cap Y| = 2\\
[\![\text{most}]\!](X, Y) &\text{ iff } |X \cap Y| > |X - Y|
\end{aligned}
$$

Generalized quantifier theory has been one of the most interesting discoveries in semantic theory, and it has yielded a number of amazing insights into the nature of natural language. For our present purposes, the most interesting feature of generalized quantifier theory is that it allows us to account for the directional entailingness of determiner positions.

Consider the distribution of facts given in (8)–(10) and summed up in Table 1.1. What accounts for this distribution of facts? Well, for one, the facts fall out from the basic set-theoretic definitions given for those determiners. This might even come as a surprise to some. We didn't, after all, define the determiners to get the entailment relations right; we defined them so as to get the basic meanings of the determiners right. The entailment facts fall out for free.

(8)	Some men run.	$\therefore$ Some $\overset{\uparrow}{\text{people}}$ run.	$\therefore$ Some men $\overset{\uparrow}{\text{move}}$.
		$\not\therefore$ Some $\overset{\downarrow}{\text{[tall men]}}$ run.	$\not\therefore$ Some men $\overset{\downarrow}{\text{[run fast]}}$.
(9)	No men run.	$\not\therefore$ No $\overset{\uparrow}{\text{people}}$ run.	$\not\therefore$ No men $\overset{\uparrow}{\text{move}}$.
		$\therefore$ No $\overset{\downarrow}{\text{[tall men]}}$ run.	$\therefore$ No men $\overset{\downarrow}{\text{[run fast]}}$.
(10)	Every man runs.	$\not\therefore$ Every $\overset{\uparrow}{\text{person}}$ runs.	$\therefore$ Every man $\overset{\uparrow}{\text{moves}}$.
		$\therefore$ Every $\overset{\downarrow}{\text{[tall man]}}$ runs.	$\not\therefore$ Every man $\overset{\downarrow}{\text{[runs fast]}}$.

Table 1.1 Directional entailment in determiner positions

determiner	restriction	scope
some	upward	upward
every	downward	upward
no	downward	downward
(at least) two	upward	upward
(exactly) two	—	upward
most	—	upward

This semantic approach is critical, but as we noted, it is only a first step. We want to push the project further. We want to develop the syntactic counterpart of this semantic project. Accordingly, after detailing the semantic—in particular, set-theoretic—approaches to inference in natural language in the form of generalized quantifier theory, we will raise the question of whether similar results might not be achieved by attending to the *form* of natural language constructions. That is, if directional entailingness is a viable candidate for semantically identifying the *dictum de omni* and *dictum de nullo* environments, could we find a suitably viable way of identifying those environments by form alone? Could we identify the *dicta de omni et nullo* solely by inspecting the syntax of natural language? And could we do it without making any ad hoc stipulations about the syntax of natural language?

Why is this question important? From the perspective of Natural Logic, it would afford a way of achieving the Holy Grail—having a syntactic way to identify the *de omni* and *de nullo* environments would provide a way of executing the project of reducing the rules of inference down to two basic rules, thus finally delivering the goal of the Medieval project. Beyond that success, we might also purchase some deeper insights into the nature of logic, inference, and negation, and indeed into the syntax of natural language (more on this in a bit).

Of course, everything depends upon whether the idea of syntactically identifying upward and downward entailing environments is feasible in natural language. It is no secret that in first-order predicate logic, these environments can be identified by inspecting *polarity*: a constituent of *positive* polarity is in an *upward* entailing environment, and a constituent of *negative* polarity is in a *downward* entailing environment. Our definition of polarity is applicable only to *Boolean* predicate logic, but this makes it very simple.

(11) a. A constituent has POSITIVE POLARITY iff it is within the scope of an *even* number of negations.
b. A constituent has NEGATIVE POLARITY iff it is within the scope of an *odd* number of negations.

Thus, if some piece of a formula is within the scope of an even number of negations, it has positive polarity and resides in an upward entailing environment, where *dictum de omni* may be applied; if it is within the scope of an odd number of negations, it has negative polarity and resides within a downward entailing environment, where *dictum de nullo* may be applied. But this link between polarity and directional entailingness is most often ignored as first-order predicate logic is thought to be inadequate for expressing the range of meanings arising in natural language. In Chapters 4 and 5 we develop a pair of formal languages, L* ("L Star") and L** ("L Nova"), which illustrate that this objection toward first-order predicate logic is unfounded, and then show how the directional entailingness environments can be read off of the syntax of these languages.

We will first describe a substitutional formal language L* from Law and Ludlow (1985), in which the crucial model-theoretic properties governing entailment relations have syntactic reflexes expressible via polarity. Then, we will offer another formal language, L**, that allows for quantification over collections of individuals, again with a similar formal result concerning entailment relations and polarity. As we will see, L** will be more elegant and syntactically austere and will eventually offer a more natural mapping to the structure of natural language.

Sometimes deep insights are valuable not for what they show, but for the discoveries that they eventually lead to. As we will see, the Medieval insights concerning the *dicta de omni et nullo* are really just a segue to some more interesting insights into the relation between polarity and logic. A formal deductive system can have pleasing meta-theoretical properties like *soundness* (following the rules of logic always produces valid arguments) and *completeness* (all valid arguments can be produced by applying these rules). Guided by polarity, *dicta de omni et nullo* are sound rules, but they do not comprise a complete system—while they produce all valid *directional* inferences (we prove that this is so in section A.3 of the Appendix), some inferences are outside their purview.

The natural deductive system formed by *dicta de omni et nullo* is thus sound but not complete. Can we envision a natural deductive system enjoying both of these properties? Can the Medieval project of reducing the syllogistic logic to two logical rules be pushed further and developed into a complete logic? In Chapters 6 and 7, we will develop a *Dynamic Deductive System (DDS)* that relies on four sub-logical syntactic operations—operations that we will call *Copy, Prune, Add*, and *Delete*—and when we combine these with a polarity-sensitive relation called *p-scope* we will see that we can deploy these basic syntactic operations in a sound and complete deductive system.

Here is a way to explain that idea. Most accounts of formal inference are somewhat abstract. They talk about moving from pattern to pattern, but they seldom get into the nitty gritty business of the mechanics of the actual operation—operations like copying elements, deleting elements, and moving elements elsewhere. You might wonder why anyone would care about such details, but believe us that if you are interested in computational theories of how these operations are actually executed these are things you have to think about. But what we discover when we get down to this nitty gritty level is that there is a single global relation—our p-scope relation—that governs where and how things can be moved and copied and deleted, and, as we will see, that relation is sensitive to polarity above all else. So the Medieval hunch that polarity is important is vindicated, but vindicated at a level of analysis that is much more fundamental.

Our interest with the Medieval project will not be restricted to the *dicta de omni et nullo*. We will also engage an insight that Medieval logicians had about the relation between polarity and quantification. We will start developing this line of thought by reviewing and generalizing a central property of determiners in the GQ theory: *conservativity*. It will turn out that our notion of p-scope, once clarified, can also be employed in an account of restricted quantification, and that this account can in turn be deployed to cash out the syntactic analogue of conservativity. *Restrictedness* will turn out to be a fundamental principle for all of natural language—not just determiner positions—and will have profound consequences for possible logical forms within natural language. Our understanding of restrictedness will allow us to show that quantificational operators in a natural language application of predicate logic are not necessary—we can get by with a simple closure rule that is sensitive to the syntactic environment and in particular to polarity.

Let us give you a glimpse into what this *quantifierless* format of predicate logic (and specifically L^{**}) will look like. As shown below, a natural language sentence will receive a logical form containing no quantificational symbols; all the variables will be free. But how are we to interpret such a formula? Our *Restricted Closure* rule underpinning this interpretation might be easiest to understand if we imagine the interpretation as a two-stage process: we "fill in" the quantifiers in the first step, and interpret the resulting standard formula in the second step. (In reality, however, we envision Restricted Closure as an interpretive procedure rather than a translation taking (12b) to (12c).) Crucially, it turns out that there is a single way to "recover the quantifiers" if we are to produce a restricted (and thus conservative) formula. If we change the type or position of any quantifier in (12c), the result will be a non-restricted formula (or a formula logically equivalent to the original).

(12) a. Every dog is hunting some cat.
b. $\neg D(x) \vee (C(y) \wedge H(x, y))$
c. $\rightsquigarrow \forall x\colon \neg D(x) \vee \exists y\colon C(y) \wedge H(x, y)$

And how does our Restricted Closure work? Finding the closure site of a variable (i.e. the position of the "recovered" quantifier) is easy. It amounts to the lowest position containing all occurrences of the variable. Determining the type of quantification is not that straightforward (we will lay out the details in Chapter 9), but it should come as no surprise that it will involve polarity. In the example above, variable x receives universal closure because $D(x)$ is negated and thus has negative polarity, and variable y receives existential closure because $C(y)$ is not negated and thus has positive polarity.

As we will see, a number of interesting linguistic and philosophical consequences tumble out of the result that quantifiers may be eliminated from natural language logical forms. One of these consequences is a novel treatment of so-called donkey anaphora. Here, deploying the quantifierless format of L^{**} essentially dissolves the issue, effortlessly deriving the core assumptions of one of the most successful accounts of the phenomenon, Discourse Representation Theory. But perhaps most importantly, quantification without quantifiers allows us to realize that the logical form of natural language may be substantially simpler than many imagine. We will show that a first-order formal language such as L^{**} can be seen to resemble natural language structures to a much greater extent than usually envisioned. In fact, we will argue that the quantifierless format of L^{**} is a perfect match to natural language structures as seen by the Minimalist Program of generative linguistics. On this view, it may be that what we call determiners are actually devices that drive structure building, and the resulting structures show no explicit quantificational operators, but merely free variables.

Above, we outlined a line of thought leading us from the simple notion of polarity all the way to the realization that given a suitable encoding of predicate logic formulas, logical forms may be not only similar, but actually isomorphic to the syntactic structure of natural language. We could have never embarked on this voyage in absence of the central idea that *form matters*. Could applying the same strategy to other topics in linguistic theory be just as fruitful? In Chapter 11, we apply the strategy to two further topics—presupposition projection and licensing of negative polarity items. We will see that the basic facts of the former phenomenon simply fall out of the technology developed for our Dynamic Deductive System, and while the latter phenomenon is much more complex we can illuminate it in helpful ways using the tools developed in this book.

Here is the core idea behind a possible syntactic account of the licensing of negative polarity items (NPIs). As indicated even by their name, these expressions, which include 'any', 'ever', 'budge an inch', 'give a damn', and so on, are sensitive to the polarity of the syntactic environment hosting them. As a first approximation, their appearance is ungrammatical in positive polarity sentences (13a) but licensed in negative polarity sentences (13b). A well-known semantic interpretation of this observation is that the licensing environment must be downward entailing, but (as is also well known) this does not cut the cake with examples such as (13c).

(13) a. *John saw anything.
 b. John didn't see anything.
 c. Most people who know anything about politics hate it.

A syntactic account, however, can make immediate sense of (13c) if one notices—as we will notice in our development of L^{**}—that 'most' contributes a negation to the logical form and that this negation scopes over the restriction of the determiner. In the L^{**} logical form for 'most', the phrase 'people who know anything about politics' occurs twice, once within the scope of negation and once without a dominating negation. In effect, the environment is not downward entailing even though it is negated, and this divergence is what yields the bare-bones syntactic account (roughly, that NPIs are licensed when at least one of their logical form occurrences is dominated by a negation).

However, as mentioned above, the bare-bones polarity-based account doesn't afford us the complete understanding of the phenomenon. There is a very broad range of facts about NPIs, including some differences between them that track interesting semantic properties like (anti-)additivity and (anti-)multiplicativity. We will want to explore whether these properties can be given syntactic accounts as well, and our preliminary findings indicate that they can. Even more than that, they indicate that the syntactic account might once again circle back to the notions developed for our Dynamic Deductive System.

Circling back to Dynamic Deductive System ourselves, one of the selling points for this deductive system is that it is not only natural (in the sense that natural deduction systems like Gentzen's are) but that it is representationally minimalistic. It carries out deductions in-line, without building new tree structures or other representations. It is dynamic, in that the deduction is carried out within the sentence itself. As we will see, the result is a surprisingly efficient deductive system. But this will raise the following question: Can a natural deduction system like this really be a model for how persons carry out deduction? Is it a viable thesis in cognitive science? Here we will look at some of the foundational questions regarding the relation between the deductive systems that humans deploy and the nature of normative systems like Natural Logic. We will draw on Chomsky's distinction between competence and performance, and some strategies developed by Lance Rips to show how predictive models of inferential capabilities can be constructed—models that not only lay out norms for correct reasoning, but are predictive of when and where people will fail in their logical reasoning.

All of this will ultimately force us to ask the following questions: Why does polarity play this deep role? Can our Dynamic Deductive System take over for semantics and the theory of meaning? That is, are proof-theoretic accounts of meaning possible? Ultimately, we want to explore the idea that our approach provides a pathway for bringing the logical structure of natural language under the province of cognitive psychology. That is, we want to explore the idea that it not merely provides a way of characterizing inference relations and meaning relations, but also provides a theory of *how* those meanings and relations are represented by human agents.

Summing up: While we take our initial cue from the Medieval project and its Holy Grail, we will see that we can push that project much further. We will see that polarity is the tool that allows us to simplify deduction to a handful of sub-logical operations via the notion of p-scope. This relation will in turn play a key role in our understanding of restrictedness and hence conservativity, which will then lead us to a quantifierless account of quantification in natural language. We will see that this simplification of the logical form of natural language dovetails nicely with recent work in the semantics of natural language, and we will also see that it can be integrated with the Minimalist Program of generative linguistics. And perhaps most importantly, we hope that our work demonstrates the fruitfulness of investigating the formal reflexes of the semantic properties of natural language.

Our chapter-by-chapter plan in this book is as follows. In Chapter 2, we describe some of the insights acquired in natural logic over the last two millennia and survey some recent attempts to incorporate these insights into contemporary generative grammar. In Chapter 3 we show how these insights can be cashed out semantically given current model-theoretic semantics—in particular in its application to generalized quantifier theory—and we survey some earlier attempts to develop syntactic "polarity logics" for natural language in the spirit we advocate here.

In Chapters 4 and 5, we show that the same insights can also be cashed out syntactically (and generalized) within the formal languages L^* ("L Star") and L^{**} ("L Nova"), and illustrate that these languages pack enough expressive power to serve as the vehicle for natural language logical forms.

In Chapter 6, we start the development of our Dynamic Deductive System (DDS). We lay out the architecture of the system and define most of the propositional rules (Delete, Add, and Copy), but most importantly, we develop the central relation of our system: p-scope. In Chapter 7, we finish the propositional part of the system by defining Prune, develop the quantificational aspects of the system, and conclude by showing that DDS is a complete deductive system. In Chapter 8, we explain what we mean by saying that DDS is representationally minimal, and compare our system to Gentzen and Deep Inference natural deduction systems.

In Chapter 9, we look at an important notion from generalized quantifiers theory—conservativity—and show how conservativity can be cashed out via the syntactic notion of restrictedness within our framework. A number of interesting consequences will flow from this result. In particular, it will afford us the resources necessary to be minimalist about quantification, effectively treating all quantification as being either existential or universal closure over open formulas. In Chapter 10, we argue that conservativity and restrictedness, in the generalized sense defined in Chapter 9, are fundamental properties of natural language logical forms, thereby laying the groundwork for deploying the results from Chapter 9 in our analysis of the syntax and semantics of natural language.

In Chapter 11, we take a look at three linguistic phenomena through the lens of the various tools developed in the preceding chapters. Our account of donkey anaphora (and discourse anaphora in general) is based on the quantifierless format of L^{**} from the previous chapter, and our discussion of presupposition projection relies on the Dynamic Deductive System. Our account of the licensing of negative polarity items involves polarity and negation, but more than that, a method for finding the syntactic reflexes of the semantic properties relevant to this phenomenon.

In Chapter 12 we explore the idea that the structure of L^{**} has a claim to being the structure of natural language. We first address some worries about using a variant of predicate logic for semantic analysis of natural language, and then we extend the discussion to the Minimalist Program of generative linguistics and show how the key ideas of our proposal may be executed within contemporary linguistic theory.

All of this will lead us to Chapter 13, in which we explore the question of whether our DDS has a kind of psychological reality and whether it can shed light on performance errors when persons misfire in their judgments on correct entailment relations. We also address remaining conceptual issues, including questions about whether a formal deductive system can take the place of a theory of meaning, and ultimately, about the deep nature of the relation between polarity and inference.

In the Appendix, we lay the mathematical groundwork for our proposal. We start with the formal definition of Dynamic Deductive System and the proof of its soundness, and we continue by proving that the *dicta de omni et nullo* comprise a complete system of *directional* entailingness, and finish with the proof of the equivalence of the semantic notion of conservativity and the syntactic notion of restrictedness.

2
Natural Logic

2.1 Sympathy for Natural Logic

In his excellent survey article on Natural Logic, van Benthem (2008) remarks that some philosophers and logicians have been unkind to Medieval logic. The common wisdom among these unkind philosophers is that Medieval logic was bogged down in Scholasticism and that the great advances of early 20th-century logic left the Medieval enterprise in the dust.

Surely not all philosophers and logicians believe this, and in recent years there has been increasing respect for these Medieval efforts (recent work by Terence Parsons being just one case in point). However, it must be recognized that the unkindness has come from some big names in philosophy.

Notoriously, Kant, in the second preface to the *Critique of Pure Reason*, remarked that "[i]t is remarkable also that to the present day this logic has not been able to advance a single step, and is thus to all appearance a closed and completed body of doctrine" (Kant [1787] 1929, p. 17).

Another fine example of this negative attitude is expressed by Wittgenstein, of all people, in a 1913 review of a logic text. Wittgenstein did not like the book he was reviewing.

> The author's Logic is that of the scholastic philosophers, and he makes all their mistakes—of course with the usual references to Aristotle. (Aristotle, whose name is taken so much in vain by our logicians, would turn in his grave if he knew that so many Logicians know no more about Logic to-day than he did 2,000 years ago). The author has not taken the slightest notice of the great work of the modern mathematical logicians—work which has brought about an advance in Logic comparable only to that which made Astronomy out of Astrology, and Chemistry out of Alchemy. (Wittgenstein 1913)

Is traditional logic really comparable to astrology and alchemy, as Wittgenstein suggests? It's a pretty strong statement about a body of work concerning patterns of inference that are still recognized as being entirely valid.

More recently, philosophers have come to understand that the Medieval logicians had their hands on a very deep project, even if it was wanting for the technical tools of contemporary logicians. The idea is that we can celebrate the recent advances while at the same time noting the fecundity and depth of the Medieval

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović,
Oxford University Press. © Peter Ludlow and Sašo Živanović 2022. DOI: 10.1093/oso/9780199591534.003.0002

program, and hopefully the projects can inform each other. It *certainly* isn't fair to say that there were no advances in natural logic between Aristotle and Frege (or Boole, for that matter). For that matter, our understanding and appreciation of the logic of Aristotle has become deeper as well.

For example, van Benthem notes that the categorical propositions of Aristotle, often ridiculed for being simplistic, were not as simple as they may seem, since the terms could themselves be complex formulas containing relative clauses and operators. To illustrate, suppose that the form of a categorical proposition always evinces the structure $Q(A, B)$, where Q is a quantifier and A and B are terms. For example: 'All men are mortal'. Van Benthem's point is that A and B might very well be quite complex, and have operators contained within them. For example, suppose that A is 'persons who know something about dogs' and B is 'prefer a collie to every poodle they have ever seen'.

Now these terms, packed as they are with operators and relative clauses can quickly become a mess if we don't have the right linguistic tools to describe them, but the point is that we have the tools! If we understand the Medieval natural logicians as being involved in exploring how to correctly unpack these complex terms, then we can see that the enterprise was certainly not trivial, and certainly not a dead project.

Of course, one contemporary view about natural language is that it hardly evinces any structure at all, much less something that could pass as logical structure. Concerns of this kind have been with us at least since Frege, and the general worry has been that natural language is too fluid and dynamic, too piecemeal and fragmented, too vague and undisciplined to allow the kind of formal description necessary in order to carry out sophisticated logical reasoning.

However, with the application of formal tools to language, beginning on the syntactic end with Chomsky in the 1950s and on the semantic and pragmatic end with Montague in the 1960s, language looks much less untamed and much more amenable to yielding sophisticated logical forms and inference patterns. For example, the notion of scope in formal languages has a natural reflex in the c-command relation in linguistics.

As an illustration of these formal tools, let's consider a specific example of formalization from Generative Linguistics in the mid-1970s, since it is pretty straightforward, if now dated. One interesting idea from that period was the introduction of the level of representation LF (suggesting a similarity to the philosopher's notion of logical form). The idea was that LF could be thought of as the syntactic level that interfaced with the semantics. The level LF involved a rule mapping from surface structure (SS) to LF. Called QR, the rule simply said "adjoin quantified NP to S" (May 1977). So, there would be a mapping from D-structure to S-structure, and from S-structure to LF. There would also be a mapping from S-structure to PF or phonetic form yielding the T-model of the grammar shown in Figure 2.1.

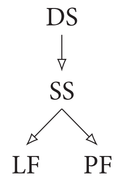
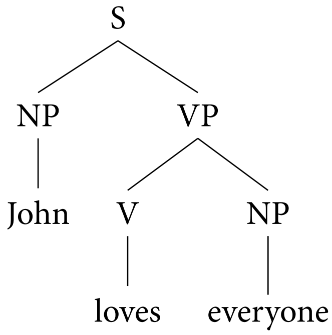


Figure 2.1 The T-model of the grammar

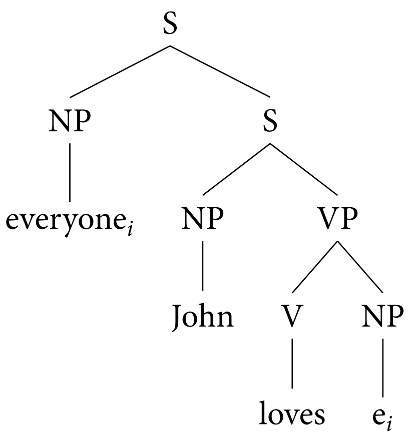
To see how quantifier movement worked, consider sentence (1a) and its SS representation (1b).

(1) a. John loves everyone.
b.



To generate the LF representations we adjoin the NP to the topmost S node (creating a new S node). Chomsky (1973) proposed treating the extraction site as being a "trace" that would be bound either by the moved wh-word or the quantified noun phrase that moved out of the position. This idea was subsequently developed in Wasow (1972) and Chomsky (1975). We can think of the co-indexed trace as being something that is functionally equivalent to a bound variable when we interpret the LF.

(2)

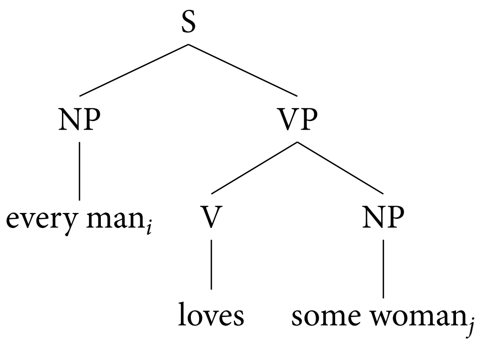


Philosophers were attracted to this idea, because it made LF representations look plausibly like tree-shaped versions of the formulas of first-order logic. In fact, Reinhart (1976) argued that one could define a notion of syntactic scope off of these tree structures using a feature of tree geometry called *c-command*. The idea works as follows.

Let's say that a node α in a tree DOMINATES a node β just in case there is a direct downward path from α to β. Then a node α in a tree C-COMMANDS a node β iff neither α nor β dominates the other and the first branching node dominating α dominates β. Or to put it in simple terms, if you want to know what a node α c-commands, you go to α's mother node. Everything below that mother node on the other branch below that node is c-commanded by α. The next step is to hypothesize that the c-command relation effectively plays the same role as scope in formal logic. In other words, perhaps c-command just is logical scope.

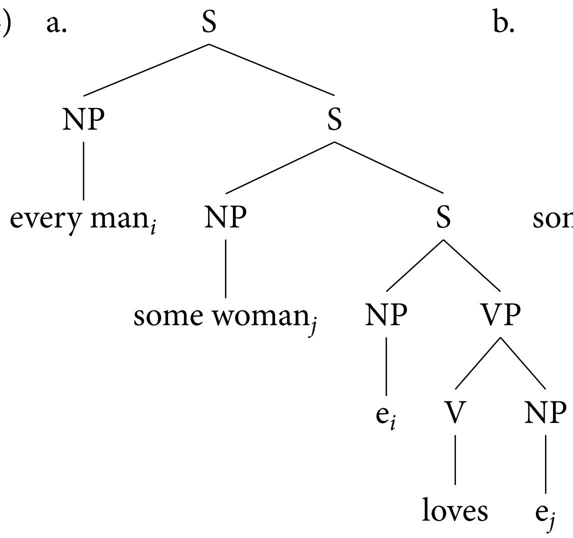
Over the next decade or so a number of arguments were offered in support of QR and LF. Quite naturally, it was seen as a way of providing structural representations that could account for quantifier scope ambiguities. For example, consider sentence (3a) and its SS representation (3b).

(3) a. Every man loves some woman.
b.

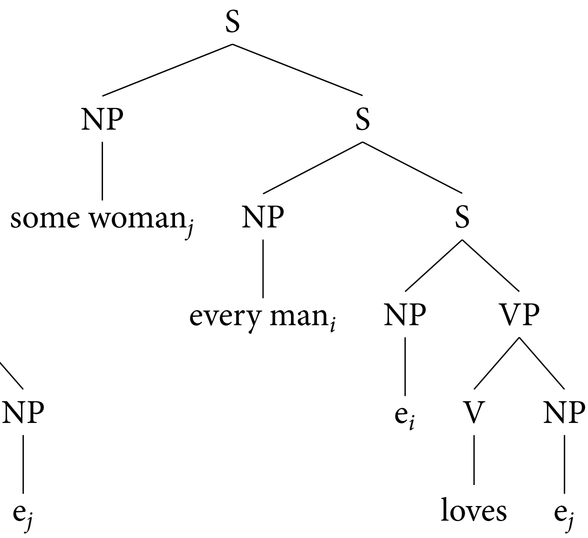


Given that either NP could raise first, this predicted two possible LF structures for the sentence, as indicated in (4).

(4) a.



b.



In short, ambiguities can be resolved structurally, including ambiguities of quantifier scope, which are treated via operations like quantifier raising (QR).

When all is said and done, if these structures are supported by linguistic theory then one might think it possible to develop a version of Natural Logic that is just as sophisticated as our formal logics. How sophisticated? There is a lot going on in formal logic these days—from modal logics and deviant logics to infinitary logics and combinations thereof. Could the logical structure of natural language be sophisticated enough (and rich enough) to do what these logics do? We believe that in many cases, the answer is yes.

There are two ways to characterize the inferences holding between logical statements. One might characterize them using a model-theoretic semantics, or one might characterize them in a formal deductive system. The project of model-theoretic semantics has been the dominant strategy for characterizing natural language inference, and with rare but important exceptions (that we will discuss) there has not been as much effort to characterize natural language inferences via some sort of formal deductive system (something we aim to do).

We consider model-theoretic semantics to be a critical component of the Natural Logic project. The complete picture of Natural Logic, however, should also include a formal deductive system for linguistic forms. Natural Logic thus has a semantic component and a syntactic component (just like formal logics).

This immediately leads to questions about completeness and soundness. If we offer a deductive system for natural logic, will it be complete? That is, will the deductive system for Natural Logic line up with all the entailment relations in the semantics? Is such a thing even thinkable for Natural Logic? You might think it is plausible for fragments of Natural Logic but not for natural language *tout cours*.

A bit later we will delve into these questions in more detail, because for sure they are interesting questions. What do we mean by completeness in this context? Is the idea even coherent for Natural Logic? If the idea is coherent, why should it be a desiderata for Natural Logic? Is there some problem if we can't show completeness? And finally, *can* we show completeness?

We will explore two languages—L^* and L^{**}—that have sufficient expressive power to handle non-elementary quantifiers and which are less expressive than second-order logic and which can, interestingly (perhaps surprisingly, for some), be equipped with a provably sound and complete deductive system.

Our goal here is ambitious on two fronts. First, we believe that many of the most interesting results achievable in formal logics are achievable in Natural Logic. Second, however, we believe that the Medieval project for natural logic—the Holy Grail of Natural Logic—can likewise be executed within this framework. As a consequence, we can construe Natural Logic as a kind of polarity logic in which there are, at bottom, a limited inventory of inference rules, and in which those rules are polarity-sensitive. If that sounds astounding or implausible, that is good, because it means you have been following the thread up to now. In sum, we will be making the case that Natural Logic is not merely a polarity logic as the Medieval logicians envisioned, but it is at the same time a contemporary logic with all of the bells and whistles and meta-logical results.

2.2 The *Dicta de Omni et Nullo* and Natural Logic's Holy Grail

As the Medieval logicians observed, the problem with classical Aristotelian logic is that it is both too constrained and too ad hoc. The theory is ad hoc because the class of valid argument forms is simply stipulated, with no apparent underlying principle or insight. It is too *constrained* because (as noted in the Introduction) it appears to cover only a relatively narrow class of natural language constructions. Inferences like the following don't cleanly fit the original paradigm.

(5) a. Every man is mortal.
  b. Socrates is a man.
  c. ∴ Socrates is mortal.

Nor is the following strictly speaking in the paradigm, as 'flies' must be rendered as 'a thing that flies'.

(6) a. No man flies.
  b. Some bird flies.
  c. ∴ Some thing that flies is not a man.

Furthermore, what does one do with sentential connectives? Sommers (1970) offered a treatment along the following lines, which most of us would consider to be ad hoc:

(7) a. Brazil will win or France will win.
  b. ⇝ All [non-[Brazil will win]] isn't [non-[France will win]].

(8) a. Brazil will win and France will win.
  b. ⇝ Some [Brazil will win] is [France will win].

The troubles don't end there. Consider the following example, adapted from De Morgan (1847):

(9) a. Every horse is an animal.
  b. ∴ Every head of a horse is a head of an animal.

This looked to De Morgan as though it was a logical inference, but how would it follow (or even be expressed) within Aristotelian logic? And as Sánchez Valencia (1994) noted, the same point was made centuries before De Morgan—by Joachim Jungius (1587–1657)—in the following example:

(10) a. Grammatica est ars.
"Grammar is an art."
b. ∴ Qui discit grammaticam discit artem.
"Who learns grammar learns an art."

It is of course a standard exercise of undergraduate logic texts to force other natural language sentences into this format. Thus we have the following translations:

(11) Socrates is a man. $\rightsquigarrow$ Every Socrates is a man.

(12) No man flies. $\rightsquigarrow$ No man is a thing that flies.

But these paraphrases are unmotivated as they stand. The Medievals saw this, and over a period of centuries pursued a research program that attempted to reduce the number of rules of inference and at the same time expand their coverage. Although the research program involved a lot of groping in the dark, one key idea appeared to be driving the best and most interesting research: One can reduce the Aristotelian inferences to two basic paradigms, and these paradigms can cover a broad class of natural language argument forms—certainly much broader than the original Aristotelian paradigm would have allowed.

The idea—what we have called the *Holy Grail* of Natural Logic—was that there are two distinct environments and that the class of inferences could be reduced to two rules, one corresponding to each of the two environments. The two basic paradigms, the *dictum de omni* and the *dictum de nullo*, are the inference paradigms which today we would think of as monotone increasing (or upward entailing) and monotone decreasing (or downward entailing) environments.[1]

As we noted in the Introduction, the basic idea can be summed up in the following way, where $A < B$ is a general way of indicating that all As are Bs, or that every A is (a) B (we are avoiding set-theoretic terminology here because it is a semantic resource, but cheating a bit and putting the generalization in set-theoretic terms, we would say that A is a subset of B).

[1] This connection has been observed in Hoeksema (1986), van Benthem (1991), and Sánchez Valencia (1991); but see Sánchez Valencia (1994) for an observation that the parallelism between downward entailingness and the *dictum de nullo* is not perfect. For now, we are going to use these notions interchangeably.

(13) A constituent A is in a *dictum de omni* environment in a sentence φ iff whenever $A < B$, $[_{\varphi} \ldots A \ldots]$ entails $[_{\varphi} \ldots B \ldots]$.

(14) A constituent A is in a *dictum de nullo* environment in a sentence φ iff whenever $B < A$, $[_{\varphi} \ldots A \ldots]$ entails $[_{\varphi} \ldots B \ldots]$.

To see that this works in the most obvious cases, simply consider some of the classical inferences discussed above. If we assume that the nuclear scope (B) position in 'every A is a B' is a *dictum de omni* environment, then we can simply swap 'mortal' for 'animal' in the argument below, since, following the second premise, 'animal' < 'mortal'. (Here we adopt the convention of underbracketing the position where the substitution will occur, and we will indicate the $A < B$ being drawn upon.)

(15) a. Every animal is mortal. ('animal' < 'mortal')
b. Every man is an animal.
c. ∴ Every man is mortal.

But if we add a negation, as in the following argument, then we have to use the *de nullo* rule. Now we can substitute 'animal' for 'mortal'.

(16) a. Every animal is mortal. ('animal' < 'mortal')
b. Some man is not mortal.
c. ∴ Some man is not an animal.

Not only can these two paradigms cover the Aristotelian syllogisms,[2] but they cover a number of other apparently valid inferences that fall outside of the Aristotelian paradigm as well. Consider the two cases discussed above. The first is simply a case of the *dictum de omni* paradigm.

(17) a. Every man is mortal. ('man' < 'mortal')
b. Socrates is a man.
c. ∴ Socrates is mortal.

[2] In most cases the actual formal derivation of the valid syllogisms is trivial, but in several cases the derivation requires the following equivalence rules: (i) No S is P. = All S is not P. (ii) Some S is P. = Some P is S. (iii) No S is P. = No P is S. So, to take the most complicated case, the syllogism EIO in the fourth figure requires the following sort of derivation.

1. No P is M.	(premise)	4. All M is not P.	(from 3, by (i))
2. Some M is S.	(premise)	5. Some S is M.	(from 2, by (ii))
3. No M is P.	(from 1, by (ii))	6. Some S is not P.	(from 4 and 5, by *dictum de omni*)

The disputed syllogisms can also be derived, but (as one would expect) only given the following additional rule: (iv) All S are P. ⊢ Some S are P.

Since ‘is a man’ is in a *de omni* environment, the first premise licenses the substitution. Had the second premise been ‘Socrates is not a mortal’ then the *de nullo* paradigm would have applied and we would have been entitled to conclude ‘Socrates is not a man’. Similar reasoning applies to the following:

(18) a. Every bird flies. (‘bird’ < ‘flies’)
b. No man flies.
c. ∴ No man is a bird.

Here, since ‘flies’ is in a *de nullo* environment we can substitute ‘bird’ for ‘flies’.

Even the rules Modus Ponens and Modus Tollens from contemporary propositional logic fall within this paradigm if we extend $A < B$ to include ‘if A then B’. Then Modus Ponens is simply a specific instance of the *dictum de omni* rule (one where S contains only A).

(19) a. If Smith is tall then Jones is short. (‘Smith is tall.’ < ‘Jones is short.’)
b. Smith is tall.
c. ∴ Jones is short.

Modus Tollens is simply an instance of the *dictum de nullo* rule (again, the instance where S contains only A).

(20) a. If Smith is tall then Jones is short. (‘Smith is tall.’ < ‘Jones is short.’)
b. Jones is not short.
c. ∴ Smith is not tall.

Earlier in this chapter we discussed an example, due to De Morgan, which he claimed fell outside the Aristotelian paradigm. De Morgan’s solution to this problematic case was to appeal to the *dicta de omni et nullo*. Thus he reasoned as follows.

(21) a. Every horse is an animal. (‘horse’ < ‘animal’)
b. Every head of an animal is a head of an animal.
c. ∴ Every head of a horse is a head of an animal.

The trick is the introduction of the tautology in the second premise and then using the *dictum de nullo* rule to substitute into that tautology (here taking the restriction of the quantifier ‘every’ to be a *dictum de nullo* environment—we will motivate this assumption later in the book). The same strategy can be applied to the argument due to Joachim Jungius. We introduce a tautology and then apply the *dictum de nullo* rule—the assumption being that the quantificational structure of the second premise would make the first position downward entailing.

(22) a. Grammatica est ars. (‘grammatica’ < ‘ars’)
b. Qui discit artem discit artem.
c. ∴ Qui discit grammaticam discit artem.

Notice what a radical departure this is from our contemporary way of thinking about the proper way of characterizing inferences. Typically, we suppose that various kinds of inference rules may be required to cover the panoply of inferences we engage in. But the headline idea of a natural logic program that incorporates the *dictum de omni et nullo* is that no other rules of inference are ultimately required.

It is a gripping project (we didn’t call it the Holy Grail for nothing!) and in the 13th century it captured the imagination of Peter of Spain, William of Sherwood, and Lambertus of Auxere. In the 14th century, Ockham (1954, p. 330) claimed that the “dicta directly or indirectly govern all the valid syllogisms.” According to Ashworth (1974, p. 232), by the 16th century, “*[d]ici* (or *dictum*) *de omni* and *dici de nullo* were the two so-called regulative principles to which every author appealed in his account of the syllogism.”[3]

Despite the great promise of the *dicta de omni et nullo* paradigm in Natural Logic, certain questions arise even in the simple cases we discussed above. We assumed without argument that the first position of ‘every’ (i.e. the noun phrase) must be a downward entailing environment, but why? Let’s return to the paradigm for determiner positions we offered in the Introduction, and then flesh it out a bit.

(23) Some men run. ‘some’
a. i. ∴ Some people run. restriction: *dictum de omni*
ii. $\not\therefore$ Some tall men run. scope: *dictum de omni*
b. i. ∴ Some men move.
ii. $\not\therefore$ Some men run fast.

(24) No man runs. ‘no’
a. i. $\not\therefore$ No people run. restriction: *dictum de nullo*
ii. ∴ No tall men run. scope: *dictum de nullo*
b. i. $\not\therefore$ No men move.
ii. ∴ No men run fast.

[3] See Sánchez Valencia (1994) for a survey.

(25) Every man runs. ‘every’

a. i. $\not\therefore$ Every person runs. restriction: *dictum de nullo*

ii. $\therefore$ Every tall man runs. scope: *dictum de omni*

b. i. $\therefore$ Every man moves.

ii. $\not\therefore$ Every man runs fast.

As we noted, there is a missing explanation for why these determiner positions have the polarity positions that they do, and just to make matters more puzzling, a determiner like ‘most’ is neither upward nor downward entailing in its restriction. What is going on here?

(26) Most men run. ‘most’

a. i. $\not\therefore$ Most people run. restriction: —

ii. $\not\therefore$ Most tall men run. scope: *dictum de omni*

b. i. $\therefore$ Most men move.

ii. $\not\therefore$ Most men run fast.

These are simple cases, and they already give rise to some puzzling questions about the motivation and criteria of identity for the *dictum de omni* and *dictum de nullo* environments. But when we consider more complex natural language constructions, those environments seem like moving targets.

One problem that became apparent to the Medieval logicians was that the substitution rules do not clearly work when we try to substitute inside of a relative clause.

(27) a. Every Greek is human. (‘Greek’ < ‘human’)

b. Every person who loves all humans is happy.

c. $\not\therefore$ Every person who loves all Greeks is happy.

‘Humans’ is in the restriction of ‘all’, but apparently not in a *dictum de nullo* environment in this case. In point of fact, the environment has flipped to a *dictum de omni*—presumably because of it being embedded in a relative clause, which is itself in the restriction of ‘every’. The problem here is that with just a few embeddings these constructions can become very complex syntactically, and one needs a firm grasp on scope in natural language and the structure of relative clauses (and a theory of polarity flipping) to work out the details (if they can be worked out). For the Medieval logicians, the relevant tools were not available.

This is especially true when we think about scope ambiguity and how that creates headaches within the Natural Logic project.

(28) a. Every collie is a dog. ('collie' < 'dog')
b. Every man who feeds a dog that I know runs a butcher shop.
c. $\therefore^{?}$ Every man who feeds a collie I know runs a butcher shop.

Is this inference valid? Well, that depends on whether or not 'a dog I know' is understood to take scope over 'every man'. If it does take wide scope then the inference is no good. If it remains tucked within the scope of 'every' man, then the inference is fine.

Perhaps you think you can see a solution path for the Medieval logician here. If so, that's good. We think there is one too, and not to give away any spoilers, if you were thinking that being in a relative clause in the scope of a *dictum de nullo* environment can flip your usual polarity, you are on the right track.

Of course, working out the details of this requires assumptions about the nature of relative clauses and ways of formalizing scope relations in natural language. In short, it requires that we have access to the tools of contemporary linguistics. Ultimately we will offer our own formal theory of the structure of these environments, but before we do that, we want to take one last deep dive into the nature of polarity in Medieval Logic, in order to show just how deep the water goes. For, as we will see in the next section, polarity (the *dicta de omni et nullo*) is also intimately entwined with the nature of quantification.

2.3 The Theory of Supposition, Distributive Terms, and Quantification

In the previous section we looked at the Medieval theory of the *dicta de omni et nullo*, and the idea that it might play some role in unifying and simplifying the theory of inference. Medieval logicians had more on their plate than inference, however, including what is often known as the theory of *supposition*. The Medieval theory of supposition is a theory of what terms refer to—or at least it is often characterized that way. But in saying this, the doctrine is really about moving from something like 'every dog barks' to some conjunction: 'Fido is a dog and Fido barks, and Lassie is a dog and Lassie barks, etc.' Similarly formulas like 'some dog barks' could unpack as a disjunction: 'Fido is a dog and Fido barks, or Lassie is a dog and Lassie barks, etc.'

As we will see, the Medieval logicians observed that there was a deep connection running between the theory of supposition and the theory of inference, and ultimately polarity in natural language. In other words, the *dicta de omni et nullo* made an appearance here too, if not always explicitly recognized.

As Dutilh Novaes (2011) observes, once the Medieval theory of supposition is worked out, one effectively has a theory of quantification in place. So, for example, Medieval Logic did not have explicit talk of existential and universal quantifiers, but because so-called distributive terms can be unpacked as infinite (or at least arbitrarily long) conjunctions, and so-called determinative or non-distributive terms can be unpacked as infinite (or at least arbitrarily long) disjunctions, they effectively become universal and existential quantifiers. To be more precise, the terms become understood as the restrictions of quantifiers (taking a quantifier to be a determiner plus its restriction, in the sense of generalized quantifier theory). Or at a minimum, they become quantifiers on some theories of quantification—including, but not limited to, the theory Wittgenstein offered in the *Tractatus* (which is more than a little ironic given Wittgenstein's aforementioned trash-talking of traditional logic).

The interesting point for our purposes is that whether a term is distributive or non-distributive, or, if you prefer, whether it behaves as a distributive or non-distributive term, depends on its surrounding syntactic environment, and at least in part depends upon the polarity of the environment in which it occurs.

This is how the Medieval logicians wed their insights about upward and downward entailing environments to the doctrine of distributive terms and ultimately quantification. We can think of a distributive term as being a quantifier restriction term that is in a downward entailing *dictum de nullo* environment, and hence one where the subset of the extension of a term could be substituted.

Matters are not exactly this tidy, however, and we will find it useful to look at the development of these ideas begining in the 14th century. We will follow T. Parsons (2008), who provides a helpful discussion of supposition theory in Medieval Logic, particularly with respect to Buridan and Ockham.

As noted earlier, supposition theory itself isn't obviously a theory of inference—it initially looks closer to a theory of quantification if anything—but whether intentionally or by accident, it ends up tied to a more general theory of inference. To see this we first want to introduce the notions of *ascent* and *descent*. Parsons offers the following analysis of distributive supposition, and describes its relation to ascent and descent as follows (attributing it to Ockham).

(29) A term F has DISTRIBUTIVE SUPPOSITION in a proposition P iff

(DESCENT) you may descend under F to a conjunction of propositional instances of all the Fs, and

(ASCENT) from any one instance you may not ascend back to the original proposition P.

Similar reasoning applies to determinate positions, which, following Buridan and Ockham, Parsons defines as follows.

(30) A term F has DETERMINATE SUPPOSITION in a proposition P iff
(DESCENT) you may descend under F to a disjunction of propositional instances of all the Fs, and
(ASCENT) from any such instance you may ascend back to the original proposition P.

Thus we can replace 'Some dog is an animal' with a disjunction: 'Fido is dog and an animal, or Lassie is a dog and an animal, etc.' Now, as we noted, there are well-known analyses of the universal and existential quantifier which unpack those quantifiers as conjunctions and disjunctions respectively, but notice that we aren't treating the determiners themselves as being quantifiers here. The determiners are instead keys to identifying certain syntactic environments that in turn control inferential relations.

In the case of ascent, we can move from cases like 'Lassie is a dog and Lassie barks' to 'Some dog barks'. We might recognize this as a traditional version of Existential Generalization. Ascent to a universal is also possible, and descent looks like traditional versions of Universal and Existential Instantiation. All the traditional rules (UI, EI, UG, EG) are supposed to fall out in this paradigm.

An observation is called for here. As noted earlier, the distributive environments look like downward entailing environments and the determinate environments look to be upward entailing. Accordingly, those environments trigger the expected inference patterns. Here is how T. Parsons (2008, pp. 256–257) describes the application of that insight.

> When the notions of modes of supposition were introduced, they brought along with them new and useful ways to assess inferences. [...] Many proposals made with the new terminology were vague, and of limited usefulness. But some applications were clear and compelling. One is the case of inference "from a superior to an inferior" with a distributed term.
>
> From a superior to an inferior: If a (non-parasitic) term T is distributed in a proposition P, then from P together with a proposition indicating that $T*$ is inferior to T, the proposition that results by replacing T by $T*$ in P follows.
>
> A common term $T*$ is inferior to a term T iff 'every $T*$ is T' is true; this is also the condition for T being superior to $T*$. And a singular term t is inferior to a common term T iff 't is T' is true; again, in this case T is superior to t.

Not surprisingly, one ends up with the distribution of facts in the *dicta de omni et nullo* paradigm. T. Parsons (2008, p. 257) illustrates this with the basic Aristotelian syllogisms we discussed earlier.

The simplest illustration of this is Aristotle's syllogism BARBARA:

Every M is P
Every S is M
∴ Every S is P

The term M is distributed in the first premise. The second premise states that S is inferior to M. The conclusion follows by replacing M with S in the

No M is P
Every S is M
∴ No S is P

Again, M is distributed in the first premise, and the second states that S is inferior to M. The conclusion follows by replacing M with S in the first premise.

As Parsons notes, this observation had uptake with the key logicians of the period. These include Paul of Venice (2002, p. III.3):[4] "from a lower-level term to its corresponding higher-level term affirmatively and without a sign of distribution and without any confounding signs impeding there is a solid inference. For example, 'man runs; therefore, animal runs'." Similarly, in Ockham (1954, pp. III.3/6) we have this: "ab inferiori ad superius sine distributione et affirmative est bona consequentia et simplex"; in Parsons' (2008, p. 258) translation: "From an inferior to a superior without distribution and affirmative is a good consequence and simple."[5] Meaning, we take it, that if you are not in a distributive position, but are in an affirmative position, you can move from an inferior term to a superior term (subset to superset).

If you think about it, the *supposito* is a special case of the *dicta de omni et nullo*. If we think of 'every dog barks' as denoting a set of worlds, then by the *dicta de omni* we should be able to swap in any superset of those worlds—for example, the set of worlds in which Lassie barks—and retain truth value. So it seems that the *dicta de omni et nullo* not only ground the inferences, but fundamentally ground the *supposito* as well (and ultimately Universal Instantiation, etc.).

We've been oversimplifying a bit, however. So far we have only been talking about terms that appear in the first position of a categorical proposition, or if you prefer using the language of generalized quantifier theory, the restriction of the determiners. What about the terms in the second position (the nuclear scope of the determiner)?

Here we will find it useful to return again to the work of T. Parsons (2008, pp. 260 sqq.), who constructs a theory grounded in the "best of Burley–Ockham–

[4] Quote from T. Parsons (2008, p. 258).
[5] As Parsons notes, Ockham later modifies this a bit.

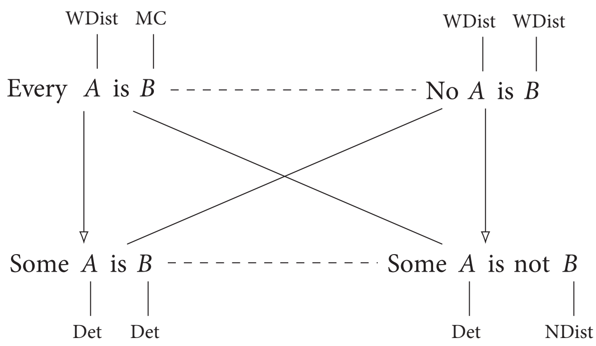


Figure 2.2 Medieval taxonomy of determiner position properties

Buridan."[6] On Parsons' reconstruction, we want to distinguish four different environments or moods: *determinate (Det), wide distributive (WDist), narrow distributive (NDist)* and *merely confused (MC)*. In this instance, the "merely confused" environment/mood would be in the second position of 'every A is B' (the scope of the determiner 'every'). The narrow distributive environment would be in the second position of 'some A is not B'. The resulting taxonomy, placed in a traditional square of opposition, is shown in Figure 2.2.

Now comes the interesting part. Once we have identified the environment as being either Det or WDist or NDist or MC, we have a guide to the availability of ascent and descent.

DETERMINATE Descent to a disjunction of propositions, and ascent back from the whole disjunction.

MERELY CONFUSED No descent to a disjunction of propositions but ascent back from the disjunction.

WIDE DISTRIBUTIVE Descent to a conjunction of propositions and ascent back from the conjunction.

NARROW DISTRIBUTIVE Descent to a conjunction of propositions and no ascent back from the conjunction.

And if we think of this in terms of the traditional understanding of UI, EI, UG, and EG, the taxonomy yields the following guidelines for the availability of those inferences.

DETERMINATE Existential Instantiation holds, and so does Existential Generalization.

MERELY CONFUSED Existential Instantiation does not hold, but Existential Generalization does.

[6] The Burley sources: Burley ([14th century] 1972, [14th century] 1955).

WIDE DISTRIBUTIVE Universal Instantiation holds, and so does Universal Generalization.

NARROW DISTRIBUTIVE Universal Instantiation holds, but Universal Generalization does not hold.

You may, like us, question whether it is really the case that the Narrow Distributive environment blocks Universal Generalization, and in point of fact, Parsons seems to suggest that it is more of a syntactic constraint than a logical constraint.

Natural language involves a lot more than simple categorical propositions. Sometimes terms are embedded—for example in the scope of negation or within a relative clause. We can speculate that a negation is going to flip polarity and we can also speculate that the mode of a term will be flipped by negation as well. But when we start embedding terms within terms matters become complicated. Historically, this is where Natural Logic starts to bog down. Parsons himself takes a stab at coming up with systematic rules but ultimately throws in the towel.

> Perhaps there are some subtle rules that will constitute an algorithm for determining the mode of supposition of a non-main term from some information about it that is evident from inspection (which is how the rules for the modes of supposition of main terms work), but I don't know how to construct them.
>
> (T. Parsons 2008, p. 276)

The question is, could we do better with contemporary linguistic tools, a better understanding of polarity, and the meta-theoretical resources at our disposal? It's a question without an obvious answer, because Parsons himself was more than fluent in contemporary linguistics and meta-theory. It stands to reason that off-the-shelf resources may not be enough. But the question remains: can the modes of supposition be systematically identified in the syntax of natural language?

One thought might be that some of the judgments involving merely confused and narrow distributive environments are pretty subtle, and we can, if we really want, ignore those judgments and go for a more elegant story. It seems that this sentiment was shared by the Port Royal Logicians, who decided to simplify the paradigm significantly. Martin (2013), in his discussion of the doctrine of

Table 2.1 The Port Royal Logic schema for identification of distributive terms

	Q	Subject	C	Predicate
(A)	all	distributive	are	non-distributive
(E)	no	distributive	are	distributive
(I)	some	non-distributive	is	non-distributive
(O)	some	non-distributive	is not	distributive

distributive terms in the Port Royal Logic, noted that the schema shown in Table 2.1 was in place for the identification of distributive terms. (If you have forgotten your categorical propositions from Aristotelian Logic, recall that an example of A would be 'all A are B', E would be 'No A are B', I would be 'Some A are B', and O would be 'some A are not B'.) You might remember, from the Introduction, that this taxonomy precisely mirrors our discussion of the upward and downward entailing environments for quantified expressions. We need merely substitute 'downward entailing' for 'distributive' and 'upward entailing' for 'non-distributive'.

Noting that more subtle stories are possible, we plan to stick with the analysis that was articulated so clearly by the Port Royal Logicians. Crucially, for now, we want to emphasize that polarity, inference, and quantification are intimately related. What we want is to understand exactly how and why they are related in this way. In later chapters, after we develop some more formal tools, we will explore these question in detail. First, however, we need to survey some contemporary versions of the natural logic project.

3
Previous Accounts of Polarity Logics for Natural Language

3.1 Syntactic Accounts of Directional Entailingness in Natural Language

One way of thinking about the Medieval efforts at defining the *dicta de omni et nullo* is that they were attempts at defining directional entailingness in a syntactic way. Clearly, success in this endeavor would have important consequences. If all of Natural Logic can be reduced to cases of the *dicta de omni et nullo* and if those paradigms can in turn be identified as features of clearly identifiable syntactic forms, then the natural logic project becomes a chapter in the syntax of natural language. And who knows? Given the advances that have been made in generative linguistics, perhaps the Medieval Natural Logic project can be successfully executed—perhaps its Holy Grail is within *our* grasp.

Admittedly, these days it is unusual to suppose that natural language syntax should have anything to do with logical entailment. Ordinarily, we take entailment relations to be the province of semantics. But this supposition can be questioned. Pre-theoretically there is no reason to suppose that either syntax or semantics has exclusive claim on entailment relations, and a handful of linguists have pursued the idea that syntax has a good deal to say about them. For example, Lakoff (1972) and McCawley (1972) both attempted to incorporate a version of the Natural Logic program within generative semantics, and Suppes (1979) offered a number of inference rules that are defined off of forms generated by context-free phrase structure grammars. The Suppes project is interesting in that the rules he defined effectively tracked the *dicta de omni et nullo*, albeit in a brute force manner. For example, consider the following two rules which identify particular instances of the *dictum de omni* and *dictum de nullo* respectively (here N and TV indicate a noun and a transitive verb; EQ and UQ are the existential and the universal quantifier 'some' and 'all', Aux is a auxiliary like the word 'do', and Cop is a copula like 'are').

(1)	a. $EQ+N_0+TV+UQ+N_2$	Some dogs bite all animals.
	b. $UQ+N_1+Cop+N_2$	All cats are animals.
	c. $\therefore EQ+N_0+TV+UQ+N_1$	∴ Some dogs bite all cats.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović,
Oxford University Press. © Peter Ludlow and Sašo Živanović 2022. DOI: 10.1093/oso/9780199591534.003.0003

(2) a. $UQ+N_0+Aux+Neg+TV+UQ+N_1$ All dogs do not love all cats.
b. $UQ+N_1+Cop+N_2$ All cats are animals.
c. $\therefore UQ+N_0+Aux+Neg+TV+UQ+N_2$ ∴ All dogs do not love all animals.

Instead of identifying two syntactic environments and providing two basic rules, Suppes offered 75 separate rules of inference, which, as a group, aproximated the *dicta de omni et nullo* environments for a fragment of English. Of course, one would like to identify these environments with fewer rules (two in the ideal case) and one wonders how the project scales when we consider larger fragments of English and other languages. Just how many rules would be necessary in the end?

Subsequent efforts attempted to identify the relevant syntactic environments through the introduction of polarity marking. In particular, Sánchez Valencia (1991, 1995), Dowty (1994), and van Benthem (2008) offered theories in which one can define inferences off of linguistic representations. All utilize categorial grammar frameworks and at the same time incorporate the insights of traditional work on upward and downward entailing environments.

The Sánchez Valencia (1991) proposal, for example, involves a three-step process that begins with the polarity marking of lexical items. The idea is that the primitive assignments give the category plus the internal polarity marking. So, whereas a determiner would ordinarily belong to the category $((e,t),((e,t),t))$ in a categorial grammar, in Sánchez Valencia's system we have the following primitive assignments.

(3) a. Every: $((e,t)^-,((e,t)^+,t))$
b. Some/a: $((e,t)^+,((e,t)^+,t))$
c. No: $((e,t)^-,((e,t)^-,t))$

Notice that these markings are of purely syntactic significance and reflect but do not replace the semantic properties of the determiners (which would be specified by generalized quantifier theory). The second step in the polarity logic is external polarity marking, which governs how the markings on the lexical items affect the marking on the phrasal categories. The following three rules are applicable.

(4) a. $$\frac{(\alpha^+,\beta) \quad \alpha}{\beta} \rightarrow \frac{\overset{(\alpha,\beta)}{+} \quad \overset{\alpha}{+}}{\beta}$$ (positive monotone marking)

b. $$\frac{(\alpha^-,\beta) \quad \alpha}{\beta} \rightarrow \frac{\overset{(\alpha,\beta)}{+} \quad \overset{\alpha}{-}}{\beta}$$ (negative monotone marking)

c. $$\frac{(\alpha,\beta) \quad \alpha}{\beta} \rightarrow \frac{\underset{+}{(\alpha,\beta)} \quad \alpha}{\beta}$$ (non-monotone marking)

Given these three rules and the primitive assignments introduced in the first step, we can generate phrasal trees with positive and negative markings at each node, as in the following structure generated for 'Abelard didn't catch every unicorn'.

(5) Abelard didn't catch every unicorn

$$((e,t),t) \quad ((e,t)^-,(e,t)) \quad (e,(e,t)) \quad ((e,t)^-,(e,(e,t))^+,(e,t)) \quad (e,t)$$

$$\frac{\underset{+}{((e,t)^-,(e,(e,t))^+,(e,t))} \quad \underset{-}{(e,t)}}{((e,(e,t))^+,(e,t))}$$

$$\frac{\underset{+}{(e,(e,t))} \quad \underset{+}{((e,(e,t))^+,(e,t))}}{(e,t)}$$

$$\frac{\underset{+}{((e,t)^-,(e,t))} \quad \underset{-}{(e,t)}}{(e,t)}$$

$$\frac{\underset{+}{((e,t),t)} \quad \underset{+}{(e,t)}}{t}$$

The final step is to determine the polarity of each constituent using the following rules.

(6) If T is a syntactic tree with root S, then
  a. A node N HAS POLARITY in T iff all the nodes in the path from N to S are marked.
  b. A node N is POSITIVE in T iff N has polarity in T and the number of nodes in the path from N to S marked "−" is *even*.
  c. A node N is NEGATIVE in T iff N has polarity in T and the number of nodes in the path from N to S marked "−" is *odd*.

For example, in the tree introduced above, 'unicorn' bottoms out in positive polarity by (6b), and may thus be replaced by a term with a larger denotation, say 'animal', by the upward monotonicity inference rule.

Dowty (1994) simplifies Sánchez's proposal by building the monotonicity marking (4) and the polarity determination (6) into the slash-elimination rules of categorial grammar. Instead of having a three-step procedure, he ends up with a single, but equivalent step.[1]

[1] We have our doubts whether Dowty's proposal can count as a genuine simplification. His proposal is easier on the eye, but as we will see below, constructing the parse tree of a sentence involves a kind of "puzzle-solving" which, we suspect, if made into an explicit algorithm, would resolve back to Sánchez's three-step procedure.

Dowty uses the following primitive syntactic categories: NP (noun phrase) of type e, S (sentence) of type t, and CN (common noun) of type (e, t). Complex categories are built by right- and left-leaning slash: the slash A/B indicates that something of type A is looking to combine with something of type B to its right; the backslash $(B\backslash A)$ indicates that something of type A is looking to combine with something of type B to its left. For conciseness, he abbreviates NP\S to VP (verb phrase) and VP/NP to TV (transitive verb). But crucially, he proposes that categories can be endowed by polarity markings (7) and that slash-elimination (i.e. functional application) rules (8) must respect these polarity markings.[2]

(7) If A/B is a category, so are A^+/B^+, A^+/B^-, A^-/B^+, A^-/B^-.

(8) Slash-elimination (/E) rules:

a. $\dfrac{A^+/B^+ \quad B^+}{A^+}$ and $\dfrac{A^-/B^- \quad B^-}{A^-}$ (polarity preserving)

b. $\dfrac{A^+/B^- \quad B^-}{A^+}$ and $\dfrac{A^-/B^+ \quad B^+}{A^-}$ (polarity reversing)

In Dowty's system, most lexical entries appear in two, polarity-wise opposite categories, but with the same semantic interpretation. As shown below, this holds for both content and functional lexical items. Determiners (9d)–(9f) are furthermore assumed to have direct object counterparts of category (TV\VP)/CN (of appropriate polarities).

(9) a. unicorn: i. CN^+, ii. CN^-
b. catch: i. $\mathrm{VP}^+/\mathrm{NP}^+$, ii. $\mathrm{VP}^-/\mathrm{NP}^-$
c. didn't: i. $\mathrm{VP}^+/\mathrm{VP}^-$, ii. $\mathrm{VP}^-/\mathrm{VP}^+$
d. every: i. $(\mathrm{S}^+/\mathrm{VP}^+)/\mathrm{CN}^-$, ii. $(\mathrm{S}^-/\mathrm{VP}^-)/\mathrm{CN}^+$
e. some/a: i. $(\mathrm{S}^+/\mathrm{VP}^+)/\mathrm{CN}^+$, ii. $(\mathrm{S}^-/\mathrm{VP}^-)/\mathrm{CN}^-$
f. no: i. $(\mathrm{S}^+/\mathrm{VP}^-)/\mathrm{CN}^-$, ii. $(\mathrm{S}^-/\mathrm{VP}^+)/\mathrm{CN}^+$

An example will illustrate how this works. Consider the sentence 'Abelard didn't catch every unicorn' again. The root of the sentence must bottom out in positive polarity and so we have a kind of puzzle to solve: which of the pairs of lexical entries must we select to end up with positive polarity for the root? There is really only one solution for this. If the VP must have positive polarity at the end, then we know that 'didn't' must combine with something of negative polarity to its right (because both lexical entries for 'didn't' are polarity flippers). That means we need the negative polarity of the TV\VP, which is going to mean we will need the corresponding

[2] Parallel definitions hold for left-leaning slash categories.

negative polarity entry for 'every' and that is going to drive selection of the positive polarity entry for 'unicorn'.

(10) Abelard didn't catch every unicorn

$$\dfrac{\dfrac{}{\text{Abelard}\atop S^{+}/VP^{+}} \qquad \dfrac{\dfrac{}{\text{didn't}\atop VP^{+}/VP^{-}} \qquad \dfrac{\dfrac{}{\text{catch}\atop VP^{-}/NP^{-}} \qquad \dfrac{{\text{every}\atop (TV^{-}\backslash VP^{-})/CN^{+}} \qquad {\text{unicorn}\atop CN^{+}}}{TV^{-}\backslash VP^{-}}}{VP^{-}}}{VP^{+}}}{S^{+}}$$

We have again forced a solution in which 'unicorn' has positive polarity, which is precisely the result we want. The restriction of 'every' ordinarily has negative polarity, and its nuclear scope ordinarily has positive polarity, but in this case those polarities are flipped by the negation.

We should be able to proceed in this manner up to an arbitrary level of complexity, even when scope relations are introduced. For example, if 'every unicorn' could take wide scope over the negation here (it perhaps cannot do so in natural language, but let's suppose it were to do so), then 'unicorn' would find itself on a path with only one polarity flipper, and we would predict that 'unicorn' would be in a downward entailing environment.

The interest of this research should be obvious. The polarity marking in the syntax allows us to identify the upward and downward entailing environments simply by inspection of the syntactic forms. In addition, since the strategy can correctly assign polarity to elements that are embedded within negations and can also reflect relative operator scopes, the strategy can readily handle the kinds of relative clause constructions that exercised Ockham and subsequent natural logicians for centuries. In effect, we are on the path to possessing the Holy Grail that eluded the Medieval natural logicians.

3.2 Recent Work on Polarity Logics for Natural Language

Recent years have shown an explosion of work in polarity logics, with a number of interesting formal results as well as applications in natural language processing and cognitive psychology. We will get to the formal results shortly, but a comment on applications of the research is worth noting here.

Work on psychology has included Geurts (2003), Geurts and van der Slik (2005), and Szabolcsi et al. (2008). Geurts and van der Slik, for example, studied the processing loads created by mixing the polarity of quantifiers in a natural language sentence. What they found is that if the monotonicity profiles of two quantifier expressions are the same (both positive or both negative), they are equally easy/hard to process, all other things being equal. Sentences containing both upward and downward entailing quantifiers are more difficult than sentences with upward entailing quantifiers only. Inferences from subsets to supersets are easier than inferences in the opposite direction. (We will come back to the issue of processing loads in Chapter 13.)

Another line of empirical investigation can be found in Szabolcsi et al. (2008), who examine the proposal from Dowty (1994) that negative polarity markers and negative concord might be used to facilitate the identification of downward entailing (*dictum de nullo*) environments. (The study by Szabolcsi et al. suggests that the empirical evidence does not support this idea.)

Another context where polarity logics have application is in natural language processing, and in particular with the problem of Recognizing Textual Entailment (RTE)—a topic that speaks to the presence of entailment relations holding between components of a text, and that has application in natural language systems that process narratives. In particular, it has had application in understanding tasks and questions, and in machine translation (if you have ever been frustrated with online translation tools, their inability to identify such structures in discourse is one reason they are typically suboptimal). MacCartney and Manning (2009) developed an RTE system that includes monotonicity reasoning as the central component and runs on a monotonicity marking algorithm like the ones from Sánchez Valencia (1991) and Dowty (1994).[3]

Of principle interest to us in this book, however, is work done obtaining formal results for polarity-based natural logics, and the work that has served as the inspiration for our formal results in Chapters 6 and 7, including Bernardi (2002), Zamansky et al. (2006), van Eijck (2007), MacCartney and Manning (2009), Muskens (2010), Icard (2012), Moss (2012), Icard and Moss (2013), and Tune (2016). There is a lot to glean from this body of work, but two ideas are relevant to us here. The first idea is that one can perfectly well exhibit the soundness and completeness of natural logics (indeed polarity-based natural logics) as the formal results in Moss (2012) and Icard, Moss, and Tune (2017) show. While we will take a different approach to proving the soundness and completeness of our system (in Chapter 7 and in section A.2 of the Appendix), their work has served as an inspiration to us, since it showed that the meta-logical results that are so prized in the study of formal languages are also available to us in the study of Natural Logic, should we approach the subject matter with sufficient care and ingenuity.

The other key idea from recent literature that we want to focus on for now is the idea that perhaps the polarity markings do not belong in the lexicon, as they were introduced in the work of Sánchez and Dowty, for example, but that we might

[3] In addition to MacCartney and Manning (2009), work of interest on the application of polarity to natural language processing includes Angeli and Manning (2014), Bowman et al. (2015), and Abzianidze (2015).

introduce them "externally," in the sense that they can be introduced in the type theory.

The shift in approach toward polarity marking is interesting in a number of respects. On the one hand, it enjoys certain processing advantages, but it also opens the door to reasoning from inferential relations down to word meanings, suggesting a way to technically execute a project that has been advocated by Brandom (1998) and others, but subject to criticism from Williamson (2010) for its lack of technical execution:

> Fifteen years after the publication of his magnum opus, *Making it Explicit*, Brandom's semantic inferentialism remains largely programmatic, unlike more orthodox semantic theories based on truth and reference. If you want an explicit theory of how some particular linguistic construction contributes to the meanings of sentences in which it occurs, the inferentialist is unlikely to have one. Better try the referentialist.

Whether this inferentialist idea is driving current research on polarity logic is unclear, but it is certainly in the background (and in some cases the foreground) of some research as the following quote from Moss (2005) suggests.

> If one is seriously interested in entailment, why not study it axiomatically instead of building models? In particular, if one has a complete proof system, why not declare it to be the semantics? Indeed, why should semantics be founded on model theory rather than proof theory?

We will come back to this issue in Chapter 13, but for now suffice it to say that our approach is going to be that neither syntax nor semantics can be fundamental, and that they can and should proceed in parallel. With respect to word meanings, we are going to approach matters in a more bottom-up way, starting with a referential account of word meanings and then exploring their inferential roles. We do support recent efforts to make the top-down approach rigorous however, and concede that at a minimum they dull the force of some of Williamson's criticism.

However, perhaps because we have a preference for proceeding in a bottom-up fashion, we have certain concerns about the nature of polarity markings and their motivations. Formally, of course, once the polarity markings and rules are established, the system is quite elegant, but the problem is that the polarity markings are not independently motivated from a truth-conditional perspective; the markings are, in effect, otherwise semantically inert. If you hand off a linguistic structure to the semantics, the semantics will have a use for every bit of information in that structure except for those polarity markings. It will assign set-theoretic denotations to the nouns, verbs, adjectives, determiners, etc., and it will be using the branching tree structure to determine the order in which the elements are interpreted. It will interpret negations in the usual way and so forth. But it will have no use for the polarity markings. The negative and positive markings would not contribute in any way to the truth conditions of the resulting grammatical form. They are only useful to the deductive mechanism.

This is troubling since they constitute syntactic elements that are all-important for defining entailment relations, but that make no apparent contribution to the truth conditions of the form. At best a wedge has been driven between entailment relations and truth conditions. At worst, this could be taken as evidence that the polarity marking strategy remains ad hoc (even if a much superior ad hoc strategy).

Even if we have moved the polarity marking from internal structures to more global rules, they are still there and they are motivated by a desire to get the inferences to work out correctly. There is nothing wrong with this per se, but it would be better if they could have motivations that are independent of entailment relations. This objection may be a little bit abstract, so in the next section we show a case where the account of directional entailingness is not ad hoc at all, but is integrated into independently motivated accounts of determiner meaning.

This will be a semantic account of determiners, and of course we are ultimately looking for a syntactic account of directional entailingness, but this detour through the semantics should give us an example of a motivated account of directional entailingness, and perhaps also some insights into what a well-grounded syntactic account might ultimately look like.

3.3 A Semantic Account of Directional Entailingness for Natural Language Determiners

Servicaeble *semantic* characterizations of natural language inferences have been available for some time. We know, for example, that the valid syllogisms of Aristotelian logic can be readily characterized using Venn diagrams. We can think of these Venn diagrams as being semantic models for characterizing the inferences of Aristotelian logic. When we rely on such models, the inferences do not flow by virtue of the argument forms, but rather by virtue of the structure of the abstract models of Venn diagrams. We can think of such diagrams as being a species of model-theoretic semantics. Similarly, we can think of truth tables as playing a similar role—characterizing inferences in terms of the notion of truth in a model (each assignment of truth values being a model). Venn diagrams work fine for characterizing Aristotelian inferences, and truth tables are excellent for characterizing inferences in propositional logic, but more complex logics require more sophisticated models. Recent work in model-theoretic semantics, in particular in generalized quantifier theory, not only gives us semantic (model-theoretic) accounts of the valid inferences for complex sentences with quantifier expressions, but it can do so in a way that illuminates the nature of *dictum de omni et nullo* environments.

Explaining how this works will require a bit of background information for those who are not familiar with generalized quantifier theory, for that theory is going to play a key role in the semantic characterization of polarity in natural language. Very simply, the idea is that determiners ('all', 'some', 'no', 'most', etc.) denote relations between sets. For example, we could have the following set-theoretic denotations for the basic determiners (here the term 'determiner' is being used very broadly—we may want to narrow that usage a bit later).

(11) a. $[\![\text{every}]\!](A, B)$ iff $|A - B| = 0$
b. $[\![\text{some}]\!](A, B)$ iff $|A \cap B| > 0$
c. $[\![\text{no}]\!](A, B)$ iff $|A \cap B| = 0$
d. $[\![\text{exactly two}]\!](A, B)$ iff $|A \cap B| = 2$
e. $[\![\text{most}]\!](A, B)$ iff $|A \cap B| > |A - B|$

There are many interesting discoveries that have been made within the generalized quantifier research program, and we will skip over most of those discoveries here. But one feature of generalized quantifier theory that will be of interest to us is that we can state—in set-theoretic terms—constraints on our theory of generalized quantifiers. One of the most interesting of those constraints is that determiner denotations are *conservative*. Informally, that means something like this: For every natural language determiner Det and a sentence of the form 'Det As are Bs', we can copy the restriction into the scope. Thus, we could just as easily say 'Det As are As that are Bs'. (We will have a lot more to say about conservativity in Chapter 9.)

This constraint on the class of natural language determiners is helpful, for it rules out many potential determiners that appear not to occur in natural language. For example, following an example discussed in Larson and Segal (1995), consider a determiner 'nall', which means something on the order of "All but the As are Bs."[4] To follow their illustration, we might point at a group of geometric figures (triangles, circles, squares) of which all are striped except for the squares. We thus truly utter 'Nall squares are striped'. Conceptually this makes perfect sense, but it is an astoundingly robust generalization across human languages that non-conservative determiners don't obtain (although there are a handful of apparent counterexamples that we will discuss in Chapter 10).

[4] You might wonder why 'all but' can't be considered a determiner in natural language having the 'nall' semantic profile. The short answer is that 'all but' is a syntactically complex construction. The restriction is probably a tacit noun (e.g. 'things', with the 'but'-phrase modifying the restriction—e.g. 'things, with the exception of squares'.) The key point is that in such a case it is not the determiner that is identifying the relevant class of objects, but the complex predicate itself.

This isn't to say we can't define such a determiner set-theoretically. Our 'nall' can be defined as follows, where E is the universe of things.

(12) $[\![\text{nall}]\!](A, B)$ iff $|(E - A) - B| = 0$

There is considerable speculation about why natural language determiners are conservative. Keenan and Stavi (1986), for example, argue that the primitive determiners are conservative, and the other determiners are constructed from the primitives by Boolean operations, which preserve conservativity. This leaves open the question of why the primitives should be conservative—why isn't 'nall' a primitive determiner? We will return to this very interesting question after we have offered our constructive proposal, but for the moment we can leave with one observation about natural language determiners: When we say 'Det As are Bs', we are saying something about As. But the problem with determiners like 'nall' is that they are about other things (too)—as revealed by the set-theoretic definition in (12), anything at all (i.e. any member of the universe E) can be relevant when it comes to 'nall'.

As we noted, there is plenty to talk about with respect to generalized quantifiers; their study has been among the most productive avenues of research in semantics since the 1980s. This is not the place to discuss all of their properties; however there are two further very important topics in generalized quantifier theory that will be relevant to our discussion going forward.

The first topic relates to the expressive power of generalized quantifier theory. Set theory is a very powerful tool, perhaps too powerful for purposes of linguistic theory, since we would like our semantic theories to be constrained—they should model human languages but not all possible languages. Constraints such as conservativity give us one way of constraining the class of constructions that can be modeled by our semantic theory for natural language.

Barwise and Cooper (1981) offered an additional kind of constraint—one driven by the set theory they deployed to model generalized quantifiers: hereditarily finite set theory. This had the effect of reducing the class of languages that can be modeled as well. What that means is that certain kinds of possible determiners (e.g. 'uncountably many') are not plausible natural language determiners. This again raises questions about *why* natural language should be constrained in this way.

The second topic is in some ways the most surprising feature of generalized quantifier theory—its ability to model upward and downward entailing environments. Directional entailingness follows directly from the set-theoretic definitions of the determiners. For example, returning to the definition of 'some' above, we know simply from the properties of set theory that if $|A \cap B| > 0$ is true, then the substitution of A and B with supersets A' and B' respectively will result in a true formula. Likewise 'no' will allow the substitution of subsets for A and B, and 'every' will allow the substitution of a subset for *A* and a superset for *B*. This just follows from basic set theory.

The exciting insight to be gleaned from this work, in our opinion, is that we can now give a good *semantic* account of the *dictum de omni et nullo* environments; we now have some grasp on why the first position of 'every' should be downward entailing (*dictum de nullo*) and the second position should be upward entailing (*dictum de omni*).

Notice that the features of the set-theoretic accounts of the determiners which we appealed to in identifying the *dicta de omni et nullo* environments are independently motivated. The determiners are defined solely so as to get the truth conditions right. We would like to see a similar result in our syntactic accounts of directional entailingness. That is, we would like to be able to identify directional entailing environments by pointing to features of natural language *syntax* that are motivated solely by getting their contributions to the truth-conditions right.

Of course some might argue that this is too much to ask for. The monotonicity accounts offered by Sánchez and Dowty are as close as we are ever going to come to the Holy Grail. Or worse, some might argue that we don't need syntactic accounts of these inferences, and more, that syntactic accounts are simply unavailable when we begin talking about non-elementary quantifiers. But this sort of pessimism strikes us as unmotivated. It is certainly the case that we cannot successfully give the logical forms for sentences like 'most men are mortal' in *standard* first-order *calculus*,[5] but this does not close the door on *all* first-order syntactic accounts. If we want to pursue the natural logic project we need to study logics which allow formal (i.e. syntactic) accounts of the non-elementary quantifiers, and which do so in a way that allows us to cash out the notion of directional entailingness in a syntactic way. Indeed, as we will see in the next two chapters, we can identify formal languages in which directional entailing environments can be identified by independently motivated features of the syntax of the language.

[5] We follow Mendelson (1997) in reserving the term (first-order predicate) CALCULUS for a (first-order) theory without proper axioms.

4

L*: A First-order Polarity Logic for Generalized Quantification

In the previous chapter, we first looked at some syntactic attempts to encode polarity markings, and we concluded that the polarity markings seemed to lack independent motivation. Then we looked at a semantic account of natural language quantification in which directional entailingness flowed directly from the analysis of determiners and no unmotivated elements were called for. We ended the chapter asking whether syntactic accounts are available in which polarity marking is more deeply motivated and is not driven solely by the need to get the inferences to work out.

In this chapter we look at a formal language, L* ("L Star"), in which formal syntactic accounts are given for the determiners. The key driver for this language is that we want to give a syntactic account of the determiners—including the non-elementary determiners such as 'most', 'infinitely many', etc. Perhaps surprisingly, we will see that once the determiners are defined in L* in a way that gets their truth conditions right, the polarity markings (now understood as being in the scope of an even or odd number of negations) will flow directly from those definitions. The effect will be a result similar to what we saw in the case of generalized quantifier theory: If the analysis of the determiners is done correctly, the entailment relations will come for free. This, it turns out, is true whether the determiner analysis is semantic or syntactic.

Before we get into details, let's remind ourselves why this result will be important. If we are interested in pursuing the Holy Grail of Natural Logic, we ultimately need to show that entailment relations can be defined off of syntactic relations (perhaps two syntactic environments—the *dicta de omni et nullo*). But before we begin exploring this question in natural language, we will find it useful to show—or better yet, *prove*—that such a thing is possible. In this chapter we will see that such a proof is possible for L*. In Chapter 5 we will make some improvements to L* that will help prepare us to construct a similar analysis for the syntax of natural language (in Chapter 12).

It goes without saying that one of the key projects in contemporary philosophy of language has been to elucidate the underlying logical form of various natural language constructions. Among the constructions of central interest have been quantified sentences. So, for example, in contemporary logic, sentences containing

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović, Oxford University Press. © Peter Ludlow and Sašo Živanović 2022. DOI: 10.1093/oso/9780199591534.003.0004

the determiners 'all', 'a', 'no', and 'the' have been argued to have the logical forms indicated in (1).[1]

(1) a. All As are Bs.
$\forall x \colon A(x) \Rightarrow B(x)$

b. An A is a B.
$\exists x \colon A(x) \wedge B(x)$

c. No A is a B.
$\forall x \colon A(x) \Rightarrow \neg B(x)$

d. The A is a B.
$\exists x \colon A(x) \wedge \big(\forall y \colon A(y) \Rightarrow x = y\big) \wedge B(x)$

Things get interesting when we turn from these standard cases of quantification to cardinal quantifiers. First-order logic is expressive enough to provide logical forms of cardinals, but the complexity of the resulting formulas rises quickly. For a given cardinal n, the conjunction expressing distinctness of x_is has $n(n-1)/2$ members.

(2) a. At least one A is a B.
$\exists x_1 \colon A(x_1) \wedge B(x_1)$

b. At least two As are Bs.
$\exists x_1 \exists x_2 \colon (x_1 \neq x_2) \wedge (A(x_1) \wedge A(x_2)) \wedge (B(x_1) \wedge B(x_2))$

c. At least three As are Bs.
$\exists x_1 \exists x_2 \exists x_3 \colon (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3) \wedge$
$(A(x_1) \wedge A(x_2) \wedge A(x_3)) \wedge (B(x_1) \wedge B(x_2) \wedge B(x_3))$

The real concern comes with the so-called non-elementary quantifiers, and it has been claimed that no first-order logical form can be provided for a number of natural language determiners, including 'most', 'few', 'infinitely many', etc.[2] It is perhaps for this reason that most research on the logic of determiners (see e.g. Wiggins 1980; Barwise and Cooper 1981; Higginbotham and May 1981; Keenan and Stavi 1986 and references therein) thus concentrated on the model theory

[1] We follow the usual bracketing convention with the following (decreasing) precedence of connectives: $\forall, \exists, \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$. Additionally, the following convention is useful in logical forms of natural language expressions: a colon following a quantifier marks wide scope of the quantifier. In other words, a quantifier followed by a colon has the weakest precedence. Without the colon, (1a) would read $(\forall x\, A(x)) \Rightarrow B(x)$; under the colon convention, it stands for $\forall x\, (A(x) \Rightarrow B(x))$.

[2] While it is often remarked that there are no first-order representations of determiners like 'most', this statement must be qualified if it is to hold. For one, it holds only for first-order languages without any bells and whistles. So it does not apply to plural logic or infinitary logic—or a language with substitutional quantifiers such as L*.

of determiners, based on the theory of generalized quantifiers (Mostowski 1957), and either skirted the question of what the logical form of determiners may be, or assumed their logical forms to be trivial. The upshot of this focus in research has been that a number of determiner properties have been explored, but that these properties were studied as exclusively semantic phenomena.

In this chapter we first want to show that determiners and their properties can also be studied as features of their logical forms. Specifically, we shall (i) develop the language L* in which logical forms can be provided for determiners such as 'most', (ii) show that the logical forms provided allow us to syntactically characterize the property of directional entailingness, and thus (iii) see that the *dicta de omni et nullo* can be formally characterized in the language.

In the next chapter we are going to be offering a more elegant and advanced version of this proposal (to be called L**, "L Nova"), but as we've done throughout this book, we are describing the historical antecedent of that proposal, both to provide some historical context and because it affords us the opportunity to explain some key concepts in an intuitive way. In a later chapter, we will take up the issue of whether such a formal language can be used to shed light on the syntax of natural language expressions. (We will argue that it can!)

4.1 A Brief History of L*

When L* was initially developed in Law and Ludlow (1985) there were two motivations for the project, both of which were grounded in ideas being pursued in philosophy at the time. One motivation had to do with epistemology, and a concern, at that time, about our epistemic access to mathematical objects, and in particular to the objects of set theory.

This epistemological concern was being driven in part by the work of Benacerraf (1973), and to an extent by work of Field (1980). The question in each case was this: If we bear no causally mediated relation to set-theoretic objects (and how could we, given that sets are abstract objects), how do we have epistemic access to them? One solution to the problem (Field's solution at the time) was to try and dispense with the abstracta altogether. Another solution, due to work by C. Parsons (1983), was to think of syntactic structures as being vehicles that mediate our access to abstracta. Either solution called for a syntactic account of the behavior of so-called non-elementary quantifiers.

The second motivation was grounded in a dispute in the foundations of cognitive science that largely pitted Jerry Fodor against Jon Barwise on the nature of inference in cognitive science. On Fodor's view at the time (primarily in Fodor 1980), human inferential capacity was construed as a formal or syntactic capacity that was indifferent to whatever it was that the syntactic forms represented (if anything). Barwise (1989), on the other hand, was arguing that human inferential abilities were not formal at all, but rather semantic, and the idea was that we are in some sense attuned to the logical structure of the world in which we are situated. One motivating factor for Barwise was the observation that formal systems of sufficient complexity are incomplete, but he also believed that the same would be true of any formal attempt to characterize the behavior of generalized quantifiers and the properties of generalized quantifiers that we discussed in the previous chapter. Barwise leaned on generalized quantifier theory as an example of features of human inferential capacity that could not be given a formal account.

L* was thus an attempt to show that Field's epistemic worries, and Barwise's logical worries might be ameliorated. As an important side note, however, identifying upward and downward entailing environments (the *dicta de omni et nullo*) was not part of the goal of L*, nor was it even on the radar of the effort. As we will see, however, L* turned out to yield a natural way of syntactically identifying the *dicta de omni et nullo* environments. In other words, Law and Ludlow got damn lucky!

The basic idea underlying the construction of L* in Law and Ludlow (1985) is that cardinality and cardinality relations can be expressed as features of the syntax of certain languages. The syntax of the original first-order language L is enriched by introducing *subscripted (objectual) quantifiers*. The primary members of the class of *quantifier-subscript terms (QS terms)* are numerals. For example, given a monadic predicate A, the set-theoretic relation $|[\![A]\!]| \geq 5$ will correspond to the L* expression $\exists_{\geq 5} x\, A(x)$, which is understood as saying "there are at least five xs such that x is A." In other words, the language is designed to directly encode cardinality properties.

(3) a. (At least) five As are Bs.
    b. $\exists_{\geq 5} x\colon A(x) \wedge B(x)$

Next, the numerical subscripts may contain *substitutional variables*, which are quantified over by two *substitutional quantifiers*: the universal Π and the existential Σ. Because the domain of substitutional quantification consists of natural numbers, this allows us to introduce non-elementary quantifiers like 'infinitely many'.[3]

(4) a. Infinitely many As are Bs.
    b. $\Pi n \exists_{\geq n} x\colon A(x) \wedge B(x)$

How are determiners like 'most' handled? The final kind of QS terms introduced in L* are *primitive recursive functions (prfs)*. They are introduced to do the work that relative cardinality relations do in set-theoretic accounts. We can think of

[3] See Shapiro (1991, pp. 243–246) for a discussion on the expressive power of substitutional quantifiers.

them as operations on inscriptions—that is, on syntactic objects.[4] Crucially for our purposes, given $f(n) := \left\lfloor \frac{n}{2} \right\rfloor + 1$,[5] we can provide a logical form for 'most'.

(5) a. Most As are Bs.
 b. $\Pi n\colon \exists_{\geq n} x\, A(x) \Rightarrow \exists_{\geq f(n)} x\, (A(x) \wedge B(x))$

Other examples of L* formulas include the following. Obviously there are alternative ways to define each of these determiners. In (6)–(9), we could have simply utilized the first-order quantifiers of L, which are clearly equivalent to quantifiers subscripted by 1.[6] Formulas (10)–(12) have no L-counterpart.

(6) a. An A is a B.
 b. $\exists_{\geq 1} x\colon A(x) \wedge B(x)$

(7) a. No A is a B.
 b. $\neg\exists_{\geq 1} x\colon A(x) \wedge B(x)$
 c. $\forall_{\geq 1} x\colon A(x) \Rightarrow \neg B(x)$

(8) a. All As are Bs.
 b. $\forall_{\geq 1} x\colon A(x) \Rightarrow B(x)$

(9) a. The A is a B.
 b. $\exists_{\geq 1} x\colon A(x) \wedge (\forall_{\geq 1} y\colon A(y) \Rightarrow x = y) \wedge B(x)$

(10) a. More As than Bs are Cs.
 b. $\Sigma n\colon \exists_{\geq n} x\, (A(x) \wedge C(x)) \wedge \neg\exists_{\geq n} x\, (B(x) \wedge C(x))$

(11) a. More As are Bs than Cs.
 b. $\Sigma n\colon \exists_{\geq n} x\, (A(x) \wedge B(x)) \wedge \neg\exists_{\geq n} x\, (A(x) \wedge C(x))$

(12) a. (Only) finitely many As are Bs.
 b. $\neg\Pi n \exists_{\geq n} x\colon A(x) \wedge B(x)$

4.2 Directional Entailingness and Polarity in L*

Ludlow (1995) observed that L* had several interesting formal properties. First, and most importantly (for our current purposes), L* affords a simple formal characterization of *dicta de omni et nullo* environments. Ludlow's characterization

[4] We postpone the formal definition until later in this chapter. For introduction to prfs, see e.g. Boolos et al. (2002, §6).

[5] $\lfloor x \rfloor$ stands for the greatest integer less than or equal to x; for a positive number x, $\lfloor x \rfloor$ is the whole part of x. $f(n)$ thus equals the smallest integer greater than half n.

[6] The logical form in (9) applies to both singular and plural definite descriptions. In fact, L* turns out to be suboptimal for distinguishing between the two.

of directional entailingness relies, as we might expect, on the notion of *polarity*, and the characterization is defined over formulas in *L* canonical form*.

(13) An occurrence of a symbol of L* in a formula has

a. POSITIVE POLARITY iff it is within the scope of an *even* number of negations; and

b. NEGATIVE POLARITY iff it is within the scope of an *odd* number of negations.

(14) A formula of L* is in L* CANONICAL FORM iff

a. it is in *prenex normal form*, i.e. all the quantifiers occur at the front of the formula, and

b. its quantifier-free portion is in *disjunctive normal form*, i.e. it is a disjunction of conjunctions of (negations of) atomic formulas.

Given these definitions, it turns out that L* has a very interesting formal property: Directional entailingness is a direct reflex of whether an element has all positive or all negative occurrences when it occurs in a formula in L* canonical form. More formally, as proved in Ludlow (1995), the following property holds in L*.

(15) For a sentence *S* in L* canonical form, and an atomic formula α in *S*:

a. If all occurrences of α in *S* are *positive*, then α is in an *upward* entailing environment in *S*.

b. If all occurrences of α in *S* are *negative*, then α is in a *downward* entailing environment in *S*.[7]

To illustrate, consider some logical forms from above but in their L* canonical form.

(16) a. An A is B.

b. $\exists_{\geq 1} x : A(x) \land B(x)$

(17) a. No A is B.

b. $\forall_{\geq 1} x : \neg A(x) \lor \neg B(x)$

[7] As suggested in Ludlow (1995), it is possible to prove the other direction of the statement as well—that is, to show that upward/downward entailing environments are just the environments with all positive/negative occurrences—but *modulo logical equivalence*: if α is in an upward/downward entailing environment in *S*, then there is a sentence S', logically equivalent to *S* (perhaps *S* itself, perhaps not), such that α has only positive/negative occurrences in S'. We will discuss this further in section 5.4; the proof of the reverse statement will be presented in A.3.

(18) a. All As are Bs.
b. $\forall_{\geq 1} x\colon \neg A(x) \vee B(x)$

Notice that in each of the above cases, we can simply inspect the formula to see if there is going to be a downward entailing environment or an upward entailing environment. If the element is in the scope of an odd number of negations, it is a downward entailing (*dictum de nullo*) environment. If it is in the scope of an even number of negations it is in an upward entailing (*dictum de omni*) environment. So, for example, (18) tells us that 'all' is downward entailing in the first position and upward entailing in the second. (17) tells us that 'no' is downward entailing in each position.

These results also apply to non-elementary quantifiers. In the following cases, the elements that occur exclusively in the scope of negation turn out to be the elements that are in downward entailing environments. All we have to do is inspect the formulas for negations. Consider the examples below.

(19) a. Infinitely many As are Bs.
b. $\Pi n \exists_{\geq n} x\colon A(x) \wedge B(x)$

(20) a. (Only) finitely many As are Bs.
b. $\Sigma n \forall_{\geq n} x\colon \neg A(x) \vee \neg B(x)$

(21) a. Most As are Bs.
b. $\Pi n \forall_{\geq n} x \exists_{\geq f(n)} y\colon \neg A(x) \vee (A(y) \wedge B(y))$

By inspecting for negations, we know that both positions of 'infinitely many' will be upward entailing. So, from 'Infinitely many collies bark', 'Infinitely many dogs make noise' follows. Similarly, because both positions of 'finitely many' are in the scope of negation, those positions are in downward entailing environments. So, if 'Finitely many dogs bark', 'Finitely many collies bark loudly' will follow. But more importantly, the analysis for 'most' in (21) shows why the first (A) position of 'most As are Bs' is neither upward nor downward entailing—since A has both a positive and a negative occurrence in the logical form.[8]

Our exploded analysis with multiple occurences of the predicate A also calls our attention to some rarely mentioned (but see e.g. Sánchez Valencia 1991, pp. 96–99) logical inferences that are a feature of directional entailingness in L*. For example, the predicate A has both a positive and a negative occurrence in (21) (and so is neither in an upward nor in a downward entailing environment). However, in

[8] This is an oversimplification; non-uniform polarity of the occurrences of a predicate does not, by itself, entail that the predicate does not occur in a directional entailingness environment—because as mentioned in footnote 7, the reverse direction of the statement in (15) holds only modulo logical equivalence.

the structure revealed by the L* analysis, the open formula $A(x)$ has all negative occurrences and $A(y)$ has all positive occurrences. It follows that if we are careful to keep track of which open formula is in the scope of negation and which isn't, we can substitute accordingly following the *dicta de omni et nullo*. And indeed, as the following example shows, there does appear to be a hidden *dictum de nullo* environment, producing (22c), and there also appears to be a hidden *dictum de omni* environment, producing (22d).

(22) Most people sing. ($\sim$ If there are n people, then at least $^{n}\!/_{2}$ people sing.)

a. $\not\therefore$ Most $\overset{\downarrow}{\text{women}}$ sing.

b. $\not\therefore$ Most $\overset{\uparrow}{\text{animals}}$ sing.

c. $\therefore$ If there are n $\overset{\downarrow}{\text{women}}$, then at least $^{n}\!/_{2}$ people sing.

d. $\therefore$ If there are n people, then at least $^{n}\!/_{2}$ $\overset{\uparrow}{\text{animals}}$ sing.

One of the other nice properties of L* is that the upward and downward entailing environments are well defined for formulas of arbitrary complexity—including complex scope interactions between multiple quantifiers—the rocky shores on which the Medieval project foundered. For example, (23) would have the following two possible (canonical) L* representations corresponding to the two scope readings.

(23) Every man loves some woman.

a. $\forall_{\geq 1}x\colon M(x) \Rightarrow (\exists_{\geq 1}x\colon W(y) \wedge L(x,y))$
$\sim \forall_{\geq 1}x\exists_{\geq 1}y\colon \neg M(x) \vee (W(y) \wedge L(x,y))$

b. $\exists_{\geq 1}y\colon W(y) \wedge (\forall_{\geq 1}x\colon M(x) \Rightarrow L(x,y))$
$\sim \exists_{\geq 1}y\forall_{\geq 1}x\colon (W(y) \wedge \neg M(x)) \vee (W(y) \wedge L(x,y))$

Interactions are a bit more interesting when we consider cases like (24). Here, in the case where 'every man' takes wide scope, it has the L* canonical form in (24a). In the case where 'no woman' takes wide scope, it has the L* canonical form in (24b). (To see where (24b) comes from, think of the negation in the definition of 'no' being passed down through $\forall_{\geq 1}x$.)

(24) Every man loves no woman.

a. $\forall_{\geq 1}x\colon M(x) \Rightarrow \forall_{\geq 1}y\colon W(y) \Rightarrow \neg L(x,y)$
$\sim \forall_{\geq 1}x\forall_{\geq 1}y\colon \neg M(x) \vee \neg W(y) \vee \neg L(x,y)$

b. $\forall_{\geq 1}y\colon W(y) \Rightarrow \neg\forall_{\geq 1}x\colon M(x) \Rightarrow L(x,y)$
$\sim \forall_{\geq 1}y\exists_{\geq 1}x\colon \neg W(y) \vee (M(x) \wedge \neg L(x,y))$

This correctly predicts that 'man' will be in a downward entailing environment when 'every man' has wide scope, but in an upward entailing environment when it has narrow scope.

Turning to the case of determiners which are embedded in relative clauses, consider the sentence (25), where we would have the following two canonical L* representations.

(25) No man who knows every dog is a veterinarian.

a. $\forall_{\geq 1}x\colon (M(x) \wedge \forall_{\geq 1}y\colon D(y) \Rightarrow K(x,y)) \Rightarrow \neg V(x)$
$\sim \forall_{\geq 1}x\exists_{\geq 1}y\colon \neg M(x) \vee (D(y) \wedge \neg K(x,y)) \vee \neg V(x)$

b. $\forall_{\geq 1}y\colon D(y) \Rightarrow (\forall_{\geq 1}x\colon (M(x) \wedge K(x,y)) \Rightarrow \neg V(x))$
$\sim \forall_{\geq 1}y\forall_{\geq 1}x\colon \neg D(y) \vee \neg M(x) \vee \neg K(x,y) \vee \neg V(x)$

Notice that this correctly predicts that the predicate 'dog' will be in an upward entailing environment when 'every dog' has narrow scope, and in a downward entailing environment when 'every dog' has wide scope. Thus L* gives us a handle on the problem that quantifier scope ambiguity posed for Medieval accounts of *dictum de omni* and *dictum de nullo* environments. This property is also gratifying since dealing with multiple quantifiers and scope has been a subtle problem for polarity marking logics.[9]

Most importantly, notice that as in generalized quantifier theory, the elements appealed to in defining directional entailing environments are independently motivated. The negations are not otherwise semantically inert. Indeed if a negation is added to the *B* position of 'all', the result is a determiner with the meaning 'no'. Thus L* shows that, at least for certain formal languages, it is possible to have a syntactic account of directional entailingness that does not require otherwise unmotivated and inert monotonicity markings.

In this chapter we want to keep our principal focus on entailment relations, but a parenthetical remark is worth making here. In Chapter 11 we are going to take a close look at some of the linguistic consequences of our view, and one of the phenomena we intend to investigate is that of negative polarity items (NPIs). These are expressions like 'any' and 'even', which are typically licensed by negation (as in 'I don't have any money'), except a more general hypothesis has been that they are licensed by downward entailing environments. That hypothesis doesn't work, however, as demonstrated by 'most people that steal any money regret it'. Our proposal will be that the key licenser isn't downward entailingness, but simply the presence of at least one negative occurence. Again, we will return to this and related questions in Chapter 11.

[9] See Sánchez Valencia (1991, ch. 6) for discussion.

4.3 Generalizing L* Results

One might wonder if the result achieved in the previous section—showing the relation between polarity and directional entailingness—is an isolated feature of L*, or whether it carries over into a larger class of systems. And one might also wonder if that result, relying as it does on conversion of logical forms into the canonical form, is optimal in the sense of being stated as generally as possible. In this section, we present the results from Živanović (2002) who showed that the relation between polarity and directional entailingness both transcends L*—it can be generalized to a particular infinitary logic (and to standard, finitary logic as well)—and does not need to rely on formulas being presented in that specific form.

Turning to the latter issue first, Živanović (2002) proves that the restriction to L* canonical form is unnecessary. We needn't convert formulas to prenex form. The formal characterization proposed for L* canonical form is valid for any L* formula limited to Boolean connectives ¬, ∧, and ∨. Furthermore, the validity extends to any first-order language using only Boolean connectives (as we will see below, the language may even be infinitary), as long as it introduces no other syntactic elements "disrupting" polarity, like a quantifier $\exists_{\leq 5}x$, which could be glossed as "there are at most five *x*s." What this suggests is that we can get similar results for many systems that are not in L* canonical form, and that we can seek out similar results for languages that are syntactically closer to natural language.

While L* was originally (Law and Ludlow 1985) envisioned as a language with substitutional quantifiers, its formulas can be reinterpreted as a notational variant of a certain class of formulas of an infinitary logic. The formulas of L* can be mapped into an infinitary formal language by two (recursively applied) rules. First, the interpretation of an existential[10] quantifier subscripted by a numeral (and containing no L* extensions in the scope) is defined via translation into the original language L. $\exists_{\geq n}x\,\varphi$ introduces a string of n objectual existential quantifiers (one for each "member" of x), assures the distinctness of the variables they bind, and asserts that φ holds for each of the variables. Below, $v_1, \ldots, v_n$ are the first n L-variables which do not occur in φ, and $A(v_i/x)$ is the result of substituting v_i for t in φ.

(26) $$\exists_{\geq n}x\,\varphi \quad \rightsquigarrow \quad \exists v_1 \ldots \exists v_n \colon \bigwedge_{\substack{i,j=0 \\ i \neq j}}^{n} v_i \neq v_j \wedge \bigwedge_{i=1}^{n} \varphi(v_i/x)$$

Next, we need to reinterpret the universal substitutional quantifier. Given an L* formula φ and substitutional variable n, $\Pi n\,\varphi$ is true iff $\varphi(m'/n)$ is true for every

[10] Universal quantifiers $\forall_{\geq n}$ can be seen as the abbreviation of $\neg\exists_{\geq n}\neg$ and thus need no separate translation rule. The same holds for the existential substitutional quantifier Σ, which we define as the abbreviation of $\neg\Pi\neg$.

numeral m, where m' stands for the successor of m.[11] (Note that n is a variable and m is a numeral, i.e. a QS term. The $'$ in $\varphi(m'/n)$ ensures that 0 is not included in the domain of substitutional quantification.)

Given a (fixed) range of quantification and terms referring to all elements of the range, a universal quantifier can be translated to a conjunction. In the case of the substitutional quantifier Π, both the requirements are fulfilled, as its range of quantification are the positive integers.

$$(27) \quad \Pi n\, \varphi \quad \sim \quad \bigwedge_{m=1}^{\infty} \varphi(m/n)$$

The resulting conjunction being infinite, it suggests that the analysis properly belongs to the realm of infinitary logics. What does this mean? Let's say $L_{\alpha\beta}$ refers to a language allowing conjunctions and disjunctions of cardinality less than α and strings of quantifiers of cardinality less than β. It is easy to see that L* is translatable into a fragment of $L_{\omega_1\omega}$, the infinitary language allowing countably infinite conjunctions and disjunctions, and finite strings of quantifiers. The conjunction in (27) is countably infinite, i.e. of cardinality $\omega < \alpha := \omega_1$. The translation of subscripted objectual quantifiers in (26) always produces a finite string of quantifiers, so we may set $\beta := \omega$.

Extending the results to $L_{\omega_1\omega}$ is an important result because $L_{\omega_1\omega}$ is a well-studied and reasonably well-behaved logic; for example, it is the only complete infinitary logic. And while L**, the successor of L* we will develop in the next chapter, will not be grounded in infinitary logic, the results about $L_{\omega_1\omega}$ are important because the generalization adds some heft to the original claim about syntactic accounts of directional entailingness in L*. It isn't some flukey result that stems from an outlier formal language. In fact, it is a result that generalizes to a very large class of languages that are well known and have impressive expressive power.

The point about expressive power deserves emphasis. One of the claims of the advantage of generalized quantifier theory was its ability to express a very large number of logical relations. Perhaps all of them, if Sher (1991) is correct. By some accounts, generalized quantifier theory has the expressive power of hereditarily finite set theory. By grounding our formal language in $L_{\omega_1\omega}$, we show that L* (or a suitable extension thereof) can have an equivalent expressive power.

Of further interest, Živanović (2002) proves that conservativity, a semantic property of determiners central to much work on generalized quantifiers, has a

[11] Following the common practice of literature on prfs, the official representation of numerals in L* uses a unary numeral system. A numeral n is represented as a 0 followed by n strokes ($'$). For example, 0, 1, and 5 are represented by 0, $0'$, and $0'''''$, respectively. The unary system is chosen is it facilitates the definition of one of the basic prfs, the successor function $S(n) := n'$.

syntactic reflex in $L_{\omega_1\omega}$ as well: restrictedness. We will go into this in some detail (but with L** in mind) in Chapter 9 and in section A.4 of the Appendix, but let us note that this work will eventually lay ground for the development of Ludlow's (2002) idea that explicit quantificational operators may be eliminated from the formal language, which in turn facilitates an interesting linguistic application of L**, where the LF of the Minimalist Program is seen as a particular encoding of L** formulas—an idea which was explored in Živanović (2007) and which will engage us in Chapter 12.

So far we have collected some strong results from our investigation of L*. We have offered a language in which there is a provable connection between polarity markings and directional entailingness, and shown how those polarity markings fall out directly from an attempt to get the logical form of determiners right, and that the polarity markings need be nothing more than the presence of even and odd numbers of negations in those determiner definitions. Finally, we've shown that these results generalize to a prominent class of infinitary languages (which is a nice result even if not crucial for our proposal).

These are interesting results, but there are some outstanding concerns about L*, and in particular its fidelity to the syntax of natural language. We turn to these concerns in the next section, and thus set the stage for the introduction of L** in Chapter 5.

4.4 From L* to L**

In the previous sections of this chapter, L* delivered quite a bit for us, including an existence proof for a language in which directional entailingness can be syntactically identified by the presence or absence of negations (our polarity markers). That's a surprising result, but we think we can do better than L*, for L* is not the optimal vehicle for a thesis about natural language.

Since its inception, the L* project has straddled the borders between philosophy, linguistics, and mathematics. However, even if the initial motivations for its development were philosophical, the focus has gradually shifted toward linguistics. The changes from L* to its successor L** ("L-nova") are thus motivated in large part by linguistic considerations. L**, which will be introduced in the following chapter, differs from L* by dropping QS terms, divorcing quantifiers and numerals, and receiving a direct (essentially mereological) interpretation. The rest of this section is dedicated to presenting several considerations motivating the forthcoming changes.

The linguistic considerations are as follows. We would like our logical results to transfer in a smooth way to natural language constructions. This means that the logical enterprise should be simple when and where natural language is simple. We want to achieve an isomorphism between the two. Such isomorphisms are not always obvious, since there are open questions about the structure and complexity of natural language, but some logical complexity seems over the top for accounts of the syntax of natural language. One place where this arises is in the introduction of QS terms in the language L*. Let's take a closer look at this issue.

As stated in the previous section, QS terms are built from numerals and substitutional variable symbols by repeated application of primitive recursive functions, whose definition is given below (Boolos et al. 2002, §6).

(28) A function is a PRIMITIVE RECURSIVE FUNCTION (PRF) if it is either one of the basic prfs (zero, successor, projections) or obtained from other prfs by composition or primitive recursion.

ZERO $z(n) := 0$

SUCCESSOR $S(n) := n'$

PROJECTION For any n and $1 \leq i \leq n$: $\mathrm{id}_i^n(x_1, \ldots, x_i, \ldots, x_n) := x_i$

COMPOSITION Given prf f (of arity m) and prfs g_i (of arity n) for $1 \leq i \leq m$, h defined below is a prf (of arity m).

$$h(x_1, \ldots, x_n) := f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

PRIMITIVE RECURSION Given prfs f (of arity n) and g (of arity $n + 2$), h defined below is a prf (of arity $n + 1$).

$$h(x_1, \ldots, x_n, 0) := f(x_1, \ldots, x_n)$$
$$h(x_1, \ldots, x_n, y') := g(x_1, \ldots, x_n, y, h(x_1, \ldots, x_n, y))$$

By using primitive recursive functions on subscripts, L* avoided the appeal to set theory and set-theoretic operations. The work was done on a purely formal level, by repeatedly reducing the subscripts of objectual quantifiers until reaching a quantifier subscripted by a numeral. For example, the subscript of the quantifier $\exists_{\geq n'} x$ is in fact $S(n)$ (the successor of n). When n is set to, say, $0'''$ (i.e. numeral 3), a single reduction step transforms it into $0''''$ (i.e. numeral 4).

In the same fashion, in the logical form of 'most', the subscript of the quantifier $\exists_{\geq f(n)} x$, where $f(n) = \left\lfloor \frac{n}{2} \right\rfloor + 1$, is ultimately reduced into numeral $0''''$ (4) when n is set to the numeral $0'''''''$ (7). However—and now the trouble starts—f is not really defined as $\left\lfloor \frac{n}{2} \right\rfloor + 1$. This is just shorthand, employing familiar mathematical notation. (In fact, in the computation of f using this definition, the intermediate result of division by 2 might even not be an integer.) One possible proper definition of f is the following (adapted from Corneli 2013). (You don't need to read it, just note its length.)

(29) a. $\mathrm{const}_1(n) := S(z(n))$
b. $\mathrm{add}(m, 0) := \mathrm{id}(m), \mathrm{add}(m, n') := S(\mathrm{add}(m, n))$
c. $\mathrm{mult}(m, 0) := z(0), \mathrm{mult}(m, n') := \mathrm{add}(m, \mathrm{mult}(m, n))$
d. $\mathrm{sub}_1(0) := z(0), \mathrm{sub}_1(n') := S(\mathrm{sub}_1(n))$
e. $\mathrm{sub}(m, 0) := \mathrm{id}(m), \mathrm{sub}(m, n') := \mathrm{sub}_1(\mathrm{sub}(m, n))$
f. $\mathrm{diff}(m, n) := \mathrm{add}(\mathrm{sub}(m, n), \mathrm{sub}(n, m))$
g. $d_0(0) := \mathrm{const}_1(0), d_0(n') := z(d_0(n))$
h. $\mathrm{sgn}(n) := \mathrm{sub}(\mathrm{const}_1(n), d_0(n))$
i. $\mathrm{rem}(0, n) := z(n), \mathrm{rem}(m', n) :=$
$\mathrm{mult}(\mathrm{sgn}(n), \mathrm{mult}(S(\mathrm{rem}(m, n)), \mathrm{sgn}(\mathrm{diff}(S(\mathrm{rem}(m, n)), n))))$
j. $q(0, n) := z(n),$
$q(m', n) = \mathrm{mult}(\mathrm{sgn}(n), \mathrm{add}(q(m, n), \mathrm{sgn}(\mathrm{diff}(S(\mathrm{rem}(m, n)), n))))$
k. $f(n) := S(q(n, S(S(z(n)))))$

There is nothing wrong with this definition from a mathematician's viewpoint. Viewed from the linguistic or cognitive perspective, however, the definition is simply too complex—particularly in the context of recent work in linguistics that prizes minimal resources and minimal effort. In this instance, for one, the computation consists of (too) many steps (we here sensibly refrain from trying to compute how many steps are required). Second, it is quite mysterious how such a complex function comes to be a part of the meaning of a determiner—even if, as we shall see in the next chapter, this determiner is cross-linguistically relatively rare. It would be one thing if we had to be taught the meaning of 'most' at great effort, but this is just not the case. Why so much logical and syntactic labor for a determiner that is used with ease? Finally, look at (29k) and suppose we strip the final layer (S) off of f. We get a simpler prf, let's call it f'. Should we not now predict that there must be (at least in some languages) a determiner like 'most' but with f' substituted for f, i.e. a version of 'most' meaning "more than half less one" instead of "more than half"? (And as f' is simpler than f, we might even predict that this determiner is a bit more commonplace than 'most'.) Of course, we get no such determiner.

Fortunately, as we will see in the develpment of L**, the logical form for our crucial example, the determiner 'most', can be provided even in the absence of prfs, if the language contains predicate "overlap," which—being one of the essential mereological predicates (Link 1983; Simons 1987)—is arguably needed for a full-blown analysis of natural language anyway. We will provide the details of this idea in the next section.

Law and Ludlow (1985) used QS terms not only to represent 'most', but fractions in general. However, the proposed logical form did not hint at how the syntax of the expression such as 'two-thirds' contributes to the construction of the QS term.

(30) More than two-thirds of As are Bs.

We believe that a more suitable analysis can be developed in L**, by recognizing the combination of distinct ordinal numerals and fraction denominators in natural language, rather than a single mathematical function. To put it another way, not only does L* introduce too much complexity, at other times it ignores actual complexity in the syntax of natural language—fractional expressions in natural language being a case in point.

The next issue concerns the tight coupling of quantifiers and numerals in L*. The only way to represent sentences containing coordinated numerals is by sentential coordination, as shown in (31b). Divorcing numerals from the quantifiers—we will do this in L** and (31c) previews the logical form that arises in that language—avoids the necessity of the sentential coordination analysis and might therefore better fit the syntax of natural language, both in the terms of the syntactic position of the numeral (among adjectives) and in the fact that natural language numerals are fundamentally predicates—and when not, they are simple referring expressions, without any clear connection to quantification. For example, they can refer to abstract objects, as in 'two plus two equals four', or act as labels for concrete objects, as in 'bus 6'. Whatever metaphysical concerns we might have about numbers *qua* mathematical objects, that should not color our theory of the syntax of natural language—or in any case it should not lead us to analyses that are wildly implausible from a syntactic point of view (Rothstein 2017).

(31) a. Three or four dogs are barking.
b. $\exists_{\geq 3} x\,(D(x) \wedge B(x)) \vee \exists_{\geq 4} x\,(D(x) \wedge B(x))$ (L*)
c. $\exists x\colon ((\underline{3}(x) \vee \underline{4}(x)) \wedge D(x)) \wedge B(x)$ (L**)

Finally, and perhaps most importantly, L* really doesn't line up with the structure of natural language determiner phrases well. In part this is due to the fact that the language was born as a response to generalized quantifier theory, which tends to treat numerical expressions as determiners. From a syntactic point of view, numerals are not determiners at all. The reality is that they are closer to predicates, and linguists have treated them as functional heads and have posited numeral phrases, etc. What this means is that if we want to prepare our theory to be a useful tool for the analysis of natural language, we need to consider some modifications.

Clearly we need a language that is not only closer to the structure of natural language, but better equipped to handle number expressions and plurals. Thus, in the next chapter we develop the details of the formally superior and more linguistically plausible langauge L**. Then, in Chapters 6–8 we will explore the proof theory of this language. In Chapters 9 and 10, we will take a deep dive into the notions of conservativity, restrictedness, and quantification in natural language, and then in Chapters 11 and 12, we will use L** to achieve some important results when we use its structure as a guide for insights into the structure of natural language syntax.

5

L**: A First-order Polarity Logic with Plural Domains

So far in this book we have been following a thread of investigation that began with Medieval Natural Logic and its project to reduce logical inferences down to two basic rules—rules sensitive to positive and negative polarity environments (the *dicta de omni et nullo*). We then looked at some recent attempts to develop theories of polarity for natural language and found them lacking in independent motivation for the polarity markings. We saw, however, that polarity marking can be motivated in the formal language L*, where the polarity markings are simply a function of the negations introduced in the definitions of determiners in order to get their truth conditions correct. We also saw that we can prove a formal relation between polarity marking and directional entailingness in L*, and that those results can be extended to a familiar class of infinitary languages.

We expressed concern, however, that L* is not optimal as a guide for seeking similar properties in the syntax of natural language. L* evinces syntactic complexity where natural language does not, and it also is not a faithful guide to the syntax of numerical expressions in natural language. We concluded the previous chapter with the promissory note that we have a better option.

In this chapter we turn to that better option, L**, and lay out its principal features. In section 5.1, we present the language informally, with the goal of showing how it can represent the meaning of determiner 'most'; a somewhat more formal presentation follows in section 5.2. In section 5.3, we turn to the logical forms of determiners in L**, and we conclude the chapter by returning to the main thread of our discussion so far: We show that polarity and directional entailingness are related in L** as well—the *dicta de omni et nullo* environments can be read off of the syntactic structure simply by inspecting the logical forms for negations.

Before we set to work, however, we want to make a note on what kind of language L** is. We sometimes describe it as a *plural* logic, but that is a little bit descriptively loose. In section 5.3 we will officially define it as a singular first-order mereology. Mereology is a general theory of parthood, and both (first-order) mereology and plural logic are contenders for being the tool for the analysis of plurals and related constructions in natural language. There is a hot debate on which approach might be better, but we do not wish to take a stance on this issue in this book.[1]

[1] McKay (2006) offers several excellent arguments favoring plural logic over mereology as vehicle for logical forms of natural language.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović, Oxford University Press. DOI: 10.1093/oso/9780199591534.003.0005

Our decision to present L** as a singular first-order mereology is simply a matter of what we felt prepared to develop in sufficient detail. We know of no reason why the same results could not be achieved by using a genuine plural logic. Indeed, as far as we can see, all our claims remain valid if L** is rebuilt as a plural logic. Proving this, however, remains homework for us and/or our readers.[2]

5.1 Getting 'Most' out of the Language

Any full account of quantification in natural language must provide logical forms for sentences describing cardinality relations. In this respect, predicate *calculus* is not optimal. While it is possible to represent simple cardinal numerals in first-order calculus, these logical forms are built out of conjunctions (often very large conjunctions!) and do not map easily to the structure of natural language expressions. Furthermore, it is well known that more complex cardinality relations are simply inexpressible using first-order predicate calculus. The determiner 'most', usually taken to mean "more than half," presents a canonical example of this ineffability (Barwise and Cooper 1981). In the previous chapter we looked at one formal way to approach this problem (the language L*), but found it wanting in certain respects. We now turn to our alternative, L**.

The general idea behind L** is to implement cardinality relations using as few resources as possible. While it is certainly possible to implement these relations in first-order logic, as witnessed by the set theory, the question is whether the entire apparatus of set theory is required to achieve this goal. As we will see, it turns out that this is not the case. L** relies on two predicates, subcollection $\subseteq$ and quantity #, and the theory underlying these predicates is arguably much simpler than standard set theory.

In L**, we tackle the problem by first thinking about the proper treatment of numerical expressions. Rather than treat them as determiners (as in generalized quantifier theory) or as subscripts on quantifiers (as in L*), we want to think of numerals as behaving as *predicates* that express cardinality and cardinality relations. Following the work of many linguists, we can think of them as being the heads of numeral phrases hosted within the extended nominal projection. If this linguistic gloss is obscure, don't worry, because we will explain the linguistics in detail in Chapter 12. For now we want to focus on the formal properties of L**.

One way to think about L** is that it is a standard first-order formal language interpreted on a non-standard domain. In a linguistic application of standard

[2] In Chapter 4, we have mentioned that L* can be interpreted as a language of infinitary logic. The same holds for L**, but we don't want to go down this path. Initially, we proved our mathematical claims about L** using infinitary logic, but because the proofs can be pushed through without relying on infinitary logic, while yielding even more general and flexible results, we don't deploy infinitary logic in this book.

logic, we usually imagine the domain to consist of individuals: people and objects and (within Davidsonian approaches) also states, events, etc. In the L** approach, the individuals do not form the domain, but what we call the PRE-DOMAIN. The DOMAIN consists of COLLECTIONS of individuals from the pre-domain; if you prefer, you can call it the PLURAL DOMAIN.

A pre-domain has no inherent internal structure. In contrast, the plural domain is structured by the subcollection relation $\subseteq$, and our account of quantification in natural language will also make crucial use of two relations derivable from $\subseteq$: overlap $\circ$ and disjointness $\wr$; we show how they can be defined off of subcollection in section 5.2. Other useful relations include equality $=$ and proper subcollection $\subset$; these can be trivially defined off of subcollection as shown below.[3]

(1) a. SUBCOLLECTION: $x \subseteq y$ iff every member of collection x is a member of collection y.

b. EQUALITY: $x = y$ iff x is a subcollection of y, and y is a subcollection of x as well.

c. PROPER SUBCOLLECTION: $x \subset y$ iff x is a subcollection of y, but y is not a subcollection of x.

d. OVERLAP: $x \circ y$ iff collections x and y have at least one common member.

e. DISJOINTNESS: $x \wr y$ iff collections x and y have no common members, i.e. if they don't overlap.

As far as this book is concerned, we can be flexible in how we think of collections. They differ from sets, in that we take sets to be abstracta in a way that collections are not. However, for our purposes, there is no harm in viewing them, from a purely mechanical point of view, as somewhat similar to sets. The distinction only becomes important when we start to think about the metaphsyical commitments of the theory.

If you choose to think of collections in terms of sets, the subcollection relation can be thought of as analogous to the subset relation, and the overlap and disjointness relations can be seen as analogous to the existence and non-existence of a non-empty intersection, respectively. We must point out a crucial difference between sets and collections, though: While there is an empty set, there is no empty collection.

Taking the domain to consist of collections of individuals affords us a straightforward implementation of counting without the detour through the syntactically complex primitive recursive operations on inscriptions that we saw in L*. As

[3] Strictly speaking, the formal language builds formulas using the *predicates* 'subcollection', etc., and these predicates can be model-theoretically interpreted as certain *relations*, which are—not surprisingly—called 'subcollection', etc., as well. Most of the time, this fine distinction is irrelevant for our discussion, so we sometimes (like above) allow ourselves some leeway in the usage of terms 'predicate' and 'relation'.

hinted earlier, numerical modifiers will come into the picture as predicates which constrain the cardinality of the collections we quantify over.

One way to approach this would be to think of cardinal numerals as predicates applying directly to collections and write $3(x)$, for example, to assert that x is assigned a collection consisting of three members. This idea turns out to be overly simple, for both formal and linguistic reasons. We will not be treating numerals as predicates holding of objects in general, but only of numbers. Specifically, we assume that the domain contains (natural) numbers and that, for example, the predicate '3' is true of (and only of) the number three. Regarding the number of individuals in a collection, we introduce the QUANTITY/COUNTING predicate # associating numbers with collections of individuals: $\#(n, x)$ is true iff n is the number of individuals in collection x. (We refer to the first and the second argument of # as the NUMERICAL and the OBJECTUAL argument, respectively.)

We could think of numerals as being names of numbers and set the value of n using the equality predicate, as in $n = 3$, but here we opt for treating the numerals as being primitive predicates, as in $3(n)$, leaving open questions whether the numeral is a simple predicate or is perhaps a name-like predicate, familiar from predicational theories of proper names. Thus we get an analysis like the following (in Chapter 12 we will discuss how to understand this analysis in terms of traditional linguistic structures).

(2) a. Three dogs are barking.

b. $\exists x\colon \exists n\,(3(n) \wedge \#(n, x)) \wedge D(x) \wedge B(x)$

There is a collection x such that: there is a number n such that n is three and n is the number of individuals in x, and x are dogs and x are barking.

Note that while n above must be three, and $\#(n, x)$ asserts that the collection x contains exactly n, i.e. three individuals, formula (2b) is nevertheless true in a model where more than three dogs are barking, because nothing forces us to assign to x the collection of all barking dogs; assigning any collection of three barking dogs will do. In other words, (2b) provides the "at least" reading of sentence (2a). This is the reading often argued to be basic (see e.g. Keenan and Stavi 1986; Ionin and Matushansky 2006), but note that opinions differ (e.g. Geurts 2006).

The final detail about the interpretation of L** formulas concerns the interpretation of non-logical predicates such as D and B above. While we will continue to gloss these simply as 'dog' and 'bark', it is important to note that these predicates don't mean 'being a dog' and 'being a thing that barks', for the simple reason that our domain consists of *collections* of individuals, and the collections can't be dogs and they can't bark. A more appropriate gloss of D is 'consists of dogs', or if we understand L** as a plural logic, simply 'are dogs'; and B should be glossed 'consists of things which bark', or perhaps simply 'are barking'. Formally, this detail will show up in the assumption that non-logical predicates of L** are *distributive*, in the sense that if a predicate is satisfied by some collection x, it is also satisfied by every subcollection of x (including singletons, i.e. collections consisting of a single individual).

Of course the real question is whether L**, equipped with the above-listed predicates applying to collections of individuals, can provide a logical form for sentences containing 'most'. It can! We provide two logically equivalent forms below, one deploying a negated existential quantifier $\exists y$, the other deploying a universal quantifier $\forall y$ with a negated scope. The logical forms also deploy the quantity predicate # and the overlap predicate $\circ$ (or its negation, the disjointness predicate $\wr$).

(3) a. Most dogs are barking.

b. $\exists x\colon (\exists n\colon \#(n,x) \wedge \neg(\exists y\colon \#(n,y) \wedge D(y) \wedge y \wr x)) \wedge D(x) \wedge B(x)$

There is a collection x of dogs of cardinality n such that there is no collection y of dogs of cardinality n disjoint from x. xs are barking.

c. $\exists x\colon (\exists n\colon \#(n,x) \wedge (\forall y\colon \neg(\#(n,y) \wedge D(y)) \vee y \circ x)) \wedge D(x) \wedge B(x)$

There is a collection x of dogs of cardinality n such that every collection y of dogs of cardinality n overlaps with x. xs are barking.

To see that the formulas given above come out as true when most dogs are barking, consider a situation where five out of nine dogs are barking. Let's say that x stands for the collection of all barking dogs, as depicted in Figure 5.1. This collection then contains more than half of all the dogs; in the given context, five. Having fixed the five dogs in x, it is clearly impossible to find five more dogs; in general, we cannot find a collection of dogs y of the same size as x, i.e. containing more than half the dogs, which would contain no dog from x. This helps us intuitively understand (3b); for (3c), observe that any two collections of dogs containing more than half dogs each will necessarily overlap.[4]

In a similar fashion, we can see that the formulas will be false when it is not the case that most dogs are barking. Consider a situation where five out of twelve dogs are barking. Here, we will show that no choice of x satisfies the truth conditions given in the scope of $\exists x$. Clearly, x must be some collection of barking dogs. But

[4] The reasoning outlined in the main text is related to a well-known argument form shown in (i)—in both cases it is crucial that any two collections of As comprising more than half of As will overlap. It is perhaps worth remarking that the properties of the predicates we deploy in the logical form of 'most' (we will axiomatize them in the following section) are crucial for a successful derivation of argument (i). That is, neither classical deductive systems nor our Dynamic Deductive System (which we will develop in Chapters 6 and 7) can derive this argument on their own, without recourse to the axioms of mereology and counting. In fact, the same holds even for some simpler deductions such as (ii). (We cannot simply use the *dictum de omni* on a logical form such as (2b) because $3(n)$ does not entail $2(n)$.)

(i) a. Most As are Bs. b. Most As are Cs. c. ∴ Some Bs are Cs.

(ii) a. (At least) three As are Bs. b. ∴ (At least) two As are Bs.

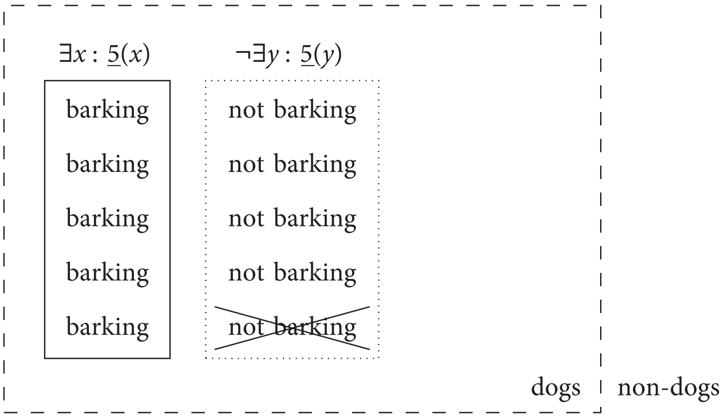


Figure 5.1 Context: Five out of nine dogs are barking

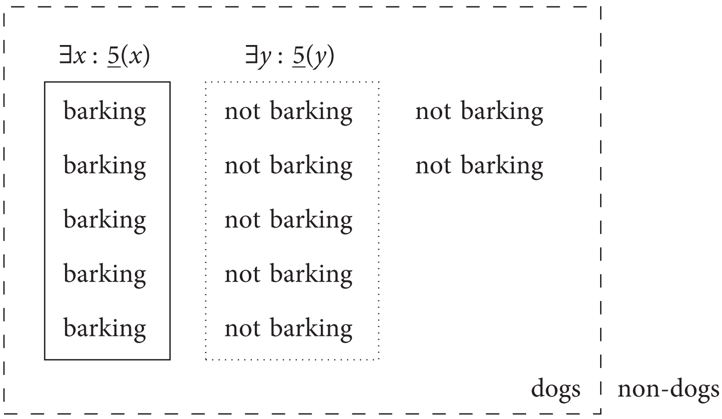


Figure 5.2 Context: Five out of twelve dogs are barking

even when x stands for all the barking dogs, as depicted in Figure 5.2, this amounts to less than or exactly half the dogs—in the given context, five. It is therefore easy to find five more dogs; in general, it is possible to find a collection y of (non-barking) dogs of the same size as x which contains no dog from x.

Here is the critical feature. In our approach, the “more than half” meaning of ‘most’ arises from the simple tools of predicate logic, without requiring the complex apparatus of set theory. Our formulas for ‘most’ rest on two instances of quantification over collections, one embedded in the other, related by two predicates, the quantity and the overlap/disjointness predicate. Furthermore, the first of these predicates seems an unavoidable feature of any account of natural language, and the second one is part and parcel of mereology, a predicate logic theory influentially argued by Link (1983) to be the appropriate tool for dealing with plurality and related phenomena in natural language.

In contrast to the paucity of resources deployed in our logical form for ‘most’, any account based on set theory deploys the full might of this cornerstone of modern mathematics. For example, a common generalized quantifiers rendering of ‘most’, shown below, deploys the cardinality function $|\ldots|$ and the relation $>$ “greater than,” which might seem innocuous at first sight, but we should remember that the set-theoretic definitions of cardinality and cardinality relations are fairly involved (see e.g. Vaught 1995).

(4) $[\![\text{most}]\!](D, B) = |D \cap B| > |D - B|$

Of course, we do not wish to imply that there is no theory—i.e. axiom system—behind L** predicates such as # and ∘. We merely wish to stress that this theory is arguably much weaker than the (standard) set theory emanating from the familiar Zermelo-Fraenkel axiom system. In the following section, we provide a semi-formal presentation of the L** language and the underlying theory.

5.2 L** Theory

If we had to describe L** in a sentence, we would most likely say that it is a relational Boolean first-order distributive mereology with counting. The purpose of this section is to explain what we mean by this, and to list the axioms underlying the theory of L** (even if the precise formulation of these axioms will play little part in the rest of the book). We will avoid the systematic formal definition of the syntax and the interpretation of L**—all of that is completely standard and can be found in any introduction to first-order logic (we find Mendelson 1997 a good source). We will instead take the time to walk, somewhat systematically, through the specifics of L** which make our approach tick.

In section 5.4, we will present a syntactic characterization of directional entailingness using polarity, and in Chapter 9, we will show that restrictedness is a syntactic reflex of conservativity. It is crucial for the validity of these correspondences between syntax and semantics (which will be proven in sections A.3 and A.4 of the Appendix) that the formal language employs only Boolean propositional connectives. We therefore admit only the following propositional connectives into L**: conjunction $\wedge$ “and,” disjunction $\vee$ “or,” and negation $\neg$ “not.” Any other propositional connectives must be defined as abbreviations. In particular, we take material implication $\varphi \Rightarrow \psi$ to be an abbreviation for $\neg\varphi \vee \psi$.

It is perhaps also worth noting that we see conjunction and disjunction as binary connectives, as this facilitates the interface to natural language (see Chapter 12). In the Appendix, however, we will take a somewhat broader view and assume that these connectives can take any number of operands. This will make the proofs more general, while of course covering the binary case.

Next, L** is a relational language, i.e. it contains no operation symbols. Simply put, terms of L** can only be variables and constants. This feature is not crucial for the above-mentioned syntactic characterizations of semantic properties; we adopt it because, first, we have no use for operation symbols in our account, and second, it simplifies the rendering of L** formulas in natural language syntax, an issue we will take up in Chapter 12.

Despite the fact that L** can represent the meaning of non-elementary determiners such as 'most', the language contains only the quantifiers of standard first-order logic: the existential quantifier $\exists$ "there is," and the universal quantifier $\forall$ "for each."

Most of the time, these *unrestricted* quantifiers will suffice for our purposes, but we will occasionally try to make logical forms easier on the eye by deploying RESTRICTED quantifiers. We emphasize, however, that these are nothing but abbreviations in our system: $\exists x\,[\varphi]\,\psi \;:\equiv\; \exists x\,(\varphi \wedge \psi)$, and $\forall x\,[\varphi]\,\psi \;:\equiv\; \forall x\,(\neg\varphi \vee \psi)$. The unrestricted quantifiers are unary; in $\exists x\,\varphi$ and $\forall x\,\varphi$, the quantifiers take a single operand, φ, called scope. In contrast, the restricted quantifiers are binary; operands φ and ψ in the abbreviations above are called the RESTRICTION and the NUCLEAR SCOPE, respectively. (We will have much more to say on restricted quantification in Chapter 9.)

While we are on the subject of quantifiers, note that we employ an additional convention on operator precedence that is not very common. In mathematical logic, the quantifiers are taken to bind very strongly. For example, the convention is to read $\exists x\, D(x) \wedge B(x)$ as $(\exists x\, D(x)) \wedge B(x)$, with the second occurrence of x free. As this example illustrates, the mathematical convention does not work very well in linguistics, as it leads to many parentheses—obviously, we want to represent 'a dog is barking' as $\exists x\,(D(x) \wedge B(x))$. This is why we adopt the colon convention. The colon following a quantifier marks a "wide-scope" reading, so that we can represent this example as $\exists x\colon D(x) \wedge B(x)$. Similarly, we will agree to assign wide scope to the nuclear scope of a restricted quantifier. This way, we can represent 'a dog is barking and jumping' as $\exists x\,[D(x)]\,B(x) \wedge J(x)$.[5]

We are ready to move on to the heart of L**, its logical predicates. As we have stated above, L** is a mereology—a theory of parts—and the central predicate of any mereology is a PARTHOOD relation $\sqsubseteq$. However, we are not interested in parthood in general, so L** employs the particular parthood relation of SUBCOLLECTION $\subseteq$—we will explain the difference in a minute.

The mereological axioms centered around the notion of subcollection are what gives L** the "plural flavor"; it is because of these axioms that we may imagine the domain of the interpretation as consisting of collections of individuals from the pre-domain. Many alternative axiom systems for mereology can be found in the literature, some producing equivalent systems, others systems of different

[5] Technically, we assume that the colon-suffixed quantifers and restricted quantifiers reside at the loose end of the operator binding strength scale, as follows:

(i) $\neg \,/\, \forall x \,/\, \exists x \quad \wedge \quad \vee \quad \Rightarrow \quad \Leftrightarrow \quad \exists x\colon / \,\forall x\colon / \,\exists x\,[\ldots] \,/\, \forall x\,[\ldots]$

We also follow the standard left-to-right parsing convention for strings of identical connectives. $A \wedge B \wedge C$ should therefore be read as $(A \wedge B) \wedge C$.

strength. The fine details of mereology are irrelevant for our purposes—feel free to skip reading (5) and (6)—in the sense that we will never need to refer to any particular axioms in this book, so we officially adopt the following particular axiom system of classical extensional mereology in (6) merely for concreteness.[6] The axiom set employs not only the subcollection relation, but also several relations definable off of this relation, so we first provide these definitions.[7]

(5) Derived mereological relations:
a. $x \circ y \Leftrightarrow \exists z\colon z \subseteq x \wedge z \subseteq y$ (overlap)
b. $x \wr y \Leftrightarrow \neg x \circ y$ (disjointness)
c. $x \subset y \Leftrightarrow x \subseteq y \wedge \neg y \subseteq x$ (proper subcollection)
d. $x = y \Leftrightarrow x \subseteq y \wedge y \subseteq x$ (equality)

(6) Classical Extensional Mereology:
a. $\forall x \forall y \forall z\colon x \subseteq y \wedge y \subseteq z \Rightarrow x \subseteq z$ (transitivity)
b. $\forall x \forall y\colon x \subset y \Rightarrow \exists z\colon z \subseteq y \wedge x \wr z$ (weak supplementation)
c. $\exists x\, \varphi(x) \Rightarrow \exists z\colon$ (fusion)
$\forall x\, (\varphi(x) \Rightarrow x \subseteq z) \wedge \forall w\colon \forall x\, (\forall x\, (\varphi(x) \Rightarrow x \subseteq w) \Rightarrow z \subseteq w)$

As we said earlier, mereology is a theory of parts and the central relation in any such theory is the parthood relation, but in L**, we rather call it the subcollection relation. The difference lies in the specific way we want (sub)collections to interact with non-logical predicates. Whereas not every part of a dog (like the tail) itself is a dog—in fact, no proper part is—we are interested in pluralities, and every part of a collection of dogs is itself a collection of dogs, as long as we allow for singleton collections, and we do.

Another way of seeing this is to say that (non-logical) monadic predicates like 'dog' are DISTRIBUTIVE. If $D(x)$ holds for some collection x, then $D(y)$ has to hold for every (proper) subcollection y of x. We therefore adopt the following axiom scheme as a part of the theory of L**.

(7) $\forall x\colon P(x) \Rightarrow \forall y\colon y \subset x \Rightarrow P(y)$, for any non-logical monadic predicate P

[6] The axiom set in (6) is the second axiomatization of classical mereology from Hovda (2009), adapted to our notation.

[7] Everything we say about L** is valid regardless of whether we view the definitions in (5) as abbreviations or as axioms of the theory. In particular, this is also true for equality, which must satisfy the following well-known axiom schemes: (i) $\forall x\colon x = x$, (ii) $t_k = u \Rightarrow (P(t_1, \ldots, t_k, \ldots, t_n) \Rightarrow P(t_1, \ldots, u, \ldots, t_n))$ for any n-place predicate P and terms t_i and u, and (iii) $t_k = u \Rightarrow (f(t_1, \ldots, t_k, \ldots, t_n) \Rightarrow f(t_1, \ldots, u, \ldots, t_n))$ for any n-place operation symbol f and terms t_i and u. (The third axiom scheme is in fact unnecessary, as we have assumed that L** is relational.)

This scheme can be seen as a special instance of a distributivity requirement on all (not only monadic) predicates, but we believe that such a requirement would be too strong. If we required even non-monadic predicates to be distributive, then it is doubtful whether (8b) is the logical form of (8a). The empirical matters are complex, and we will not dwell on them here (see e.g. Scha 1984; Schein 1993; Landman 1996; Kratzer 2002), but note that (8a) can be easily thought of as true in a situation where not every dog is hunting every cat, and not every cat is hunted by every dog. At least in one reading, the sentence seems to only require that each of the dogs (from x) is hunting some cat (from y), and that each cat (from y) is hunted by some dog (from x). This reading is called CUMULATIVE. Each dog participates in the hunt (as a hunter), and each cat participates in the hunt (as hunted); in other words, some dogs (x) are cumulatively hunting some cats (y).

(8) a. Some dogs are hunting some cats.
b. $\exists x\colon D(x) \wedge \exists y\colon C(y) \wedge H(x, y)$

Formally, we can require cumulativity with the axiom scheme below (again, feel free to skip ahead). It should be clear that for monadic predicates, cumulativity is in fact the same as distributivity (when $k = n = 1$, the axiom contains no existential quantifiers). Also note that while we present a general scheme applying to predicates of any arity, we actually believe that only monadic and dyadic predicates are necessary in a linguistic setting (cf. Pietroski 2018).

(9)
$$\begin{aligned} \forall x_1 \ldots \forall x_n\colon P(x_1, \ldots, x_n) \Rightarrow\ & \forall y_k \left[y_k \subset x_k\right] \\ & \exists y_1 \left[y_1 \subset x_1\right] \ldots \exists y_{k-1} \left[y_{k-1} \subset x_{k-1}\right] \\ & \exists y_{k+1} \left[y_{k+1} \subset x_{k+1}\right] \ldots \exists y_n \left[y_n \subset x_n\right] \\ & P(y_1, \ldots, y_n) \end{aligned}$$
for any n-place predicate P and $k \in \{1, \ldots, n\}$

As we have hinted above, the empirical matters involving distributivity, cumulativity, collectivity, plurals vs. mass terms, etc., are complex, and it is out of the scope of this book to address them in any detail. In broad strokes, we would like to follow proposals such as Schein (1993) and Landman (1996) and treat collectivity using the tools of event semantics. For example, in our account, the logical form of (10a) cannot be simply $\exists x\colon P(x) \wedge G(x)$, as this would entail that every person is also gathering on their own—we would rather assert that the individuals in x are all agents of some event of gathering.

(10) a. Some people are gathering.
b. $\exists x\colon P(x) \wedge \exists e\colon G(e) \wedge \mathrm{Ag}(x, e)$

We finally turn to the axiomatization of counting. Remember that we want to have (11b) as the logical form of (11a). So number predicates such as 5 should apply to natural numbers, and we want $\#(n, x)$ to be true iff n is a singleton referring to the number of individuals in collection x.

(11) a. Five dogs are barking.
b. $\exists x\colon \exists n\,(5(n) \wedge \#(n, x)) \wedge D(x) \wedge B(x)$

To this end, we first assume that there is an L^{**} predicate corresponding to every natural number N. We will abuse (decimal) representations of numbers to stand for L^{**} predicates as well; for example, we use 5 both as a representation of the number five and of the predicate of L^{**}. In the axiom schemes in (12) below, N (and N_1 and N_2) thus stand for an arbitrary natural number predicate. Note that the sum symbol in $N_1 + N_2$ in (12d) belongs to the meta-language, i.e. $N_1 + N_2$ denotes the predicate of L^{**} corresponding to the sum of numbers N_1 and N_2.

We are now ready to present the axioms. Axiom schemes (12a) and (12b) give us the existence and uniqueness of natural numbers as individuals in the domain. Axiom (12c) defines the "meaning" of 'one', i.e. it tells us when a collection has a single member. Axiom scheme (12d) states that any collection x of size $N_1 + N_2$ can be partitioned into collections x_1 and x_2 of sizes N_1 and N_2. (Remember that we have defined $\underline{N}(x)$ to be the abbreviation for $\exists n\colon N(n) \wedge \#(n, x)$.)[8]

(12) The counting theory:
a. $\exists n\colon N(n)$
b. $\forall n_1 \forall n_2\colon N(n_1) \wedge N(n_2) \Rightarrow n_1 = n_2$
c. $\forall n\colon 1(n) \Rightarrow \forall x\colon \#(n, x) \Rightarrow \forall y\colon y \subseteq x \Rightarrow y = x$
d. $\forall x\colon \underline{(N_1 + N_2)}(x) \Rightarrow \exists x_1 \exists x_2\colon x_1 \subseteq x \wedge x_2 \subseteq x \wedge x_1 \wr x_2 \wedge$
$(\forall y\colon y \subseteq x \Rightarrow y \circ x_1 \vee y \circ x_2) \wedge$
$\underline{N_1}(x_1) \wedge \underline{N_2}(x_2)$

Our intention here is merely to provide the bare bones of the idea of the axiom system for counting. In a more robust account, we would most likely follow Wągiel (2018) by supplanting (12c) by some mereotopological definition of unity, and supplanting (12d) by some measure function approach applicable to mass terms as well.

With these preliminaries out of the way, we can now turn to the logical forms of various determiners in L^{**}.

[8] Note that we do not assume that every collection can be counted. The axiom system ensures that every finite collection can be counted, but there is no need to force the domain to be finite—given our assumption that the domain contains all natural numbers, this would in fact be contradictory.

5.3 L** and the Logical Form of Determiners

In the previous sections we laid out the basics of our new formal language, L**. We now want to show the payoff from that effort. First, we want to show how determiners will be handled within the language. We already looked briefly at the analysis of 'most', but obviously there is much more to say about the determiner system. Once we have the logical forms of determiners in place, we will turn our attention to developing a syntactic account of directional entailingness for L**. Keep in mind that this development of L** is an intermediate step. Ultimately we want to use L** as a template for thinking about and investigating the structure of directional entailingness *in natural language*. As it is just a template, for now we ignore many morphological, syntactic, semantic, and pragmatic details.

5.3.1 Indefinites and cardinal numerals

Let's start by looking at simple indefinite descriptions and cardinal predicates (which, recall, are not treated as determiners in L**). Notice first, that the truth conditions given for an indefinite in (13b) are the same regardless of whether the formula is interpreted in L** or in standard predicate logic, even though x stands for a collection in L**—because the cardinality of x is not restricted, nothing forces this collection to contain more than one member. Therefore, (13b) *could* serve as the logical form of a sentence with an indefinite quantifier in L**. However, since English nouns (and verbs) carry grammatical number, we reflect the number explicitly in the logical form using the predicates "Sg" and "Pl," representing the singular and the plural morpheme, respectively. We thus prefer to use (13c) over (13b) for the singular; the logical form of a plural indefinite is then as in (14b).[9]

(13) a. Some dog is barking.
b. $\exists x\colon D(x) \wedge B(x)$
c. $\exists x\colon \mathrm{Sg}(x) \wedge D(x) \wedge B(x)$

(14) a. Some dogs are barking.
b. $\exists x\colon \mathrm{Pl}(x) \wedge D(x) \wedge B(x)$

[9] To a first approximation, the semantic contribution of the singular and the plural morpheme, which we represent using predicates 'Sg' and 'Pl', can be taken to be equal to that of numerals 'exactly one' and '(at least) two'. However, note that semantic and grammatical number do not always coincide, see e.g. Corbett (2000) and Harley and Ritter (2002). There is furthermore research indicating that semantically, plural is in fact number neutral, and that the '(at least) two' interpretation arises pragmatically through a higher-order implicature; see e.g. Spector (2013) and references therein.

As remarked in the previous section, the naive way of asserting the cardinality of a collection is by taking cardinal numerals to be predicates applying directly to collections, as in (15b). However, we opted for an indirect approach, as in (15c), where the numerical predicate applies to a number, which is then associated with x by a counting predicate #. We shall at times abbreviate $\exists n\,(5(n) \wedge \#(n,x))$ as $\underline{5}(x)$, as shown in (15d)—not to be confused with (15b)!

(15) a. Five dogs are barking.
b. $\exists x\colon 5(x) \wedge D(x) \wedge B(x)$ (rejected)
c. $\exists x\colon \exists n\,(5(n) \wedge \#(n,x)) \wedge D(x) \wedge B(x)$
d. $\exists x\colon \underline{5}(x) \wedge D(x) \wedge B(x)$ (abbreviation)

Clearing up a detail about the numerical argument of # is in order here. To be completely consistent, we should assume that, like objectual terms, numerical terms refer to collections as well (in this case, collections of numbers). This raises some interesting questions; for example, what is the meaning of $\#(n,x)$ when n is assigned the collection of numbers forty and two? Here we assume that in such a case, $\#(n,x)$ is simply false, the logic being that no collection (x) can have both exactly forty and exactly two members. Other assumptions regarding complex numerals are possible, but we pass over them here.[10]

It might appear that (15c) is less faithful to the sytax of natural language than (15b), but that isn't consistent with our understanding of the literature on numeral phrases. We will go into details on this in Chapter 12, when we delve deeper into the syntax of natural language, but for now we point out two issues. First, our logical form is compatible with the view of a cardinal numeral as a composition of a numeral root with a cardinal morpheme (cf. Wągiel 2018). Second, linguists often hypothesize a dedicated position of (cardinal) numerals within the extended noun phrase (see e.g. Ihsane and Puskás 2001; Aboh 2004), whose head is an abstract function quite similar to our counting relation. $\#(n,x)$ would then transpose into a phrase structure tree in which # is an abstract phrasal head and the numeral is generated in (or perhaps moves into) the specifier position of the phrase. The variable x ties the quantity to the noun phrase.

This of course opens the door to other kinds of qualities which we might attribute to numbers—most of them the product of mathematical discovery (which is fine; many predicates in natural language are the product of discovery).

[10] Note that we might employ different reasoning here. What if # is "additive" and the assertion is actually that x has forty-two members? If we also adopt the idea that the domain can contain more than one instance of any number, then we could take $\exists m(3(m) \wedge \#(m,n)) \wedge 100(n)$ to be the logical form of 'three hundred'. These ideas could well lead to a representation of complex numerals based on addition and multiplication *within predicate logic*. We leave the issue for further research, and henceforth assume that all referents of numerical variables are singletons. For previous work on complex numerals, see e.g. Ionin and Matushansky (2006).

Thus we might have even(n) and odd(n), prime(n) and perhaps even advanced mathematical properties like imaginary(n).[11]

Options are available for the treatment of 'infinitely many' and 'finitely many'. We could introduce them directly as predicates on numbers, for example finite(n). Alternatively, we might think that these flow directly from the introduction of quantification over n (as was the case in L*) and propose the following logical forms.

(16) a. Infinitely many dogs are barking.
b. $\forall n\colon \neg\text{natural-number}(n) \vee \exists x\colon \#(n, x) \wedge D(x) \wedge B(x)$

(17) a. (Only) finitely many dogs are barking.
b. $\neg\forall n\colon \neg\text{natural-number}(n) \vee \exists x\colon \#(n, x) \wedge D(x) \wedge B(x)$

What would decide between these approaches? One thought would be that the answer turns on whether the concept of infinite must be learned (for example in the context of a mathematics class), or whether it flows naturally from simply being able to understand numerical determiners. We leave this as an open question for work in developmental psychology.[12]

5.3.2 Universal quantifiers

The treatment of universals is straightforward in L**. By the logical duality of the universal and the existential quantifier, (18b) means that there is no collection x such that all its members are dogs which are not barking. Any collection x of dogs we take, it contains a dog which is barking. As the range contains singleton collections, we get the usual meaning: every dog is barking. In fact, as 'every' takes a singular complement, the restriction to singletons might be hard-coded into the logical form as in (18c).

(18) a. Every dog is barking.
b. $\forall x\colon \neg D(x) \vee B(x)$ $\sim \neg\exists x\colon D(x) \wedge \neg B(x)$
c. $\forall x\colon \neg(\text{Sg}(x) \wedge D(x)) \vee B(x)$

[11] Like in the case of 'most' mentioned in footnote 4 on page 62, any reasoning involving such mathematical properties will of course have to deploy explicitly provided relations between these properties (which will most likely again be a product of discovery). For example, there is no way to conclude '(Exactly) two As are Bs' from '(Exactly) a prime number of As are Bs' and '(Exactly) an even number of As are Bs' without externally given information that two is the only even prime number.

[12] Note, however, that the (wide) scope of numerical quantification in (16) and (17) is unlike the (narrow) scope we find with other numerals. This might speak against treating infinity via quantification.

The negative determiner 'no' receives its standard logical form as well, with the note that the plural version only yields the correct truth conditions if the plural morpheme has no semantic contribution (see e.g. Spector 2007).

(19) a. No dog is barking.
b. $\neg\exists x\colon (\mathrm{Sg}(x) \wedge D(x)) \wedge B(x) \sim \forall x\colon \neg(\mathrm{Sg}(x) \wedge D(x)) \vee \neg B(x)$

(20) a. No dogs are barking.
b. $\neg\exists x\colon D(x) \wedge B(x) \sim \forall x\colon \neg D(x) \vee \neg B(x)$

In cases where it entails (or perhaps presupposes) existence, determiner 'all' might receive the following logical form (cf. McKay 2006). Note that as the matrix quantifier below is existential, modification by a numeral can be represented as it was with indefinites.

(21) a. All (of the) dogs are barking.
b. $\exists x\colon \mathrm{Pl}(x) \wedge \forall y\, (\neg D(y) \vee y \subseteq x) \wedge D(x) \wedge B(x)$

(22) a. All five dogs are barking.
b. $\exists x\colon \underline{5}(x) \wedge \forall y\, (\neg D(y) \vee y \subseteq x) \wedge D(x) \wedge B(x)$

Note that even if the matrix quantifier ($\exists x$) in (21b) is existential, the logical form still contains a universal quantifier ($\forall y$). This universal quantifier is crucial to arrive at the correct truth conditions—and it is also crucial to yield the correct predictions about directional entailingness, demonstrating once again how the latter property directly flows from the truth-conditional analysis in our system.[13]

5.3.3 Definite descriptions

While we have doubts about the uniqueness criterion inherent in the Russellian analyses of descriptions (see Ludlow 2018), we nevertheless want to show how it can be introduced here. The idea is to wed the analysis of indefinites and cardinals of the previous section with some form of Russell's (1905) (singular) first-order uniqueness condition shown below.

(23) a. The dog is barking. (standard FOL)
b. $\exists x\colon \forall y\, (\neg D(y) \vee y = x) \wedge D(x) \wedge B(x)$

[13] It might be worth noting that even if 'all' could be, at least intuitively, considered a (universal) numeral, universal quantification over numbers in the analysis of 'all' (or, for that matter, 'every') won't do—a collection simply cannot be of "every size."

We can treat the definite determiner in a manner similar to other approaches to plural definite descriptions (e.g. Sharvy 1980; McKay 2006). The key idea is that the uniqueness condition of the singular definites is just a special case of totality. In (24), the collection x of dogs must be such that any ys which are dogs are a part of it. In the singular case (of x), this obviously boils down to equality of x and y.

(24) a. The dog is barking.
b. $\exists x\colon \mathrm{Sg}(x) \wedge \forall y\,(\neg D(y) \vee y \subseteq x) \wedge D(x) \wedge B(x)$

(25) a. The dogs are barking.
b. $\exists x\colon \mathrm{Pl}(x) \wedge \forall y\,(\neg D(y) \vee y \subseteq x) \wedge D(x) \wedge B(x)$

We are not completely satisfied with these logical forms, however—because we have already proposed (25) as the logical form of 'all'. If at all possible, we would prefer to have distinct logical forms for words with clearly differing syntactic distribution. And indeed, it is easy to think of an alternative rendering of 'the'; it turns out that substituting $\circ$ (overlap) for $\subseteq$ (subcollection) in (24) and (25) yields equivalent formulas shown below. (We actually provide two logically equivalent forms using either the universal quantifier and the overlap predicate, or the negated existential quantifier and the disjointness predicate; we also enrich the list of logical forms by adding a formula for 'the' plus a numeral.)

(26) a. The dog is barking.
b. $\exists x\colon \mathrm{Sg}(x) \wedge \forall y\,(\neg D(y) \vee y \circ x) \wedge D(x) \wedge B(x)$
c. $\exists x\colon \mathrm{Sg}(x) \wedge \neg\exists y\,(D(y) \wedge y \wr x) \wedge D(x) \wedge B(x)$

(27) a. The dogs are barking.
b. $\exists x\colon \mathrm{Pl}(x) \wedge \forall y\,(\neg D(y) \vee y \circ x) \wedge D(x) \wedge B(x)$
c. $\exists x\colon \mathrm{Pl}(x) \wedge \neg\exists y\,(D(y) \wedge y \wr x) \wedge D(x) \wedge B(x)$

(28) a. The five dogs are barking.
b. $\exists x\colon \underline{5}(x) \wedge \forall y\,(\neg D(y) \vee y \circ x) \wedge D(x) \wedge B(x)$
c. $\exists x\colon \underline{5}(x) \wedge \neg\exists y\,(D(y) \wedge y \wr x) \wedge D(x) \wedge B(x)$

One might wonder why we have decided to deploy $\subseteq$ in the logical form for 'all' and $\circ/\wr$ in the logical form for 'the', rather than vice versa. This is because not only the syntactic, but also the cross-linguistic distribution of these items is radically different. It turns out that the cross-linguistic distribution of 'the' (but certainly not that of 'all', which, as far as we know, is present in all langauges) is actually related to the distribution of 'most'. We will reveal more details in the following section, after telling you a bit more about 'most'.

5.3.4 Superlatives and comparatives

Recall the analysis of 'most' from section 5.1 and the idea behind that analysis: any two subcollections, each consisting of more than half the members of the whole, will necessarily overlap.[14]

(29) a. Most dogs are barking.
b. $\exists x\colon (\exists n\colon \#(n,x) \wedge \neg(\exists y\colon (\#(n,y) \wedge D(y)) \wedge y \wr x)) \wedge D(x) \wedge B(x)$
c. $\exists x\colon (\exists n\colon \#(n,x) \wedge (\forall y\colon \neg(\#(n,y) \wedge D(y)) \vee y \circ x)) \wedge D(x) \wedge B(x)$

However, not all languages share this reading of the English superlative determiner, which we shall henceforth call the MAJORITY reading. In fact, this reading is cross-linguistically quite rare, while the PLURALITY reading is available in all languages with a superlative determiner (Coppock, Bogal-Allbritten, et al. 2020). In the plurality reading, illustrated below, the superlative is associated with focus (the focused constituent is underlined).

(30) Most dogs are barking.

The function of focus is to invoke contextually salient alternatives to the focused constituent. In (30), it invokes contextually salient alternatives to barking, say sleeping and jumping. The superlative then asserts that the number of barking dogs surpasses the number of sleeping dogs and the number of jumping dogs. For example, the sentence is true in a situation with forty barking dogs, thirty sleeping dogs, and twenty jumping dogs. This is the case as more dogs are barking than sleeping and more dogs are barking than jumping, regardless of the fact that the number of the barking dogs is less than the number of the sleeping dogs and the jumping dogs, together.

To get the idea on how to represent the plurality reading in L**, we will first have to take a look at comparatives. But before we do that, let us pause to remark on the cross-linguistic rarity of the majority reading mentioned above. Why would the majority reading be confined to a handful languages? Is there a pattern to which

[14] Note a possible simplification of the logical form for 'most' introduced in the main text. Instead of introducing an existentially bound numerical variable occuring in the quantity predicates linked to both x and y, (i) directly compares the cardinality of these terms, i.e. we assume $x \approx y$ iff x and y are of the same cardinality. Indeed, under a suitable conceptualization of numbers, $\approx$ could completely take over the function of #. Imagine that a (natural) number n is represented as a collection consisting of n members. Then $n \approx x$ is true iff the collection x consists of n members, which is precisely the meaning of $\#(n,x)$.

(i) a. Most dogs are barking.
b. $\exists x\colon (\neg\exists y\colon x \approx y \wedge D(y) \wedge y \wr x) \wedge D(x) \wedge B(x)$

languages allow this reading? It seems that the answer is yes, and even more, that L** can provide an explanation of why this is the case.

Živanović (2007, 2008) observed that the logical form of the majority superlative determiner is exactly like the logical form of the definite determiner, but with cardinality conditions added on top. This is shown below, where the parts of 'most' shared with 'the' are grayed out.

(31) a. The/most As are Bs.
b. $\exists x\colon (A(x) \land \exists n\colon \#(n,x) \land \neg\exists y\,(A(y) \land \#(n,y) \land y \wr x)) \land B(x)$

Now let's assume that the following general hypothesis holds: If a language deploys some complex structure, it will also deploy a simplified version of this structure. Given that 'the' is the simpler cousin of 'most' (in the majority reading), we get the following *unidirectional* prediction.

(32) If a language has a superlative determiner with the majority reading, definitess is an active grammatical category in this language.[15,16]

Živanović (2007, 2008) tested (and corroborated) the prediction on twenty languages from several language families, and several authors citing his work provided further corroborations (and no counterexamples).[17]

[15] There are many languages, with and without definite determiners, which can paraphrase the "more than half" reading of English 'most' using the noun 'majority'. However, our prediction says nothing about nominal means of expressing majority. The prediction is limited to items belonging to the determiner system of a language, or more broadly, to the grammatical rather than lexical categories.

[16] We phrase the prediction a bit more generally than Živanović (2007, 2008), who states that if a language has a majority superlative determiner, it also has a definite determiner. As convincingly argued for Bengali (Syed 2017; Syed and Simpson 2017), some languages express definiteness by word order rather than a definite determiner. Given that we have arrived at the prediction through the analysis of logical forms, it is of course completely insensitive to whether definiteness is expressed by morphological or syntactic means.

[17] These further corroborations include ASL (Bernath 2009), Lithuanian (Pereltsvaig 2015), and Bengali (Syed 2017). An opportunity for further tests is provided by Coppock, Bogal-Allbritten, et al. (2020), who conducted an extensive typological study of the readings of superlatives and the means to express these readings. Seventeen out of ninety-two languages in their survey exhibit the majority superlative determiner and could therefore potentially falsify our prediction (languages without this determiner cannot possibly present a counterexample). Checking the descriptive grammars and *The World Atlas of Linguistic Structures* (WALS is freely accessible at wals.info) for the existence of a definite determiner in these languages reveals a single potential counterexample, i.e. a language without a definite determiner: Finnish. However, it turns out that the Finnish grammars (Sulkala and Karjalainen 1992; Karlsson 1999), describing *standard* Finnish, are not up-to-date in this respect. Laury (1997, p. 1) informs us that "since at least the late 1800s, linguists working on *spoken* Finnish have known that the demonstrative pronoun 'se' is being used in a way which closely resembles the use of definite articles" (emphasis ours). The cited work is an extensive study of the diachronic development of Finnish 'se', showing without a shade of doubt that modern Finnish *does* have a definite determiner and therefore does *not* falsify our prediction. Conclusion? When our theory asks for a definite determiner, the language develops one. Who could ask for more than that?

Having received some empirical evidence that our logical form for the majority superlative determiner is on the right track, let us turn to the analysis of the comparative determiner, which will later lead us to the logical form of the plurality superlative determiner. We have provided the L* logical forms for comparatives in Chapter 4, so let us take these as the starting point. Formulas (33b) and (34b) present an L** rendering of the idea behind the L* logical forms, but note that unlike in the L** logical forms involving cardinality we have introduced so far, $\exists n$ has wide scope in these formulas. In L*, this was unavoidable, as quantification and cardinality restriction were done simultaneously by means of subscripted quantifiers. In L**, quantification and cardinality are divorced, so the comparison class can be moved into the scope of $\exists x$, as in (33c) and (34c), yielding logical forms that better resemble the syntax of natural language expressions.

(33) a. More dogs than cats are sleeping.
b. $\exists n\colon (\exists x\colon \#(n,x) \wedge D(x) \wedge S(x)) \wedge \neg(\exists y\colon \#(n,y) \wedge C(y) \wedge S(y))$
c. $\exists x\colon (\exists n\colon \#(n,x) \wedge \neg\exists y\colon \#(n,y) \wedge C(y) \wedge S(y)) \wedge D(x) \wedge S(x)$

(34) a. More dogs are barking than sleeping.
b. $\exists n\colon (\exists x\colon \#(n,x) \wedge D(x) \wedge B(x)) \wedge \neg(\exists y\colon \#(n,y) \wedge D(y) \wedge S(y))$
c. $\exists x\colon (\exists n\colon \#(n,x) \wedge \neg\exists y\colon \#(n,y) \wedge D(y) \wedge S(y)) \wedge D(x) \wedge B(x)$

We now transform these formulas so that they can serve as an inspiration for the logical form of the plurality reading of superlatives. Starting with (34c) above, we first convert it to use (neo-)Davidsonian event semantics, as shown in (35b) (Davidson 1967; T. Parsons 1990, 1995).[18] Instead of thinking of 'bark' as a predicate applying to individuals $B(x)$, we decompose it into the assertion that a certain event e (introduced by an existential quantifier) is an event of barking $B(e)$ and associate the individual x and the event e by the predicate "is an agent of" $\mathrm{Ag}(x,e)$, the semantic reflex of the thematic role Agent from the syntax. The transition to event semantics affects both B and S, yielding two event variables, e and f. As the second step, we add the requirement that the two collections of events are disjoint (35c). As events of dogs sleeping and dogs barking are disjoint anyway, the addition is vacuous; it will be crucial for superlatives, though.

(35) a. More dogs are barking than sleeping.
b. $\exists e \exists x\colon$
$(\exists n\colon \#(n,x) \wedge \neg\exists f \exists y\colon \#(n,y) \wedge D(y) \wedge \mathrm{Ag}(y,f) \wedge S(f))$
$\wedge\, D(x) \wedge \mathrm{Ag}(x,e) \wedge B(e)$

[18] Even though we believe that natural language expressions should be represented using event semantics, most of the logical forms in this book to not deploy it, for clarity of presentation. We discuss event semantics in the context of conservativity and restrictedness in section 10.2.

c. $\exists e \exists x\colon$
$(\exists n\colon \#(n, x) \wedge \neg\exists f \exists y\colon \#(n, y) \wedge D(y) \wedge \mathrm{Ag}(y, f) \wedge S(f) \wedge f \wr e)$
$\wedge\, D(x) \wedge \mathrm{Ag}(x, e) \wedge B(e)$

Under our analysis, one crucial difference between comparatives and superlatives is that the latter omit an explicit comparison class, which is provided contextually. The logical form of a superlative on the plurality reading is given below: $S(y)$ from (35c), featuring predicate 'sleep', is replaced by $P(y)$, where P is a variable. The position of the quantifier over P is the same as the position of the quantifier over the inner event variable f. Translating (36b) to plain English: there is a collection of events e, which are events of n dogs barking, such that there is no contextually salient activity P and a collection of events f disjoint from e which are events of n dogs performing P.[19]

(36) a. Most dogs are barking.
b. $\exists e \exists x\colon$
$(\exists n\colon \#(n, x) \wedge \neg\exists P \exists f \exists y\colon \#(n, y) \wedge D(y) \wedge \mathrm{Ag}(y, f) \wedge P(f) \wedge f \wr e)$
$\wedge\, D(x) \wedge \mathrm{Ag}(x, e) \wedge B(e)$

This is not a book about comparatives and superlatives, so now is not the time to go into further detail on this class of constructions. Our immediate goal, after all, has been to show how superlatives and comparatives interact with the determiner system and ultimately, of course, encode the polarity markings required to have a robust polarity logic for natural logic. We can, however, conclude this section with rudimentary proposals of logical forms for several other constructions involving comparatives and superlatives. Hopefully these examples will provide enough clues for readers interested in pursuing these constructions in more detail.

(37) a. At most five dogs are barking.
b. $\forall x\colon \neg(D(x) \wedge \exists n\, (5(n) \wedge \#(n, x)) \wedge \exists y\, (D(y) \wedge B(y) \wedge y \wr x)) \vee \neg B(x)$

(38) a. More than five dogs are barking.
b. $\exists x\colon (D(x) \wedge \exists n\, (5(n) \wedge \#(n, x)) \wedge \exists y\, (D(y) \wedge B(y) \wedge y \wr x)) \wedge B(x)$

(39) a. Less than five dogs are barking.
b. $\forall x\colon \neg(D(x) \wedge \exists n \exists y\colon 5(n) \wedge \#(n, y) \wedge y \subseteq x) \vee \neg B(x)$

(40) a. At least five dogs are barking.
b. $\exists x\colon (D(x) \wedge \exists n \exists y\colon 5(n) \wedge \#(n, y) \wedge y \subseteq x) \wedge B(x)$

[19] Of course, (36b) looks suspiciously second-order. However, monadic lexical predicates can be interpreted as plural individuals, in line with Boolos ([1984] 1998), who shows that monadic second-order logic is equi-interpretable with plural logic. Under this interpretation, $P(f)$ reads $f \subseteq P$ and the status of $\exists P$ equals the status of other quantifiers. See also section A.4 of the Appendix.

Before moving on, we want to draw the reader's attention to the austerity of L**. Perhaps contrary to expectations of many semanticists, truth conditions associated with many (if not all) determiners can be represented with few resources. For example, there is no need for a greater-than relation as a primitive; as shown above, it can be "constructed" within first-order predicate logic employing the subcollection (or overlap) and the counting relation. In fact, it is the paucity of resources in L** which forces the negation to do all the work it can, and this, in turn, is the primary reason that we are able to successfully reach our principal goal, the development of a polarity logic for natural language. Or, as we have described the project throughout this book, we can pursue the question of whether the Medieval idea of two basic inference rules, the *dicta de omni et nullo*, can be executed for natural language. We have enough on the table now so we can begin investigating this question in earnest. It is time to turn to the theory of directional entailingness and polarity in L**.

5.4 A Syntactic Characterization of Directional Entailingness Environments in L**

We've covered a lot of ground in our discussion of L**, but there is one very important topic that we have touched on only briefly thus far—directional entailingness. As we will see, the results we achieved in L* will be available for us here as well, but here we are going to go into a bit more detail, and we are going to highlight an open problem.

Let's begin with a review. First, let's remind ourselves what the semantic notion of directional entailingness is. Applied to determiner positions, we can define directional entailingness as follows.

(41) A determiner D is upward/downward entailing in its first/second argument iff $[\![\mathrm{D}]\!](A, B)$ plus the given condition produces the given entailment, as specified in Table 5.1.

Table 5.1 The definition of directional entailingness for determiners

direction	argument	condition	entailment
upward	first	$A \subset A'$	$[\![\mathrm{D}]\!](A', B)$
upward	second	$B \subset B'$	$[\![\mathrm{D}]\!](A, B')$
downward	first	$A \supset A'$	$[\![\mathrm{D}]\!](A', B)$
downward	second	$B \supset B'$	$[\![\mathrm{D}]\!](A, B')$

Recall now the L** definitions of some basic determiners and compare them to their directional entailingness properties.

(42) a. $[\![\text{some}]\!](\overset{\uparrow}{A}, \overset{\uparrow}{B})$ iff $\exists x\colon A(x) \wedge B(x)$

b. $[\![\text{five}]\!](\overset{\uparrow}{A}, \overset{\uparrow}{B})$ iff $\exists x\colon \exists n\,(5(n) \wedge \#(n, x)) \wedge A(x) \wedge B(x)$

c. $[\![\text{every}]\!](\overset{\downarrow}{A}, \overset{\uparrow}{B})$ iff $\forall x\colon \neg A(x) \vee B(x)$

d. $[\![\text{no}]\!](\overset{\downarrow}{A}, \overset{\downarrow}{B})$ iff $\forall x\colon \neg A(x) \vee \neg B(x)$

The generalization, as in the case of L*, is obvious. The directional entailingness properties of a determiner depend on the *form* of the definition. More precisely, they depend on one simple feature of that form—on whether an argument is negated. If the argument is within the scope of a negation, it is in a downward entailing (*dictum de nullo*) environment; if it is not within the scope of a negation, it is in an upward entailing (*dictum de omni*) environment.

Obviously, whether an argument is within the scope of a negation or not depends on the set of logical connectives that we use. For example, if we used implication in the logical form of 'every', no negation would be present in the formula: $\forall x\colon A(x) \Rightarrow B(x)$. For our account of directional entailingness to work, it is crucial that the only propositional connectives used by the formal language are conjunction $\wedge$, disjunction $\vee$, and negation $\neg$. (When we write down some other connective, we do it merely for clarity, for example to highlight the conditional form of $A(x) \Rightarrow B(x)$. Officially, all non-Boolean connectives are defined as abbreviations.)

The symbols A and B we have used for arguments may lead one to believe that our analysis only applies to determiner arguments that are simple predicates. This is not the case. These arguments may be as complex as we wish. That is, we may substitute (for) not only words, but arbitrary *constituents* of a sentence. So, as long as it is established that 'black dog in this yard' $<$ 'pet belonging to my neighbor', we can substitute the former phrase by the latter in a *de omni* environment such as the first argument of 'some' below.

(43) a. Every black dog in my yard is a pet belonging to my neighbor.

b. Some black dog in my yard is barking.

c. ∴ Some pet belonging to my neighbor is barking.

Next, is it really the *definition* of a determiner that is the final arbiter of directional entailingness? Certainly not, since determiners may interact with other elements in a sentence, and it is quite common to encounter, for example, an indefinite with an argument that is in a downward entailing environment. A case in point:

(44) a. No student owns a car. ('Ferrari' < 'car')
$\neg\exists x\colon S(x) \wedge \exists y\colon C(y) \wedge O(x, y)$
b. ∴ No student owns a Ferrari.

Clearly, the trick is to inspect not the definitions of the determiners but the logical form of the entire sentence (which the determiner definitions contribute to, but certainly do not exhaust). Above, we find $C(y)$—an argument of 'a'—within the scope of a negation, and this negation makes $C(y)$ downward entailing, even if the negation itself comes from the negative determiner 'no'.

Once we start looking at entire logical forms rather than definitions of determiners, it is easy to find examples where a predicate occurs within the scope of two or more negations. In the example below, we see that being dominated by two negations has the same effect as being dominated by no negations (starting from the middle term in 'easy exam in semantics' < 'exam in semantics' < 'exam', we may go up but not down).

(45) a. No student passed every [exam in semantics]$_E$.
$\neg\exists x\colon S(x) \wedge \forall y\colon \neg E(y) \vee P(x, y)$
b. ∴ No student passed every exam.
c. ∴̸ No student passed every easy exam in semantics.

The point is, of course, that the two negations above 'exam in semantics' "cancel" each other. Thus, instead of talking about a predicate being within the scope of a negation or not, we use the (by now familar) notion of *polarity*, whose definition we repeat below.[20]

(46) A constituent has
a. POSITIVE POLARITY iff it lies within the scope of an *even* number of negations, and
b. NEGATIVE POLARITY iff it lies within the scope of an *odd* number of negations.

The updated generalization is then that a constituent of positive polarity is in an upward entailing environment, and that a constituent of negative polarity is in a downward entailing environment.

Deploying polarity, and having shifted our attention from argument positions of determiners to arbitrary constituents of logical forms, *dicta de omni et nullo*

[20] The definition of polarity should be understood as relative to a given formula. Clearly, $D(x)$ has different polarities in $\exists x\colon D(x) \wedge B(x)$ "some dog barks" (positive polarity) and in $\neg\exists x\colon D(x) \wedge B(x)$ "no dog barks" (negative polarity).

can now perform inferences beyond the wildest dreams of Medieval logicians. For example, we can substitute for a noun modified by adjectives and relative clauses, as in (47), and we can perform substitutions within noun phrases embedded in other noun phrases, as in (48).

(47) a. Every young $\underbrace{\text{person}}_{-}$ who [wears a hat]$_W$ is a Norwegian.
$\forall x\colon \neg(Y(x) \wedge P(x) \wedge W(x)) \vee N(x)$
b. $\therefore$ Every young $\overline{\text{man}}$ who wears a hat is a Norwegian.

(48) a. No man who knows every $\underbrace{\text{friend}}_{+}$ of his bookie is a [good father]$_G$.
$\neg\exists x\colon M(x) \wedge \forall y \left(\neg(\exists z\colon P(x,z) \wedge B(z) \wedge F(y,z)) \vee K(x,y)\right) \wedge G(x)$[21]
b. $\therefore$ No man who knows every $\overline{\text{acquaintance}}$ of his bookie is a good father.
c. $\not\therefore$ No man who knows every $\overline{\text{boyfriend}}$ of his bookie is a good father.

Let us move on to the interesting cases where an argument occurs more than once in the definition of a determiner. These cases require that we state the relation between polarity and directional entailingness more precisely. What can happen now is that the occurrences of the argument can have opposing polarities. As we can see below, such a situation results in the absence of directional entailingness for that argument.

(49) a. $[\![\text{the}]\!](\overset{\times}{A}, \overset{\uparrow}{B})$ iff $\exists x\colon \forall y\,(\neg A(y) \vee y \circ x) \wedge A(x) \wedge B(x)$
b. $[\![\text{the five}]\!](\overset{\times}{A}, \overset{\uparrow}{B})$ iff $\exists x\colon \underline{5}(x) \wedge \forall y\,(\neg A(y) \vee y \circ x) \wedge A(x) \wedge B(x)$
c. $[\![\text{most}]\!](\overset{\times}{A}, \overset{\uparrow}{B})$ iff $\exists x\colon (\exists n\colon \#(n,x) \wedge (\forall y\colon \neg(\#(n,y) \wedge \neg A(y)) \vee y \circ x)) \wedge$
$A(x) \wedge B(x)$

(50) a. The dog is barking.
b. $\not\therefore$ The collie is barking.
c. $\not\therefore$ The animal is barking.

(51) a. The five dogs are barking.
b. $\not\therefore$ The five collies are barking.
c. $\not\therefore$ The five animals are barking.

(52) a. Most dogs are barking.
b. $\not\therefore$ Most collies are barking.

[21] We analyze 'friend' as a dyadic predicate F; $F(y,z)$ means that y is a friend of z. Predicate P stands for the grammatical relation of possession. Let us disentangle $P(x,z)$: the possessive pronoun 'his' is bound by 'no man', so it is represented by x, and z stands for the bookie, so $P(x,z) \wedge B(z)$ is the logical form of 'his bookie'.

c. $\not\therefore$ Most animals are barking.

The final syntactic characterization of directional entailingness must therefore take into account the possibility that a single determiner argument has multiple occurrences in the definition, or more generally, it must take into account the possibility that "what you see is *not* what you get"—the logical form of a sentence may contain several occurrences of a predicate corresponding to a certain word.[22]

(53) An *occurrence of a predicate* has

a. POSITIVE POLARITY iff it lies within the scope of an *even* number of negations, and

b. NEGATIVE POLARITY iff it lies within the scope of an *odd* number of negations.

(54) A *predicate* has

a. POSITIVE POLARITY iff all its occurrences have positive polarity, and

b. NEGATIVE POLARITY iff all its occurrences have negative polarity.

If a predicate has either positive or negative polarity, we say it has UNIFORM polarity; otherwise its polarity is MIXED.

(55) A syntactic characterization of Directional Entailingness

a. If a predicate has *positive* polarity, it is in an *upward* entailing environment.

b. If a predicate has *negative* polarity, it is in a *downward* entailing environment.

c. If a predicate has *mixed* polarity, it is *not* in a *directionally entailing* environment.

Issues of quantifier scope present another example which warns us that our syntactic characterization of directional entailingness applies to logical forms of sentences, and not to their surface forms. The preferred reading of (56) is the one where 'every' (covertly) takes wide scope over 'no'. The logical form must reflect this reading if our characterization is to correctly predict the directional

[22] The definition of polarity in (46) talked about the polarity of *a constituent* of a formula. Below, we define the polarity of *(an occurrence of) a predicate*. While the new definition is an improvement in the sense that it addresses the issue of mixed polarity, it is also a step backwards, as it does not cover cases such as (43), where we substitute (for) a complex argument. To be fully general, we would have to abandon definition (53) in favor of the former (46), and—as complex predicates are really open formulas—define the polarity of an open formula rather than a predicate in (54). However, this would open up some technical issues related to renaming of (free) variables (which produces a logically equivalent formula). We will not address these details here, as they will become clear in Chapter 7, during the development of our Dynamic Deductive System.

entailingness properties. (Briefly, we assume that the surface form undergoes the syntactic operation of Quantifier Raising. We will say more on this topic in Chapter 12.)

(56) No man who knows every $[\underbrace{\text{dog in town}}_{-}]_D$ [runs a butcher shop]$_R$.
$\forall y\colon \neg D(y) \vee (\neg\exists x\colon (M(x) \wedge K(x,y)) \wedge R(x))$

a. ∴ No man who knows every $\overline{\text{collie in town}}$ runs a butcher shop.

b. $\not\therefore$ No man who knows every $\overline{\text{pet in town}}$ runs a butcher shop.

Clause (55c) of the characterization also makes it clear why we have proposed (57b) rather than, say, (57c) as the logical form of 'most'. In (57c), the disjointness of x and y flows from contradictory requirements on these variables, namely $B(x)$ and $\neg B(y)$, much in the spirit of the GQ denotation of 'most' (58), and also in a fashion similar to comparatives. However, B has mixed polarity in (57c). Given this logical form, our generalization would thus predict, contrary to the fact, that the second argument is not in a directionally entailing environment.

(57) a. Most dogs are barking.

b. $[\![\text{most}]\!](\overset{\times}{D}, \overset{\uparrow}{B})$ iff $\exists x\colon (\exists n\colon \#(n,x) \wedge \neg\exists y\colon \#(n,y) \wedge D(y) \wedge y \wr x)$
$\wedge D(x) \wedge B(x)$

c. *$[\![\text{most}]\!](\overset{\times}{D}, \overset{\times}{B})$ iff $\exists x\colon (\exists n\colon \#(n,x) \wedge \neg\exists y\colon \#(n,y) \wedge D(y) \wedge \neg B(y))$
$\wedge D(x) \wedge B(x)$

(58) $[\![\text{most}]\!](D,B) = |D \cap B| > |D - B|$ (GQ)

Given one formula, the generalization predicts upward entailment in B, given the other, no entailment. How can this be? After all, the formulas are logically equivalent!

The answer is that (55c) will not do as it stands. After all, given a formula which fulfills the condition of either (55a) or (55b), it is trivial to produce a logically equivalent formula which falls under (55c): one simply needs to attach a tautology such as $B(x) \vee \neg B(x)$ anywhere within the scope of the quantifier over x.

This is a problem that was noted by Ludlow (1995), and it is a problem that any syntactic account of directional entailingness needs to address. It is one thing to argue from having all positive (negative) occurrences to upward (downward) entailingness, and it is quite another to argue from mixed polarity to neither upward nor downward entailingness. Similarly, the problem infects any attempt to argue from directional entailingness to polarity. As indicated by the example above, the key issue here is logical equivalence—provably so, as we show in the Appendix—so we want to recast (55c) as (59).

(59) a. If a predicate P has mixed polarity in every formula S' logically equivalent to S, then P is not in a directionally entailing environment in S. Or equivalently:

b. If a predicate P is in a directionally entailing environment in formula S, then there is a formula S', logically equivalent to S, such that P has uniform polarity in S.

And this is where things get interesting, because one needs to be cautious about the notion of logical equivalence here. It is, for lack of a better word, *cheating* to employ the notion of logical equivalence without having a proof theory available. After all, we are looking for a syntactic account of directional entailingness, and so we cannot rightly rely on a semantic notion of logical equivalence to secure our understanding of directional entailingess. We need a proof theory to go with our language L**.[23]

It is thus time to develop the proof theory for our formal language L**. It will be, after all, the guts of our proposal, as it would be for any system that purports to provide a syntactic account of the logic of natural language. Accordingly, in the next three chapters we turn our attention to the deductive system, show how it is complete, and compare it to other natural deduction systems. Our deductive system will lie at the heart of this project, and it will also deliver on our project to cash out the quest for the Holy Grail of Natural Logic, for what we will find is that our deductive system will take the Medieval project farther than even its progenitors could have imagined.

[23] An alternative idea would be that there are multiple ways in which agents can syntactically represent these formulas, and when simplification fails the resulting formula may not license directional entailingness, even though it is logically equivalent to something that does. We will get to the issue of these theories as proposals for how humans carry out inferences, but looking ahead, the idea here is that persons do reject some inferences that appear to be perfectly sound, and when they do so it is a function of the forms of the premises in the arguments. What we are suggesting here is that it would not be unreasonable to say that upward entailingness fails for cases like $B \wedge (B \vee \neg B)$. Or at least it fails when agents can't come up with a successful simplification.

6
An Introduction to the Dynamic Deductive System

In the previous chapter we claimed that L** allows us to run natural deduction off of polarity, and that in doing so it can cash out the Medieval logicians' idea of the *dicta de omni et nullo*. Ultimately, of course, we want to show that we can do exactly the same thing in natural language, and laying the ground for this enterprise is in fact the major contribution of L* and L**—after all, the received wisdom, which we hope to have dispelled, is that first-order predicate logic is not apt for natural language analysis.

Before we get there, however, we want to work out the idea of a polarity-based deductive system in earnest. The *dicta* rules presented in the previous chapters cash out a large chunk of entailment in natural language—and thanks to the plural nature of L**, this chunk is perhaps larger than many would have thought possible—but as they stand, the *dicta* rules need supplementation. That is, the *dicta* rules do not constitute a *complete* system, and developing a complete polarity-based system is one of the major goals of our enterprise. So in this chapter, we start answering the question: What do we need to supplement the *dicta* rules with, to transition from a system covering only *directional entailment* to a complete deductive system cashing out entailment in general?

One of the surprising insights we will glean is that expanding our theory beyond the dicta rules will ultimately lead to an important simplification. Perhaps that shouldn't be surprising, since sometimes deeper insights come from expanding our scope of interest. In this case we will see that the basic syntactic operations underlying our syntactic system are all guided by our polarity-sensitive notion of *p-scope*. In effect, the role of polarity in our system will be even greater than in the Medieval systems. Getting to that conclusion will take some time, however, and it is not for nothing that we have dedicated three chapters to our deductive system and showing the important role that polarity plays at its most fundamental level of analysis.

We have much ground to cover in the next three chapters, and in this chapter, we begin to explain the basics of our Dynamic Deductive System (DDS), which works not only for L**, but for standard first-order predicate logic as well. In fact, we know it works for L** because we can prove that DDS is sound and complete for first-order logic in general and because L** can be interpreted as a first-order language supplemented by the axioms of mereology and counting. Let us start exploring what, precisely, our formal deductive system looks like.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović,
Oxford University Press. © Peter Ludlow and Sašo Živanović 2022. DOI: 10.1093/oso/9780199591534.003.0006

Perhaps the best place to start is with a proposal that didn't quite work out, but which contained the kernel of a good idea. Ludlow (2002) suggested that a deductive system could get by with the following two rules, to be used in upward entailing and downward entailing environments, respectively.

Dictum de Omni	*Dictum de Nullo*
$\alpha < \beta, [_S \ldots \alpha^+ \ldots] \vdash [_S \ldots \beta \ldots]$	$\alpha < \beta, [_S \ldots \beta^- \ldots] \vdash [_S \ldots \alpha \ldots]$

In this case, $+$ and $-$ signs indicate polarity within a sentence *S* and the $<$ symbol indicates a generalized way of describing statements that take you from subsets to supersets; examples would include 'collies are dogs', 'all collies are dogs', and 'if dogs bark, then dogs bark or cats meow'. With respect to that last example, think of the set of worlds denoted by 'dogs bark' vs. the set of worlds denoted by 'dogs bark or cats meow'. The former is a subset of the latter.

This characterization of the deductive rules needs to be tightened up to account for inferences like Addition and Simplification, but there is a bigger concern. Sometimes the premises to our inference are not neatly laid out as separate sentences. They might be contained within a single sentence, as follows.

(1) $[_S \ldots (\alpha < \beta)^+ \ldots \beta^- \ldots]$

On such embedded premises, the rules proposed by Ludlow are mute. So, for example, we might encounter something like the following, where the surrounding linguistic context is elided:

(2) $[_S \ldots$ all dogs bark... Fifi doesn't bark...]

If you are tempted to use the *dictum de nullo* rule to substitute α for β in this example (concluding that the occurence of '(doesn't) bark' can be replaced with '(isn't) a dog'), you should be careful, because sometimes you can do that and sometimes you absolutely can not. To take a simple case, consider the following.

(3) a. All cats meow or all dogs bark, but Fifi doesn't bark.
    b. $\not\therefore$ Fifi is not a dog.

It is not enough to know what local or global polarities terms are in. A special relationship has to obtain between inline premises—a relation that we will call premise scope, or p-scope for short. The rough idea of p-scope is that it is a relationship between locations within a logical tree structure that can determine the reach of a premise within that structure. It can do so within an arbitrarily complex formula by keeping track of polarity (and a few other things) in the intervening nodes of the tree structure of the formula.

We think this is an important consideration for Natural Logic in particular, because in natural language there is no reason to assume that premises are laid out neatly. We quite naturally combine them into complex constructions. One option is to explode those constructions into separate premises, but that isn't always trivial, and in any case we will see it merely adds an extra layer of work. The rules of the *dicta* can apply *in-line*—in the sense that they can perform the substitution anywhere within the sentence—so why not just do it all in-line and dispense with the long list of premises?

If we take this approach to Natural Logic then the resulting picture is what we call a *Dynamic Deductive System*. There is no static list of premises from which we generate a longer list of theorems, etc. There is rather a natural language structure to which we can apply successive operations of the *dicta* rules, yielding a structure that changes on each rule application. It is in this sense that the deductive system is *dynamic*.

So, our immediate task is to rectify two concerns. First, we need a systematic account of axioms and rules of the theory. Second, we need to address the issue of p-scope, and provide a formal definition of it. Once we do, we will see (as a bonus) that p-scope turns out to be interesting for more than its role in our deductive system. It will also turn out to be crucial for understanding the notion of restrictedness in the syntax, and that in turn is going to help us develop a syntactic counterpart to conservativity (recall the discussion of generalized quantifiers in section 3.3), and ultimately our understanding of determiners and the nature of quantification in natural language itself.

Meanwhile, we haven't forgotten that we promised to deliver on a serious deductive system for natural logic. Viewed through the lens of contemporary logic, that was quite the promissory note, because since the development of 20th-century logic we have come to expect quite a lot from our deductive systems. We not only want to know if the system is complete. We also want to know if the system achieves any advantages for us as logicians—does it make our lives easier, or more complicated? If we are to embed the deductive system within linguistics, or at least make sure that it dovetails with linguistics, we want to know if it enjoys the relevant attributes of simplicity, understood in the sense of a proposal that prepares linguistics for a unification with other sciences (for example as part of a chapter in biolinguistics). If it is a deductive system for Natural Logic we might also want to know about its psychological reality—is it plausible as an analysis of how humans carry on actual deductions (in any case, as an account of their competence, if not their performance)?

The deductive system we propose in this chapter ends up having all of the positive features wished for above. The system is, in fact, complete (or rather, it has the same deductive power as standard deductive systems), and it does simplify a number of deductions. Perhaps more interesting is the fact that it gives us a new way of approaching deduction, a way which we believe to be cognitively much more plausible than even the most natural of natural deductive systems currently on the market. Dynamic Deductive System operates "inline"—taking premises from anywhere in the formula and dropping the conclusion elsewhere in the formula—which leads to various welcome features, like its "dynamic" character and the ability to transparently apply in conditional environments. The secret sauce, as we shall see, is that by keeping track of the positive and negative polarity environments we can radically simplify the deductive process. Doing so allows us to apply inference rules directly within the formula—indeed it allows us to apply them arbitrarily deep within the formula.

6.1 Formalizing the *Dicta* Rules

Recall from section 5.4 that we distingushed the *dictum de omni et nullo* environments, which are related to the polarity of a term or region of a logical tree, from the rules which apply in those environments. We called those rules the *dicta* rules, and we said that there are two of them: one for *dictum de omni* environments, and one for *dictum de nullo* environments.

We have also already observed that a number of familiar rules from propositional logic are simply special cases of *dicta de omni et nullo*. For example, Modus Ponens is just a special case of the *dictum de omni*, and Modus Tollens is a special case of the *dictum de nullo*. For marketing purposes it might serve us well to simply call traditional rules by their *dicta* names, but for purposes of exposition we will find it helpful to distinguish flavors of *dicta* rules by referencing their traditional designations. This can help us see how the traditional rules are being eliminated. But we won't be *directly* eliminating traditional inference rules like Modus Ponens—first we will be generalizing the traditional rules to in-line versions of those rules, and then we will show how those inline rules can be replaced by very basic syntactic operations.

A word about in-line rules is perhaps in order here. Traditional rules like Modus Ponens (MP) require "isolated" premises, and also produce "isolated" conclusions. For example, MP cannot apply directly to (4a) to yield (4b). In order to use MP, we need to first "extract" the two premises ($A \Rightarrow B$ and A) from (4a) using two applications of Simplification (aka Conjunction Elimination), one for each conjunct. Only then can we deduce B from these premises by MP, and this step must be followed by an instance of Conjunction Introduction, which conjoins $A \Rightarrow B$ and B into (4b).

(4) a. When Ann is here Bill is here too, and Ann is here. $(A \Rightarrow B) \wedge A$

b. ∴ When Ann is here Bill is here too, and Bill is here. $(A \Rightarrow B) \wedge B$

We will say that traditional rules are instances of ROOT RULES because we like to imagine logical structures as (parse) trees, and another way of phrasing that the premises and the conclusion of traditional rules must be isolated sentences is to say that these rules can only apply to the *root* of a formula.[1] In contrast, our INLINE RULES can apply *within* trees (as long as they are applied in a polarity-sensitive way). So an inline version of MP can simply inspect (4a) for the premises $A \Rightarrow B$ and A and, seeing that they are available within that formula, drop the conclusion B into the formula, thereby concluding (4b). (Of course, not just any subformula will do as a premise, and the conclusion cannot be dropped just anywhere. Figuring out what precisely may happen here is what this chapter is all about.)

The canonical examples of *dicta de omni et nullo* are (5) and (6), where the TARGET is replaced by a hypernym and a hyponym, respectively. But there are other examples that can also be considered a part of the *dicta* paradigm. In contrast to the core cases, these additional examples change the structure of the sentence. (7) and (8) add a disjunction and a conjunction, respectively. (9) and (10) remove a conjunction and a disjunction, respectively.[2]

(5) Inline Modus Ponens (iMP) $\subset$ *dictum de omni*

a. Every pet is a dog. $\forall x\colon \neg P(x) \vee D(x)$

b. ∴ Every pet is an animal. $\forall x\colon \neg P(x) \vee A(x)$

(6) Inline Modus Tollens (iMT) $\subset$ *dictum de nullo*

a. Every animal is barking. $\forall x\colon \neg A(x) \vee B(x)$

b. ∴ Every dog is barking. $\forall x\colon \neg D(x) \vee B(x)$

(7) Inline Addition of a positive polarity target $\subset$ *dictum de omni*

a. Every pet is a dog. $\forall x\colon \neg P(x) \vee D(x)$

b. ∴ Every pet is a cat or dog. $\forall x\colon \neg P(x) \vee (C(x) \vee D(x))$

(8) Inline Addition of a negative polarity target $\subset$ *dictum de nullo*

a. Every dog is barking. $\forall x\colon \neg D(x) \vee B(x)$

b. ∴ Every angry dog is barking. $\forall x\colon \neg(A(x) \wedge D(x)) \vee B(x)$

(9) Inline Simplification of a positive polarity target $\subset$ *dictum de omni*

a. Every pet is an angry dog. $\forall x\colon \neg P(x) \vee (A(x) \wedge D(x))$

b. ∴ Every pet is a dog. $\forall x\colon \neg P(x) \vee D(x)$

[1] In fact, natural deductive systems usually include an inline rule as well. Under the assumption that two formulas are logically equivalent, Replacement may substitute an occurrence of one for the other anywhere in the formula.

[2] The conjunction in (8)/(9) is explicitly added/removed only in the logical form. In the sentence, it is an attributive intersective adjective that is added/removed.

Table 6.1 Breakdown of *dicta de omni et nullo*

	dictum de omni		*dictum de nullo*	
target polarity	positive		negative	
substitute	Inline Modus Ponens		Inline Modus Tollens	
add structure	Inline Addition	$\vee$	Inline Addition	$\wedge$
remove structure	Inline Simplification	$\wedge$	Inline Simplification	$\vee$
		$\uparrow$	structure added or removed	$\uparrow$

(10) Inline Simplification of a negative polarity target ⊂ *dictum de nullo*
a. Every dog or cat is angry. $\forall x\colon \neg(D(x) \vee C(x)) \vee A(x)$
b. ∴ Every dog is angry. $\forall x\colon \neg D(x) \vee A(x)$

There is a good reason for grouping these inference patterns under the headings of *dictum de omni* and *dictum de nullo*. All the rules comprising *dictum de omni (de nullo)* replace the target by an expression with a wider (narrower) meaning. However, from the perspective of a formal deductive system, the various inference patterns of *dicta de omni et nullo* should be understood as separate rules (or rule manifestations, if you prefer), as they modify the target in structurally different ways.

As indicated by the headings of the examples above, we begin by splitting *dicta de omni et nullo* into four formal inference rules: Inline Modus Ponens, Inline Modus Tollens, Inline Addition, and Inline Simplification. Table 6.1 shows whether they correspond to *dictum de omni* or *dictum de nullo* (i.e. whether they apply to a target of positive or negative polarity), in what way they change the structure (substitution, addition, or removal), and what kind of structure they add or remove (if they do).

Let's go into some more detail about the inline rules presented above. As our rule format will be an extension of the standard rule format, let's refresh our memory of the latter first. The root rules are usually expressed in the following format: premises ⊢ target. The purpose of this formalization is to encode form-matching requirements. For example, Modus Ponens is usually expressed as $\mathcal{A}$, $\mathcal{A} \Rightarrow \mathcal{B} \vdash \mathcal{B}$. This tells us that in order to apply MP, the antecedent premise ($\mathcal{A}$) must match the antecedent of the conditional premise ($\mathcal{A} \Rightarrow \mathcal{B}$), and that the conclusion should be formally identical to the consequent of the conditional premise, i.e. that we should conclude $\mathcal{B}$.

In standard rules, all premises are treated equal. Not so in DDS, where one of the premises of an inline rule always gets dubbed as the target. Or perhaps it is better to say that the target has two roles; on one hand, it is the replacement site, on the other hand it is also one of the premises. For example, the root Simplification

($\mathcal{A} \wedge \mathcal{B} \vdash \mathcal{A}$) requires a single premise $\mathcal{A} \wedge \mathcal{B}$, which is "transformed" into the conclusion $\mathcal{A}$.[3] The same transformation is applied by Inline Simplification, but additionally, the location of the target ($\mathcal{A} \wedge \mathcal{B}$), which can occur anywhere in the formula as long as it has positive polarity, is where the conclusion $\mathcal{A}$ ends up; in effect, the conclusion replaces the target. What we need is a way to single out the target in the formal expression of a DDS rule.

It turns out that the only information about the target which is relevant for an application of a rule is its form (as with classical rules) and its polarity. We will note the required form and polarity of the target as the subscript and superscript on $\vdash$, respectively, where the polarity can be either positive ($+$) or negative ($-$). These conventions already suffice to formalize Inline Modus Ponens and Tollens. They both require a premise of form $\mathcal{A} \Rightarrow \mathcal{C}$. The former applies to positive polarity targets, the latter to negative polarity targets. iMP requires the target to formally match the antecedent of the non-target premise; iMT, the consequent. The conclusion of iMP formally matches the consequent of the non-target premise; the conclusion of iMT, the antecedent.[4,5]

<u>Inline Modus Ponens (iMP)</u>	<u>Inline Modus Tollens (iMT)</u>
$\mathcal{A} \Rightarrow \mathcal{C} \vdash^{+}_{\mathcal{A}} \mathcal{C}$	$\mathcal{A} \Rightarrow \mathcal{C} \vdash^{-}_{\mathcal{C}} \mathcal{A}$

Inline Simplification and Addition, or DELETE and ADD, as we shall call them from now on, apply to targets of any polarity, but require either the form of the target (Delete) or the conclusion (Add) to vary with respect to the polarity of the target. As we can observe in examples (9) and (10), Delete eliminates a conjunction in a positive polarity environment, but it eliminates a disjunction in a negative polarity environment. And as we can observe in examples (7) and (8), Add introduces a disjunction in a positive polarity environment, but it introduces a conjunction in a negative polarity environment. We could formalize this dependence on polarity by composing each of the rules from two sub-rules, one applying to targets of positive polarity, the other to targets of negative polarity.[6]

[3] Logicians typically ignore whatever is going on "under the hood" here when we talk about transformation, but as we will see, this is not a trivial matter when we are investigating a natural deduction system. As we will see a bit later, when we study this process more closely, we will gain some insights into the control of sub-logical operations and the role that polarity plays in that control.

[4] Strictly speaking, these rules cannot (yet) be applied to derive inferences like (5) and (6), because those inferences contain quantifiers. As we will see in subsection 6.5.1, it is not the rules themselves which are the issue here. What is missing is the notion of p-scope, which will explain the "transparency" of the universal quantifier in these examples.

[5] In the following chapter, we will abandon iMP and iMT in favor of a single, simpler, and more general rule: Prune.

[6] In fact, we would need four sub-rules for each of Delete and Add, because it also makes a (formal) difference whether we eliminate/introduce the left or the right conjunct/disjunct. We will only pay attention to this after we introduce the notation which will allow us to compactly express the dependence on polarity.

Delete		Add	
$\vdash^{+}_{\mathcal{A}\wedge\mathcal{B}} \mathcal{A}$	(positive pol. target)	$\vdash^{+}_{\mathcal{B}} \mathcal{B}\vee\mathcal{C}$	(positive pol. target)
$\vdash^{-}_{\mathcal{A}\vee\mathcal{B}} \mathcal{A}$	(negative pol. target)	$\vdash^{-}_{\mathcal{B}} \mathcal{B}\wedge\mathcal{C}$	(negative pol. target)

However, having a separate sub-rule for each target polarity does not only make the formal expression of the rules unnecessarily verbose, it also misses a deeper point. The duality of $\wedge$ and $\vee$ evident in the above rules—in both rules, we refer to conjunction at one target polarity and to disjunction at the other polarity—is not a fluke of Delete and Add. It is rather a property we will have occasion to observe at every single point of the development of our system, and not surprisingly, the existential and universal quantifier will exhibit the very same duality.

It is actually easy to see how this duality comes about—in the case of $\wedge$ and $\vee$, it is powered by De Morgan's laws. Consider Delete. In a positive polarity environment, it behaves just like the familiar root Simplification: it eliminates a conjunction. So why does it eliminate a disjunction in a negative polarity environment? Well, this disjunction is EFFECTIVELY a conjunction! Imagine we have a formula like $\neg(A \vee B)$ containing a negative polarity disjunction. We can transform this formula into the logically equivalent $\neg A \wedge \neg B$ by "pushing" the negation over the disjunction. We have thus "transformed" the negative polarity disjunction into a positive polarity conjunction, but now we can perform the positive polarity sub-rule of Delete, which will eliminate the conjunction. We end up with $\neg A$, which is of course precisely what we would get if we directly eliminated the original negative polarity disjunction.

To concisely express the duality of $\wedge$ and $\vee$ (and $\forall$ and $\exists$), we introduce a piece of terminology accompanied by some formal notation. Summing up the discussion above, both a conjunction of positive polarity and a disjunction of negative polarity are EFFECTIVELY a conjunction, or an E-CONJUNCTION for short. In a similar fashion, both a disjunction of positive polarity and a conjunction of negative polarity are effectively a disjunction, or an E-DISJUNCTION for short. Extending the notion of effectiveness to quantifiers, the well-known Quantifier Negation equivalences $\forall x \neg\varphi \sim \neg\exists x\, \varphi$ and $\exists x \neg\varphi \sim \neg\forall x\, \varphi$ give rise to the notions of effectively universal and effectively existential quantifier. An E-UNIVERSAL quantifier is either a universal quantifier of positive polarity, or an existential quantifier of negative polarity, and an E-EXISTENTIAL quantifier is either an existential quantifier of positive polarity, or a universal quantifier of negative polarity.

Let us now turn to the notation we introduce to emphasize the duality of $\wedge$ and $\vee$, and the duality of $\forall$ and $\exists$. One could say that the definitions below introduce "aliases" of these symbols: $\wedge^{+}$ is an alias of $\wedge$, and so is $\vee^{-}$; $\forall^{+}$ is an alias of $\forall$, and so is $\exists^{-}$; and so on.

(11) Conjunction and disjunction:
  a. $\wedge^{+} := \vee^{-} := \wedge$
  b. $\vee^{+} := \wedge^{-} := \vee$

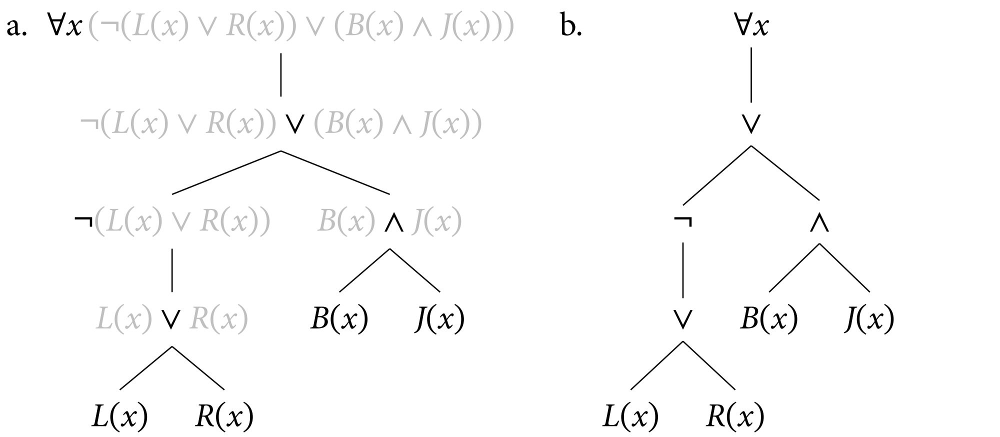
(12) Universal and existential quantifier:
a. $\forall^+ := \exists^- := \forall$
b. $\exists^+ := \forall^- := \exists$

To see how the terminology and the notation we have introduced above help us concisely formalize our polarity-dependent inline rules, we turn back to Delete and Add. It is now quite simple to express what these rules do. Delete eliminates effective conjunctions; more precisely, it replaces an effective conjunction with one member of the junction. Add introduces effective disjunctions; more precisely, it replaces the target with an effective disjunction consisting of the target and an arbitrary formula. In the symbolic formalization below, τ is a variable standing for either $+$ or $-$. In the statement of Delete, τ superscripts both $\vdash$ and $\wedge$. The former superscript stands for the polarity of the target, while the latter decides whether the target should be a conjunction ($\wedge^+ = \wedge$) or a disjunction ($\wedge^- = \vee$). In effect, we state that Delete applies to conjunctions of positive polarity and to disjunctions of negative polarity (i.e. to e-conjunctions in short).

Delete		Add	
$\vdash^{\tau}_{\mathcal{A}\wedge^{\tau}\mathcal{B}}\ \mathcal{A}$	(right)	$\vdash^{\tau}_{\mathcal{B}}\ \mathcal{B} \vee^{\tau} \mathcal{C}$	(right)
$\vdash^{\tau}_{\mathcal{A}\wedge^{\tau}\mathcal{B}}\ \mathcal{B}$	(left)	$\vdash^{\tau}_{\mathcal{B}}\ \mathcal{C} \vee^{\tau} \mathcal{B}$	(left)

Now, let us show you how we can visualize rule application in DDS. Instead of writing down the formulas linearly, we'll write them down as (parse) trees. This should help us better see what is happening. The trees below both encode exactly the same information as the formula; we normally use the concise variant shown on the right.

(13) Every labrador or rottweiler is barking and jumping.
$\forall x\colon \neg(L(x) \vee R(x)) \vee (B(x) \wedge J(x))$

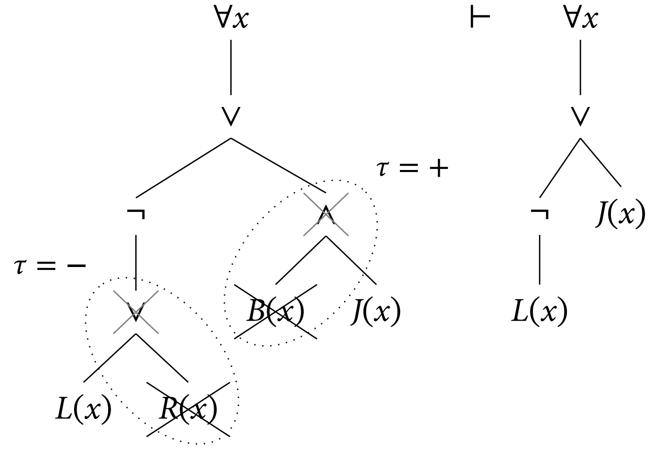


First, let's take a look at a (double) application of Delete. We find a conjunction of positive polarity and remove the node, along with one of its children. We find a disjunction of negative polarity and remove the node, along with one of its children. In any order. Done.

(14) a. Every labrador or rottweiler is barking and jumping.
$\forall x\colon \neg(L(x) \vee R(x)) \vee (B(x) \wedge J(x))$

b. ∴ Every labrador is jumping. $\forall x\colon \neg L(x) \vee J(x)$

c.

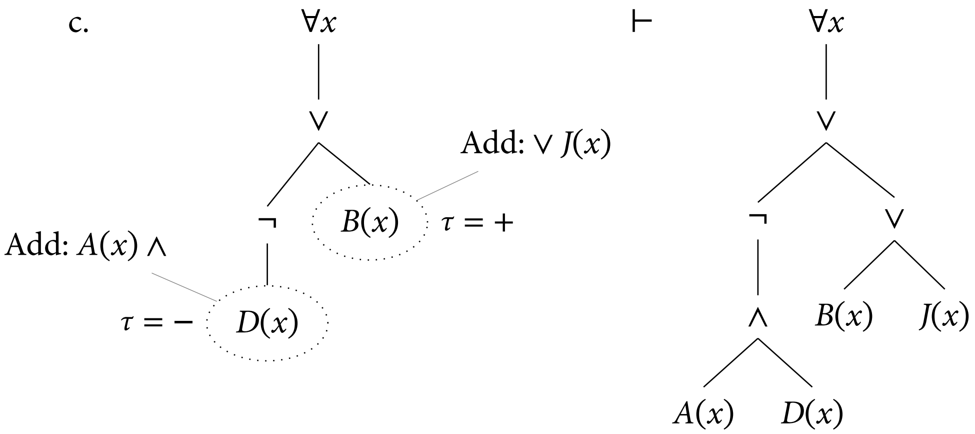


Next, a (double) application of Add. It introduces a disjunction in the positive polarity environment, and its dual, conjunction, in the negative polarity environment.

(15) a. Every dog is barking.

b. ∴ Every angry dog is barking or jumping.

c.



Actually, we have another example of Add, exemplifying a very useful conditional variant of the rule. If anything ($\mathcal{C}$) can be disjoined to a positive target, one can also disjoin a negation of anything ($\neg\mathcal{C}$), and $\neg\mathcal{C} \vee \mathcal{B}$ can be abbreviated as $\mathcal{C} \Rightarrow \mathcal{B}$—but note that this only works under positive polarity, thus $\vdash^{+}$ in the rule.

<u>Add</u>

$\vdash^{+}_{\mathcal{B}} \mathcal{C} \Rightarrow \mathcal{B}$ (conditional)

(16) a. [Two plus two is four.]$_P$
b. ∴ If [you know any math]$_K$, then [two plus two is four]$_P$.
c. $P \vdash \neg K \vee P \equiv K \Rightarrow P$

In this section, we have broken down *dicta de omni et nullo* into several inline rules (Inline Modus Ponens and Tollens, Add and Delete), and introduced some terminology and notation which will help us talk about the role of polarity in these inline rules. It is now time to take a broader view on how DDS works.

6.2 Going Dynamic and Simplifying Deduction

The simplest familiar kind of deductive system, in terms of conceptual complexity, is a Hilbert-style deductive system. In that system, a deduction is represented as a sequence of formulas (lines). Every formula is either an axiom (or a hypothesis) or is justified by some rule of inference, with some previous lines functioning as premises of that rule (and the current line being the conclusion). What is important for us to recognize here is that, more often than not, we need to refer to lines in the deduction that do not immediately precede the line being inferred. For example, take a look at the deduction below. On the right of every line, there is a justification of that line, which tells us which rule was deployed and which lines were used as its premises. You can see that reference to non-immediately preceding lines is commonplace.

(17)	a. All aliens are cats.	$\forall x\colon A(x) \Rightarrow C(x)$
	b. Some alien barks.	$\exists x\colon A(x) \wedge B(x)$
	c. ∴ Some cat barks.	$\exists x\colon C(x) \wedge B(x)$

(18)	1. $\forall x\,(A(x) \Rightarrow C(x))$	(hypothesis)
	2. $\exists x\,(A(x) \wedge B(x))$	(hypothesis)
	3. $A(a) \wedge B(a)$	(EI 2, $x \mapsto a$)
	4. $A(a)$	(Simp 3)
	5. $A(a) \Rightarrow C(a)$	(UI 1, $x \mapsto a$)
	6. $C(a)$	(MP 5, 4)
	7. $B(a)$	(Simp 3)
	8. $C(a) \wedge B(a)$	(Conj 6, 7)
	9. $\exists x\,(C(x) \wedge B(x))$	(EG 8, $a \mapsto x$)

In the system we're developing, this will not be the case. An application of a rule will only use the immediately preceding line. What this really means is that we'll dispense with the image of a deduction as an ever-growing list of formulas in favor of seeing it as a DYNAMIC FORMULA changing with every step of the deduction, with no record of the earlier history of the deduction. We thus arrive at the promised DYNAMIC DEDUCTIVE SYSTEM (DDS).

How do we use Inline Modus Ponens and Tollens in such a system? They require two premises after all, the conditional premise ($A \Rightarrow B$), and the antecedent premise (a positive polarity A for iMP, and a negative polarity B for iMT), and they cannot both equal the immediately preceding line! Look at the following one-step deduction of (17): the conditional premise does not immediately precede the conclusion.[7]

(19)	1. $\forall x\,(A(x) \Rightarrow C(x))$	(hypothesis)
	2. $\exists x\,([A(x)]_i \wedge B(x))$	(hypothesis)
	3. $\exists x\,(C(x) \wedge B(x))$	(iMP 1, 2_i)

On our approach, the antecedent premise is not the immediately preceding line, either, it's not even a line, it's a *constituent* within the (immediately preceding) line, which we mark by a subscript (i) in example (20) below. Can both premises be constituents of the immediately preceding line? Sure! Let's imagine that we don't start with two isolated hypotheses as above, but that we are given the same information as a single hypothesis in the form of a conjunction, as in (20.1). Then, all the information a rule such as iMP needs is present in the immediately preceding line: iMP takes two constituents of the previous line (marked as i and j below) as premises, and replaces one of them (the antecedent premise, i) by the conclusion. In other words, the non-target, conditional premise of iMP ($\mathcal{A} \Rightarrow C$) is a constituent of the preceding line, just like the target. (iMP yields (20.2), which is not yet the conclusion of (19)—but it is trivial to get the intended conclusion (20.3) out of this formula by Delete.)

(20)	1. $\exists x\,([A(x)]_i \wedge B(x)) \wedge [\forall x\,(A(x) \Rightarrow C(x))]_j$	(hypothesis)
	2. $\exists x\,(C(x) \wedge B(x)) \wedge \forall x\,(A(x) \Rightarrow C(x))$	(iMP $1_i, 1_j \mapsto 1_i$)
	3. $\exists x\,(C(x) \wedge B(x))$	(right-Delete 2)

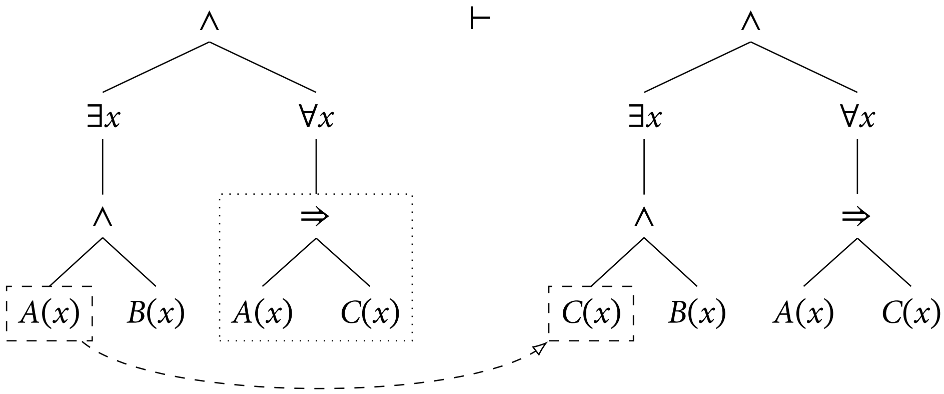
Compare this essentially single-step deduction to the seven-step Hilbert-style deduction in (18).[8] The only step that remains is the application of Modus Ponens, now inlined. What this tells us is that the rest was nothing but bureaucratic bookkeeping. The prior steps disassemble the hypotheses to arrive at the premises of Modus Ponens; the subsequent steps reassemble the hypothesis containing the target, but with the target replaced by the conclusion of Modus Ponens. Crucially, all this bookkeeping was needed just because Modus Ponens is a root rule. It cannot apply to premises embedded within the hypotheses. Inline Modus Ponens, on the other hand, is formulated to do just that.

[7] Yet again, we ignore the universal quantifier for reasons of exposition. See also footnote 4.

[8] We are actually being generous when we say that (18) consists of only seven steps. Existential Instantiation used in this deduction is in fact a derived rule invoking Deduction Theorem, which means it introduces a subderivation; for details, see subsection 8.2.2 and section A.2, and Mendelson (1997, pp. 74–75). Writing down this deduction without EI is non-trivial and *much* longer.

The ad hoc subscript notation used in (19) and (20) is not very pretty. The ideal way of viewing deductions of Dynamic Deductive System is by observing how the dynamic formula, represented as a tree, evolves through time.[9] Here we show the step from (20.1) to (20.2).

(21)



Of course, we can't really do this in a book all the time—the publisher would never let us get away with that! Luckily, the fact that we will only need to refer to the immediately preceding line of the deduction allows us to introduce some notational flourishes which will make the notation for justifications much more visual. The basics of our new notation are as follows.

- The $\underbrace{\text{target}}_{\substack{+/-\\ \text{RULE}}}$ is indicated by underbracketing. Its polarity is noted above the horizontal line of the bracket. The name of the applied rule is typeset below the line.
- A non-target $\underline{\text{premise}}$ is indicated by underlining.
- The $\overbrace{\text{replacement}}$ is indicated by overbracketing (in the next line).

We can now rewrite the deduction of inference (17), repeated below, in the new notation.

(22) a. All aliens are cats. Some alien barks.
b. ∴ (All aliens are cats.) Some cat barks.

(23) 1. $\forall x\,(\underline{A(x) \Rightarrow C(x)}) \wedge \exists x\,(\underbrace{A(x)}_{\substack{+\\ \text{iMP}}} \wedge B(x))$
2. $\forall x\,(A(x) \Rightarrow C(x)) \wedge \exists x\,(\overbrace{C(x)} \wedge B(x))$

[9] The dynamic nature of our system is best displayed in live graphical presentations. We plan on developing a website where you can play with DDS in a more visual manner. Once done, it will be accessible at https://spj.ff.uni-lj.si/dds.

While iMP and iMT substitute the entire target for the conclusion, Add adds material to the target, and this material can be added either to the left or to the right of the target. We indicate this in two ways. First, we align the name of the rule either to the left or to the right of the target underbracket. Second, we split the replacement overbracket into the thinner part and the thicker part: The material corresponding to the original target can be found under the thinner part; the added material and the junction we have introduced can be found under the thicker part. (We will use the same convention for Copy, which we will introduce in the following section.)

(24) a. Every dog is barking.
  b. ∴ Every angry dog is barking or jumping.

(25) 1. $\forall x\colon \neg \underset{\textsc{add}}{\underbrace{D(x)}_{-}} \vee B(x)$
  2. $\forall x\colon \neg(\overbrace{A(x) \wedge D(x)}) \vee \underset{\textsc{add}}{\underbrace{B(x)}_{+}}$
  3. $\forall x\colon \neg(A(x) \wedge D(x)) \vee (\overbrace{B(x) \wedge J(x)})$

We use a similar alignment and thin–thick convention with Delete. Here, the rule name is aligned to the side where the material is deleted, and the underbracket is split so that the material which will be deleted (including the connective) occurs above the thicker part. (We will use the same convention with Prune, which we will develop in the next chapter.)

(26) a. Every labrador or rottweiler is barking and jumping.
  b. ∴ Every labrador is jumping.

(27) 1. $\forall x\colon \neg(\underset{\textsc{delete}}{\underbrace{L(x) \vee R(x)}_{-}}) \vee (B(x) \wedge J(x))$
  2. $\forall x\colon \neg\overbrace{L(x)} \vee (\underset{\textsc{delete}}{\underbrace{B(x) \wedge J(x)}_{+}})$
  3. $\forall x\colon \neg L(x) \vee \overbrace{B(x)}$

Equipped with this new view on deductions and some notational conventions, let us see what we can accomplish.

6.3 The Rule Copy and the Need for Premise Scope

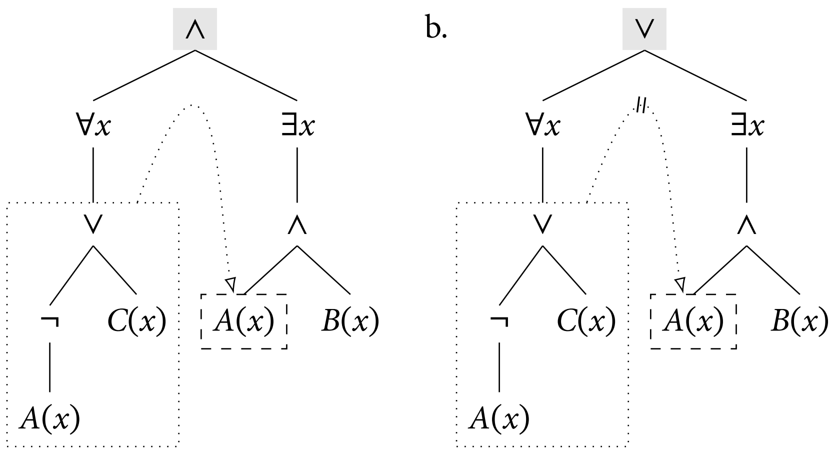
By now, we are familiar with the general architecture of DDS—we see a deduction as an unfolding of a dynamic formula governed by inline rules—and we know that these inline rules can take their premises from any location within the formula, and also drop the conclusion elsewhere in the formula. In principle. However, as you have probably guessed, not every constituent will be able to function as a premise for the given target. For example, while we can substitute 'cat' for 'alien' in (28), we can't do it in (29).

(28) a. Every alien is a cat, and some alien barks.
b. ∴ (Every alien is a cat, and) some cat barks.

(29) a. Either every alien is a cat, or some alien barks.
b. $\not\therefore$ Either every alien is a cat, or some cat barks.

What this means is that 'every alien is a cat' can function as a premise of iMP for the target 'alien' (of 'some alien barks') in (28a) but not in (29a). These sentences have logical forms (30a) and (30b), respectively. The only difference between them is the conjunction vs. disjunction at the root. While it is clear, semantically, that the disjunction in that place should block the inference, what is the precise syntactic formulation of this?

(30) a. b.



Figuring out the answer to the above question is the main challenge of the development of DDS. Stated more generally and formally, the question goes like this.

Let us say that constituent β is in the PREMISE SCOPE (P-SCOPE) of constituent α (of the same formula)—we will also say that α p-scopes over β—iff α can function as a premise for target β. So the first conjunct, 'every alien is a cat' p-scopes over 'alien' (of 'some alien barks') in (28a) but in (29a), the first disjunct does not p-scope over 'alien'.

The question is, then, what is the precise *syntactic* characterization of p-scope? We provide this characterization in sections 6.4 and 6.5. But before we start explaining it, it makes sense to introduce a very simple but important inline rule, Copy, as it will help us investigate the syntax of p-scope.[10]

[10] In principle, the availability of a premise at a certain target, and thus the notion of p-scope, might depend on the particular rule, i.e. different constituents might be available as premises for iMP, iMT

Take a look at the (clearly valid) inferences in (31) and (32).[11] We are going to offer a new perspective on how they are derived. The logical forms of the sentences determine whether we can Copy 'dog' next to 'barking' (ignoring some syntactic details for now). In our usage, "copying next to" can be understood as conjoining a formula of the same form as the premise to the target.

(31) a. Some angry dog is barking. COPY — $\exists x\colon (A(x) \wedge D(x)) \wedge \underbrace{B(x)}_{\text{COPY}}$

b. ∴ Some angry dog is a dog and is barking. — $\exists x\colon (A(x) \wedge D(x)) \wedge (D(x) \wedge B(x))$

(32) a. Every angry dog is barking. COPY — $\forall x\colon \neg(A(x) \wedge D(x)) \vee \underbrace{B(x)}_{\text{COPY}}$

b. ∴ Every angry dog is a dog and is barking. — $\forall x\colon \neg(A(x) \wedge D(x)) \vee (D(x) \wedge B(x))$

Let us continue with some examples that can be analyzed by propositional logic alone. They will make it easier to understand what's going on. To wit: the following cases are just like the examples above, minus the quantifiers. Copying *R* next to *H* is valid.

(33) a. It is raining. We're at home. COPY — $R \wedge \underbrace{H}_{\text{COPY}}$

b. ∴ It is raining. It is raining and we're at home. — $R \wedge (R \wedge H)$

(34) a. When it is raining, we stay at home. COPY — $\neg R \vee \underbrace{H}_{\text{COPY}}$

b. ∴ When it is raining, it is raining and we stay at home. — $\neg R \vee (R \wedge H)$

Now we finally present an example where Copying is not valid: (35). However, if we start from the same hypothesis but negate the copied premise, as in (36), we do reach a valid conclusion!

(35) a. It is raining, or we're on a trip. *COPY — $R \vee \underbrace{T}_{\text{*COPY}}$

b. ∴̸ It is raining, or it is raining and we're on a trip. — $R \vee (R \wedge T)$

and Copy. However, it turns out that this is not the case. *What* we can use is independent of *how* we use it; the proof of soundness in the Appendix makes this abundantly clear.

[11] These are actually examples of the familiar tests for conservativity of determiners. We'll say much more about conservativity in Chapter 9.

(36) a. It is raining, or we're on a trip. $\underset{-}{R} \lor T$

"COPY+NEGATE" "COPY+NEGATE"

b. ∴ It is raining, or it is not raining and we're on a trip. $R \lor (\neg R \land T)$

What this shows is that premise scope comes in two varieties. Up until now, we had been considering POSITIVE P-SCOPE, where it is possible to Copy the premise as it is. (36) is our first example of NEGATIVE P-SCOPE, where we may Copy the premise, but we need to negate it while doing so. In the justifications (to the right of the examples), we will indicate the polarity of the p-scope of the premise by putting the polarity marker above the underline, as shown below, where we repeat two deductions using the upgraded notation.

(37) a. It is raining. We're at home. $\underset{+}{R} \land H$

COPY COPY

b. ∴ It is raining. It is raining and we're at home. $R \land (R \land H)$

(38) a. It is raining, or we're on a trip. $\underset{-}{R} \lor T$

COPY COPY

b. ∴ It is raining, or it is not raining and we're on a trip. $R \lor (\neg R \land T)$

Before we continue, we need to emphasize a point about polarity—or rather about manifestations of polarity. We are by now quite familiar with the notion of polarity. The polarity of a constituent is determined by the number of negations dominating it; it has positive/negative polarity iff it is dominated by an even/odd number of negations. But now, we have defined a new kind of polarity a constituent can have: the polarity of its p-scope at a certain target, or P-SCOPE POLARITY for short. Although they are very closely related on a conceptual level, these two notions are not the same thing from a technical perspective. Therefore, whenever there is a chance of confusion, we shall refer to the original notion as CONSTITUENT POLARITY.

We also emphasize that both p-scope and constituent polarity are relative notions. P-scope polarity is relative to the given target, as described above. Constituent polarity is relative to an ancestor. Usually, we take that ancestor to be the root of the formula, but not always. So, in $\neg[\neg(A \land B) \land C]_i$, A has positive constituent polarity within the entire formula, but negative constituent polarity within constituent i.

Returning to Copy, the examples so far have all conjoined the premise to the target. A bit more formally, the replacement generated by the rule was a conjunction of the premise and the target. There is another variant of Copy, however: we can simply replace the target by the copy of the premise, as below. (In the justification, this is indicated by centering the rule name under the underbracket.)

(39) a. Some $\underset{+}{\underline{\text{dog}}}$ is $\underbrace{\text{barking}}_{\text{COPY}}$. $\qquad \exists x\colon \underset{+}{\underline{D(x)}} \wedge \underbrace{B(x)}_{\text{COPY}}$

b. ∴ Some dog is a $\overline{\text{dog}}$. $\qquad \exists x\colon D(x) \wedge \overline{D(x)}$

Below, we formalize Copy as developed thus far. The first notational novelty is the superscript on the premise, indicating its p-scope polarity with respect to the target. Second, these rules use a notational aid (40) similar to the dual connectives from section 6.1: $\neg^\sigma \mathcal{A}$ in the formal statement of Copy means that when a premise p-scopes over the target positively, it is Copied as it is; if the p-scope is negative, the copy is negated.

Copy (for a positive polarity target)

$\mathcal{A}^\sigma \vdash^+_{\mathcal{B}} \mathcal{B} \wedge \neg^\sigma \mathcal{A}$ (right)

$\mathcal{A}^\sigma \vdash^+_{\mathcal{B}} \neg^\sigma \mathcal{A}$ (center)

$\mathcal{A}^\sigma \vdash^+_{\mathcal{B}} \neg^\sigma \mathcal{A} \wedge \mathcal{B}$ (left)

(40) Absence and presence of negation (for any formula φ):[12]

a. $\neg^+ \varphi :\equiv \varphi$

b. $\neg^- \varphi :\equiv \neg\varphi$

However, by requiring a positive polarity target (the superscript on $\vdash$), the above formulation of Copy clearly tells only one half of the story. And indeed, we have not yet provided any examples of Copy targeting a constituent of negative polarity! Let's remedy that.

Let's start with center Copy to avoid the complications arising from introducing a connective. (41) shows the obviously valid conclusion we get when we Copy 'it is raining' (R) over 'we're not on a trip' ($\neg T$). Now, when we Copy 'it is raining' (R) over 'we're on a trip' (T; note the ommision of 'not'!) in (42), we expect to get a logically equivalent sentence. However, Copying (or applying any inline rule, for that matter) can only change the target. In particular, the negation above the target must stay there. Clearly, to get a logically equivalent sentence, we must negate the copy before integrating it in the target position.

(41) a. $\underset{+}{\underline{\text{It is raining}}}$ and $\underbrace{\text{we're not on a trip}}_{\substack{+\\ \text{COPY}}}$. $\qquad \underset{+}{\underline{R}} \wedge \underbrace{\neg T}_{\substack{+\\ \text{COPY}}}$

b. ∴ It is raining, and $\overline{\text{it is raining}}$. $\qquad R \wedge \overline{R}$

(42) a. $\underset{+}{\underline{\text{It is raining}}}$ and we're not $\underbrace{\text{on a trip}}_{\substack{+\\ \text{COPY}}}$. $\qquad \underset{+}{\underline{R}} \wedge \neg \underbrace{T}_{\substack{-\\ \text{COPY}}}$

b. ∴ It is raining, and it is not the case that $\overline{\text{it is not raining}}$. $\qquad R \wedge \neg\overline{\neg R}$

[12] The notation is chosen so that the polarity of φ in $\neg^\pi \varphi$ equals π.

The target has positive and negative polarity in (41) and (42), respectively. We conclude, then, that Copying over a negative polarity target introduces a negation (immediately) above the copy. (Once again, we can't just simply delete the upper negation, as it is not a part of the target. Furthermore, the negation inducing the negative polarity in the target is *not* always *immediately* above it!)

We can, however, set up the Copy operation so that the negations induced by the negative p-scope polarity of the premise and by the negative constituent polarity of the target cancel each other.[13] For example, the application of Copy in (43) introduces no negation. (Incidentally, Copying a disjunct over the other branch—modulo negations on either the premise or the target—always gets us an instance of the Law of Excluded Middle.)

(43) a. It is $\underbrace{\text{raining}}_{-}$, or we're not $\underbrace{\text{at home}}_{-}$. $\qquad \underset{-}{\underline{R}} \vee \neg \underset{\substack{-\\ \text{COPY}}}{\underline{H}}$

b. ∴ It is raining, or it is not ~~the case that it is not the case that it is not~~ raining. $\qquad R \vee \neg \overline{R}$

Moving on to structure-introducing variants of Copy, observe that in the case of a positive polarity target, Copy complemented Add. The latter introduces a disjunction. Copy introduces a conjunction. We have seen that for a negative polarity target, Add introduces a conjunction. Could Copy then introduce a disjunction for a negative polarity target?

(44) a. $\underbrace{\text{It is raining}}_{+}$ and we're not $\underbrace{\text{on a trip}}_{\substack{-\\ \text{COPY}}}$. $\qquad \underset{+}{\underline{R}} \wedge \neg \underset{\substack{-\\ \text{COPY}}}{\underline{T}}$

b. ∴ It is raining, and it is not the case that we're on a trip or it is not raining. $\qquad R \wedge \neg(T \vee \neg R)$

Clearly, this follows even if the abundance of negations taxes our minds a bit. Summing up, while Add introduces an effective disjunction, Copy introduces an effective conjunction. Of course, Add operates without a non-target premise. Copy, on the other hand, requires an "additional license" in the form of the non-target premise (p-scoping over the target), which gets duplicated in the non-target member of the effective conjunction introduced by the rule.

We are almost ready to write down the full formal statement of Copy. To capture the cancelling effect of negative p-scope accompanied by negative polarity target, exhibited in (43) above, we introduce a very natural operation of "multiplying" polarities. The idea is that if α has polarity π within β, and β has in turn polarity ρ within γ, then α has polarity $\pi\rho$ within γ. As positive/negative polarity corresponds to an even/odd number of dominating negations, the products are as shown below.

[13] This property of Copy will buy us a way to effect Double Negation Elimination in subsection 6.5.2.

(We call them products because we get the same results if we imagine $+1$ for $+$ and -1 for $-$.)

(45) $++ := +$
$+- := -$
$-+ := -$
$-- := +$

The superscript of $\neg$ in the formulation of Copy below contains the product of the p-scope polarity of the premise (σ) and the target polarity (τ). The cancelling effect arising when both are negative then obtains because $-- = +$.

Copy

$$\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \mathcal{B} \wedge^{\tau} \neg^{\sigma\tau}\mathcal{A} \qquad \text{(right)}$$

$$\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \neg^{\sigma\tau}\mathcal{A} \qquad \text{(center)}$$

$$\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \neg^{\sigma\tau}\mathcal{A} \wedge^{\tau} \mathcal{B} \qquad \text{(left)}$$

It might be interesting to note that supplementing the rules introduced in the preceding sections (iMP, iMT, Add, and Delete) with Copy actually already results in a complete system (of propositional logic, as we haven't talked about quantification yet). In other words, this rule is the only thing that the *dicta de omni et nullo* rules lack in order to be complete![14] But of course, we haven't yet given you all the details on how Copy and other inline rules work—we have yet to develop a syntactic characterization of p-scope. This is a task we set upon in the following section.

6.4 Premise Scope

So far in this chapter we have presented the general architecture of our Dynamic Deductive System. The central idea was that basing the deductive system on polarity-sensitive inline rules affords us a way to conceptualize the proofs as dynamically evolving formulas. We have also defined several inline rules, but more importantly, we have started to develop the notion of premise scope—a notion that tells us where an inline premise can be applied.

That accomplishment was significant, because it allows us to apply very basic syntactic operations that are not in themselves logical. The "logic" comes out of the role of polarity in the application of the basic syntactic operations. Our treatment

[14] We will never actually prove this statement, because by the time we turn to completeness in section 7.3, we will have abandoned iMP and iMT in favor of Prune.

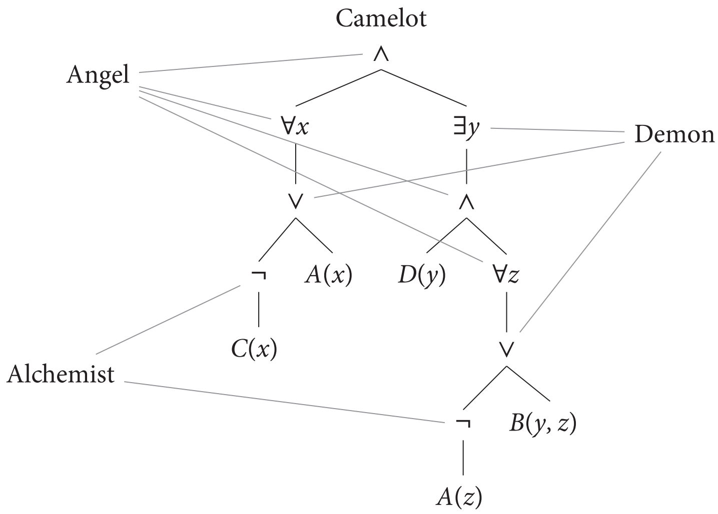


Figure 6.1 The Kingdom of Logic 'Every cat is an alien. Some dog barks at every alien.' $\forall x\,(\neg C(x) \vee A(x)) \wedge \exists y\,(D(y) \wedge \forall z\,(\neg A(z) \vee B(y,z)))$

of Copy is thus the first of several such moves. The key idea is that the basic rules, like Copy, are non-logical syntactic operations. The logic extrudes from our attention to the polarity of the environments in which those syntactic operations are applied. One way to look at this project in broad view, is that matters are even simpler than the Medieval logicians imagined when they proposed two basic rules of inference that are polarity-sensitive. We believe that the *logical* rules might actually dissolve to zero in number if we handle polarity correctly.

We hope that sounds like an audacious claim, because it is nothing if not audacious. For it to work, everything depends on the successful handling of polarity, and part of that success depends on correctly developing the idea of p-scope, which guides the application of our syntactic operations. While the formal definition of p-scope may look daunting to non-technical readers, the basic idea and execution of the rule is actually quite simple. To ease our way in, we thus begin with a story that illustrates how p-scope works.

Since this book begins with the Holy Grail of Medieval Logic, we will use a Medieval allegory. Let's imagine a Medieval Kingdom that has the downward branching structure of a logical tree, like in Figure 6.1. At the topmost node, is the royal city, which we will call Camelot, because why not? Camelot represents the root node of our tree. Roads (the branches of our tree) head South from Camelot, passing through towns on the way. Those towns are the logical connectives $\wedge$, $\vee$, $\forall$, $\exists$, and $\neg$;[15] in some of those towns ($\wedge$, $\vee$) the roads will branch.

[15] We put the quantifiers in the picture as well, although we have not really discussed them yet. We will briefly talk about them in subsection 6.5.1, and then again, in more detail, in section 7.2.

The Southern points where the roads terminate are the villages. We can think of the villages as being the terminal nodes or leaves of the tree, and they will be atomic formulas, like $A(x)$.

Each town (but no village) is controlled by an angel, a demon, or an alchemist. In our allegory, the towns that the angels control correspond to conjunctions ($\wedge$) and universal quantifiers ($\forall$). The towns that the demons control are disjunctions ($\vee$) and existential quantifiers ($\exists$) in the allegory. Alchemist towns are negations ($\neg$) in our story.

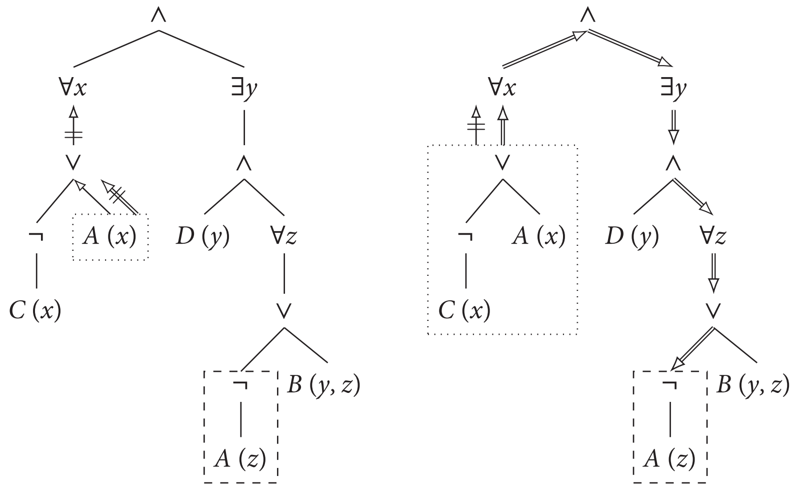
Here is where our knight—let's call him Perceval—comes into the story. From time to time, Perceval is called upon to deliver the map of some region (the premise in our story) to another location in the kingdom (the target). Let's assume that this region map is only kept in the capital of the region (a region of the kingdom is a subtree in our story, so the capital is of course the root of this subtree) and let's pretend that Perceval lives in this very town or village.[16] So Perceval's mission is to deliver the map of his home region (the premise) from the capital of this region (the location of the premise) to some other town or village in the kingdom (the target). The details of his mission will be of importance a bit later; but first and foremost, we are interested in seeing if Perceval can even reach the destination he was asked to deliver the map to—the success of his mission will indicate that his hometown p-scopes over the target, while mission failure will indicate lack of p-scope between these locations.

And here is where the story gets interesting, for earlier we mentioned that some of the towns are controlled by angels and some by demons, and as it turns out the angels and the demons do not let just anybody into their town. You see, angels are quite fond of gold, and they will only let Perceval enter if he is wearing gold armor (which represents positive polarity). Demons, on the other hand, are quite fond of silver, and they will only let Perceval enter if he is wearing silver armor (which represents negative polarity). The armor checkpoints (in both angel and demon towns) are located only at the southern gates into town. (The conjunction and disjunction towns have two gates in the south, the south-west and the south-east gate. Both of them are checkpoints.) Perceval's armor is not checked when he enters a town from the north, nor it is checked when he exits the town, in any direction.

The alchemist towns are a different story altogether. There are no checkpoints in these towns, but the alchemists controlling them take great delight in changing the metallic composition of Perceval's armor. If he shows up in gold, they will change his armor to silver. If he shows up in silver, they will transmute it to gold.

When Perceval starts his journey, he has two options, he can put on golden armor or silver armor. And perhaps sadly, that is the only thing he really has

[16] If we didn't pretend that Perceval's hometown is the capital of the "premise region," we would soon run out of names for our knights.



(a) Unsuccessful mission (b) Successful mission

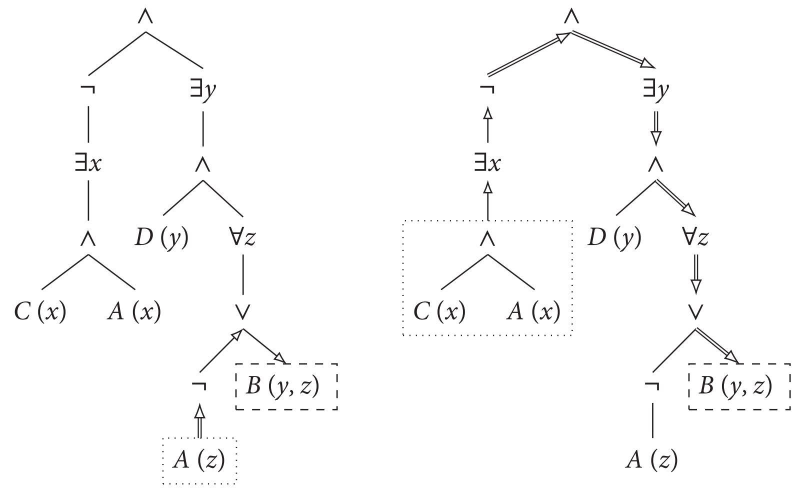
Figure 6.2 Perceval's mission: Get from the premise to the target

control over. Whether he will be successful in his mission or not is already determined. Perhaps on the bright side, if he gets stuck somewhere he can turn back and try the other kind of armor (e.g. silver instead of gold), but that isn't much consolation. For example, suppose he was leaving the village $A(x)$ in Figure 6.2a, his intended destination being $\neg A(z)$. If he leaves his village in golden armor (double arrow) he will be turned back in the very first town he encounters (the disjunction node), which is a demon town. If he starts with silver armor (single arrow) he will get through the first town but not the second (the universal quantifier).[17] He is definitely not making it to Camelot, which he needs to pass through on his way to the target in Figure 6.2a.

But if, in the same Kingdom, Perceval starts his journey elsewhere, for example from $\neg C(x) \vee A(x)$ as shown in Figure 6.2b, he can make it to $\neg A(z)$. Sure, starting with the silver armor (single arrow) does not get him through $\forall x$, but donning the golden armor (double arrow), he can pass through both $\forall x$ and Camelot, as both of these are Angel towns, and afterwards, he can travel unobstructed South to his destination—remember there are no armor checkpoints at the northern gates into towns!

Perhaps you thought that knights of the round table were clever and resourceful, but we are sorry to inform you that Perceval and the other knights are, for a lack of a better expression, dumb. All they can really do is don their armor, head North, and hope for the best. And we emphasize this point not to be mean to Perceval and the other knights, but to illustrate what a very simple (indeed, bone dumb)

[17] When we need to indicate more than one of Perceval's attempts at the journey in a single tree, we overlay the arrows corresponding to one attempt with the branches of the tree; the arrows representing the other attempt are shifted a bit to the side. And remember, a double arrow means that Perceval is wearing the golden armor on that leg of the journey, while a single arrow corresponds to the silver armor.



(a) Starting with golden armor: positive p-scope

(b) Starting with silver armor: negative p-scope

Figure 6.3 The effect of Alchemists, and p-scope polarity

operation this really is. And just to leave our allegory for a moment, we also want to emphasize that the combination of this astoundingly dumb knight and some astoundingly simple syntactic operations are all that we are going to need to cash out our theory of logical deduction, *if* we pay attention to polarity. We will develop this thought in detail in the next chapter, but first we need to return to our allegory and explore some of the consequences of sending such dumb knights on such perilous missions.

It is important to consider the role of the alchemists in Perceval's travels. Say that the first checkpoint town (that is, a town controlled by either an angel or a demon) on his way north is a demon town, but there is an alchemist (precisely one) between his hometown and this demon town, as in Figure 6.3a. Then if Perceval dons golden armor at home, the alchemist will transmute it to silver, and the demon will let him pass. And it can also happen that an alchemist is not immediately north of Perceval's hometown but somewhere else on Perceval's way north, like in Figure 6.3b. Here, Perceval is very lucky, because the Alchemist is positioned just right, between the Demon existential quantifier and the Angel conjunction, so if Perceval starts with silver armor to get through the existential, the Alchemist will transmute it to gold just in time for the Perceval to be allowed to pass the conjunction.

The time has come to tell you more about Perceval's mission. The mission is to deliver the map (which, recall, is the premise in our story) to another location in the kingdom. However, as this is a secret map, he carries it concealed on his person, etched on the inside of his armor. Gold being the metal liked by angels, the map can be etched onto the armor as it is, but etching the map onto silver, the choice of demons, can only be achieved if it is etched in the negative. The effect of all this is that if Perceval starts out with golden armor, the map will arrive at the destination as it is, but if he starts out with the silver armor, the map will arrive at the destination in the negative. But the map is really the premise, and in the previous section, we have seen that a premise can have either positive or negative p-scope, depending on whether it is seen as-is or negated by the rule. So the p-scope polarity corresponds to Perceval's starting armor: if he starts out with golden armor, p-scope polarity is positive; if he starts out with silver armor, p-scope polarity is negative.

As we noted, if Perceval gets blocked by an angel or a demon at a gate, maybe he has put on the wrong armor. Then he can return home and put on the other kind or armor, but if that doesn't work he's just plain out of luck. If, for example he was at $A(x)$ in Figure 6.2a and trying to travel to $\neg A(z)$, his mission will be a failure. There is no way for him to reach that point, which means (on our allegory) that the premise does not p-scope over the target. The mission fails. But if the mission is successful, it is usually only one of the starting choices of armor that works. In Figure 6.3a, starting with the golden armor gets Perceval to his destination, but starting with the silver armor would not; in Figure 6.3b, the situation is just the reverse. But as we have explained above, p-scope polarity is determined by Perceval's starting armor. Figures 6.3a and 6.3b therefore illustrate instances of positive and negative p-scope, respectively.

The picture we have painted for p-scope should be clear by now, and fundamentally this is all you need to grasp what p-scope is. However, in the chapters to follow we will sometimes find it useful to appeal to some borderline situations not covered by the allegory presented above. We will fill these tiny gaps in the next section, while getting some hands-on experience in deploying the p-scope relation in DDS deductions.

6.5 Premise Scope in Action

In the previous section we got acquainted with Perceval, the knight of the round table going about various missions in the Kingdom. The main idea was that if Perceval can reach his destination from his hometown (and we have pretended that he resides in the town or village where he receives the map he needs to deliver), then his hometown (the premise) p-scopes over the destination (the target), and a little detail about Perceval's journey also told us about the polarity of the p-scope: it was positive if Perceval started his journey with golden armor, and negative if he started with silver armor.

But if we want this to be more than just an amusing story, we have to show you that the syntactic characterization of p-scope which extrudes from Perceval's missions actually matches the intended semantics of p-scope: if Perceval can reach node β from node α, α should be able to function as a premise of an inline rule targeting β.

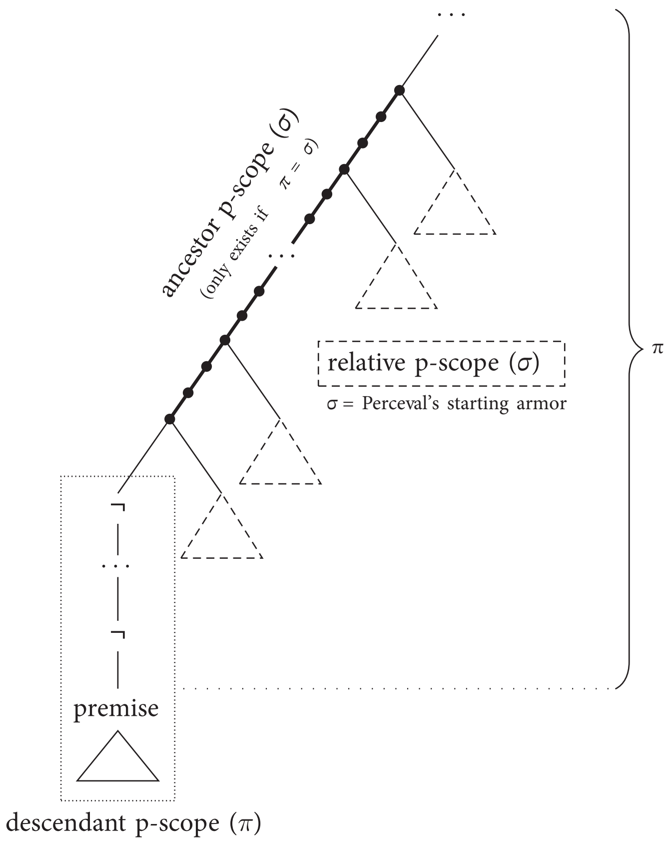


Figure 6.4 Premise scope parts (and their polarities)

Illustrating that the syntax and semantics of p-scope indeed match is one of the goals of this section. We will walk you through various examples of deductions in DDS with the help of our allegorical quest, observing that Perceval's ability or inability to reach his destination reliably matches our judgments about the validity of the deduction. This should give you a taste of why our p-scope-based inline rules are *sound*—but just a taste: we provide the real proof of soundness only in the Appendix—even if we will mostly keep to deductions involving only the simplest inline rule, Copy.

Along the way, we will introduce several fine-grained concepts related to p-scope. In particular, we will see that the p-scope domain of a premise (i.e. the nodes a premise p-scopes over) can be neatly divided into three parts, each of which has uniform p-scope polarity (i.e. the p-scope polarities of the premise at any two nodes in the same part match). The three parts and their particular properties are listed below and illustrated in Figure 6.4.

1. The most important part is the *relative p-scope*, and this is also the part that the allegory in the previous section fully applies to. It consists of the towns Perceval can only reach when his journey contains both a North-bound part and a South-bound part. The p-scope polarity of relative p-scope (σ in Figure 6.4) is determined by the kind of armor Perceval starts out with.

2. The *descendant p-scope* contains the towns which Perceval can reach without going through any checkpoint. Most of these towns lie South of Perceval's hometown. The peculiar property of the descendant p-scope is that its polarity is not determined by Perceval's starting armor, but matches the global constituent polarity of the premise (π in Figure 6.4).
3. The *ancestor p-scope* contains the towns which Perceval can reach by moving only North, but at the same time needs to pass through some checkpoint on the way. As shown in Figure 6.4, ancestor p-scope forms a sort of a bridge between the relative and the descendant p-scope. The interesting thing about the ancestor p-scope is that it only exists (or more precisely, the ancestor p-scope domain is only non-empty) when the polarities of the relative and the descendant p-scope match, i.e. when $\pi = \sigma$.

6.5.1 Relative p-scope

Relative p-scope is certainly the most common type of p-scope, and also the most useful one in terms of performing deductions. We say that β is in the RELATIVE P-SCOPE of α iff α p-scopes over β, and β is neither a descendant nor an ancestor of α—this also explains the name, as in this case α and β are merely *relatives*. From Perceval's perspective, this of course means that when he travels from the premise α to the target β, he will first need to journey North a bit, and then turn South at some junction ($\wedge$ or $\vee$). This junction, which you can think of as the point where the roads from α and β to Camelot meet, is often called the *lowest common ancestor* of α and β, and abbreviated to LCA(α, β). Figure 6.5 should make the idea clear.

Let us take a look at our first example, (46). We are Copying 'song' to the right of 'well known'—why are we allowed to do so? Well, clearly donning the golden armor at $S(y)$ gets Perceval as far North as necessary to reach $W(y)$; in

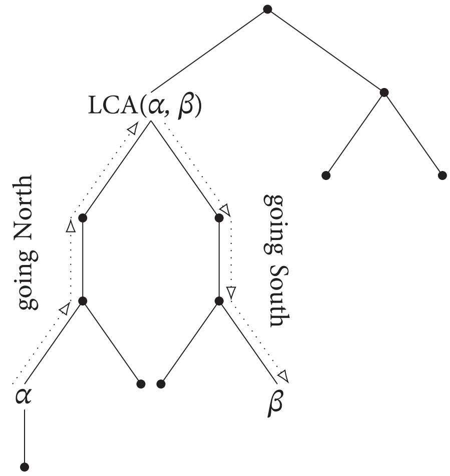


Figure 6.5 Perceval's journey in the case of relative p-scope

particular, it gets him through the two conjunctions just above $S(y)$—remember that conjunctions are Angel towns.

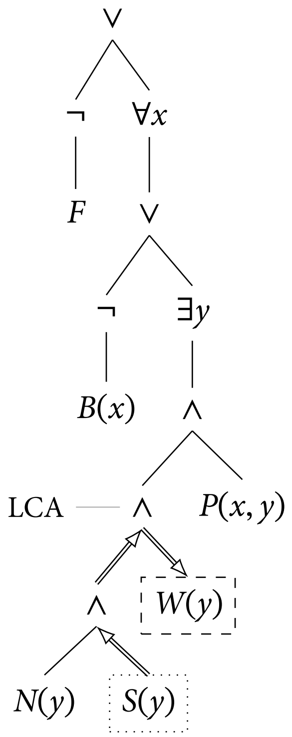
(46) a. At a festival, every band plays a nice song that is $\underbrace{\text{well known}}_{\text{COPY}}$.

$\neg F \vee \forall x\colon \neg B(x) \vee \exists y\colon ((N(y) \wedge S(y)) \wedge \underbrace{W(y)}_{\text{COPY}}) \wedge P(x, y)$

b. ∴ … plays a nice song that is a well-known song.

$\exists y\colon ((N(y) \wedge S(y)) \wedge (W(y) \wedge S(y))) \wedge P(x, y)$

c.



This example was fairly complicated, but for a reason: to show, beyond a shadow of a doubt, that relative p-scope is a property *local* to the LCA of the premise and the target. The "site of action" is deeply embedded in the tree. It occurs (listing top-down) within a disjunction, universal quantifier, another disjunction, an existential quantifier and a conjunction—but Perceval does not care about these. Relative p-scope is insensitive to both the depth of embedding and to the types of nodes above the LCA of the premise and the target—it does not even depend on the polarity of the LCA!

Now remember that Perceval is encumbered by the checkpoints only on his way North. Traveling South, he never encounters any obstacles. In the example below, it is clear that no matter what time it is and what John is doing, John is still a student in that situation; in effect, $S(j)$ can be copied arbitrarily deep into the second sentence. Another way to put this is that if a constituent p-scopes over some target (the entire second sentence below), it also p-scopes over all its descendants (like the target clause).

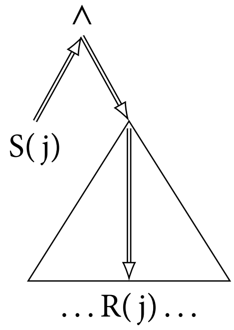
(47) a. John is a student. He's either at home (it's either morning and he's having breakfast or it's evening and he's reading a book), or he's at the university, or it is night and he's out partying.

$S(j) \wedge \big((H(j) \wedge ((M \wedge B(j)) \vee (E \wedge R(j)))) \vee U(j) \vee (N \wedge P(j))\big)$

b. ∴ … he's a student reading a book …

$S(j) \wedge R(j)$

c.

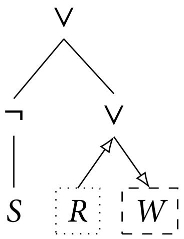


In (48), Perceval makes a short journey from R to W: one step North, one step South. Of course, to be let through Demon ∨, he needs to wear—and therefore start with—silver armor. But remember that the starting armor is significant for the polarity of p-scope: if, on a successful mission, Perceval starts out with golden/silver armor, p-scope polarity is positive/negative. Below, p-scope of R at W is thus negative, so Copying R does not introduce R itself, but $\neg R$.

(48) a. In spring, it's either raining or warm. $\neg S \vee (R \vee W)$

b. ∴ In spring, it's either raining, or it is warm and not raining. $\neg S \vee (R \vee (W \wedge \neg R))$

c.



Imagine now that we had sent Perceval on two (separate) missions, both starting in the same town, but with different destinations. Furthermore assume that both missions were successful (so we have p-scope) and that in both missions, Perceval's journey had both the North-bound and the South-bound part (so we have relative p-scope). What can we say about Perceval's starting armor (the p-scope polarity)? Obviously, the starting armor in both missions must have been the same! For example, we had sent Perceval from R to W in (48), but after he is done with that job, let us also send him from R to S in the same Kingdom. But the first checkpoint on his second travel is the same as before (a Demon disjunction), so to get through, he needs to start out with the same (silver) kind of armor! Similarly, if we send him to $N(y)$ or $P(x,y)$ in (46), he will yet again need to pass through the bottom-most Angel conjunction. The general point here is that the p-scope polarity is determined by the first checkpoint (a junction or a quantifier) above Perceval's hometown, and as he needs to pass this checkpoint on all his successful *relative* p-scope missions, it follows that the polarity of relative p-scope is *uniform*.

Consecutive junctions of the same type present no obstacle to Perceval, or more precisely, they are no more trouble than a single junction of that type. If one Angel town lets Perceval through (because he is wearing golden armor) and the town just North of it is an Angel town, he will obviously be allowed to pass through the latter town as well—this is precisely what happens in (49). Of course, the same holds for a sequence of Demon towns, as shown in (50), but as Perceval must wear his silver armor to get through those towns, p-scope has negative polarity in (50), so the conjunct introduced by Copying C must be negated. The situations are schematized in (51a) and (51b): a constituent will positively/negatively p-scope "out of" any number of conjunctions/disjunctions. From a mathematical perspective, one could say that Perceval's ease of getting through consecutive junctions of the same type is really how *associativity* and *commutativity* of conjunction and disjunction are reflected in the notion of p-scope.

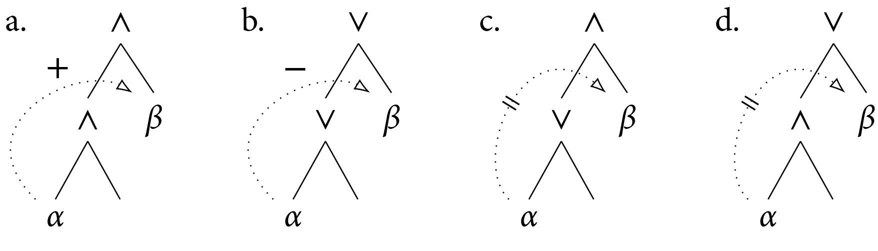
(49) a. Some $\underbrace{\text{hungry}}_{+}$ dog is $\underbrace{\text{angry}}_{+\ \text{COPY}}$. $\qquad \exists x\colon (\underbrace{H(x)}_{+} \wedge D(x)) \wedge \underbrace{A(x)}_{+\ \text{COPY}}$

b. ∴ Some hungry dog is hungry and angry. $\qquad \exists x\colon (H(x) \wedge D(x)) \wedge (H(x) \wedge A(x))$

(50) a. It is either $\underbrace{\text{cold}}_{-}$ or freezing outside, or $\underbrace{\text{it is summer}}_{+\ \text{COPY}}$. $\qquad (\underbrace{C}_{-} \vee F) \vee \underbrace{S}_{+\ \text{COPY}}$

b. ∴ It is either cold or freezing outside, or it is summer, which is not cold. $\qquad (C \vee F) \vee (S \wedge \neg C)$

(51)



On the other hand, the "mixed" situation, depicted in (51c)–(51d), is always fatal for Perceval's mission. Whenever the North-bound part of the road from the premise to the target contains an Angel and a Demon town (in any order) one right after the other, Perceval's mission is bound to fail, as it is clear that he will be denied entry at one of the checkpoints, whichever armor he wears. But the important issue here is that this failure to p-scope, either positively or negatively, neatly corresponds to the fact that the inferences below are invalid: conjoining either the premise (positive p-scope) or its negation (negative p-scope) to the target results in an invalid conclusion.

(52) a. This car is either red or white, but it is surely fast. $(R(c) \vee W(c)) \wedge F(c)$
*COPY

b. ∴̸ This car is either red or white, and it is red and fast. $(R(c) \vee W(c)) \wedge (R(c) \wedge F(c))$

c. ∴̸ This car is either red or white, and it is not red but fast. $(R(c) \vee W(c)) \wedge (\neg R(c) \wedge F(c))$

(53) a. It is either morning and John is sleeping, or it is afternoon. $(M \wedge S(j)) \vee A$
*COPY

b. ∴̸ It is either morning and John is sleeping, or it is afternoon and John is sleeping. $(M \wedge S(j)) \vee(A \wedge S(j))$

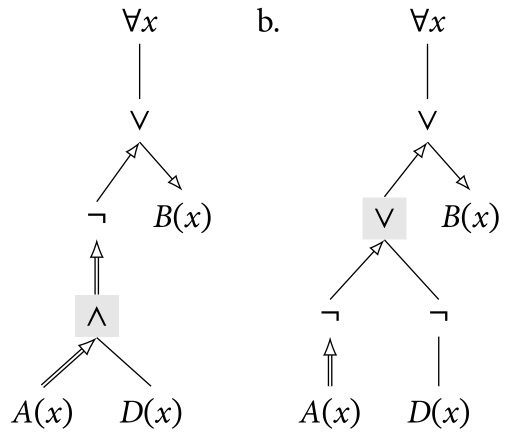
c. ∴̸ It is either morning and John is sleeping, or it is afternoon and John is not sleeping. $(M \wedge S(j)) \vee(A \wedge \neg S(j))$

Let's now move to the examples where Perceval travels through an Alchemist town in the North-bound part of his journey. What these examples show is that De Morgan's laws fall out from the notion of p-scope as well. Let us see how starting with the golden armor (positive p-scope) gets Perceval through all the towns in (55a). The first town (∧) is controlled by angels. They require golden armor and Perceval is wearing it, so they let him through. Moving further north, Perceval meets an alchemist (¬), who transmutes his golden armor into silver, and that is a good thing, as the final town (∨) on Perceval's way north is manned by demons, who want him to wear silver armor. After this town, Perceval starts moving south (east) and nobody cares about his armor anymore, so he successfully reaches the destination $B(x)$.

(54) a. Every angry dog is barking. $\forall x\colon \neg(A(x) \wedge D(x)) \vee B(x)$
COPY

b. ∴ Every angry dog is angry and barking. $\forall x\colon \neg(A(x) \wedge D(x)) \vee (A(x) \wedge B(x))$

(55) a. b.



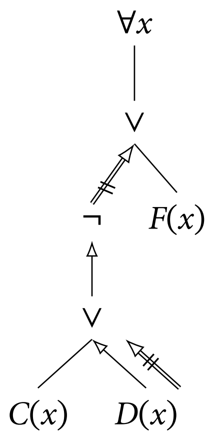
Tree (55b) is logically equivalent to (55a) under De Morgan's laws—we have replaced a negation of a conjunction by a disjunction of negations—so we expect $A(x)$ to have the same p-scoping properties in both cases. And indeed, in (55b), $A(x)$ once again positively p-scopes over $B(x)$—Perceval's initial golden armor gets transmuted to silver right away, but that is fine, because all the junctions on his way north are now Demon disjunctions.

Example (56) shows us a failed mission, one involving two junctions of the same (Demon) type, but separated by an Alchemist. If Perceval, starting in $D(x)$ and attempting to reach $F(x)$, starts out wearing golden armor, he gets nowhere at all (which means that $D(x)$ doesn't positively p-scope over $F(x)$). But donning the silver armor cannot get him to his intended destination either (which means that $D(x)$ doesn't negatively p-scope over $F(x)$ either). While he may now pass through the lower ∨, his armor gets transmuted to gold in ¬, so he is not allowed to pass through the upper ∨.

(56) a. Every cat or $\underset{*}{\underline{\text{dog}}}$ is $\underset{+\ \text{COPY}}{\underline{\text{furry}}}$. $\quad \forall x\colon \neg(C(x) \vee \underset{*}{\underline{D(x)}}) \vee \underset{+\ \text{COPY}}{\underline{F(x)}}$

b. $\not\therefore$ Every cat or dog is a furry dog. $\quad \forall x\colon \neg(C(x) \vee D(x)) \vee (F(x) \wedge D(x))$

c.

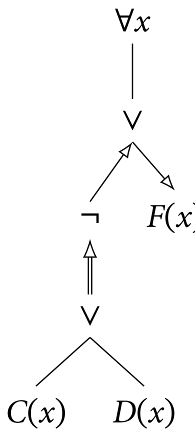


But if Perceval's mission in the same kingdom was to carry the map from the lower $\vee$ to $F(x)$, he would have succeeded—and this is why (57) is a valid argument. As the lower disjunction is now Perceval's starting point, he does not need to enter it—he lives there!—so he is free to start out with golden armor, which the Alchemist transmutes to silver before Perceval reaches the upper Demon $\vee$.

(57) a. Every $\underbrace{\text{cat or dog}}_{+}$ is $\underbrace{\text{furry}}_{+}$. COPY $\qquad \forall x\colon \neg(\underbrace{C(x) \vee D(x)}_{+}) \vee \underbrace{F(x)}_{+}$ COPY

b. $\therefore\!\!\!/$ Every cat or dog is a furry [cat or dog]. $\qquad \forall x\colon \neg(C(x) \vee D(x)) \vee (F(x) \wedge (C(x) \vee D(x)))$

c.



So far in our discussion of relative p-scope, we have not sent Perceval on any journeys where he would need to pass through a *quantifier* checkpoint on his way North. It is time to do so, because we want to show you that we were right to say that universal quantifiers are Angel towns and that existential quantifiers are Demon towns. In other words, universal quantifiers line up with conjunctions, while existential quantifiers pattern with disjunctions.

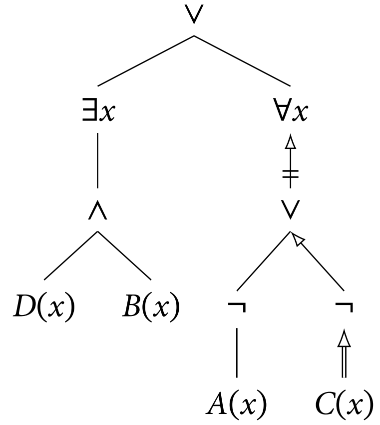
Let us start by an example of a failed journey. Below, the idea is to Copy 'cat' over 'dog', which of course should not be possible; in other words, Perceval should be unable to reach $D(x)$ from $C(x)$. The first checkpoint on Perceval's way North is a disjunction. To get through this checkpoint, Perceval must start out with the golden armor—the alchemist just above Perceval's hometown will transmute his armor to silver, so Perceval can pass through the Demon disjunction which comes next. But now, if he could also pass through the universal quantifier wearing the silver armor which he now wears, we know he could complete his journey, as the only further obstacle in his way would be Camelot, which is again a Demon disjunction in this kingdom; in other words, if universal quantifiers were Demon towns, the inference would be valid. As it is not valid, however, the universal quantifier must be an Angel town, and Perceval is stopped in this town because he is wearing silver armor.[18]

[18] In Chapter 9 we will see that the combination of a Demon junction and an Angel quantifier, or vice versa, is ubiquitous in natural language, and forms the base of the so-called restricted quantification.

(58) a. Some dog barks, or no alien is a cat. $\exists x\,(\underbrace{D(x)}_{*\text{COPY}} \wedge B(x)) \vee \forall x\,(\neg A(x) \vee \neg\underline{C(x)})$

b. .·/. Some cat barks, or ... $\exists x\,(\overline{C(x)} \wedge B(x)) \vee \dots$

c.



If the previous example illustrates that the universal quantifier and the disjunction host different kinds of supernatural beings, the following example will show that the universal quantifier and the conjunction host the same kind: Angels. In this example, we will (finally!) send Perceval on a mission in the service of some rule other than Copy. He will be asked to deliver a premise for Inline Modus Ponens (iMP).

Let's refresh our memory about iMP. We formally stated it as $\mathcal{A} \Rightarrow C \vdash^{+}_{\mathcal{A}} C$, where we see that iMP targets a node of positive polarity ($\mathcal{A}$) and additionally requires the conditional premise (of form $\neg\mathcal{A} \vee C$). When we first met iMP (and iMT) at the beginning of this chapter, we did not know about p-scope yet—but now we do, and it is clear that the conditional premise must positively[19] p-scope over the target for the rule to be applicable; so a more precise statement of iMP is $(\mathcal{A} \Rightarrow C)^{+} \vdash^{+}_{\mathcal{A}} C$.

Let us see how this plays out in (59). We want iMP to target $A(x)$ and use $\neg A(x) \vee C(x)$ as the (conditional) premise. The target plays a role of the (antecedent) premise as well, but as it is already "in the right place," we need not dwell on it now (see subsection 6.5.2 for details). What we need Perceval to do is to bring the conditional premise to the target. Can he do that? If universal quantifiers hosted Demons, the sequence of the universal quantifier and the root conjunction would present an unpassable obstacle for Perceval. However, if we assume—and we do assume—that universal quantifiers are Angel towns, starting out with the golden armor gets Perceval not only through the Angel universal quantifier, but also through the Angel conjunction; he turns south there, so the rest of his journey is easy. (Remember that Perceval's journey is unaffected by the fact that his

[19] Consider the following example to see that a constituent (of an appropriate form) cannot function as the conditional premise of iMP if it *negatively* p-scopes over the target.

(i) a. It's not the case that when it rains, it pours. It rains. $\neg(\underline{\neg R \vee P}) \wedge \underbrace{R}_{*\text{IMP}}$

b. .·/It pours. $\overline{P}$

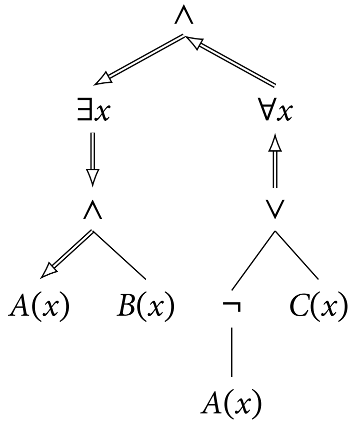
hometown is a disjunction. He does not need to enter this disjunction as he lives there, so he can happily start out with the golden armor.)

(59) a. Some $\underbrace{\text{alien}}_{\text{iMP}}$ barks. $\exists x\,(\underbrace{A(x)}_{\text{iMP}} \wedge B(x)) \wedge \forall x\,(\underline{\neg A(x) \vee C(x)})$

Every alien is a cat.

b. ∴ Some $\overline{\text{cat}}$ barks. $\exists x\,(\overline{C(x)} \wedge B(x)) \wedge \ldots$

c.



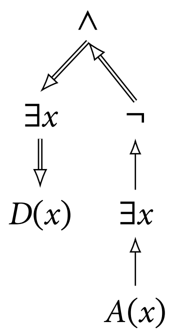
Having seen that universal quantifiers are truly Angel towns (and that existential quantifiers must therefore be Demon towns), we can finally explain a lingering issue from section 6.1. We started off that section with a traditional view on *dicta de omni et nullo*: to function, they needed a premise of a form $\forall x\colon \neg\mathcal{A} \vee C$. But then we had formalized them by Inline Modus Ponens and Tollens, which required $\neg\mathcal{A} \vee C$—notice the absence of the quantifier. The issue is patently clear now. As witnessed by the above example, the quantifier does not need to be a *formal* part of the conditional premise because it is taken care of by p-scope.

As the final example in our discussion of relative p-scope, we offer a deduction where Perceval is sent on a journey passing through an existential quantifier (on his way North). Starting at $A(x)$ with silver armor, Perceval first passes through the Demon existential quantifier (on the right), then gets his armor transmuted to gold at the Alchemist negation, and finally turns South at the Angel conjunction in Camelot. As he started out with silver armor, the p-scope polarity of $A(x)$ over $D(x)$ is negative, so its copy must be negated.

(60) a. There is a $\underbrace{\text{dog}}_{\text{COPY}}$ here, but no $\underline{\text{alien}}$. $\exists x\,\underbrace{D(x)}_{\text{COPY}} \wedge \neg\exists x\,\underline{A(x)}$

b. ∴ Some dog is not an alien, … $\exists x\,(D(x) \wedge \neg A(x)) \wedge \ldots$

c.



This concludes our discussion of relative p-scope, i.e. instances of p-scope where neither of the premise and the target dominates the other. We have seen that the computation of relative p-scope is local to the lowest common ancestor of the premise and the target: only the towns Perceval meets on his travel are relevant for the success of his mission. Even more, Perceval can only get in trouble in the North-bound part of his journey—traveling South is easy!—the typical obstacle being a sequence of an Angel conjunction and a Demon disjunction, or vice versa. Of course, other obstacles exist as well. Some involve quantifiers: a common obstacle is formed by the combination of an Angel/Demon junction and a Demon/Angel quantifier above it. Other unpassable combinations involve Alchemists (negations), who change the metal composition of Perceval's armor and thereby effectively reverse the role of all subsequent Angel and Demon towns.

We have noted the significance of Perceval's starting armor—golden starting armor indicates positive p-scope, and silver starting armor indicates negative p-scope—and we had also noticed that given a fixed starting point (Perceval's hometown), the starting armor is the same for any destination that Perceval can successfully reach when his travels include both a North-bound part and a South-bound part. The relative p-scope polarity of the premise is therefore the same for any target, or in other words, the polarity of relative p-scope is uniform. But crucially, this only comes about because for relative p-scope, Perceval always needs to travel North before turning South, which means that he will always pass through at least one checkpoint (the lowest common ancestor of the premise and target is surely a junction and thus a checkpoint). In the next section, we turn to cases of Perceval's travels that do not include any checkpoints. We will see that in these cases, the material of his starting armor is irrelevant and that the p-scope polarity is determined in another way.

6.5.2 Descendant p-scope

We have already mentioned that traveling South is a lot easier for Perceval than traveling North, in the sense that he is never checked for armor when he enters a town via the north gate, or when he exits the town. On some missions, Perceval does not pass through any checkpoints at all. These missions are the topic of the present subsection. As most of them involve a destination situated south of Perceval's hometown, i.e. the target which is a descendant of the premise, we call this part of p-scope DESCENDANT P-SCOPE.

One thing to note about these missions is that they are always successful. If there is no checkpoint between Perceval's hometown and his destination, Perceval's journey is not obstructed in any way, so he will always reach the destination. In other words, if the road from the premise to the target involves no checkpoints, the premise surely p-scopes over the target.

Given the absence of checkpoints, it is also completely irrelevant what armor Perceval chooses to start out with on these missions. However, this does not mean that descendant p-scope has no polarity, or that that polarity is arbitrary; it means that our allegory, as it stands, is a bit misleading in these cases. It turns out that while the p-scope polarity of descendant p-scope is not determined by Perceval's starting armor, it nevertheless equals a very familar property, the global (constituent) polarity of the premise. (Incidentally, it follows that the p-scope polarity within the descendant p-scope is uniform as well.)

So let us take our allegory a bit further to explain what is going on here. We have talked a lot about angel and demon towns, but of course the towns (and even villages) themselves have polarities that are quite independent of whether angels or demons have taken up residence there. This is their global (constituent) polarity, which is a function of how many alchemists (negations) are found north of the town. Of course if an even number appear above them the towns have positive polarity and if it is an odd number they will have negative polarity. Let us suppose that positive polarity towns and villages fly *golden banners* and that negative polarity towns and villages fly *silver banners*.

The bottom-line is that whenever Perceval encounters no checkpoints on his journey, it is not his initial armor which is important, what matters is what kind of banner his hometown flies: if his hometown flies a golden banner, the (descendant) p-scope polarity is positive, and if it flies a silver banner, the p-scope polarity is negative.

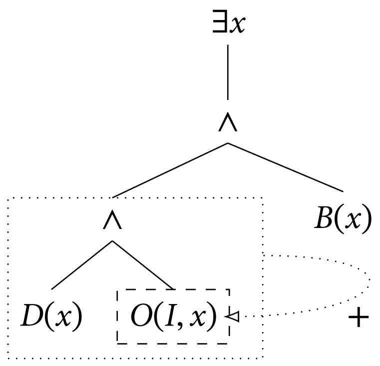
Truth be told, while descendant p-scope derives its name from the configuration where the target is a descendant of the premise, actual deductions deploying this configuration are very rare (in fact, we could do without them). Here's an example of Copying the premise "into itself":

(61) a. $\underbrace{\text{Some dog which } \underbrace{\text{I own}}_{+}}_{+}$ is barking. $\quad \underbrace{\exists x\colon (D(x) \wedge \underbrace{O(I, x)}_{+}}_{+}) \wedge B(x)$

COPY COPY

b. ∴ Some dog which I own and is a dog which I own is barking.

$\exists x\colon (D(x) \wedge (O(I, x) \wedge (D(x) \wedge O(I, x)))) \wedge B(x)$

c. 

However, other descendant p-scope configurations are very useful. Let us first consider the situation where there is an Alchemist town, or a string of Alchemist towns, just north of Perceval's hometown, and Perceval is sent to deliver the map to one of them. You see, because there are no checkpoints in Alchemist towns, Perceval can reach the Alchemist towns just north of his hometown (if there are any) without passing any checkpoints. So despite the name, the negations immediately dominating the premise belong to its descendant p-scope!

This is the situation we find ourselves in when we want to effect Double Negation Elimination. As shown below, we can do this by Copying the premise over the grandparent negation. The trick works both in the positive and in the negative polarity environment. In the second example, we rely on the negative target polarity and negative p-scope polarity cancelling each other—remember from section 6.3 that given target polarity τ and p-scope polarity σ of premise $\mathcal{A}$, Copying replaces the target by $\neg^{\sigma\tau}\mathcal{A}$.

(62) a. It is not the case that it is not raining. $\qquad \neg\neg R$

COPY $\qquad$ COPY

b. ∴ It is raining. $\qquad R$

(63) a. If it is not the case that it is not raining, $\qquad \neg\neg\neg R \vee \neg S$

COPY $\qquad$ COPY

then it can't be sunny.

b. ∴ If it is raining, then it can't be sunny. $\qquad \neg R \vee \neg S$

Finally, let us tell you about Perceval's favorite kind of mission. The fact is that the knights of the round table are, deep down, a bit lazy, and Perceval, being one of them, enjoys a task that does not involve much travel. Luckily, such is his very common mission, one where he is asked to deliver the map within his hometown.

In section 6.1, we saw that the target plays two roles in an application of a rule. On one hand, it is the constituent that gets replaced by the conclusion of the rule. On the other hand, it is also a premise. As such, Perceval is called to deliver it, but of course, this is a "local" delivery, because Perceval's starting point (his hometown) is at the same time also his destination.

Let us take Inline Modus Ponens and Tollens as an example. These rules have two premises, really. One of them (the conditional premise, of form $\neg\mathcal{A} \vee C$) may occur anywhere in the tree, as long as it positively p-scopes over the target. But the other premise is the target itself. In the trees below, we illustrate this by looping an arrow around the target—(66a) and (66b) depict the situation in (64) and (65), which exemplify the application of iMP and iMT, respectively.

(64) a. Every pet is a dog. $\qquad (\forall x\colon \neg P(x) \vee D(x))$

iMP $\qquad$ iMP

Every dog is an animal. $\qquad \wedge(\forall x\colon \neg D(x) \vee A(x))$

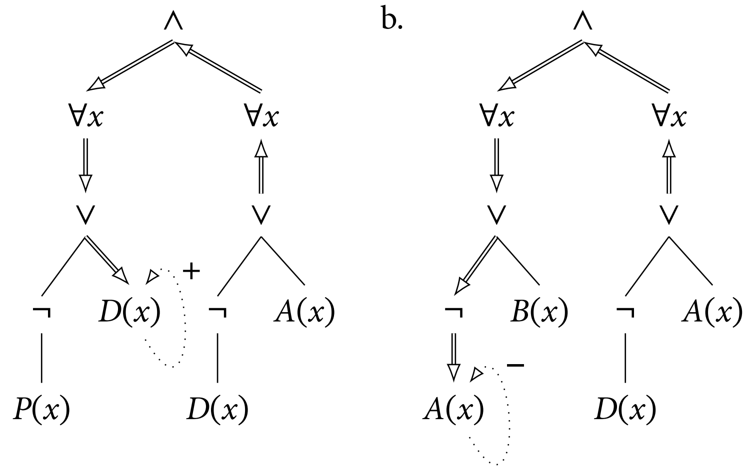
b. ∴ Every pet is an animal. $\qquad (\forall x\colon \neg P(x) \vee A(x)) \wedge \ldots$

(65) a. Every $\underbrace{\text{animal}}_{\text{iMT}}$ is barking. $\qquad \big(\forall x\colon \neg \underbrace{A(x)}_{\text{iMT}} \vee B(x)\big)$

Every dog $\underset{+}{\text{is}}$ an animal. $\qquad \wedge\big(\forall x\colon \neg D(x) \underset{+}{\vee} A(x)\big)$

b. ∴ Every $\overline{\text{dog}}$ is barking. $\qquad \big(\forall x\colon \neg \overline{D(x)} \vee B(x)\big) \wedge \ldots$

(66) a. b.



Let us see how the various polarities play out in the above examples. Following the idea that *dictum de omni* can be applied in positive polarity environments, and that *dictum de nullo* can be applied in negative polarity environments, we required that the target of iMP must have positive (constituent) polarity, and that that the target of iMT must have negative (constituent) polarity, as repeated below. But how does this relate to the local delivery of the target premise?

Inline Modus Ponens (iMP)	Inline Modus Tollens (iMT)
$(\neg\mathcal{A} \vee C)^{+} \vdash^{+}_{\mathcal{A}} C$	$(\neg\mathcal{A} \vee C)^{+} \vdash^{-}_{C} \mathcal{A}$

Well, as with all premises, the target premise is delivered by Perceval (even if the delivery is local), so it is delivered as-is when its p-scope polarity is positive, and it is delivered in the negated form when its p-scope polarity is negative. But the p-scope of the target over itself is an instance of descendant p-scope, so the p-scope polarity of the target over itself equals its constituent polarity! So a positive polarity target $\mathcal{A}$ will be delivered as $\mathcal{A}$, which works well for applying Modus Ponens, which requires premises of form $\mathcal{A} \Rightarrow C$ and $\mathcal{A}$. But a negative polarity target C will be delivered as $\neg C$, which works well for applying Modus Tollens (requiring $\mathcal{A} \Rightarrow C$ and $\neg C$). In the trees above, the p-scope polarity of the target premise ($D(x)$ in (66a) and $A(x)$ in (66b)) is indicated by the label on the looped arrow, and it is easy to see that it matches the constituent polarity of the target.

Summing up, the descendant p-scope obtains when Perceval passes through no checkpoints on a successful journey from the premise to the target. All the descendants of the premise are within its descendant p-scope, and so are any negations immediately dominating the premise, but the most important instance of descendant p-scope is surely the p-scope of the target over itself, which is what makes every target a potential premise—we have been using descendant p-scope all the time, without knowing it!

Polarity-wise, descendant p-scope is very different from relative p-scope. While the relative p-scope polarity is determined by Perceval's starting armor, the descendant p-scope polarity depends on the banner of his hometown; the descendant p-scope polarity of a premise equals its constituent polarity, for any target. We now turn to the final part of the p-scope domain, called ancestor p-scope, where these two ways of determining p-scope polarity meet.

6.5.3 Ancestor p-scope

When we last left Perceval, he was delivering premises to places he could reach without passing through any checkpoints. We have seen that these descendant p-scope missions are a bit special in terms of p-scope polarity. This section is about another kind of special missions—missions where Perceval has to not only enter a checkpoint town, but also deliver the map there! Of course, as Perceval's armor is only ever checked in the first, North-bound part of his journey, all such towns are located directly North of his hometown, and this is why we are talking about ANCESTOR P-SCOPE here (the target is an ancestor of the premise).

We will see that ancestor p-scope is not crucial for our deductive system, but we keep it, both to give you a complete picture of the p-scoping possibilities and because it simplifies some deductions.

So, what is special about Perceval's destination being one of the checkpoint towns? As Perceval travels North, he can pass through the checkpoints in the angel and demon towns if he has the right armor, but it turns out he needs special permission to stop and deliver the map (the premise) in those towns. In the previous section, we told you that each town flies either a golden or a silver banner: towns of positive polarity (with an even number of alchemists above them) fly a golden banner, and towns of negative polarity (with an odd number of alchemists above them) fly a silver banner. When he is moving North, Perceval only gets permission to deliver the map in a checkpoint town if his armor matches the town's banner.[20]

But what about the alchemist towns that he encounters on the way North? Well, if he encounters an alchemist town before he hits a checkpoint (a junction or a quantifier), then that town will be in his descendant p-scope, and we have already

[20] In fact, we can phrase this "permission" in several ways, all completely equivalent:
- Perceval's armor upon entering the target town must match the target town's banner.
- The banner of the target town must match the metal liked by the controllers of the town (Angels or Demons).
- Perceval's starting armor must match the banner of his hometown.
- The relative and the descendant p-scope polarity of the premise must match.
- The target must be an e-conjunction or an e-universal quantifier (see section 6.1).

shown that he can deliver his map there. But suppose he encounters a junction town or quantifier town, receives permission to deliver his map there but then proceeds to the alchemist town (actually, any town in the string of alchemist towns) to its North and wants to deliver the map there. Well, in that case he retains his permission to deliver the map, as the junction and quantifier towns have a kind of jurisdiction over the alchemist towns to their immediate North. Staying within the allegory, you can think of it this way: If a town with a checkpoint lets him pass, then it is responsible for what Perceval does until he reaches the next checkpoint. Thus, similarly, if he does not have authorization to deliver the map in a checkpoint town (because his armor does not match the checkpoint town's banner), he will be unable to deliver his map to an alchemist town to the immediate North of the checkpoint town.

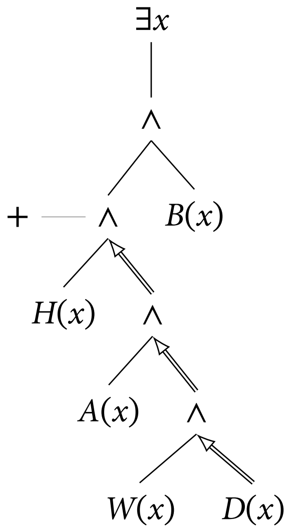
Can we see this in action? How can we deduce (67)? Well, we could apply Delete three times (to 'huge', 'angry', and 'white'). Or we can simply Copy 'dog' over the entire noun phrase![21] This Copying is clearly licensed by ancestor p-scope. Perceval starts out with golden armor, so he can pass through the three Angel conjunction checkpoints. He is still wearing the golden armor at his destination, the final conjunction, and because this town flies a golden banner (has positive polarity), Perceval is allowed to deliver the map there.

(67) a. Some $\underbrace{\text{huge angry white } \underline{\text{dog}}^{+}}_{\text{COPY}}^{+}$ barks.

$$\exists x\colon (\underbrace{H(x) \wedge (A(x) \wedge (W(x) \wedge \underline{D(x)}^{+}))}_{\text{COPY}}^{+}) \wedge B(x)$$

b. ∴ Some $\overline{\text{dog}}$ barks. $\qquad \exists x\colon \overline{D(x)} \wedge B(x)$

c.



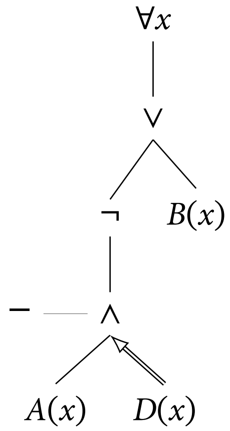
[21] This example illustrates that we can derive Delete from Copy—but only from the center variant, which overwrites the target. And clearly, we can also derive center Copy from right (or left) Copy and Delete. So, our basic rules could include either center and right Copy, or right Copy and Delete. We will choose the latter option in section 7.3 for symmetry.

The following example illustrates the failure to p-scope over an ancestor. To reach the parent conjunction (which is also his final destination), Perceval must start out with the golden armor, but this town flies a silver banner, having exactly one alchemist to the north.

(68) a. Every $\underbrace{\text{angry } \overset{?}{\text{dog}}}_{*\text{COPY}}$ barks. $\forall x\colon \neg(\underbrace{A(x) \wedge \overset{?}{D(x)}}_{*\text{COPY}}) \vee B(x)$

b. $\not\therefore$ Every $\overline{\text{dog}}$ barks. $\forall x\colon \neg\overline{D(x)} \vee B(x)$

c.



We will deploy ancestor p-scope again in section 7.2, where it will help us juggle quantifiers. Here, we want to show you why we really like ancestor p-scope, even if it throws additional obstacles at our poor Perceval. Usually, the conclusion of a derivation is taken to be the final formula of the derivation, but having learned about ancestor p-scope, we can actually go for a somewhat different definition. In fact, we will not define *the* conclusion, but *a* conclusion:

(69) A CONCLUSION of a derivation is any constituent positively p-scoping over the root of the dynamic formula.[22]

Clearly, the root is an ancestor of every constituent, so the p-scope in the above definition is an instance of ancestor p-scope (except when we consider the root itself as a conclusion, or when the root and the intended conclusion are separated only by negations; these cases are instances of descendant p-scope, as explained in the previous chapter).

If we stick to the standard definition of conclusion, 'every pet is an animal' is not the conclusion of the following one-step deduction—to deduce it, we would need to Delete the second root conjunct. But with our (ancestor) p-scope-based definition of conclusion, we can stop after the first step: the left conjunct of the

[22] We could go even further and say that the negation of a constituent negatively p-scoping over the root is a conclusion as well.

root positively p-scopes over the root and thus counts as a conclusion (we have marked this by underlining it as if it was a premise).

(70) a. Every pet is a $\underbrace{\text{dog}}_{\text{iMP}}$. $\underline{\text{Every dog } \underset{+}{\text{is}} \text{ an animal}}$.

$$\big(\forall x\colon \neg P(x) \vee \underbrace{D(x)}_{\text{iMP}}\big) \wedge \big(\forall x\colon \underline{\neg D(x) \underset{+}{\vee} A(x)}\big)$$

b. $\therefore$ $\underline{\text{Every pet } \underset{+}{\text{is}} \text{ an } \overbrace{\text{animal}}}$. Every dog is an animal.

$$\big(\underline{\forall x\colon \neg P(\underset{+}{x}) \vee \overbrace{A(x)}}\big) \wedge \big(\forall x\colon \neg D(x) \vee A(x)\big)$$

That is hopefully a sufficiently detailed look at p-scope to serve our ends in this book; the formal definition can be found in the Appendix. The first and most critical thing to understand is that p-scope was designed to be a polarity-sensitive mechanism by which we can carry out in-line deductions in L^{**} and ultimately natural language. Its principal role is to keep track of permissible and impermissible applications of our inline rules; so far, we have used it mainly with Copy, but it will also feature in another fundamental rule, which we will call Prune. But, perhaps surprisingly, the role of p-scope is not limited to the deductive system. Critically, it is going to provide us a tool for a syntactic account of aboutness and conservativity, and thus is going to help us to better understand the nature of quantification.

But that is ahead of us. We are not yet finished with our Dynamic Deductive System. In the next chapter, we will take a closer look at our DDS machinery with several goals in mind. One is to break down Inline Modus Ponens and Tollens into simpler rules, another to investigate the behavior of quantifiers. But we also have the goal of delivering on an early promissory note from the Introduction: completeness.

7
Completing the Dynamic Deductive System

In the previous chapter we introduced our inline Dynamic Deductive System, showed the role of p-scope within that system, and showed how p-scope was sensitive to polarity. You may have wondered, however, why we still use (Inline) Modus Ponens and Modus Tollens, even though we had previously argued that they were special cases of the *dictum de omni* and *dictum de nullo* respectively. Wasn't the whole idea of the Holy Grail to simplify traditional logical principles down to two simple polarity-sensitive rules? If that is the case, then why do we need iMP and iMT to be stated explicitly, even if they are in some sense fundamentally versions of the *dicta*?

It is a good question, and the answer is that we want you to see how, with p-scope now in place, we can use p-scope and thus polarity to dissolve iMP and iMT into operations that we consider to be more fundamental than even the *dicta* rules. We met two of these operations—Copy and Delete—in the previous chapter, and in this chapter we are going to introduce the third, and central, ingredient that we call *Prune*, which is a rule that cleans up internal polarity conflicts—if you like, you can think of it as a generalized inline version of *reductio ad absurdum*.

Once this final piece of (the propositional part of) DDS is in place, we will be able to push the Holy Grail of Medieval Logicians deeper than they could have envisioned. Remember that their idea was to reduce all of logic down to two basic rules of inference, *dictum de omni* and *dictum de nullo*, guided by polarity. Our execution of the Medieval program fully follows the spirit of their idea in that the inline inference rules of DDS are guided by the p-scope relation, in which polarity plays the central role. But by breaking down the *dicta* into what we consider to be non-logical syntactic operations (Copy, Prune, Add, and Delete),[1] we gain an even more detailed understanding of the role of polarity in a deductive system. At the same time, our execution of the idea broadens the scope of its application—while the *dicta* account only for directional entailingness, DDS is a complete system in the sense that it can deduce any valid argument.

As indicated above, our four syntactic operations (Copy, Prune, Add, and Delete) manipulate the propositional structure of a formula. What about the

[1] Perhaps Prune is the least non-logical of these syntactic operations. We suspect that omitting it would point us toward a kind of constructive logic—a matter we leave to further research.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović, Oxford University Press. DOI: 10.1093/oso/9780199591534.003.0007

quantificational aspects of the system? What are the DDS counterparts of the traditional quantificational rules (Universal and Existential Generalization and Instantiation)? In section 7.2, we will show you that we don't need these as separate rules. Thanks to p-scope and polarity, we can integrate these quantificational rules into every application of Copy, Prune, Add, and Delete. In fact, this integration, supported by the inline nature of DDS, makes it very natural for our system to work exclusively on closed formulas (i.e. formulas without free variables)—one of the desiderata for a deductive system according to Quine (1981).[2]

Of course we haven't forgotten that we also promised to address the completeness of our system. We will get to that in section 7.3, where we will show that DDS can simulate a well-known complete deductive system, a Hilbert-style system. Besides the proof of completeness, the section will also include the very final update to the list of DDS rules. However, the rule we will add in that section is more of an "interface" rule than a rule of inference—*Import* will allow us to bring axioms and hypotheses into a derivation. But before we get to all that, we first want to lay some groundwork for Prune.

7.1 Prune

It is time to introduce our final fundamental rule—Prune—which will (in tandem with Copy and a little help from Delete) take over the function of Inline Modus Ponens and Tollens. We will start by showing that a generalized statement of iMP and iMT applicable for premises and target of any polarity is too complex to be seriously considered as a basic rule of inference. This will lead us to the idea that it might be preferable to break down iMP and iMT into simpler components, and we will execute this idea by developing Prune in subsection 7.1.2. Finally, the examples in subsection 7.1.3 will show how Prune can not only replace Inline Modus Ponens and Tollens, but also extend their domain of application.

7.1.1 Why do we need Prune?

Below, we recapitulate the rules that we have equipped our Dynamic Deductive System with so far. It could be shown (by a trivial adaptation of the proof in section 7.3) that these rules yield a complete deductive system. So why do we then want to introduce an additional rule, Prune?

[2] Even more, the closed formula variant of DDS should make for a perfect companion to the quantifierless approach to quantification which will be the final result of our investigations in Chapter 9. We will not wed these two results in this book—we believe that one new deductive system per book is quite enough—but we are excited by the further research directions that seem to be opened by the combination of these results (for one, DDS on the quantifierless formal language appears to be perfectly suited for formalization of inclusive logic, i.e. logic which allows the empty domain).

Inline Modus Ponens (iMP)	Inline Modus Tollens (iMT)
$(\neg\mathcal{A} \vee \mathcal{C})^{+} \vdash^{+}_{\mathcal{A}} \mathcal{C}$	$(\neg\mathcal{A} \vee \mathcal{C})^{+} \vdash^{-}_{\mathcal{C}} \mathcal{A}$

Delete	Add	Copy
$\vdash^{\tau}_{\mathcal{A}\wedge^{\tau}\mathcal{B}} \mathcal{A}$ (right)	$\vdash^{\tau}_{\mathcal{B}} \mathcal{B} \vee^{\tau} \mathcal{C}$ (right)	$\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \mathcal{B} \wedge^{\tau} \neg^{\sigma\tau}\mathcal{A}$ (right)
$\vdash^{\tau}_{\mathcal{A}\wedge^{\tau}\mathcal{B}} \mathcal{B}$ (left)	$\vdash^{\tau}_{\mathcal{B}} \mathcal{C} \vee^{\tau} \mathcal{B}$ (left)	$\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \neg^{\sigma\tau}\mathcal{A}$ (center)
		$\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \neg^{\sigma\tau}\mathcal{A} \wedge^{\tau} \mathcal{B}$ (left)

The answer is that while the rules above suffice to deduce anything that can be deduced in classical deductive systems, the deductions are sometimes clumsier than we would like (though still simpler than in the classical systems). This is problematic, because humans find some of those inferences very easy—after all, we are in search of a *natural* deductive system. If we put this in more general philosophical terms, we would prefer a rule system which has more of what Ludlow (2011) called P-simplicity, for "Peirce Simplicity," which means that it will make our lives easier, both in terms of theory and application, and it will also have advantages in S-simplicity, for "Sober Simplicity," in that it allows a preparation of the natural deductive theory for a reduction to more basic principles. And truth be told, we also think the alternative we are going to offer in subsection 7.1.2 is more aesthetically pleasing.

One issue with our current formulation of iMP and iMT is that those rules aren't polarity flexible, as are the other rules afforded within DDS. All the other rules fully integrate polarity: the target may be of any polarity, the premise may p-scope over the target either positively or negatively, and the connectives manipulated by the rules automatically "adjust" to those polarities. iMP and iMT, on the other hand, are tied to fixed polarities and forms of the premise, the target and the conclusion (like their traditional counterparts MP and MT).

The example below illustrates how the fixed p-scope polarity of the premise prevents us from applying iMP in a case where we could reasonably expect it to be applicable. All is well in (1) if we represent the negative determiner 'no' by a universal quantifier. But if we choose the alternative representation using a negated existential, iMP is inapplicable: the "failed premise" is a conjunction (rather than a disjunction) and its p-scope polarity is negative (rather than positive).

(1) a. $\underline{\text{No alie}\underset{-}{\text{n}}\text{ is a cat.}}$ $\qquad \neg\exists x\,(\underline{A(x) \underset{-}{\wedge} C(x)}) \sim \forall x\,(\underline{\neg A(x) \underset{+}{\vee} \neg C(x)})$

b. Some visitor is an $\underset{\text{IMP}}{\underset{+}{\underset{+}{\text{alien}}}}$. $\qquad \wedge\exists x\,(V(x) \wedge \underset{\text{IMP}}{\underset{+}{\underset{+}{A(x)}}})$

c. ∴ Some visitor is $\overline{\text{not a cat}}$. $\qquad \ldots \wedge \exists x\colon V(x) \wedge \overline{\neg C(x)}$

This is unacceptable. Our deductive system is *designed* to transparently deal with arbitrary polarities, and with dual connectives such as conjunction and disjunction.

Can't we state iMP so that it will require the connective of the conditional premise to be a disjunction when the premise has positive p-scope, and a conjunction when its p-scope is negative? Of course we can. We just need to say that the connective must be an *effective* disjunction (with respect to the p-scope polarity), i.e. that when the p-scope polarity equals σ, the premise should contain $\vee^\sigma$. (Note the similarity to Add, which introduces an e-disjunction.)

But before we write down the iMP rule based on the e-disjunction idea, are there any additional generalizations we could make? That is, are there any other examples of inferences that feel natural but the current formulation of iMP and iMT does not let through? Yes! For example, one could reasonably expect a polarity-based system to subsume the traditional rule of Disjunctive Syllogism (DS) under iMP. After all, isn't the point of iMP that the two occurrences of $\mathcal{A}$ in the rule have opposite polarities? Well, so do the two occurrences of $A(f)$ below.

(2) a. Felix is either an alien or a cat. $A(f) \vee C(f)$
b. Felis is not an alien. $\neg A(f)$
c. ∴ Felix is a cat. $C(f)$

In fact, the realization that MP and DS can be seen as one and the same rule with a flipped polarity of the antecedent immediately leads to the realization that we can unify not just MP ($\neg\mathcal{A} \vee C$) and DS ($\mathcal{A} \vee C$), but also MT ($\mathcal{A} \vee \neg C$) and an unnamed rule with conditional premise of $\neg\mathcal{A} \vee \neg C$. We thus arrive at a rule we call Generalized Disjunctive Syllogism (GDS).[3]

Generalized Disjunctive Syllogism (GDS)

$$(\neg^{\pi\sigma}\mathcal{A} \vee^\sigma \neg^{\rho\sigma}C)^\sigma,\ \mathcal{A}^{-\pi} \vdash^\tau \neg^{\rho\tau}C \qquad \text{(left)}$$

$$(\neg^{\rho\sigma}C \vee^\sigma \neg^{\pi\sigma}\mathcal{A})^\sigma,\ \mathcal{A}^{-\pi} \vdash^\tau \neg^{\rho\tau}C \qquad \text{(right)}$$

However, while GDS is perfect in the sense that it has an impressively wide range of application, and furthermore possesses a certain attractive symmetry, we find it highly unlikely that it could be one of the *basic* rules of inference used in natural language. It seems too complex to fit that role. To apply it, we need to juggle no fewer than four interacting polarities: target polarity (τ), p-scope polarity of the conditional premise (σ), and constituent polarities of $\mathcal{A}$ and C within the conditional premise ($\pi\sigma$ and $\rho\sigma$).[4]

[3] We have sneaked one further detail into GDS. As with every target, the target of iMP/iMT has two functions: it is both a premise and a replacement site. In GDS, these two functions are separated—and this is why GDS has two premises—resulting in an arguably more aesthetic rule.

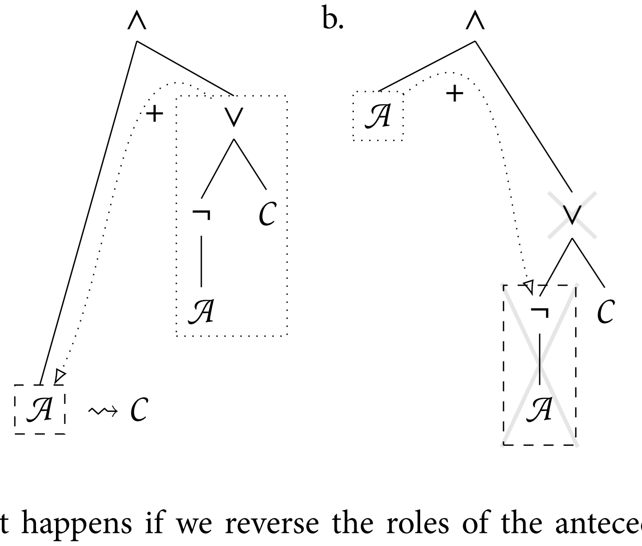
[4] To the reader who wants to apply it to (1): use the left variant of GDS, setting the target polarity $\tau = +$, the p-scope polarity $\sigma = -$, and $\pi = \rho = -$. We even chose the predicate symbols well for you.

The strategy of unifying rules we have employed above is not unreasonable, but in the specific case of iMP and iMT it actually takes us even further away from our desideratum—a simple rule system (in the sense of both P-simplicity and S-simplicity). However, reflection shows there are more interesting opportunities here if we push to make the deductive system as deep as possible. The strategy of providing unified, generic formulations of rules, applicable under various polarities, works in general, but the strategy is not optimal here because the starting points, i.e. the pair of iMP and iMT, are too complex to start with. The idea we will develop in the rest of this section will begin with simpler, more fundamental pieces. We already have two of them, Copy and Delete, and we're about to start working on the final piece, the rule Prune.

7.1.2 Understanding Prune

In the *dicta* rules, as well as in our take on these rules in the form of Inline Modus Ponens and Tollens, the two premises $\mathcal{A}$ and $\neg\mathcal{A} \vee C$ are inherently asymmetric. The antecedent premise doubles as the target, which has two consequences, both shown in (3a). First, the antecedent premise must be in the positive p-scope of the conditional premise. Second, the antecedent premise $\mathcal{A}$ is *replaced* by the consequent C.

(3) a. b.



What happens if we reverse the roles of the antecedent and the conditional premise, letting the latter take over the function of the target? For one, the antecedent premise must now p-scope over the conditional premise, as it does in (3b). But more importantly, what should be the effect of the rule? As shown in (3b), the antecedent of the conditional premise is *eliminated*—where by ELIMINATING a child of a binary node we mean replacing the binary node with its other child, or perhaps more intuitively: we delete the constituent we are eliminating (in (3b), $\neg\mathcal{A}$) along with the parent junction node ($\vee$), and "reconnect" the other child of the junction (C) to the grandparent ($\wedge$).

Our new rule, Prune, will embody this shift in perspective. Rather than replacing the antecedent premise by the consequent, it will eliminate the "antecedent half" of the conditional premise. Prune will have all the desirable attributes discussed in the previous subsection. It will be simple both to formulate and to use, and it will deal with various polarities of all the components in a completely transparent way. These properties will be achieved because Prune, unlike iMP/iMT, will fully exploit the resources available in DDS. But to finish paving the way toward this, we need to eliminate the final vestige of dependence on form, implicit in the application of proto-Prune in (3b).

In (3b), $\mathcal{A}$ (positively) p-scopes over $\neg\mathcal{A}$. We can imagine this as a *conflict* between $\mathcal{A}$ and its negation $\neg\mathcal{A}$: p-scope brings the premise $\mathcal{A}$ to the target $\neg\mathcal{A}$, and the appearance of a formula ($\mathcal{A}$) and its negation $\neg\mathcal{A}$ in the same location constitutes a conflict. (Of course, nothing really happens just by virtue of the existence of some p-scope relation, but the conflict could be made explicit by Copying the premise next to the target, which creates a *contradiction*, i.e. a formula of the form $\mathcal{A} \wedge \neg\mathcal{A}$.)

However, basing the idea of a conflict on the opposition between a formula and its negation takes away the possibility of fully deploying the DDS toolbox, or at least complicates the precise formulation of the rule. You see, the situation in (3b) constitutes a conflict only because both the p-scope (of the premise at the target) and the target polarity are positive. If the premise p-scoped negatively over the target, or if the target was of negative polarity (but not both), the conflict would not arise. To formulate Prune with precision, we need to pay attention to polarities! But if we have to pay attention to polarities anyway, why not let them do the job currently assigned to negation? That is, why insist that the conflict must involve a *formal* opposition between a formula and its negation? Can we not express this opposition by the concept of polarity alone?

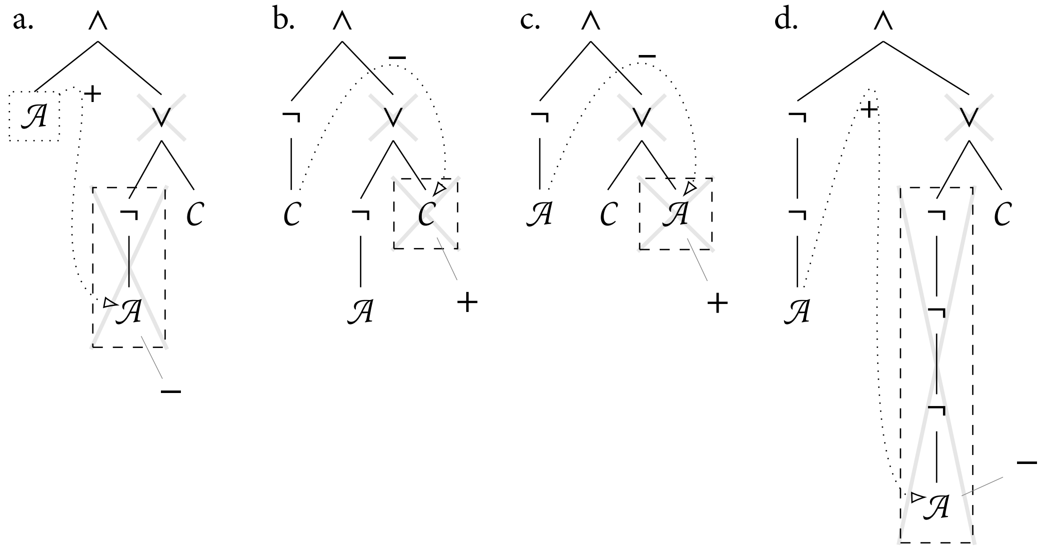
Sure we can: we hereby define CONFLICT as a situation where a constituent of some form ($\mathcal{A}$) p-scopes over a target of the same form ($\mathcal{A}$), but where additionally the p-scope polarity and the target polarity are opposite.[5]

Clearly, the concept of conflict cannot be completely divorced from form, but the above definition reduces the formal requirements to the absolute minimum: the formal identity of the premise and the target. All the rest is left to polarity, in fact to p-scope. The premise must p-scope over the target, with some polarity, and the polarity of the target must be just the opposite. By 'the polarity of the target' we of course refer to the target's constituent polarity, corresponding to the number of negations above it—but remember that as any constituent is in the descendant p-scope of itself, the target's constituent polarity equals the p-scope polarity of the target over itself.

[5] One other (pleasingly symmetric) way to define conflict would be to say that it arises when two constituents of the same form p-scope over the same target (of any form) with opposite p-scope polarities. The conflict would of course arise at the target.

Let's now see how the two shifts of perspective, from replacement to elimination, and from the formal to the polarity-based expression of opposition, result in a fully general and aesthetically pleasing formulation of Prune. (4a) shows the same tree as (3b) above—and consequently, we want to eliminate the same constituent—but the idea behind (4a) is that the premise p-scopes over $\mathcal{A}$ rather than $\neg A$. The negation above the target $\mathcal{A}$ affects the situation by contributing toward the (constituent) polarity of the target, which is negative and thus in opposition to the p-scope polarity of the premise.

(4) a. b. c. d.



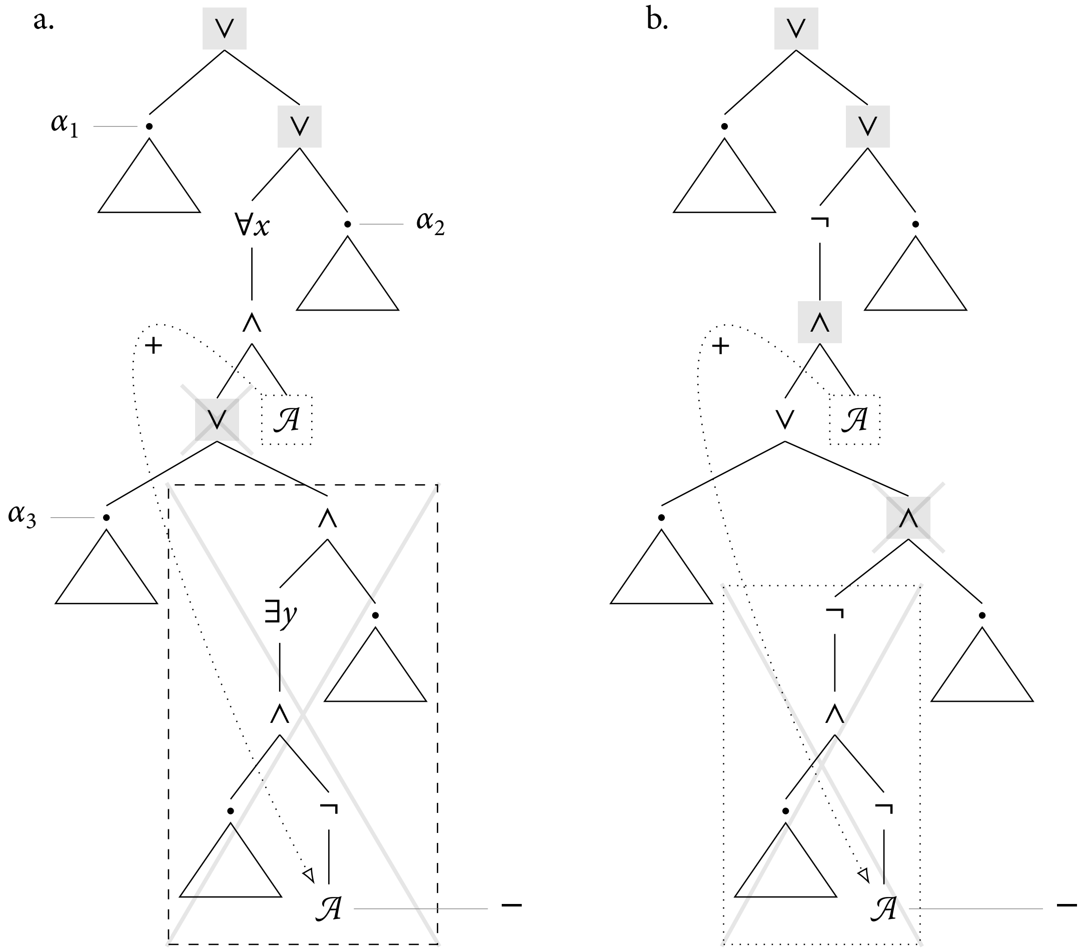
(4a) is an example of Modus Ponens, but this approach also subsumes Modus Tollens (4b) and Disjunctive Syllogism (4c) with the same ease, and it is not bothered by multiple negations (over either the premise or the target) or the order of disjuncts (4d). The only thing that matters is that we have a constituent p-scoping, with some polarity τ, over another constituent of the same form but with constituent polarity $-\tau$. If you will, imagine Prune as a sort of anti-Copy: we "try to Copy" the premise over the target of the same form, but if the target has the wrong polarity—Prune!

Let's now reveal the final detail of our rule Prune: which constituent do we eliminate? According to examples in (4), clearly not the conflict site, i.e. the target of p-scope! This makes Prune somewhat unusual compared to other inline rules but it is not problematic since the rule will turn out to be sound, as we shall prove in the Appendix.

In all the examples in (4), we eliminated the disjunct containing the conflict site (remember that by 'disjunct', we mean a child of a disjunction node, and that eliminating the disjunct *constituent* also involves removing the disjunction *node*, but the other disjunct is left intact). This situation can be complicated by two factors: multiple disjunctions, and negation.

Let's first turn to a situation involving several disjunctions in the formula. Such a situation is presented in (5a), highlighting all the disjunctions. Which disjunct do we eliminate? It turns out that, in general, we may only eliminate the smallest disjunct containing the conflict, which we will call the LOCAL DISJUNCT. To find it, we move up the tree, starting at the p-scope target, until we hit a disjunction. Once we get to the disjunction, we remove both the disjunction node, and the entire constituent on the branch we have arrived from.

(5) a. b.



The reader familiar with *reductio ad absurdum* has probably noticed that there is more than just a passing resemblance between our Prune and this form of argument. Imagine we are traveling from the root of the tree down to the site of conflict. Disjunctions "introduce possibilities," so every time we come upon a disjunction, we decide to explore one possible scenario—or rather, we decide to *not* follow the other possibility. In effect, at each disjunction on our way down, we make the assumption that the other branch is false, or that its negation is true; in (5a), we therefore first assume $\neg\alpha_1$, then $\neg\alpha_2$ and finally $\neg\alpha_3$.[6] The conflict is

[6] In formal languages which admit implication $\alpha \Rightarrow \beta$, the assumption is the antecedent α. In a polarity logic such as ours, $\alpha \Rightarrow \beta$ unabbreviates to $\neg\alpha \vee \beta$, so when we decide to go down the β fork, we are really assuming the negation of $\neg\alpha$, i.e. α. Modeling assumptions using implications or disjunctions therefore comes down to the same thing.

therefore relative to all these assumptions, and we can only conclude that assuming $\neg\alpha_3$ was the final straw triggering the conflict. And indeed, by eliminating the local disjunct of the target in (5a) we leave α_3 in place of the eliminated disjunction, which amounts to rejection of the final assumption $\neg\alpha_3$.

We now know that Prune should eliminate the local disjunct of the conflict site. But as announced above, we have one final issue to address: the effect of negations.

At this point in our discussion of the Dynamic Deductive System, it should come as no surprise that a negation reverses the roles of conjunctions and disjunctions it dominates. If a disjunction is in the scope of an odd number of negations, it is not really a disjunction; it is effectively a conjunction, or an e-conjunction for short, and is as such irrelevant in our search for the local disjunct to eliminate. And conversely, if our search for the local disjunct, starting at the conflict site and proceeding up the tree, encounters a conjunction of negative polarity, this conjunction is effectively a disjunction, meaning our search is finished. This is illustrated in (5b), where all the e-disjunctions are highlighted.

Summing up our discussion of Prune, if we have two constituents of the same form such that one (the premise) p-scopes over the other (the target) and that the polarity of the p-scope and the target are opposite, we may eliminate the local e-disjunct of the target.

<u>Prune</u>

$\mathcal{A}^{-\tau} \vdash^{\tau}_{\mathcal{A}}$ eliminate the local e-disjunct of the target

Before moving on to the next subsection and using Prune in earnest, let us take a moment to remember why we wanted to develop Prune in the first place. In subsection 7.1.1, we saw that it makes sense to generalize Inline Modus Ponens and Tollens if we want to use them when their conditional premise has negative p-scope polarity. But then we saw that if we pay attention to polarity, we can do better than those rules. With polarity in hand, we need only the rudimentary syntactic operations of Copy, Prune, and Delete (we will not need Add to simulate iMP and iMT), yielding a more aesthetically elegant and deeper deductive system.

7.1.3 Prune at work

Let's look at some applications of Prune. Obviously, the first thing we want to do is show how to carry out the inferences where we formerly employed Inline Modus Ponens and Inline Modus Tollens. After that we will look at some new applications.

Let's review the procedure for using Prune. We take a target of some polarity, and find a formally identical constituent (the premise) which p-scopes over the target, but with a p-scope polarity opposite of the constituent polarity of the target.

This makes the target a conflict site and marks it for deletion. We erase the target and start moving up the tree, erasing everything until we hit an e-disjunction. We do not erase the e-disjunction completely: we have already erased one of its child constituents, and now we erase the connective (either $\lor$ or $\land$, depending on the polarity) as well, but we leave the other e-disjunct intact (and fix the bit of the link between the disjunct and its ex-grandparent that is missing). When we are doing a DDS deduction on a whiteboard, the eraser is often more useful than the marker!

On that note, it is worth remarking that the operations we are engaged in are in some sense, sub-logical. By that we mean that logical rules of inference (like Modus Ponens and the *dicta* rules) typically are stated at a certain level of abstraction in which we do not worry about how substitutions take place. But here, now that we are using Copy and Prune, and Add and Delete, we are dealing with rules that are so low level that they speak to how the inference mechanically takes place. It isn't exactly at the level of hardware implementation, but it is certainly at the level of how we carry out the operation on a whiteboard (as noted in the previous paragraph). The interesting observation to take away from this, in our opinion, is that if we let polarity (and in particular p-scope) guide our application of these sub-logical rules, we have effectively taken the Medieval logicians' idea of reducing all of logic down to two basic rules of inference and made the idea even deeper—one can reduce all of logic down to a handful of rudimentary non-logical syntactic operations, plus polarity.

To see how this works, it will be helpful comparing a deduction of the syllogisms using the old tools (iMP or iMT) with the new (non-logical) tools (we will need Copy, Prune, and Delete). Let's start with (6), where we might earlier deploy Inline Modus Ponens—a traditional syllogism of the form Barbara, or what today we might call a Hypothetical Syllogism. First, let's consider that deduction using Inline Modus Ponens, as in (7).[7] What we need to keep in mind from this example is that, as we are targeting the antecedent premise, the rule is applicable because the conditional premise (positively) p-scopes over the antecedent premise, and the effect of the rule is to *replace* the target by the consequent.

(6) a. Every pet is a dog. $\forall x\colon \neg P(x) \lor D(x)$

b. Every dog is a animal. $\forall x\colon \neg D(x) \lor A(x)$

c. $\therefore$ Every pet is an animal. $\forall x\colon \neg P(x) \lor A(x)$

(7) 1. $\forall x\,(\neg P(x) \lor \underbrace{D(x)}_{\text{IMP}}) \land \forall x\,(\underline{\neg D(x) \lor A(x)})$

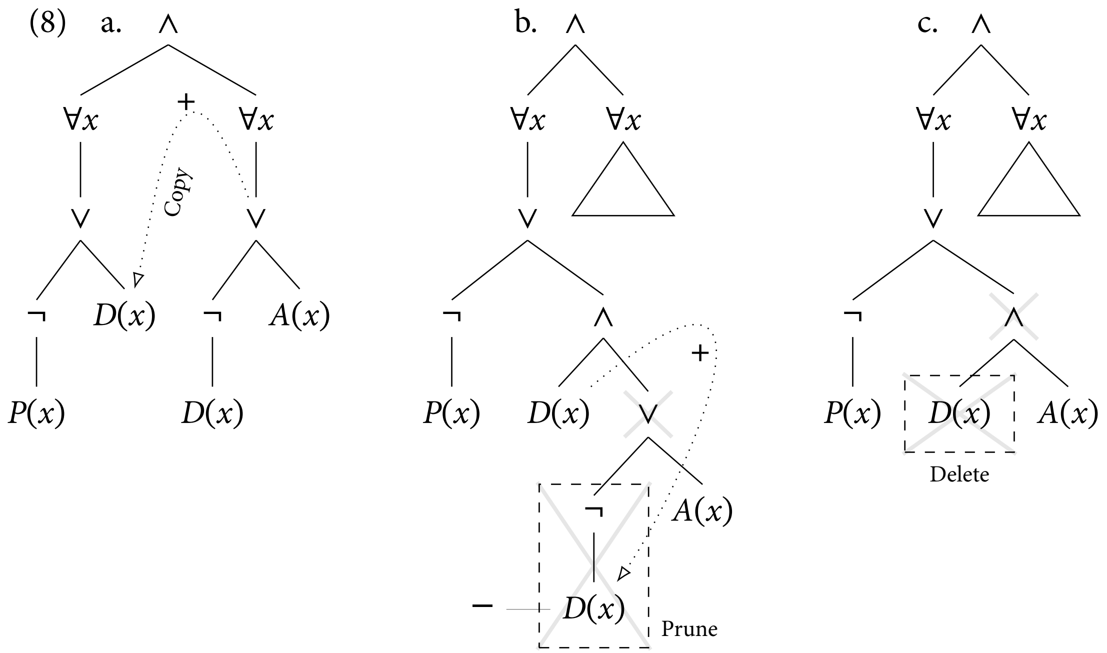
2. $\underline{\forall x\,(\neg P(x) \lor \overline{A(x)})} \land \forall x\,(\neg D(x) \lor A(x))$

[7] Throughout this section, we will take advantage of the idea from subsection 6.5.3 that any constituent positively p-scoping over the root can be seen as *a* conclusion. We do that for clarity of exposition—in absence of this convention, it is trivial to isolate the intended conclusion by center Copy or Delete(s).

We have already discussed how this is an instance of the *dictum de omni* inference paradigm, but in this case, as noted, we are going to push our analysis even deeper than the *dicta* analysis did. As we know from the previous subsection, deducing the same conclusion using the Prune rule requires a different approach. But the schematic examples (3) and (4) we used to introduce Prune were not fully realistic: it is not often that we find the antecedent and the conditional premise of iMP/iMT p-scoping over one another. And indeed, the antecedent premise $D(x)$ does not p-scope over the conditional premise $\neg D(x) \vee A(x)$ in (7.1), so we cannot apply Prune directly. Simulating iMP using Prune in general requires a preemptive step, Copy, and a clean-up step, Delete.

The deduction of the above inference using Prune is shown in tree form in (8). (8a) shows the preemptive Copy. The idea behind this step is that, if we want to apply Prune, we need to bring the conditional premise within the p-scope of the antecedent premise, and this is precisely what Copy achieves in (8a). In (8b), we apply Prune itself. We can apply it because the upper $D(x)$ positively p-scopes over the lower $D(x)$ of negative polarity; the eliminated e-disjunct is $\neg D(x)$. In (8c), we perform the clean-up. Unlike in the application of iMP, the antecedent premise $D(x)$ is not replaced by the consequent $A(x)$. To arrive at the same conclusion as with iMP, we should therefore Delete it: this step is applicable because the consequent ended up e-conjoined to the antecedent premise.

(8)



By its low level nature, Prune requires that we be more explicit (i.e. more honest) about what is happening in the syntax of these inferences. And if we examine the inference at fine enough grain, it is easy to see that there are three syntactic operations instead of one: Copying brings the conditional premise within the p-scope of the antecedent premise; Prune removes the antecedent within the conditional premise; Delete eliminates the original antecedent premise. If we represent these operations in linear fashion, the steps are as in (9).[8]

(9)
1. $\forall x\,(\neg P(x) \vee \underbrace{D(x)}_{\text{COPY}}) \wedge \forall x\,(\neg D(x) \vee A(x))$
2. $\forall x\,(\neg P(x) \vee (D(x) \wedge (\neg \underbrace{D(x) \vee}_{\text{PRUNE}} A(x)))) \wedge \forall x\,(\neg D(x) \vee A(x))$
3. $\forall x\,(\neg P(x) \vee (\underbrace{D(x) \wedge}_{\text{DELETE}} A(x))) \wedge \forall x\,(\neg D(x) \vee A(x))$
4. $\forall x\,(\neg P(x) \vee A(x)) \wedge \forall x\,(\neg D(x) \vee A(x))$

We proceed in a similar fashion with Inline Modus Tollens. The deduction using Prune shown in (12) is exactly as for iMP above, except that we now have a positive polarity target (the right $A(x)$) in a negative p-scope of the antecedent (the left $A(x)$) in (12.2). Consequently, it is the non-negated member ($A(x)$) of the conditional that gets Pruned, leaving us with the negated member ($\neg D(x)$).

(10)
a. Every animal is barking. $\quad \forall x\colon \neg A(x) \vee B(x)$
b. Every dog is a animal. $\quad \forall x\colon \neg D(x) \vee A(x)$
c. ∴ Every dog is barking. $\quad \forall x\colon \neg D(x) \vee B(x)$

(11)
1. $\forall x\,(\neg \underbrace{A(x)}_{\text{IMT}} \vee B(x)) \wedge \forall x\,(\neg D(x) \vee A(x))$
2. $\forall x\,(\neg D(x) \vee B(x)) \wedge \forall x\,(\neg D(x) \vee A(x))$

(12)
1. $\forall x\,(\underbrace{\neg A(x)}_{\text{COPY}} \vee B(x)) \wedge \forall x\,(\neg D(x) \vee A(x))$
2. $\forall x\,((\neg A(x) \wedge (\underbrace{\neg D(x) \vee A(x)}_{\text{PRUNE}})) \vee B(x)) \wedge \forall x\,(\neg D(x) \vee A(x))$
3. $\forall x\,((\underbrace{\neg A(x) \wedge}_{\text{DELETE}} \neg D(x)) \vee B(x)) \wedge \forall x\,(\neg D(x) \vee A(x))$
4. $\forall x\,(\neg D(x) \vee B(x)) \wedge \forall x\,(\neg D(x) \vee A(x))$

The Prune-based simulation of iMP/iMT certainly contains more steps than the original, but that is to be expected, given that we are being transparent about the nitty gritty elements involved in the syntactic execution of the inference. We are, after all, breaking things down to the level of copying and deleting bits of syntax.

[8] For an application of Prune such as (9.2), we need to mark both the p-scope target and the eliminated constituent. We indicate both by underbracketing; the underbracket of the eliminated constituent also includes the eliminated connective. The markings are never ambiguous, as the eliminated constituent always dominates the p-scope target.

However, there are occasions where, even respecting this level of nitty gritty syntactic detail, Prune gets by with fewer steps. For example, to deduce the inference in (13), an iMP-based approach in (14) needs two preemptive applications of Delete: without them, the formal matching requirement of iMP (that the target must match the antecedent within the premise) would not be satisfied. These preemptive Deletions are not necessary in the Prune-based approach in (15). The formal requirements of Prune are minimized, their function taken over by p-scope: observe how the left $D(x)$ in (15.2) (positively) p-scopes through the parent conjunction into the negated disjunction containing the right (negative) occurrence of $D(x)$.

(13) a. Some pet is a black dog. $\exists x\colon P(x) \land (B(x) \land D(x))$
b. Every dog or cat is here. $\forall x\colon \neg(D(x) \lor C(x)) \lor H(x)$
c. ∴ Some pet is here. $\exists x\colon P(x) \land H(x)$

(14)
1. $\exists x\,(P(x) \land (\underbrace{B(x) \land D(x)}_{\text{DELETE}})) \land \forall x\,(\neg(D(x) \lor C(x)) \lor H(x))$
2. $\exists x\,(P(x) \land \overline{D(x)}) \land \forall x\,(\neg(\underbrace{D(x) \lor C(x)}_{\text{DELETE}}) \lor H(x))$
3. $\exists x\,(P(x) \land \underbrace{D(x)}_{\text{IMP}}) \land \forall x\,(\underline{\neg\overline{D(x)} \lor H(x)})$
4. $\underline{\exists x\,(P(x) \land \overline{H(x)})} \land \forall x\,(\neg\overline{D(x)} \lor H(x))$

(15)
1. $\exists x\,(P(x) \land (\underbrace{B(x) \land D(x)}_{\text{COPY}})) \land \forall x\,(\underline{\neg(D(x) \lor C(x)) \lor H(x)})$
2. $\exists x\,(P(x) \land ((B(x) \land \underline{D(x)}) \land (\underbrace{\neg(D(x) \lor C(x)) \lor H(x)}_{\text{PRUNE}}))) \land \ldots$
3. $\exists x\,(P(x) \land (\underbrace{(B(x) \land D(x)) \land \overline{H(x)}}_{\text{DELETE}})) \land \ldots$
4. $\underline{\exists x\,(P(x) \land \overline{H(x)})} \land \forall x\,(\neg(D(x) \lor C(x)) \lor H(x))$

Now consider our "problematic" example from subsection 7.1.1. We had decided to represent the negative determiner 'no' using a negated existential quantifier and were therefore unable to use iMP or iMT directly, because the conditional premise (or rather, the constituent that we know should function as one) had negative p-scope polarity. However, the derivation using Prune is no different than above—it is certainly of the same length—it is just that some polarities are different now.

The "conditional premise" $A(x) \land C(x)$ has negative p-scope polarity over the left $A(x)$ in (17.1), so Copying it negates it. The new occurrence of $A(x)$ in (17.2) thus has negative polarity, and as it is within a positive p-scope of the original, left occurrence of $A(x)$, we are allowed to apply Prune. We will eliminate the local e-disjunct of the target $A(x)$, but which constituent is this? To see that this constituent is the target $A(x)$ itself, observe that its parent, being a conjunction of negative polarity, is an e-disjunction. Performing the elimination thus leaves us with $C(x)$, which "inherits" the negation from the earlier Copy. As usual, the clean-up step (17.3) eliminates the original antecedent premise.

(16) a. Some visitor is an alien. $\wedge\exists x\,(V(x) \wedge A(x))$
b. No alien is a cat. $\neg\exists x\,(A(x) \wedge C(x))$
c. ∴ Some visitor is not a cat. $\exists x\colon V(x) \wedge \neg C(x)$

(17) 1. $\exists x\,(V(x) \wedge \underbrace{A(x)}_{\text{COPY}}) \wedge \neg\exists x\,(A(x) \wedge C(x))$

2. $\exists x\,(V(x) \wedge (A(x) \wedge \neg\underbrace{(A(x) \wedge C(x))}_{\text{PRUNE}})) \wedge \ldots$

3. $\exists x\,(V(x) \wedge \underbrace{(A(x) \wedge \neg C(x))}_{\text{DELETE}}) \wedge \ldots$

4. $\exists x\,(V(x) \wedge \neg C(x)) \wedge \ldots$

Let's review the precise formulation of Prune one more time. Given a target $\mathcal{A}$ of polarity τ, and a premise of the same form ($\mathcal{A}$), which p-scopes over the target with polarity $-\tau$, we may eliminate the local e-disjunct of the target. But what happens if there is no e-disjunction above the polarity conflict? For example, what if we are given a formula in the form of a contradiction $\mathcal{A} \wedge \neg\mathcal{A}$, which would constitute a "global" conflict in our system? The short answer is that we can't apply Prune then. But we know that given a contradiction in a traditional deductive system, we can deduce *anything*. Can we achieve that using DDS? Of course we can. All we need to do is to first apply Add to the root—$\mathcal{B}$ below can be any formula at all—Pruning the polarity conflict then leaves us with $\mathcal{B}$ alone.

(18) 1. $\underbrace{\mathcal{A} \wedge \neg\mathcal{A}}_{\text{ADD}}$

2. $\underbrace{(\mathcal{A} \wedge \neg\mathcal{A})}_{\text{PRUNE}} \vee \mathcal{B}$

3. $\mathcal{B}$

While Prune is the final piece of the DDS puzzle as far as propositional logic is concerned, we still have some unfinished work to do in the predicate logic department. We have mentioned quantification briefly at the end of the previous chapter, but there are still several details left to be worked out. We move on to these details in the next section.

7.2 Quantifier Introduction and Elimination

At the end of Chapter 6, we explained how quantifiers fit into the computation of p-scope, seeing that universal quantifiers host Angels (same as conjunctions), while existential quantifiers are home to Demons (same as disjunctions). But this said nothing about how quantifiers are *manipulated* (eliminated and introduced) in Dynamic Deductive System, and this is the task we will take up in this section.

In subsection 7.2.1, we will show how we achieve the effect of Universal Instantiation (UI), and then turn to Universal Generalization (UG) in subsection 7.2.2. Foreshadowing the discussion in these sections, it should come as no surprise that the DDS implementation of UI and UG doesn't always manipulate the *universal* quantifier in DDS. In negative polarity environments, we will be eliminating and introducing an *existential* quantifier. In other words, our UI and UG are really e-UI and e-UG, because they are eliminating and introducing e-universal quantifiers.

Another thing about e-UI and e-UG is that we will not actually formulate them as independent rules. We will rather implement e-universal instantiation and generalization as the initial and the final part of any application of Copy, Prune, Add, and Delete. In subsection 7.2.3, this will lead us to the realization that we don't ever need to see an open (dynamic) formula in a DDS deduction (unless we want to, for some reason)—and we will also explain why we find the idea of such, closed, deductive systems appealing.

As the section overview above exhausts the list of subsections, you might wonder what ever happened to e-*existential* elimination and introduction? Existential Instantiation (EI) and Existential Generalization (EG) are, after all, part and parcel of many natural deduction systems (sometimes as admissible rules). Shouldn't one expect they will turn up, in the e-existential incarnation typical for DDS, in our system as well? The answer is that these rules are completely absent from our system. Really, the traditional EI and EG are not a part of DDS in any shape or form. Of course, this doesn't imply that DDS cannot deduce the inferences where other deductive systems typically deploy these rules—we have actually already presented many such deductions. We will explain how we *can* have our cake and eat it in section 8.2 of the next chapter. Right now, it is time to explain exactly how we want to implement the e-universal part of the equation.

7.2.1 Implicit e-Universal Instantiation

Let us let you in on a secret: Universal Instantiation was part of many DDS deductions we have presented so far. You perhaps haven't noticed it because we have carefully selected examples where it has no visible effect.

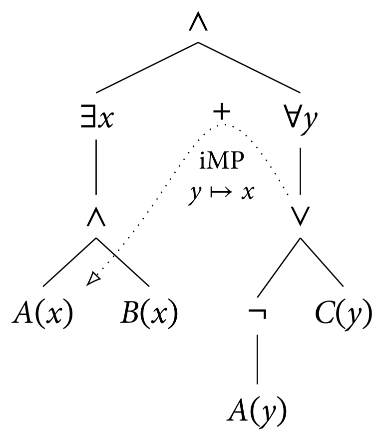
Consider the inference below (we stick to Inline Modus Ponens for familiarity of exposition). The middle and the right column of (19) present two alternative sets of logical forms of the sentences, the only difference being in the choice of the universally bound variable of (19b)—x in the middle column and y in the right column. It is clear that renaming the variable should have no effect on the deduction—it certainly does not affect classical deductions. So how does this work in DDS?

(19)	a. Some $\underbrace{\text{alien}}_{\overset{+}{\text{IMP}}}$ barks.	$\exists x\,(\underbrace{A(x)}_{\overset{+}{\text{IMP}}} \wedge B(x))$	$\exists x\,(\underbrace{A(x)}_{\overset{+}{\text{IMP}}} \wedge B(x))$
	b. $\underline{\text{Every alien is a cat.}}$ (+)	$\wedge\forall x\,(\neg A(x) \underset{+}{\vee} C(x))$, $x \mapsto x$	$\wedge\forall y\,(\neg A(y) \underset{+}{\vee} C(y))$, $y \mapsto x$
	c. ∴ Some $\overbrace{\text{cat}}$ barks.	$\exists x\colon \overbrace{C(x)} \wedge B(x)$	$\exists x\colon \overbrace{C(x)} \wedge B(x)$

If we were to simulate an application of an inline rule in a classical deductive system,[9] we would perform a universal instantiation for every quantifier Perceval meets in the North-bound part of his journey.[10] This holds even for existential quantifiers on this path: the properties of p-scope ensure that any quantifier town (universal or existential) Perceval meets on his journey North is *effectively* universal.[11] In fact, that is the whole point of p-scope, at least from the quantificational perspective. And this is why we are allowed to do (for these quantifiers) what classical Universal Instantiation does: replace the bound variable with any term.[12]

Therefore, all we need to do to solve the riddle of the right column of (19) is instantiate y to x. We expressed this by writing $y \mapsto x$ under the premise; see also (20) for visualization. Strictly speaking, the instantiation happens in the middle column of (19) as well, but as we instantiate x to itself, there is no visible effect.

(20)



[9] In fact, this is precisely what we do in section A.2 to prove the soundness of DDS.

[10] In a classical deduction, these universal instantiations would be applied in the reverse order of Perceval's walk, i.e. top–down, as we would be disassembling the premise.

[11] Strictly speaking, this extends the notion of effectiveness introduced in section 6.1. There, we talked about the effective form of the target (and the replacement), and we have seen that it depends on the constituent polarity of the target. But as any constituent always p-scopes over itself (with the p-scope polarity equal to the constituent polarity), this is just a special case of effectiveness with respect to some p-scoping relation. In other words, the generalized notion of effectiveness applies to any pair of a premise and a target, not just to situations where they are one and the same constituent.

[12] There is a technical complication (in both DDS and classical systems). Given $\forall x\,\varphi$ as a premise, UI allows us to conclude $\varphi(t/x)$, where t is an arbitrary term, with the caveat that t must be free for variable x in formula φ, i.e. there should be no variable y occurring in t such that y is bound by some quantifier in φ. Plainly put, the instantiation should not create any new binding relations.

What is important to realize is that Implicit Universal Instantiation described above is *not a rule* of DDS. It is a part of the procedure which “prepares” each premise for the form-matching mechanism of the rule. What we did in (19) can only work if the implicit instantiation transforms the premise *before* the premise is “seen” by the rule; iMP used in (19) requires a formal match between the target $A(x)$ and the antecedent of the conditional premise. Prior to the implicit UI, there is no match (in the right column), as the antecedent of the conditional premise is $A(y)$. It is only after the implicit UI that the forms match.

Below, we show how the implicit universal instantiation plays a crucial role in deriving (inline) Existential Generalization.[13] Once we Add the existential quantifier, the Demonic nature of that quantifier and its parent disjunction make sure that $D(x) \wedge B(x)$ negatively p-scopes over $D(f) \wedge B(f)$, laying grounds for Prune. And now, the punch-line: the premise is only formally identical to the target due to the implicit universal instantiation of x to f.

(21) a. Fifi is a dog. Fifi is barking. $D(f) \wedge B(f)$
b. ∴ Some dog is barking. $\exists x\colon D(x) \wedge B(x)$

(22) 1. $\underbrace{D(f) \underset{+}{\wedge} B(f)}_{\text{ADD}}$

2. $\underbrace{(D(f) \underset{+}{\wedge} B(f))}_{\text{PRUNE}} \vee \exists x\colon \underbrace{D(x) \underset{-}{\wedge} B(x)}_{x \mapsto f}$

3. $\exists x\colon D(x) \wedge B(x)$

7.2.2 Implicit e-Universal Generalization

Having shown how Universal Instantiation (UI) is realized in DDS, we now turn to Universal Generalization (UG). The result will be similar: UG is not a *rule* of DDS, but rather a part of an application of any rule, just as with UI. However, while the implicit UI is performed at the beginning of a rule application, the Implicit Universal Generalization is performed at the end.

Let us jump straight to an example, which shows that the implicit UG is instrumental in raising the universal quantifier over the existential. We “raise” $\forall x$ (and its restriction) in two steps. We first (24.1) apply Add, joining $\neg D(z)$ to the hypothesis. Crucially, as the final part of Add, we introduce the universal quantifier

[13] The procedure is in fact completely analogous to how one would derive Existential Generalization in a Hilbert-style system, except that it works inline. One could even say that our procedure is more transparent, because it does not involve unabbreviating the existential quantifier $\exists x$ in to $\neg\forall x\neg$. In fact, the existential quantifier is *not* defined as an abbreviation in DDS! For further discussion on Existential Instantiation, see subsection 8.2.2.

$\forall z$ atop the conclusion of Add (in the justification, we note this by adding $\forall z$ in front of the rule name). The second step (24.2) is to Copy the conditional over its parent quantifier (we can do this because $\forall x$, being an e-universal quantifier, belongs to its ancestor p-scope), which eliminates the original $\forall x$. The step starts by the implicit UI, which renames variable x to the recently introduced z (note that Perceval visits $\forall x$ in his single, North-bound leg of the journey) and therefore rebinds it to the quantifier introduced in the first step. The final step (24.3) of the deduction is a straightforward application of Prune.

(23) a. Some cat is being barked at by every dog. $\exists y\colon C(y) \land \forall x\colon \neg D(x) \lor B(x,y)$

b. ∴ Every dog is barking at some cat. $\forall x\colon \neg D(x) \lor \exists y\colon C(y) \land B(x,y)$

(24) 1. $\underbrace{\exists y\colon C(y) \land \forall x\colon \neg D(x) \lor B(x,y)}_{\forall z\ \text{ADD}}$

2. $\forall z\colon \neg D(z) \lor \exists y\colon C(y) \land \underbrace{\forall x\colon \underbrace{\neg D(x) \lor B(x,y)}_{x\mapsto z}}_{\text{COPY}}$

3. $\forall z\colon \neg D(z) \lor \exists y\colon C(y) \land (\underbrace{\neg D(z) \lor B(z,y)}_{\text{PRUNE}})$

4. $\forall z\colon \neg D(z) \lor \exists y\colon C(y) \land B(z,y)$

Clearly, there are restrictions on the implicit UG. One thing we can safely anticipate, given our experience with DDS, is that we may only introduce a universal quantifier in a positive polarity environment. In a negative polarity environment, we may introduce an existential quantifier. In general, the implicit UG may therefore introduce an effectively universal quantifier.

Let's illustrate this by negating the sentences of the previous example. When we do this, the direction of the entailment reverses. We proceed with the deduction exactly as above, but as the premise is negated, all the polarities will be reversed. Add in (26.1) now introduces a conjunction, and the implicit UG at the end gets us an existential quantifier. We remove the existential (but e-universal) quantifier by Copying in (26.2); note that the negative target polarity and the negative p-scope polarity of the premise cancel each other out. In (26.3), the e-disjunct we eliminate is a conjunct of negative polarity.

(25) a. It is not the case that every dog is barking at some cat. $\neg\forall x\colon \neg D(x) \lor \exists y\colon C(y) \land B(x,y)$

b. ∴ It is not the case that some cat is being barked at by every dog. $\neg\exists y\colon C(y) \land \forall x\colon \neg D(x) \lor B(x,y)$

(26) 1. $\underbrace{\neg\forall x\colon \neg D(x) \lor \exists \underset{-}{y}\colon C(y) \land B(x,y)}_{\exists z\ \textsc{add}}$

2. $\neg\overline{\exists z\colon C(z) \land \forall x\colon \neg D(x) \lor}\ \underbrace{\exists y\colon \underset{\underset{-}{y\mapsto z}}{\underline{C(y) \underset{-}{\land} B(x,y)}}}_{\textsc{copy}}$

3. $\neg\exists z\colon \underset{+}{\underline{C(z)}} \land \forall x\colon \neg D(x) \lor (\overline{\underbrace{\underset{-}{C(z)}}_{\textsc{prune}} \land B(x,z)})$

4. $\neg\exists z\colon C(z) \land \forall x\colon \neg D(x) \lor \overline{B(x,z)}$

In the previous two examples, we deliberately chose to introduce a new variable (z)—but we could have safely introduced x and y—just to have a reason to show how we can rename variables. To rename a variable bound by an e-universal quantifier, we Copy the scope over the quantifier, renaming the variable in the premise of Copy, and perform the implicit UG (of the new variable) as the final part of Copy.

(27) 1. $\exists y\colon C(y) \land \underbrace{\forall z\colon \underset{\underset{+}{z\mapsto x}}{\underline{\neg D(z) \underset{+}{\lor} B(z,y)}}}_{\forall x\ \textsc{copy}}$

2. $\exists y\colon C(y) \land \overline{\forall x\colon \neg D(x) \lor B(x,y)}$

Note that the one thing we may *not* do above is rename x to y, as a quantifier binding y already exists above the target. This is a general restriction on the implicit UG: we can introduce any e-universal quantifier, but only if it binds a variable not bound by a quantifier dominating the target (or a variable which is free in the dynamic formula; free variables are implicitly universally quantified, in a sense).

To rename the variable bound by an e-existential quantifier, the procedure is more complicated on first sight. The Copy approach does not work, because the scope of the existential does not p-scope over its quantifier (the descendant and the relative p-scope polarity do not match, so there is no ancestor p-scope). We need to add the renamed version using Add and then use Prune to remove the original.

(28) 1. $\underbrace{\exists y\,(\underset{+}{C(y)} \land B(y))}_{\textsc{add}}$

2. $\overline{\underbrace{\exists y\,(\underbrace{C(y) \underset{+}{\land} B(y)}_{+})}_{\textsc{prune}} \lor \exists x\,(\underset{x\mapsto y}{\underline{C(x) \underset{-}{\land} B(x)}})}$

3. $\overline{\exists x\,(C(x) \land B(x))}$

However, (27) only appears to be simpler than (28). Remember that we really consider the center Copy to be a derived rule, an abbreviation of the sequence of the left/right Copy plus Delete. The application of Copy in (27) thus really consists of two steps, just as (28). In fact, the symmetry that we now see—using Copy + Delete in a positive polarity environment, and Add + Prune in a negative polarity environment—is typical of many deductions using DDS.

We know that negative determiner 'no' can be analyzed in two ways. Can we deduce one logical form from the other? (30) shows how to do that (for one direction, in a positive polarity environment). There are two parts to the deduction. We are primarily interested in the first step (Copy), which transform the negated existential quantifier into a universal quantification over a negation; note that the ancestor p-scope in this step is of the kind where Perceval gets the permission to deliver the map at the e-universal $\exists x$, but only uses it in the immediately dominating negation town. The subsequent steps derive one of De Morgan's laws using a combination of Add and Prune.

(29) No dog is barking. $\neg\exists x\colon D(x) \wedge B(x) \sim \forall x\colon \neg D(x) \vee \neg B(x)$

(30) 1. $\underbrace{\neg\exists x\,(D(x) \underset{-}{\wedge} B(x))}_{\forall x \text{ COPY}}$

2. $\forall x\,\underbrace{\neg(D(x) \underset{+}{\wedge} B(x))}_{\text{ADD}}$

3. $\forall x\,(\neg D(\underset{+}{x}) \vee \neg(\underbrace{D(\underset{-}{x})}_{\text{PRUNE}} \wedge B(x)))$

4. $\forall x\,(\neg D(x) \vee \neg B(x))$

7.2.3 Farewell to open formulas

An occurrence of a variable not bound by a quantifier is a free occurrence, and if a variable has a free occurrence in a formula, we say it is free in that formula. A formula containing free variables is OPEN and conversely, a CLOSED formula, also called a SENTENCE, contains no free variables. We know how to interpret a closed formula—the quantifiers tell us how. But how do we interpret an open formula?

(31) $D(x) \Rightarrow A(x)$

There is a *convention* to interpret any free variables in a formula as universally closed.[14] (This convention is often used when stating the axioms of a theory, for

[14] There is an alternative convention—popular among linguists rather than logicians—to interpret all free variables as referring expressions, where the reference is fixed by the assignment function otherwise used in the recursive definition of truth. For example, we could analyze the sentence 'Every dog likes him' as $\forall x\colon D(x) \Rightarrow L(x, y)$, which contains the free variable y as the reflex of the pronoun. The idea is that the context will fix the value of the assignment function at y. So if the assignment function maps y to John, the sentence will be understood as saying that every dog likes John.

example.) Under this convention, the formula above would be interpreted the same as the closed formula $\forall x\colon D(x) \Rightarrow A(x)$. But as we said, this is a convention—and the source of this convention are deductive systems! Remember that classical deductive systems typically need to "peel" the quantificational layers off of their hypotheses before they can use them. As a quick example, say we want to use the Hypothetical Syllogism (HS), which reads $\mathcal{A} \Rightarrow \mathcal{B}, \mathcal{B} \Rightarrow \mathcal{C} \vdash \mathcal{A} \Rightarrow \mathcal{C}$, to derive the argument in (32). Clearly, we cannot use the hypotheses as such. We first need to use Universal Instantiation on each premise to arrive at $D(x) \Rightarrow A(x)$ and $A(x) \Rightarrow M(x)$. (And afterwards, we need to use Universal Generalization on $D(x) \Rightarrow M(x)$, the conclusion of HS.)

(32) a. Every dog is an animal. $\forall x\colon D(x) \Rightarrow A(x)$
b. Every animal is mortal. $\forall x\colon A(x) \Rightarrow M(x)$
c. ∴ Every dog is mortal. $\forall x\colon D(x) \Rightarrow M(x)$

Clearly, it is the detour through open formulas which allows us to use propositional inference rules in a classical predicate logic deductive system. (And the universal closure convention is just another way of saying that Universal Generalization is a generally applicable rule.) However, we are not really interested in open formulas as such; we are interested in inferential relations between closed formulas, i.e. sentences. As pointed out by Quine (1981, p. iv), "these are what logic is for; schemata and even open [formulas] are technical aids along the way," and in fact, Quine goes to some lengths to develop an axiom system for first-order predicate logic which avoids using these technical aids.[15]

Let us say that a deductive system which cannot work without the technical aid of open formulas is an OPEN system, and let us call a system which cannot ever produce an open formula a CLOSED system. A classical deductive system with Universal Generalization is an open system, and Quine (1981) develops a closed system. Which category does our DDS belong to? We will see that, as it stands at the moment, it belongs to yet another category, which we will call SEMI-CLOSED: DDS can produce open formulas but it doesn't need to (in order to validly deduce any closed formula from other closed formulas).

As demonstrated by virtually every predicate logic example in Chapters 6 and 7, our Dynamic Deductive System doesn't need to "peel" away the quantificational layers of a formula in order to manipulate its propositional structure. It is therefore

[15] We could say that Quine develops a little bit of DDS. In our terminology, he proves that Inline Modus Ponens is applicable to premises of form $\forall(X \cup Y)\,(\varphi \Rightarrow \psi)$ and $\forall X\,\varphi$, yielding $\forall Y\,\psi$, where X and Y are the sets of free variables of φ and ψ, and $\forall Z$ is a shorthand for a string of quantifiers $\forall z$ consisting of all $z \in Z$; see Quine (1981, *111) for details.

clear that DDS is *not* an open system—but before we explain why it is not (yet) a closed system either, let us first think about what makes it non-open. Clearly, the inline nature of DDS rules of inference plays a major role here. Without inline rules, we could never perform deductions directly within the scope of quantifiers as we do. However, one further design property of DDS plays a role in bringing DDS closer to being a closed system. If we didn't implement quantifier introduction as we did in subsection 7.2.2, our deductive system would still occasionally require a detour through an open formula—a short detour, as we will see, but a detour nevertheless.

Let's put our implementations of quantifier elimination (as implicit e-universal instantiation) and quantifier introduction (as implicit e-universal generalization) from subsections 7.2.1 and 7.2.2 into the full context of an application of an inline rule in DDS. We have proposed to have e-universal instantiation and generalization as the initial and the final part of an application of any inline rule. In full detail, the procedure of applying an inline rule thus stands as follows.

(33) a. For each premise:
   i. Check whether it p-scopes over the target. It must.
   ii. Perform instantiation of the variables bound by quantifiers occurring in the North-bound part of Perceval's journey.
  b. Apply the rule, i.e. check if the forms of the target and the instantiated premises match the requirements of the rule, and produce the form of the CONCLUSION.
  c. E-universally generalize the conclusion with respect to any free variables of the conclusion not bound by a quantifier dominating the target (or free in the dynamic formula). This produces the REPLACEMENT.
  d. Substitute the replacement for the target.

The following example will explain why we decided to implement quantifier generalization as a part of rule application rather than a rule on its own. If step (33c) didn't offer the opportunity to bind the free variables of the conclusion, some applications of a rule would result in an open formula. Clearly, this would be the case even if we had a separate rule—let's call it Quantifier Introduction (QI)—which we could use to bind those free variables in the following step of the deduction. Let's illustrate this with (the initial part of) the deduction of the universal variant of the logical form for 'no' from the existential variant. If there was no step (33c), Copying in the first step of the deduction would force us on the one-step detour through the open formula (34.2). In the presence of step (33c), however, we can perform the universal generalization as a part of the Copy, effectively bypassing the open-formula step (34.2) of the deduction.

(34) No dog is barking. $\neg\exists x\colon D(x) \wedge B(x) \sim \forall x\colon \neg D(x) \vee \neg B(x)$

1. $\neg\exists x\,(D(x) \wedge B(x))$

   COPY WITHOUT STEP (33C)

2. $\neg(D(x) \wedge B(x))$

   QI

3. $\forall x\,\neg(D(x) \wedge B(x))$

4. ...

If we read (33c) carefully, we see that this clause does not *force* us to perform any e-universal generalizations; it only *allows* us to do so. This is why DDS (as it is currently defined) is not a closed, but merely a semi-closed deductive system. As things stand right now, DDS can still produce deductions such as (34) containing open formulas—if we wish. And in fact, sometimes this is something that we do wish. We will encounter one such occasion in the following section, where we will simulate a classical (Hilbert-style) deductive system. That system crucially relies on open formulas, so if we wish to simulate it using DDS, it must be possible for DDS to produce open formulas as well.

But closing off the possibility of the open-formula detour in DDS is exceedingly simple. All we need to do is change a single word in the definition of DDS: We must substitute 'all' for 'any' in (33c). As you can imagine, this is the variant of DDS we foresee as the most useful in linguistics; note that we have satisfied (33c′) in all the linguistically flavored DDS deductions presented so far—and we will continue to do so.

(33c′) E-universally generalize the conclusion with respect to *all* free variables of the conclusion not bound by a quantifier dominating the target.[16]

Substituting (33c′) for (33c), we have arrived at the closed variant of DDS, i.e. at a deductive system which works *exclusively* on closed formulas. There are two design features of DDS which make this possible. First and foremost, DDS deploys inline rather than root rules in its deductions, but it is also crucial that e-universal generalization is implemented as a part of an application of any rule rather than an inline rule in and of itself.

You might wonder if we went through all this trouble of differentiating open, semi-closed, and closed deductive systems, and showing that DDS can be (very naturally) formulated as a closed system, only to achieve Quine's desideratum. We did not. We have another, more important reason for emphasizing that we can easily define a truly closed variant of DDS.

Our ultimate goal (which we will pursue in Chapter 12) is to have a deductive system which runs off of the syntactic structure of natural language. To that

[16] We may now omit the addendum "or free in the dynamic formula" from (33c), as the dynamic formula never contains any free variables.

end, a part of Chapter 9 will be devoted to the implementation of a particular format of predicate logic (a format which aligns well to the structures proposed by contemporary linguistics)—we will call it the *quantifierless* format,[17] and as indicated by the name, there will be no quantifiers in quantifierless formulas. The function of explicit quantifier symbols will be taken over by a closure rule for free variables. In effect, even if all variables will be formally free, the quantificational force over each variable will be uniquely determined. In a way, it will be then equally appropriate to say that no variable in a quantifierless formula is ever free, meaning that only sentences can be represented in the quantifierless format.

It should now be clear why we insisted on having a closed variant of DDS. If we want to run DDS off of the quantifierless formal language, which can only represent sentences, then our deductive system should be able to work exclusively with sentences, i.e. without ever producing an open formula in the course of a deduction.

In the interest of full disclosure, another modification of DDS is necessary if it is really to work on the quantifierless format. Above, we addressed the situation of free variables, which cannot (in a certain sense) occur in the quantifierless format and whose introduction must therefore be avoided by the deductive system. Now, a free variable is of course nothing but a variable without a quantifier to bind it. But the converse situation of a quantifier without a variable to bind can be encountered as well. This is known as *vacuous quantification*, and clearly, a deductive system designed to work on the quantifierless formal language should also avoid producing formulas containing vacuous quantifiers.

It is again relatively easy to adjust DDS not to produce any vacuous quantifiers, by simply adding to (33) one further clause instructing us to remove any vacuous quantifiers that arose as the result of the application of the rule. However, the real issue here is of course whether the resulting system is sound. It is, but we will ask you to trust us on this, as going into the details would take us into the world of inclusive logic (the logic allowing the empty domain) and its relation to restrictedness (a property which, as we will see in Chapter 9, powers our quantifierless format). Rather than straying too far from the main topics of our project, let us finally deliver on a promissory note from the Introduction. Let us show you that DDS is complete.

7.3 Completeness of DDS

We've come far enough in the development of our Dynamic Deductive System that we can begin addressing the issue of completeness—an issue that we promised

[17] Not every formula of predicate logic can be recast into our quantifierless format. This format is only available to restricted formulas, i.e. formulas which only deploy restricted quantification (we will define restrictedness in Chapter 9). But this limitation presents no obstacle—in Chapter 10, we will argue that restrictedness is a fundamental property of natural language logical forms.

to address way back in the Introduction. But before we get into the details, we should pause for a couple of pages and ask some questions about the nature of completeness as it applies to natural language. What exactly is completeness? Is there some problem if we don't have it? And is there anything that great about having it if we do have it? These questions are not at all trivial.[18]

Let's begin with the question of what completeness is. When we use the term COMPLETENESS, we are using it in the sense of "complete with respect to some semantic model"—this is sometimes called SEMANTIC COMPLETENESS. So, for example, we might use truth tables to model the propositional calculus, and we could talk about a deductive system as being complete with respect to that model (truth tables) just in case our deductive system could derive any entailment that could be shown to hold in our model.

To use another example, we have talked about using generalized quantifier theory, which is a semantic model. That model utilizes the power of set theory to show certain entailment relations. A complete deductive system would thus show that a deduction could be provided for every entailment that generalized quantifier theory provides.

To this way of thinking, nothing is really complete *simpliciter*, but it really only makes sense to talk about completeness with respect to a model (this is a point stressed in Kreisel 1952). And this raises a question about what counts as an acceptable model. As Sánchez Valencia (1991) observes, we need to proceed with caution here. For example, while second-order logic is incomplete with respect to standard models it has been shown to be complete with respect to *general models*, in the sense of Henkin (1950). We don't rely on Henkin's non-standard models for the fragment of natural language we are interested in (a fragment that is already vast, given that it incorporates the non-elementary quantifiers), but suppose we needed to rely on non-standard models at some point. Is there some compelling reason why general models *à la* Henkin are inappropriate for the interpretation of natural language inferences? It is hard to see what that reason would be.

Sometimes discussions of completeness get muddled, because when people think about completeness they think in terms of Gödel's (1931) incompleteness result. But it is important to keep in mind that Gödel's famous proof was aimed at a very specific project in the foundations of mathematics—a formalist project that had been put forward by Hilbert. The aim of that project was to try and place mathematics on purely formal, or if you prefer, syntactic foundations, so that everything would be a play of symbols. (Why pursue such a project? Well, perhaps you might have an epistemological concern about abstract objects like models.) But if that is the project then you can't be in the business of helping yourself to models of the usual sort, since they are not syntactic objects (typically

[18] Thanks to Juhani Yli-Vakkur for helpful discussion of this section.

they involve set-theoretic objects). What then becomes of completeness when you can no longer define it in terms of models? Gödel's idea was that you could think of completeness as holding when every statement or its negation could be derived by the deductive system—this is sometimes called SYNTACTIC COMPLETENESS.[19] It is a perfectly reasonable approach, but really only necessary if you are pursuing a very specific anti-Platonist project.[20]

In this chapter we are going to show that our system is (semantically) complete, but turning to our second question above, would there be a problem if it wasn't complete? This is an interesting question. Results like completeness are prized for formal systems because we want to think of formal systems as having a kind of perfection to them. An infinite mind could use the system for an unlimited amount of time and never encounter an inference that couldn't be formally derived. It is easy to see the attraction of such a system for mathematics, but is such a property really necessary for Natural Language? To borrow an example from Chomsky (1980, p. 127), suppose that our system was only complete for sentences of a length less than one million words. Would that somehow be a problem? Humans would be endowed with a system that could compute every inference made in the history of their existence.

The question gets an even sharper edge when we consider the result from Church (1936), that although complete, first-order logic is not DECIDABLE; even though we know that a derivation exists for every valid inference, there is no effective procedure that can determine whether some arbitrary inference is valid or not. This then raises the question about why we should care about completeness, since even if we had infinite minds and an unlimited amount of time, we don't have a guarantee that we can determine whether a given inference is truly invalid, or if we simply haven't yet managed to find a formal derivation deducing this inference.[21,22]

[19] Syntactic completeness is a stronger property than semantic completeness, in the sense that a sound, syntactically complete system is semantically complete as well.

[20] For the record, we don't think such a project is attractive, in part of course because of Gödel's results. We do think the epistemological concerns about abstracta are real (we would add that expression types are abstracta as well, so formal approaches don't exactly escape this problem.) We are attracted to ideas expressed in C. Parsons (1983) which suggest that syntactic objects might mediate our access to abstracta. In that case, L^{**} could be thought of as mediating our access to the abstract objects (e.g. sets) of generalized quantifier theory and thus anchoring our epistemic access to them.

[21] However, we do have a guarantee that (if we had infinite minds and an unlimited amount of time) we cound contruct a deduction of an inference known to be valid. This is so because even if first-order logic is not decidable, it is SEMI-DECIDABLE; given a *valid* inference, there is an effective procedure that can (constructively) confirm its validity, or in other words, there is an effective procedure which, given unlimited time, will eventually generate every valid (but no invalid) inference.

[22] Notwithstanding the undecidability of first-order logic, one can ask reasonable questions about whether Natural Logic might be decidable. How could this be? Well, we know that significant fragments of first-order logic are decidable, so it is certainly conceivable that Natural Logic, being a fragment of first-order logic that is restricted (see subsequent chapters), could turn out to be decidable. We leave this as an open question. (Thanks to Larry Moss for discussion of this point.)

What makes these questions all the more poignant is that whatever the core system of inference infinite beings might have, if that system has some holes in it, those holes are small potatoes compared to the breakdowns in reasoning that actual humans exhibit on a daily basis. Whether from memory lapses or concentration failures or distractions or just flat out being bad at logic (students do fail logic exams, after all), one wonders why people interested in how humans reason should care one bit about formal completeness. In section 13.1 we are going to briefly take up the issue of human inferential capacity, and not to give too much away, we can say that we will lean on Chomsky's distinction between competence and performance. So the question can be reframed in this way: Is there some reason that the human competence system for logical reasoning should be complete, given that the performance of humans gives exactly zero evidence that it should be?

This leads us almost directly to our final question from above: Would there be anything good about formal completeness for our system if we could show it? Would a proof of completeness not make such a system look *less* plausible as an account of human logical competence? We can thus roll our last two questions into one: If we are interested in the science of human logical competence, why bother with completeness?

Let us set aside the issue of human logical failures for the time being. Science is all about considering the natural world under certain idealizations, from frictionless planes to ideal gasses. Our theories incorporate concepts that could never be realized in the real world, but which nevertheless drive our deepest theorizing about the world. So, the question is, in an idealized model of human inferential capacities, would completeness be an important desideratum? We think there is cause to think so.

Recent work in the philosophy of science has developed the idea that scientific explanation involves much more than covering the data; it also involves the appeal to certain prized theoretical properties. These properties might range from enhancing our ability to unify the sciences, to abstract conceptions of simplicity and aesthetic elegance. This view has been put forward by philosophers like Harman (1965), Boyd (1981, 1984), Harré (1986, 1988), McMullin (1992), Psillos (1999), Lipton (2004), and Williamson (2017), who sometimes call the approach Inference to the Best Explanation and sometimes Abduction. Williamson (2017) extended its application to the philosophy of logic and argued that we should not give up classical logic in the face of the paradoxes because classical logic has a number of desirable theoretical features. Williamson insisted that "[w]e do not fully understand why this methodology works so well," but "[t]he abductive methodology is the best science provides, and we should use it" (2017, p. 15). McMullin (1992), for what it is worth, called abduction "the inference that makes science."

What does this have to do with our project? The idea is that a theory of human inferential capacity will hopefully provide us a way of thinking about the phenomenon under certain idealizations, and there are likewise certain theoretical properties independent of data coverage that we should seek out. Whether we want to call these properties simplicity or elegance or conceptual profundity doesn't matter. Our point is that if properties like completeness are to be prized for formal logical systems, there is no reason they should not be likewise prized in our study of natural systems. Or to put it another way, if it is a desideratum for logic, it is no less a desideratum for Natural Logic.

Now we will be honest here. If it could not be shown that Natural Logic was complete, we might very well shrug our shoulders and, following some of the considerations raised above, say "no big deal." On the other hand, the fact that Natural Logic turns out to be complete strikes us as remarkable—as though a pristine theoretical property, prized elsewhere but not expected here, suddenly appeared to us. It is not a proof of the correctness of our empirical theory, but it is nevertheless a sign that we are on the right track.[23]

Finally turning to the business of this section, our strategy will be as follows. The classical deductive systems for first-order predicate calculus are both sound and (semantically) complete. Proving that any deduction of DDS can be simulated in some classical system therefore shows that DDS is sound, and conversely, simulating some classical system in DDS exhibits that DDS is semantically complete for predicate calculus.[24] Jointly, the two simulations of course prove that Dynamic Deductive System has precisely the same deductive power as classical deductive systems.

Regarding soundness, the construction of DDS, which we started in Chapter 6 and completed in the preceding sections of this chapter (and accompanied with examples from natural language), should have given you a preliminary sense of why the system is sound. While the formal proof of soundness is not really hard, it is still rather intricate. The complexity can be partially avoided by choosing an appropriate classical system in which to simulate DDS; we have chosen a Hilbert-style deductive system conservatively extended to Boolean connectives and supplemented with several admissible rules like Existential Generalization to do this. The proof of soundness needs to deal with many little details in the definition of p-scope and other notions relevant to DDS, so we are postponing it to section A.2 of the Appendix. In this section, we deal with the other side of the coin: completeness.

[23] Of course we are aware that pushback is possible here. Why shouldn't such a result show us that we are on the *wrong* track? But this is a species of a more general question about the scientific enterprise. Why shouldn't simplicity and elegance in *all* of our theories be a sign that they are on the wrong track? Shouldn't real science be "hacky" and inelegant? There is a time and a place to address this question, but for now we only wish to observe that if we are to treat Natural Logic like any other science, these results would be prized and considered confirmatory under abductive theories of explanation.

[24] Of course, just as the classical deductive systems, DDS is complete for predicate *calculus*. For a particular first-order *theory*, the chips will fall where they may—what is important is that as DDS is equi-derivable with the classical systems, DDS is complete for a given theory exactly when the classical systems are.

When done from scratch, the proof of completeness is usually much harder than the proof of soundness. However, in our case the situation will be reversed. As mentioned above, what we plan to do here is show that DDS has at least as much deductive power as some classical system, by simulating the latter in the former. Here, we chose to simulate a simple and well-known classical system: the Hilbert-style system with Modus Ponens and Universal Generalization as the only rules of inference (and the standard axiom schemes that go along with the rule set of the system). Our task is to simulate the application of its rules and deduce its axioms.

We will turn to this simulation in subsection 7.3.2. But before we can do that, we need to introduce the very final detail about DDS. In the first subsection, we will show how to introduce axioms and start a deduction from scratch, i.e. in the absence of any extra-logical hypotheses.

7.3.1 Polarity-sensitive axiom introduction

So far, we have been presenting fragments of deductions which begin with some extra-logical hypothesis. (When given multiple hypotheses, we have imagined that we receive them conjoined into a single statement.) In this subsection, we thus need to answer two questions. First, what are the details of how we introduce a hypothesis into the derivation? In principle, this should be achievable at any point in the deduction, not just at the very beginning. In fact, the same question concerns the introduction of axioms, and will (as is the usual practice) receive the same answer, as there is no formal difference between the two. Second, how do we start a deduction in the absence of any hypotheses or proper axioms? (You might have noticed that we have not yet introduced a single logical axiom!) In other words, how do we start deducing a theorem of propositional or predicate calculus?

Starting with the introduction of hypotheses and axioms (whether logical or proper), we clearly need an additional rule to achieve that—think of it as an interface rule, if you will. Now, how do we formulate axiom introduction, or IMPORT, as we shall call it? We could define a simple variant which conjoins the axiom to the root of the dynamic formula. That would suffice as from there, we can clearly Copy the axiom to wherever we need it—a root conjunct positively p-scopes over the entire dynamic formula. But let's be more faithful to the project and define Import so it is a proper inline rule. Given what we said above, it will clearly share its basic properties with Copy.

In particular, the inline variants of axiom introduction are sensitive to the polarity of the target, and this sensitivity is expressible using the dual connectives notation. When targeting a positive polarity constituent, Import introduces an axiom (or a hypothesis); when targeting a negative polarity constituent, the negation of an axiom is introduced. We develop three variants of Import. The *center* variant replaces the target by (the negation of) an axiom. (As the form of the target is irrelevant in this case, we omit the subscript on $\vdash$ below.) The *right* and the *left* variant join (the negation of) an axiom to the target. Specifically, the axiom is conjoined to a positive polarity target and its negation is disjoined to a negative polarity target. In other words, the right and the left variant introduce an e-conjunction.

Import

$\vdash^{\tau}_{\mathcal{B}} \mathcal{B} \wedge^{\tau} \neg^{\tau} \mathcal{A}$ (right)

$\vdash^{\tau} \neg^{\tau} \mathcal{A}$ (center)

$\vdash^{\tau}_{\mathcal{B}} \neg^{\tau} \mathcal{A} \wedge^{\tau} \mathcal{B}$, where $\mathcal{A}$ is an axiom or a hypothesis (left)

As an example of introducing extra-logical hypotheses (introducing axioms works exactly the same), consider a deduction that yields the classical form Cesaro in the second figure. Unlike the examples presented so far, we start the deduction from a single premise (35c) and add other premises as they are needed.

(35) a. No dog is an alien. $\neg\exists z\colon D(z) \wedge A(z)$
b. Every visitor is an alien. $\forall y\colon \neg V(y) \vee A(y)$
c. (There is a visitor.) $\exists x\colon V(x)$
d. ∴ Some visitor is not a dog. $\exists x\colon V(x) \wedge \neg D(x)$

(36) 1. $\exists x\colon V(x)$
IMPORT (35b)

2. $\exists x\colon V(x) \wedge \forall y\colon \neg V(y) \vee A(y)$
$y \mapsto x$
COPY

3. $\exists x\colon V(x) \wedge (\neg V(x) \vee A(x))$
PRUNE

4. $\exists x\colon V(x) \wedge A(x)$
IMPORT (35a)

5. $\exists x\colon V(x) \wedge (A(x) \wedge \neg\exists z\colon D(z) \wedge A(z))$
$z \mapsto x$
COPY

6. $\exists x\colon V(x) \wedge (A(x) \wedge \neg(D(x) \wedge A(x)))$
PRUNE

7. $\exists x\colon V(x) \wedge (A(x) \wedge \neg D(x))$
DELETE

8. $\exists x\colon V(x) \wedge \neg D(x)$

We now turn to the second issue announced above. In DDS, we are faced with an interesting, if somewhat technical problem. As inline rules *transform* the dynamic formula, the initial state of a derivation cannot be empty (as the list of proof lines is empty at the start of a Hilbert-style deduction), but must be some formula. Above, we started the deduction with a hypothesis, and starting with an axiom would have been fine as well. But what do we do in the case of the pure propositional or predicate calculus, i.e. a theory without proper axioms? The "problem" is that like some classical natural deduction systems, DDS does not require any logical axioms.[25]

The tidiest solution might be to adopt a "dummy" logical axiom. To avoid redundancy, we propose adopting the simplest axiom possible: a formula consisting of a single 0-place connective ⊤ (true). Armed with this axiom, we can work in propositional and the predicate calculus by starting every deduction by ⊤, as exemplified by the deduction of Double Negation Introduction (stated as an implication to be fed to MP) below.[26]

(37)
1. $\underbrace{\top}_{\substack{+\\ \text{ADD}}}$
2. $\overline{\underbrace{\top}_{\substack{+\\ \text{COPY}}}} \vee \overline{\neg\neg\underset{-}{\mathcal{A}}}$
3. $\overline{\neg\mathcal{A} \vee \neg\neg\mathcal{A}} \quad \sim \quad \mathcal{A} \Rightarrow \neg\neg\mathcal{A}$

7.3.2 Simulating a Hilbert-style deductive system

It is clear that if we can simulate some complete deductive system, we are home free as far as proving completeness is concerned. Of course, we choose to simulate the Hilbert-style deductive system (without any extensions), because that is the simplest deductive system on the market. A deduction in that system is nothing but a sequence of formulas (lines). Every formula is either an axiom (or a hypothesis), or it is justified by some rule of inference. In the latter case, the premises of that rule must be found among the previous lines of the proof. To simulate a Hilbert-style system we therefore need to:

[25] The initial state problem does not arise for classical natural deduction systems even if they adopt no logical axioms. Because they adopt Deduction Theorem (DT) as one of the rules, they can derive theorems of propositional and predicate calculus by starting with an arbitrary assumption for DT. While DT holds in DDS, it is trivial to the point of being useless. In fact, the ease of working with DDS without employing DT is one of the virtues of the system, so we prefer to adopt some other solution to the initial state problem.

[26] The initial state problem could have been resolved in at least one other way. Instead of assuming that ⊤ is an axiom and requiring a deduction to start from an axiom or a hypothesis, we could simply stipulate that ⊤ is the initial state of any deduction. In this case, ⊤ does not need to be an axiom. It is a theorem of the system, specifically, the conclusion of the shortest possible deduction.

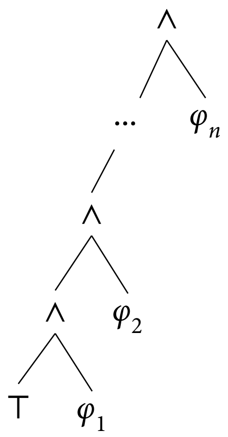
We retain the system presented in the main text for two reasons. First, we can then directly start a deduction by a hypothesis. Second, having ⊤ as an axiom affords us to easily "build" structure by Import without introducing any superfluous material. This feature of the system also plays well with a (minor) assumption from Chapter 12 that ⊤ corresponds to a syntactic node without semantic content.

Regardless of the status of ⊤ (an axiom or the initial state), it can be viewed as a mere extension of the original formal language, in the sense that we can require that it does not occur in the conclusion of a derivation.

- encode the list of lines of the proof in our dynamic formula,
- show that, given this encoding, the previous lines of proof are available as premises of further lines, i.e. that the previous lines positively p-scope over the location where they will be used,
- achieve the effect of the inference rules of the Hilbert system in DDS, and perhaps most importantly,
- show that we can deduce the axioms of the Hilbert system in DDS.

The "growing" list of proof lines of a Hilbert-style system can be encoded as a dynamic formula which extends at every application of a classical rule. In particular, each conclusion is conjoined to the root. Formally, for every n, the list of formulas $\varphi_1, \ldots, \varphi_n$ corresponds to the dynamic formula Φ_n such that $\Phi_0 \equiv \top$ and $\Phi_i \equiv \Phi_{i-1} \wedge \varphi_i$ for $i \in \{1, \ldots, n\}$.

(38) $\Phi_n \equiv ((((\top \wedge \varphi_1) \wedge \varphi_2) \wedge \ldots) \wedge \varphi_n)$



Clearly, the constituents of the dynamic formula corresponding to the proof lines (φ_i) positively (ancestor) p-scope over the root of the dynamic formula (Φ_n). Consequently, they can serve as premises of the inline rules targeting the root of the formula (and the final line can serve as the conclusion of the proof, as indicated in subsection 6.5.3). Furthermore, once we start simulating the addition of a new line (φ_{n+1}) of the classical proof, the DDS deduction will add a "new" conjunct to the root. This conjunct will be eventually developed into φ_{n+1}, but crucially, the previous lines positively (relative) p-scope into this new conjunct and are thus available as premises to any inline rules targeting some node within it. (We will show how this works in a minute.)

An "unadorned" Hilbert-style system employs only Modus Ponens and Universal Generalization (the propositional deductive system of course employs only the former). The steps of the inline derivation simulating their application are presented in (39) and (40), respectively. MP is applied to lines $\varphi_i \equiv \neg\mathcal{A} \vee C$ and $\varphi_j \equiv \mathcal{A}$, and derives $\varphi_{n+1} \equiv C$; UG over x is applied to φ_i (of any form), and derives $\forall x\, \varphi_i$. In steps (39.1) and (40.1), the inline rules target the root, which is in the ancestor p-scope of φ_i; in (39.2), the target of Prune ($\mathcal{A}$) occurs within the new root conjunct, but that is fine, as this location is within the relative p-scope of φ_j.

(39) 1. $\underbrace{\overbrace{\ldots \underset{+}{\varphi_i} \ldots}^{\Phi_n}}_{\text{COPY}}$

2. $(\overbrace{\ldots \underset{+}{\varphi_j} \ldots}^{\Phi_n}) \wedge \underbrace{(\neg \underset{-}{\mathcal{A}} \vee \mathcal{C})}_{\text{PRUNE}}$

3. $\Phi_n \wedge \mathcal{C}$

(40) 1. $\underbrace{\overbrace{\ldots \underset{+}{\varphi_i} \ldots}^{\Phi_n}}_{Qx\ \text{COPY}}$

2. $\Phi_n \wedge \forall x\, \varphi_i$

To simulate an introduction of a hypothesis or a *proper* (i.e. externally given) axiom (below, α), we simply use Import targeting the root.

(41) 1. $\underbrace{\underset{+}{\Phi_n}}_{\text{IMPORT}}$

2. $\Phi_n \wedge \alpha$

What about the logical axioms of the Hibert system? Those are not given, so they cannot be imported. How do we then get those axioms within DDS, where we only have rules like Prune and Copy, and a single (dummy) axiom, $\top$? Well, we need to *deduce* those axioms within DDS. In effect, the Hilbert axioms will become theorems of DDS, derived via our sub-logical rules, guided by p-scope. (One logician's axiom is another logician's theorem!)

Specifically, we will deduce the following, commonly adopted axiom system for propositional and first-order predicate logic (Mendelson 1997, pp. 27, 62). Schemes (A1)–(A3) axiomatize propositional logic; supplementing them by (A4) and (A5), we arrive at the axiomatization of predicate logic.[27]

(A1) $\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{A})$,
(A2) $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C}))$
(A3) $(\neg\mathcal{A} \Rightarrow \neg\mathcal{B}) \Rightarrow ((\neg\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow \mathcal{A})$
(A4) $\forall x\, \mathcal{A} \Rightarrow \mathcal{A}(t/x)$, where t is free for x in $\mathcal{A}$
(A5) $\forall x\, (\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \forall x\, \mathcal{B})$, where x is not free in $\mathcal{A}$

[27] Mendelson (1997, p. 142) also offers an axiom system which works exclusively on closed formulas (the system works without UG). This alternative system is just as easily derivable in DDS—this becomes important if we decide to work with the closed formula variant of DDS.

We have written down the axiom schemes in the familiar form deploying connective $\Rightarrow$—and for this reason, the deductions below use the conditional form of Add ($\vdash^{+}_{\mathcal{B}} \mathcal{A} \Rightarrow \mathcal{B}$), introduced in section 6.1—but remember that in our system, $\mathcal{A} \Rightarrow \mathcal{B}$ is merely an abbreviation for $\neg\mathcal{A} \vee \mathcal{B}$.

(42)
1. $\top$ ADD
2. $\mathcal{A} \Rightarrow \top$ ADD
3. $\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \top)$ COPY
4. $\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{A})$

(43)
1. $\top$ ADD
2. $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow \top$ ADD
3. $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow \top)$ COPY
4. $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{B}))$ COPY
5. $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow (\mathcal{B} \wedge (\neg\mathcal{A} \vee (\neg\mathcal{B} \vee \mathcal{C})))))$ PRUNE
6. $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow (\mathcal{B} \wedge (\neg\mathcal{B} \vee \mathcal{C}))))$ PRUNE
7. $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow (\mathcal{B} \wedge \mathcal{C})))$ DELETE
8. $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C}))$

(44)
1. $\top$ ADD
2. $(\neg\mathcal{A} \Rightarrow \neg\mathcal{B}) \Rightarrow \top$ ADD
3. $(\neg\mathcal{A} \Rightarrow \neg\mathcal{B}) \Rightarrow ((\neg\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow \top)$ COPY
4. $(\neg\mathcal{A} \Rightarrow \neg\mathcal{B}) \Rightarrow ((\neg\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\neg\neg\mathcal{A} \vee \mathcal{B}))$ COPY
5. $(\neg\mathcal{A} \Rightarrow \neg\mathcal{B}) \Rightarrow ((\neg\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\neg\neg\mathcal{A} \vee (\mathcal{B} \wedge (\neg\neg\mathcal{A} \vee \neg\mathcal{B}))))$ PRUNE
6. $(\neg\mathcal{A} \Rightarrow \neg\mathcal{B}) \Rightarrow ((\neg\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\neg\neg\mathcal{A} \vee (\mathcal{B} \wedge \neg\neg\mathcal{A})))$ PRUNE

7. $(\neg\mathcal{A} \Rightarrow \neg\mathcal{B}) \Rightarrow ((\neg\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow \underset{\text{COPY}}{\neg\neg\mathcal{A}}$

8. $(\neg\mathcal{A} \Rightarrow \neg\mathcal{B}) \Rightarrow ((\neg\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow \mathcal{A}$

(45) 1. $\underset{\text{ADD}}{\top}$

2. $\neg\forall x\, \underset{x \mapsto t}{\mathcal{A}} \vee \underset{\text{COPY}}{\top}$

3. $\neg\forall x\, \mathcal{A} \vee \mathcal{A}(t/x)$

(46) 1. $\underset{\text{ADD}}{\top}$

2. $\neg\forall x\, (\neg\mathcal{A} \vee \mathcal{B}) \vee \underset{\text{ADD}}{\top}$

3. $\neg\forall x\, (\neg\mathcal{A} \vee \mathcal{B}) \vee (\neg\mathcal{A} \vee \underset{\text{COPY}}{\top})$

4. $\neg\forall x\, (\neg\mathcal{A} \vee \mathcal{B}) \vee (\neg\mathcal{A} \vee \underset{\text{PRUNE}}{(\forall x\, (\neg\mathcal{A} \vee \mathcal{B}))})$

5. $\neg\forall x\, (\neg\mathcal{A} \vee \mathcal{B}) \vee (\neg\mathcal{A} \vee \forall x\, \mathcal{B})$

We have seen that we can deduce the classical axioms within DDS, but we also have to show that these deductions can be integrated into the simulation. This is really a question of how to "create space" for these deductions. We can do that by Importing our dummy axiom $\top$ next to the root.[28] The deduction of a classical axiom can then proceed as shown in (42)–(46), because the new position is of positive polarity and (this is relevant only for predicate logic) because the new position is not dominated by any quantifier nodes. As an example, let's simulate the introduction of $A(x) \Rightarrow (B(x) \Rightarrow A(x))$.

(47) 1. $\underset{\text{IMPORT}}{\Phi_n}$

2. $\Phi_n \wedge \underset{\text{ADD}}{\top}$

3. $\Phi_n \wedge (A(x) \Rightarrow \underset{\text{ADD}}{\top})$

4. $\Phi_n \wedge (A(x) \Rightarrow (B(x) \Rightarrow \underset{\text{COPY}}{\top}))$

5. $\Phi_n \wedge (A(x) \Rightarrow (B(x) \Rightarrow A(x)))$

[28] We could also create the new position by Copying something (anything) next to the root, or by using Import to conjoin some proper axiom or hypothesis to the root. The idea of Importing $\top$ seems to be the most elegant, as it avoids the unnecessary temporary introduction of some material—the subsequent steps of the inline deduction of the classical axiom eradicate whatever is present in the new position at first anyway.

Table 7.1 The basic rules Dynamic Deductive System

	introduction	elimination
e-conjunction	Copy	Delete
e-disjunction	Add	Prune
	+ Import	

This concludes our two-chapter-long development of Dynamic Deductive System, so let us take the time to take a look at the final product. As promised, DDS is a deductive system highly sensitive to *polarity*. The central ingredient of DDS is *p-scope*—a polarity-sensitive relation between locations in a logical tree which tells us about the reach of a premise within the tree. But we have followed the Medieval idea of *dicta de omni et nullo* in more than just paying attention to polarity. Like the *dicta* (and in fact even more so), the inference rules of DDS are *inline* rules—they can take their premises from any location within the formula and (paying attention to p-scope) target any location as well. For one, this allows us to see a deduction as an unfolding of the dynamic formula, but more importantly, a deductive system deploying inline rules typically yields much shorter (and therefore arguably more cognitively plausible) deductions than a traditional deductive system. Thanks to p-scope and inline rules, there is no need to "peel" the layers off a constituent and then "rebuild" the formula—in DDS, one can get straight to work on any location.

The final list of the DDS inline rules can be found in Table 7.1 (and their formalizations are repeated below). However, it might be worth noting that the rules comprising this list form but one of the many possible instantiations of DDS. We have decided to present the list in Table 7.1 as the "official" variant of DDS not because other rule systems would not form a complete system, for example, but because these particular rules seem to offer optimal balance between non-redundancy and usability, while furthermore being pleasantly symmetric.[29]

<u>Copy</u>

$\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \mathcal{B} \wedge^{\tau} \neg^{\sigma\tau}\mathcal{A}$ (right)

$\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \neg^{\sigma\tau}\mathcal{A} \wedge^{\tau} \mathcal{B}$ (left)

<u>Add</u>

$\vdash^{\tau}_{\mathcal{B}} \mathcal{B} \vee^{\tau} \mathcal{C}$ (right)

$\vdash^{\tau}_{\mathcal{B}} \mathcal{C} \vee^{\tau} \mathcal{B}$ (left)

<u>Delete</u>

$\vdash^{\tau}_{\mathcal{A}\wedge^{\tau}\mathcal{B}} \mathcal{A}$ (right)

$\vdash^{\tau}_{\mathcal{A}\wedge^{\tau}\mathcal{B}} \mathcal{B}$ (left)

<u>Prune</u>

$\mathcal{A}^{-\tau} \vdash^{\tau}_{\mathcal{A}}$ eliminate the local e-disjunct of the target

[29] For example, we can drop one side of Copy, Add, and Delete and still end up with a complete system (if we properly coordinate which side, left or right, is dropped for each rule). This would arguably result in a simpler system, but note that as a price, some deductions would get longer.

We have concluded this chapter by simulating a Hilbert-style system and thus proving that DDS is complete. But as hinted at the start of the section, completeness is perhaps the least interesting feature of DDS, at least from the perspective of a linguist or a cognitive scientist. In the next chapter, we will turn to a property which we believe to be more important if we are interested in the integration of the deductive system within linguistics, or in having a proposal that prepares linguistics for unification with other sciences. We will call this property *representational minimalism*, and we will explain what we mean by that in the next chapter, where we compare DDS to other natural deductive systems.

8
Representational Minimalism and Comparison with Other Natural Deductive Systems

In the previous two chapters we introduced our Dynamic Deductive System and showed how it could take the Medieval logical project and push it further, by reducing the rules of logic to a handful of sub-logical syntactic operations guided by p-scope. We also saw that we could prove the completeness of DDS by simulating a Hilbert-style deductive system, and we did this by showing how the axioms and operations of a Hilbert system can be derived within ours.

That is a pretty interesting result, we think, but there are more interesting results to be had. One of the most interesting features of DDS is that it is *stateless*—it doesn't keep a record of theorems proved as does the Hilbert system we looked at earlier. Perhaps even more importantly, it is *representationally minimal*—the only representation that needs to be constructed is the formula that is being operated upon.

Despite being representationally minimal, DDS will have the deductive power of other natural deduction systems, such as Gentzen's. More than that, we will see that it even has certain advantages compared to those other systems—precisely because it is representationally minimal.

Here is the plan for this chapter. In section 8.1, we will introduce the concepts of statelessness and representational minimality. In section 8.2, we will compare DDS to other natural deduction systems, in particular the Gentzen natural deduction system, with the aim of illustrating how DDS can achieve the same level of deductive power as these systems despite the fact that it has no internal state like a proof tree. Finally, in section 8.3, we compare DDS to a deep inference system known as Calculus of Structures (CoS). As we know, DDS is characterized by two features: inline rules and sensitivity to polarity. CoS has the former property, but not the latter, so the comparison will bring us back to the topic of polarity, and make its contribution vivid.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović,
Oxford University Press. © Peter Ludlow and Sašo Živanović 2022. DOI: 10.1093/oso/9780199591534.003.0008

8.1 Statelessness and Representational Minimalism

One of the hallmark properties of the Dynamic Deductive System is that it is *stateless*. What do we mean by this?

We borrow the term from computer science, which categorizes communication protocols as either stateful or stateless. Following the general usage of the term, we define statefulness and statelessness as follows.

In a STATEFUL communication, at least one of the parties keeps and updates the STATE of the conversation, which comprises any information relevant to the purpose of the communication. No state is recorded in a STATELESS communication.

A nice example of a stateless communication protocol is Hypertext Transfer Protocol (HTTP). Your computer uses this protocol when you visit a simple webpage which contains some text and maybe images but does not interact with the user. You request a page at some address, the server sends you the contents, and then forgets about you. That is how web browsing works in theory, but in practice, we wonder if there are any such static webpages left at all, as these days most webpages not only interact with the user in some way, but keep a record of those interactions. Whenever you log in somewhere, you can be sure that your session is managed using some stateful protocol (most often built on top of HTTP). For example, the server might assign you some random identifier and store it in *your* computer—that is what those notorious cookies are. Of course, the server will typically store further information from the session, like the contents of your shopping basket and your browsing habits, but the bottom line is that some information about the interaction is stored, so the communication between the parties is stateful.

Moving back to deductive systems, it is easy to see that a Hilbert system is stateful. Using the computer metaphor (and keep in mind that this *is* a metaphor), we can see an application of a rule as a "request–response" pair: we ask the server to apply a certain rule using certain lines as the premises, and expect it to reply with the conclusion (or an error, if the rule is inapplicable to the given premises). Say we are performing the deduction in (1) below and are about to apply Modus Ponens in (2.6). When we ask the deduction server to apply MP to lines 5 and 2, the server can only compute the conclusion if it has stored the history of the deduction (lines 1–5), otherwise it has no idea what line 5 and line 2 are. In effect, each deduction is a "session" and the communication protocol is stateful.

(1) a. In sunny weather, we always visit the seaside. $S \Rightarrow V$
b. Visiting the seaside, we always stroll the promenade. $V \Rightarrow P$
c. Strolling the promenade, we always buy ice-cream. $P \Rightarrow I$
d. The weather was sunny. S
e. ∴ We bought ice-cream. $\therefore I$

(2)

1.	$S \Rightarrow V$	(hypothesis)
2.	$V \Rightarrow P$	(hypothesis)
3.	$P \Rightarrow I$	(hypothesis)
4.	S	(hypothesis)
5.	V	(MP 4, 1)
6.	P	(MP 5, 2)
7.	I	(MP 6, 3)

In contrast, DDS is clearly stateless. At (3.5), we ask the server, can you apply Inline Modus Ponens[1] targeting the second conjunct *of this formula that I'm sending* with the second conjunct of the first conjunct of the first conjunct of the formula as the premise,[2] and the server replies, sure, I can do that, here is the result: (3.6).

(3)

1. $\underbrace{S \underset{+}{\Rightarrow} V}_{\text{IMPORT}}$
2. $\underbrace{(S \Rightarrow V) \underset{+}{\land} (V \Rightarrow P)}_{\text{IMPORT}}$
3. $\underbrace{((S \Rightarrow V) \land (V \underset{+}{\Rightarrow} P)) \land (P \Rightarrow I)}_{\text{IMPORT}}$
4. $(((\underline{S \underset{+}{\Rightarrow} V}) \land (V \Rightarrow P)) \land (P \Rightarrow I)) \land \underset{\text{IMP}}{\underset{+}{S}}$
5. $(((S \Rightarrow V) \land (\underline{V \underset{+}{\Rightarrow} P})) \land (P \Rightarrow I)) \land \underset{\text{IMP}}{\underset{+}{V}}$
6. $(((S \Rightarrow V) \land (V \Rightarrow P)) \land (\underline{P \underset{+}{\Rightarrow} I})) \land \underset{\text{IMP}}{\underset{+}{P}}$
7. $\underbrace{(((S \Rightarrow V) \land (V \underset{+}{\Rightarrow} P)) \land (P \Rightarrow I)) \land}_{\text{DELETE}} I$
8. I

The crucial point here is that the server can do the job by only looking at the formula we just sent (and following the rules, of course). When it is applying Inline Modus Ponens to (3.5), it does not know nor care how we arrived at it—we might have even started the deduction with it! All the information it needs to compute the conclusion is given right inside the formula itself, and there is no other way to get further information—for example, if we delete a part of the dynamic formula, it is gone forever. (Incidentally, our justification notation is only possible because DDS is stateless.)

[1] For clarity, we don't use Prune here. The choice of the rule does not affect our argument.
[2] Clearly, we can phrase this request in some more efficient way!

Let's borrow another term from computer science: DATA STRUCTURE. You are actually familiar with the meaning of the term even if you had never heard of it, because you surely know about file formats like pdf, odt, doc (for text documents), jpg, png, gif (for images), etc. File formats are complex, high level data structures, and in general, a data structure defines *how* the data is stored. Common (low level) data structures include lists, ordered lists, trees, graphs, hash tables, etc. For example, a formula of a formal language can be represented either as a string (sequence, list) of symbols or as a tree.[3]

There is an immediate and trivial consequence to statelessness: a stateless system like DDS needs no data structure dedicated to keeping the system state—as there is no system state! In contrast, consider the Hilbert deduction server, which is stateful because it needs to keep the history of the deduction. True, it only employs a very simple data structure—a list—but that is beside the point, which is that it *needs* a data structure. It simply cannot do without.[4]

Of course, every deductive system must be aware of one particular data structure, the formal language of the formulas it is constructing. This is a matter of conceptual necessity. The question is whether a system also deploys some other data structure, a data structure internal to the deductive system. We will call a deductive system which requires no such additional data structure REPRESENTATIONALLY MINIMAL. A Hilbert-style system is not representationally minimal; the additional data structure is the list of formulas comprising the deduction. A Gentzen-style system (we will turn to these presently) is not representationally minimal either; here, the deduction is represented in the form of a tree, called the proof tree. Our Dynamic Deductive System, however, *is* representationally minimal—all the information it needs to apply any rule at any step of the deduction is contained within the (dynamic) formula itself.

The simplicity of the data structure of Hilbert-style systems comes at a price: these systems yield long and complex proofs. Nobody ever seriously deduces anything in a plain Hilbert system. The utility of these systems lies in their

[3] While this is orthogonal to our point, note that the same data can be presented in various ways. For example, the same piece of music can be stored either in the mp3 or in the ogg format. Furthermore, the choice of the data structure typically depends on the task at hand. For example, your shopping list needs to be, well, a list. When you think of another thing you need to buy, you simply add it to the end of the list. And this is good enough for your purpose, but imagine that dictionaries were written like this. You clearly want the dictionary entries to appear alphabetically: a useful dictionary is organized as an ordered list. For formulas, the string representation is useful as the input format, but a computer program will surely parse it into a tree before using the formula in any way.

[4] Of course, one could implement a stateless Hilbert deduction protocol. We could simply send the history of the deduction to the server every time we wanted to apply a rule. But this is cheating, as it delegates the burden of keeping the history to the client, and anyway, the cheat does not obviate the need to keep the history or to deploy some data structure to store it. In fact, not only the client, who actually stores the history in this scenario, but also the server needs to know about the data structure, otherwise the references to lines would be meaningless to the server.

conceptual simplicity, which facilitates investigation of (some of) their meta-theoretical properties. So let us take a brief look at a well-known alternative, the Gentzen natural deduction system. We will illustrate how the more complex data structure it introduces helps it reach the goal of being closer to human reasoning—or at least the reasoning familiar from mathematical practice—and then show how DDS achieves the same, or even greater, level of naturalness despite being representationally minimal—in fact, because of it.[5]

8.2 Comparison with Gentzen Natural Deduction Systems

In this section, we look at the standard way to make deductions more "natural" than Hilbert-style deductions, and then show how DDS achieves the same (if not greater) level of naturalness even without the bells and whistles of traditional natural deduction systems. For concreteness, we focus on the most familiar natural deduction system: the Gentzen natural deduction system (Gentzen 1934).[6]

As we can see in (5), the outstanding feature of the system is that the proof has a form of a tree, the idea being that the leaves of the tree (the top-most nodes) are the hypotheses leading, via application of inference rules, to the conclusion at the root (the bottom-most node). The essential *data structure* of a Gentzen system is thus the PROOF TREE. Given that we like to think of formulas as trees as well, a Gentzen deduction is in fact a tree of trees!

(4) a. Some alien visited every inhabited planet.
$$\exists x\colon A(x) \wedge \forall y\colon I(y) \wedge P(y) \Rightarrow V(x,y)$$
b. ∴ If every planet is inhabited, then some alien visited every planet.
$$\forall z\,(P(z) \Rightarrow I(z)) \Rightarrow \exists x\colon A(x) \wedge \forall y\colon P(y) \Rightarrow V(x,y)$$

[5] Josh Dever (personal communication) asks whether it is entirely fair of us to claim statelessness, since, after all, all the initial premises of an argument are folded into a big conjunction. Even more, don't we carry the entire history of the deduction with us? We *could*, but we usually don't. It is important to keep in mind that Delete and Prune eliminate a part of the dynamic formula, leaving no trace of the original structure—and it's not as though we make a "backup" of the tree before applying Delete or Prune. Furthermore, the state of deduction can be more than just its history. What we find really impressive is that DDS, a stateless and representationally minimal system, can operate in the simple sequential fashion of a Hilbert-style system while keeping the "high level" state familiar from Gentzen-style systems (the next section should make clear what we mean by that).

[6] There are two families of Gentzen systems, the natural deduction and the sequent calculus. The latter introduces further data structure in the form of sequents, which replace (with respect to natural deduction) formulas as residents of the nodes of the proof tree. We will not talk about the sequent calculus here, as it is further removed from natural language deduction—and for good reason, as it was designed to facilitate investigation of meta-theoretical properties of deductive systems.

In fact, one of the three broad approaches to proving the interpolation theorems, which our proofs of the validity of the syntactic characterizations of directional entailingness (section A.3) and conservativity (section A.4) are based on, relies on the Cut Elimination theorem of Gentzen sequent calculi.

(5)

$$
{\scriptstyle\text{DT-1}}\dfrac{
{\scriptstyle\text{EI-2}}\dfrac{
{\scriptstyle\text{hyp-a}}\dfrac{}{(4a)}
\qquad
{\scriptstyle\text{EG}}\dfrac{
{\scriptstyle\text{CI}}\dfrac{
{\scriptstyle\text{CE}}\dfrac{{\scriptstyle\text{ass-2}}\dfrac{}{A(a)\wedge\forall y\colon I(y)\wedge P(y)\Rightarrow V(a,y)}}{A(a)}
\qquad
{\scriptstyle\text{UG}}\dfrac{
{\scriptstyle\text{DT-3}}\dfrac{
{\scriptstyle\text{MP}}\dfrac{
{\scriptstyle\text{CI}}\dfrac{
{\scriptstyle\text{MP}}\dfrac{{\scriptstyle\text{ass-3}}\dfrac{}{P(y)} \quad {\scriptstyle\text{UI}}\dfrac{{\scriptstyle\text{ass-1}}\dfrac{}{\forall z\colon P(z)\Rightarrow I(z)}}{P(y)\Rightarrow I(y)}}{I(y)}
\quad {\scriptstyle\text{ass-3}}\dfrac{}{P(y)}
}{I(y)\wedge P(y)}
\quad
{\scriptstyle\text{UI}}\dfrac{{\scriptstyle\text{CE}}\dfrac{{\scriptstyle\text{ass-2}}\dfrac{}{A(a)\wedge\forall y\colon I(y)\wedge P(y)\Rightarrow V(a,y)}}{\forall y\colon I(y)\wedge P(y)\Rightarrow V(a,y)}}{I(y)\wedge P(y)\Rightarrow V(a,y)}
}{V(a,y)}
}{P(y)\Rightarrow V(a,y)}
}{\forall y\colon P(y)\Rightarrow V(a,y)}
}{A(a)\wedge\forall y\colon P(y)\Rightarrow V(a,y)}
}{\exists x\colon A(x)\wedge\forall y\colon P(y)\Rightarrow V(x,y)}
}{\exists x\colon A(x)\wedge\forall y\colon P(y)\Rightarrow V(x,y)}
}{\forall z\,(P(z)\Rightarrow I(z))\Rightarrow\exists x\colon A(x)\wedge\forall y\colon P(y)\Rightarrow V(x,y)}
$$

As the tree form of a deduction is clearly a complication, one might ask: why is it there? To understand that, we need to remember what a SUBDERIVATION is. In a Gentzen system, subderivations are subtrees of the proof; more precisely, certain branches above certain rules. These rules license using an additional hypothesis in the subderivation(s) above them (we'll call them assumptions); on a subderivation branch, a leaf may be either a hypothesis (same as outside the subderivation) or the assumption of the subderivation rule. In (5), we find (highlighted) two Deduction Theorem (DT) subderivations and one Existential Instantiation (EI) subderivation.

Rule DT states that if we can find a subderivation concluding ψ, with φ as the assumption, then we may conclude $\varphi \Rightarrow \psi$.[7] For DT-3 in (5), the assumption $P(y)$ leads to sub-conclusion $V(a, y)$, so we can conclude $P(y) \Rightarrow V(a, y)$.

Rule EI states that, given an existential statement $\exists x\, \varphi(x)$ and a derivation from $\varphi(a)$ (where a is a constant, hitherto not used in the proof) to some conclusion ψ (in which a does not occur), we may conclude ψ.[8] In (5), the existential premise (4a) licenses the assumption ass-2 in the right branch of EI-2.

Proof tree (5) also illustrates that subderivations may be nested. The EI-subderivation EI-2 occurs within the DT-subderivation DT-1. This is why the assumption of DT-1, marked ass-1 in the proof tree, is available within the subderivation EI-2, which itself licenses ass-2. (There are actually three levels of embedding in (5), as DT-3 is further embedded in EI-2.)

The point of having the proof organized as a tree is to keep track of which subderivations are "open" at some point, so that we know which assumptions we may make. In other words, the state of a Gentzen system is essentially a position in the tree. In our deduction server metaphor, we would be asking the server about the applicability of a rule relative to a position.

[7] In predicate logic, the subderivation must respect certain requirements regarding universal generalization as well; see A.2 for details.

[8] Again, see A.2 for details regarding universal generalization.

Given the importance of subderivations for natural deduction, how come our DDS does not have all this proof tree machinery? How can we deduce precisely the same statements as Gentzen's natural deduction without proof trees (or some other implementation of subderivations)? The idea is to realize that all the information contained in the proof tree is in fact already present in the formulas of the formal language, *if* we know how to access it!

It turns out that natural deduction subderivations correspond to particular syntactic environments of a dynamic formula. DT-subderivations correspond to e-disjuncts, i.e. children of e-disjunctions. EI-subderivations correspond to children of e-existential quantifiers.

If this seems somewhat accidental, remember that e-disjunctions and e-existential quantifiers are precisely the nodes (barring negations) that can never belong to some ancestor p-scope. In our Medieval allegory for ancestor p-scope from subsection 6.5.3, e-disjunctions and e-existential quantifiers were precisely the checkpoint towns where Perceval could *not* stop to deliver plans on his way North. The problem was that his armor did not match the town's banner (the town's constituent polarity), although we should really not blame this on Perceval—he needs a particular kind of armor to pass through a checkpoint, so the problem with e-disjunctions and e-existential quantifiers is really in the towns themselves: their Angel/Demon type does not match their town's banner.

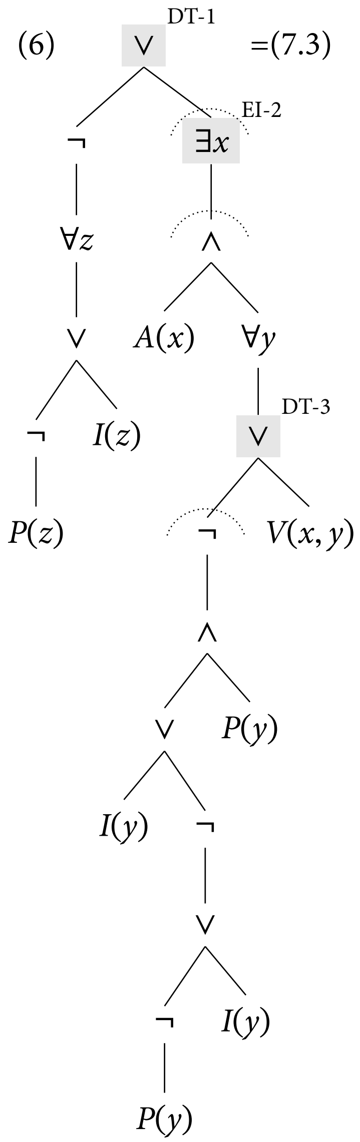
We could, in homage to the Medieval supposition theory, call these towns/nodes CONFUSED. Unifying several paradigms, we could therefore say that every subderivation in a natural deduction proof tree corresponds to some child of some confused town in some stage of the evolution of the dynamic formula. (Of course, a given child of a confused town corresponds to some subderivation only when something actually happened within that constituent, i.e. only when some of its descendants were the target of an inline rule in the course of the DDS deduction.)

Let's illustrate the correspondence by comparing the proof tree (5) and the DDS deduction (7). We will also be keeping an eye on tree (6), which vizualizes the dynamic formula at (7.3)—after applying all the introduction rules but before applying any elimination rules.

The proof tree (5) contains three (nested) subderivations: the branches above the highlighted rules.[9] The highlighted rules correspond to the highlighted nodes in (6), and the subderivations correspond to the subtrees under the arches below these nodes.

[9] In the case of EI, only the right branch is the subderivation.

(6)



(7)

1. $\underbrace{\exists x\colon A(x) \wedge \forall y\colon \neg(I(y) \wedge P(y)) \vee V(x,y)}_{\text{ADD}}$

2. $\neg\forall z\,(\neg P(z) \vee I(z)) \vee \ldots$
$\ldots\ \exists x\colon A(x) \wedge \forall y\colon \neg(\underbrace{I(y)}_{\text{COPY}} \wedge P(y)) \vee V(x,y)$ ($z \mapsto y$)

3. $\neg\forall z\,(\neg P(z) \vee I(z)) \vee \exists x\colon A(x) \wedge \forall y\colon$
$\neg((I(y) \vee \underbrace{\neg(\neg P(y) \vee I(y))}_{\text{PRUNE}}) \wedge P(y)) \vee V(x,y)$

4. $\neg\forall z\,(\neg P(z) \vee I(z)) \vee \exists x\colon A(x) \wedge \forall y\colon$
$\neg(\underbrace{(I(y) \vee \neg I(y))}_{\text{PRUNE}} \wedge P(y)) \vee V(x,y)$

5. $\neg\forall z\,(\neg P(z) \vee I(z)) \vee \exists x\colon A(x) \wedge \forall y\colon$
$\neg P(y) \vee V(x,y)$

In DDS, the real action, the elimination of 'inhabited', happens in steps (7.3) and (7.4), which both target a constituent under the innermost arch of (6), and this corresponds to the fact that the natural deduction needs the assumptions of all three subderivations to arrive at the conclusion above DT-3.

The starting point in the DDS deduction is the hypothesis (4a), which contains only two of the highlighted nodes of (6). The missing node is created by Adding the antecedent in (7.1), a step which not only creates the missing "subderivation constituent" but introduces the missing part of the final conclusion as well. The important point here is that while the correspondence between these is encoded by the rule in a Gentzen deduction and can be thus considered somewhat arbitrary, the two effects are inseparable in DDS, creating an intimate link between the structure of the proof and its conclusion.

Summing up, in DDS, subderivations are not really some part of the proof, but are rather syntactic contexts that we can restrict our attention to, or "zoom in on," if you will. A subderivation is not "opened" in DDS, it is rather "entered," simply by targeting a constituent (properly) dominated by a "confused" town. Whenever we apply some rules which target (a constituent within) the scope of an effectively existential quantifier, we have, without lifting a finger, opened an EI-subderivation. Whenever we apply some rules that target (a constituent within) some effective disjunct, we have, again without lifting a finger, opened a DT-subderivation.

All subderivations, arbitrarily nested, are available to be entered at any time in DDS, and there is a single piece of information we need in order to know whether a child of a given quantifier (or binary connective) can correspond to a subderivation for EI (or DT). We need to know what polarity it has, and this information is encompassed in our notion of effectiveness, in particular in the notions of effective disjunction and effectively existential quantifier. However, at the end of the day, we don't care in the least whether we target a constituent within an e-disjunction or an e-existential quantifier, i.e. whether we work within an EI- or a DT-subderivation. For DDS, all subderivations are born equal.

Identifying the DDS correspondents of natural deduction subderivations is only the first part, and in fact the less important part, of the story. The *raison d'être* of subderivations are the assumptions they license. So how do we keep track of active assumptions in DDS if we don't explicitly open subderivations and mark constituents as additional hypotheses? The answer is p-scope!

Remember that premise scope (p-scope) is a relation between constituents of the formula that tells us when we may use one of them as the premise of a rule targeting the other. The point here is that the target can be in the p-scope of constituents within a subderivation constituent, but it can also be in the p-scope of constituents *outside* the subderivation constituent. The latter constituents correspond to the assumptions of natural deduction systems.[10]

In the following subsections, we illustrate how this applies to DT- and EI-subderivations and also discuss some details pertaining to each of these environments.

8.2.1 Deduction Theorem

Imagine we have just zoomed into ψ of $\varphi \Rightarrow \psi$, which we actually see as $\neg\varphi \vee \psi$. In such a case, φ (positively) p-scopes over ψ and its descendants. This is an instance of relative p-scope, which is completely independent of the context, including the polarity of $\varphi \Rightarrow \psi$ within the dynamic formula, so the p-scope obtains without

[10] As we shall see, this statement is not exactly correct, but it should suffice as a general guideline. The point is that in the case of EI, the assumption of a natural deduction system actually corresponds to the subderivation constituent, not something outside it. However, if we were to use a semi-natural deductive system which would feature DT- but not EI-subderivations, then the effect of EI would be performed via DT, and our statement in the main text would turn out to be fully precise.

exception. And this is all there is to the issue. Once we zoom into the consequent, the antecedent is automatically available to us as a premise—there is no need to explicitly mark it as an assumption (of DT).

This is a good point for us to deliver on a promise regarding the Deduction Theorem. We hinted that DT holds in DDS, but also claimed that it is too trivial to ever use in an actual deduction. And indeed, DT clearly holds. First, if we can perform some deduction, then we can also perform it inside an arbitrary positive polarity context. Second, if we remove one of the hypotheses of the deduction but set up the context so that the ex-hypothesis positively p-scopes over it, we can amend the deduction by replacing all Imports of the ex-hypothesis by Copy. We provide a more formal statement below.[11]

(8) Deduction Theorem for DDS:
Let φ and ψ be some sentences, and Γ a set of sentences. If ψ can be deduced from $\Gamma \cup \{\varphi\}$, then $\neg\varphi \vee \psi$ can be deduced from Γ.

To prove this theorem, we modify the original deduction to arrive at the new one. The initial step is the same. As the new second step, we Add $\neg\varphi$ to (the left of) the root. This prefixes $\neg\varphi\vee$ to all but the first stage of the dynamic formula. We then replace all Imports of φ by Copying φ from the antecedent of the root "implication." This works because the antecedent always p-scopes over the consequent and its descendants. The other rules can be applied because the polarity of the constituents of the original deduction is the same when they occur within the root consequent of the new deduction.

The following example illustrates both the validity and the otiose nature of Deduction Theorem for DDS, where by "otiose" we mean that DT does not have any utility—it does not make the deduction any shorter. (Remember that for a Hilbert system, allowing DT makes the deductions *much* shorter.) To relate the example to the Deduction Theorem, set $\Gamma := \{(9a)\}$, $\varphi = (9b)$ and $\psi = (9c)$. Then, (10) exemplifies the original deduction of ψ from $\Gamma \cup \{\varphi\}$, and (12) exemplifies the deduction of $\neg\varphi \vee \psi$ from Γ.

(9)	a.	Some alien visited Earth.	$\exists x\colon A(x) \wedge V(x)$
	b.	Every alien is a bird.	$\forall y\colon \neg A(y) \vee B(y)$
	c.	∴ Some bird visited Earth.	$\exists x\colon B(x) \wedge V(x)$

(10) 1. $\exists x\colon \underbrace{A(x)}_{\substack{+\\ \text{IMPORT}}} \wedge V(x)$

[11] As we only aim to illustrate the principle, we ignore some technicalities concerning universal generalization which arise if we do not restrict our attention to sentences.

2. $\exists x\colon (A(x) \wedge \forall y\colon \neg A(y) \vee B(y)) \wedge V(x)$
COPY

3. $\exists x\colon (A(x) \wedge (\neg A(x) \vee B(x))) \wedge V(x)$
PRUNE

4. $\exists x\colon (A(x) \wedge B(x)) \wedge V(x)$
DELETE

5. $\exists x\colon B(x) \wedge V(x)$

(11) a. Some alien visited Earth. $\exists x\colon A(x) \wedge V(x)$
b. ∴ If every alien is a bird, then some bird visited Earth.
$(\forall y\colon \neg A(y) \vee B(y)) \Rightarrow (\exists x\colon B(x) \wedge V(x))$

(12) 1. $\exists x\colon A(x) \wedge V(x)$
ADD

2. $(\neg\forall y\colon \neg A(y) \vee B(y)) \vee \exists x\colon A(x) \wedge V(x)$
COPY

3. $(\neg\forall y\colon \neg A(y) \vee B(y)) \vee \exists x\colon (A(x) \wedge \forall y\colon \neg A(y) \vee B(y)) \wedge V(x)$
COPY

4. $(\neg\forall y\colon \neg A(y) \vee B(y)) \vee \exists x\colon (A(x) \wedge (\neg A(x) \vee B(x))) \wedge V(x)$
PRUNE

5. $(\neg\forall y\colon \neg A(y) \vee B(y)) \vee \exists x\colon (A(x) \wedge B(x)) \wedge V(x)$
DELETE

6. $(\neg\forall y\colon \neg A(y) \vee B(y)) \vee \exists x\colon B(x) \wedge V(x)$

8.2.2 Existential Instantiation

The DDS treatment of Existential Instantiation (EI) is so simple that it can be hard to see how it relates to the classical implementation of EI. For starters, given $\exists x\, \varphi$ in a natural deduction system, EI opens a subderivation with the assumption $\varphi(a/x)$ for some fresh constant a, i.e. a constant which does not yet occur in the proof. How is this reflected in DDS?

First, no introduction of a fresh constant is necessary in DDS. We can simply continue to work with the variable x. This means the DDS deduction can proceed without extending the original formal language—another instance of adherence to Representational Minimalism. In classical systems, this is generally not the case.

This result has desirable consequences. For example, all the constants of the language might be "burdened" by occuring in some hypothesis—individuals a and b might be dogs, individual c might be sleeping and d might be reading e—so no fresh constant would be available *within* the formal language. In fact, all constants might be burdened even if infinite in number, although in that case, compactness would of course guarantee that not all of them play a role in the deduction.

Be that as it may, the formal language might not even admit constants at all—natural language analysis can certainly proceed without them! In that case, the extension of the language required by EI is not just quantitative but even qualitative: in order to apply EI, we must add a new *kind* of object to our vocabulary. This is not desirable.

As we have seen in the examples so far (and as we will illustrate yet again below), DDS does not require instantiation of the existentially bound variable to a constant, fresh or not, to proceed with reasoning. No extension of the language is necessary: we can work with what we have.

Given that we perform no substitution of the bound variable x, we would expect to have φ (most probably containing some free occurrences of x) as the assumption of "EI" of $\exists x\, \varphi$. Do we? Of course. In DDS, performing EI is really just targeting some constituent within an effectively existential quantifier, so φ, which forms the entire scope of the e-existential, clearly contains, and thus p-scopes, over any such target—remember descendant p-scope here: a constituent p-scopes over all its descendants, with p-scope polarity equal to its constituent polarity.

So, in the case of an EI-subderivation, the entire subderivation constituent p-scopes over the target. However, this fact is rarely used in an actual deduction. Typically, it is not the entire subderivation constituent that needs to act as a premise of some rule, but some constituent within it. Unlike a classical deductive system, we do not need to disassemble the subderivation constituent manually, p-scope does that for us automatically.

Let us illustrate this with an example.[12] First, the classical deduction (14) contains an EI-subderivation, (14.3)–(14.9). The fact that we may assume (14.3) in the classical system (thereby opening the subderivation) corresponds to the fact that $V(x) \wedge \neg A(x)$ p-scopes over itself in DDS. Next, in the classical derivation, the assumption (14.3) must be simplified to (14.4) before it can be used as the premise of MT. In DDS, this corresponds to the first $A(x)$ in (15.2) serving as a premise of Prune—it is just that we did not have to do anything to make this possible.

(13)	a.	All cats are aliens.	$\forall x\colon \neg C(x) \vee A(x)$
	b.	Some visitors as not aliens.	$\exists x\colon V(x) \wedge \neg A(x)$
	c.	$\therefore$ Some visitors are not cats.	$\exists x\colon V(x) \wedge \neg C(x)$

(14)	1.	$\forall x\,(C(x) \Rightarrow A(x))$	(hypothesis)
	2.	$\exists x\,(V(x) \wedge \neg A(x))$	(hypothesis)
	3.	$V(a) \wedge \neg A(a)$	(open EI 2, $x \mapsto a$)
	4.	$\neg A(a)$	(Simp 3)
	5.	$C(a) \Rightarrow A(a)$	(UI 1, $x \mapsto a$)

[12] For simplicity, we switch back to a sequential presentation of natural deduction derivations.

6.	$\neg C(a)$	(MT 5, 4)
7.	$V(a)$	(Simp 3)
8.	$V(a) \wedge \neg C(a)$	(Conj 6, 7)
9.	$\exists x\,(V(x) \wedge \neg C(x))$	(EG 8, $a \mapsto x$)
10.	$\exists x\,(V(x) \wedge \neg C(x))$	(close EI 3–9)

(15) 1. $\forall x\,(\underline{\neg C(x) \underset{+}{\vee} A(x)}) \wedge \exists x\colon V(x) \wedge \underbrace{\neg A(x)}_{\text{COPY}}$

2. $\ldots \wedge \exists x\colon V(x) \wedge (\overline{\underline{\neg A}(x) \wedge (\neg C(x) \vee A(x))})$ $\underbrace{\neg C(x) \vee A(x)}_{\text{PRUNE}}$

3. $\ldots \wedge \exists x\colon V(x) \wedge (\underbrace{\neg A(x) \wedge}_{\text{DELETE}} \overline{\neg C(x)})$

4. $\ldots \wedge \exists x\colon V(x) \wedge \overline{\neg C(x)}$

Finally, while the concept of closing off a subderivation is completely alien to DDS, a classical EI-subderivation must be closed and it may only be closed once it reaches a conclusion in which the introduced constant (a) does not occur. Above, the result of the Existential Generalization in (14.9) does not contain a, so we may close the subderivation in the next step, producing the same formula but in the context of the matrix derivation. (In the above example, this concludes the entire deduction, but in principle, the deduction can continue, as it does in (5) on page 170.)

Summing up, unlike classical deductive systems, DDS is stateless (it keeps no history of the deduction) and representationally minimal (it has no need for a data structure dedicated to storing the state of the deduction)—it gets all the information it needs from a data structure that any deductive system needs to know about anyway: the sentence it is operating on.

In particular, the subderivations, which are the primary workhorses in natural deduction systems like Gentzen's (regardless of whether they are implemented through proof trees or in some other way), can be reconstructed in DDS, but they don't need to be. All the work is done by our friend p-scope, in such an efficient manner that the concept of a subderivation is rendered otiose.

This state of affairs is desirable not only for its conceptual simplicity, but also from a cognitive and theoretical viewpoint. After all, we have shown that if you have the right deductive system, building and storing additional data structures is redundant at best. It is certainly counter to current approaches to language that prize economical systems that minimize effort as much as possible.

But we aren't the only ones to think dynamic, stateless deductions are superior. There is another stateless approach to deductive systems—the so-called Deep Inference approach—and in the next section we want to examine that approach, and compare it to our DDS.

8.3 Comparison with Deep Inference Deductive Systems

One of the characteristic features of the Dynamic Deductive System is the inline nature of the inference rules. But, as we noted, DDS is not the only deductive system on the market exhibiting this property. It shares it with a family of deductive systems known as *deep inference*.

Deep inference is a relatively new development in proof theory, the first papers appearing at the start of the century (Brünnler and Tiu 2001; Guglielmi and Straßburger 2001). It was motivated by issues in computer science (formal modeling of concurrent systems in particular) and continues to be primarily applied to that field. At the same time, the investigations into deep inference offer many new insights into the proof theory itself, in particular by developing entirely novel normalization methods. While it was linear logic that first received the deep inference treatment, deep inference systems have now been developed and investigated for various logics, including modal logics, intuitionistic logic, and—importantly for our humble purposes—classical logic, both propositional and predicate (Brünnler and Guglielmi 2002, 2004; Brünnler 2003, 2004, 2006a,b,c; Brünnler and McKinley 2008; Guglielmi, Gundersen, and Straßburger 2010).[13]

Our plan in this section is to investigate certain properties of DDS by comparing it with deep inference systems. To date, there are three incarnations of deep inference: *Calculus of Structures (CoS)* (Guglielmi 2007), *Open Deduction* (Guglielmi, Gundersen, and Parigot 2010), and *Nested Sequents* (Brünnler 2010). We limit the comparison to CoS, both because it is arguably the simplest incarnation of deep inference and because the basic structure of a deduction in both CoS and DDS is the same.

As mentioned above, the essential property shared by CoS and DDS is the adoption of inline rules. Several other major properties of the two systems are a consequence of this single feature.

- Both DDS and CoS are stateless and therefore require no additional data structures. In other words, they both adhere to Representational Minimalism.
- Both DDS and CoS are LINEAR, where "linear" is the term sometimes used for our "dynamic":[14] for any two adjacent stages φ_n and φ_{n+1} of the deduction, φ_n entails φ_{n+1} *without any additional assumptions.*

[13] A recent primer for deep inference is Tubella and Straßburger (2019), and an overview can be found in Guglielmi (2015). An up-to-date list of deep inference literature is maintained at http://alessio.guglielmi.name/res/cos/index.html.

[14] The very first linear deductive system seems to be Craig's (1957) Linear Reasoning. A similar system can also be found in Henkin's (1963) proof of an extension of the Craig-Lyndon Interpolation Theorem. (Our polarity–monotonicity correspondence is based on this theorem; for details, see section A.3.)

However, the two systems also differ in two major respects—the treatment of polarity, and the locality of the system. The principle aim of this section is to present these differences in some detail. We will argue that polarity-sensitivity (a property of DDS) and locality (a property of CoS) are mutually exclusive, and that the choice between them is dictated by the main intended application of the system: linguistics for DDS, and computer science for CoS.

8.3.1 Polarity versus locality

Since we are already familiar with DDS, let's present CoS by utilizing the contrast between the systems. We start with polarity, in fact with the very definition of negation in CoS, as it is not what we are used to from classical logic. This begins with the notation: in CoS, the negation of $\mathcal{A}$ is written as $\overline{\mathcal{A}}$ in place of $\neg\mathcal{A}$. An example will make this clear:

(16) $\overline{\exists x\colon A(x) \wedge B(x)} \equiv \forall x\colon \overline{A(x)} \vee \overline{B(x)}$

To understand this, let's remember that we transform a formula into NEGATION NORMAL FORM (NNF) by pushing the negations all the way down to atomic formulas. The NNF of a formula is of course logically equivalent to the original.

(17) $\neg\exists x\colon A(x) \wedge B(x) \rightsquigarrow \forall x\colon \neg(A(x) \wedge \neg B(x)) \rightsquigarrow \forall x\colon \neg A(x) \vee \neg B(x)$

Intuitively, we can see the CoS negation as the negation normal form of the standard negation. We could say that formulas of CoS are auto-normalized with respect to negation. The formal, recursive definition of negation in CoS is given in Table 8.1a. (The other parts of the Table will be discussed in due course.)

The treatment of negation is enough to see that polarity is going to have a very different status in CoS than in DDS. As we know, the latter is all about polarity. In CoS, the concept of polarity plays virtually no role. Only an atomic formula can have negative polarity in CoS, and even in this case, we know that it occurs in the scope of a single negation, which is furthermore immediately above it. All other constituents are not only of positive polarity, they are not dominated by a negation at all.

Before we turn to locality of inference rules, let us present a sound and complete CoS rule system for predicate logic. There are several such systems, of course: Table 8.1b presents system SKSgr,[15] defined by Brünnler (2006a). The rules of

[15] SKSgr: a *s*ymmetric classical (ger. *k*lassisch) calculus of *s*tructures for *g*eneral formulas with the *r*etract rule.

Table 8.1 Calculus of Structures for predicate logic

(a) Negation

(a) $\overline{\overline{a}} :\equiv a$, where a is an atom (either an atomic formula or its negation)
(b) $\overline{\bot} :\equiv \top, \overline{\mathcal{A} \vee \mathcal{B}} :\equiv \overline{\mathcal{A}} \wedge \overline{\mathcal{B}}, \overline{\exists x \mathcal{A}} :\equiv \forall x \overline{\mathcal{A}}$
(c) $\overline{\top} :\equiv \bot, \overline{\mathcal{A} \wedge \mathcal{B}} :\equiv \overline{\mathcal{A}} \vee \overline{\mathcal{B}}, \overline{\forall x \mathcal{A}} :\equiv \exists x \overline{\mathcal{A}}$

(b) The inference rules (SKSgr)

Identity (i↓)	Identity (i↑)
$\vdash^0_{\top} \mathcal{A} \vee \overline{\mathcal{A}}$	$\vdash^0_{\mathcal{A} \wedge \overline{\mathcal{A}}} \bot$
Weakening (w↓)	Weakening (w↑)
$\vdash^0_{\bot} \mathcal{A}$	$\vdash^0_{\mathcal{A}} \top$
Contraction (c↓)	Contraction (c↑)
$\vdash^0_{\mathcal{A} \vee \mathcal{A}} \mathcal{A}$	$\vdash^0_{\mathcal{A}} \mathcal{A} \wedge \mathcal{A}$
Switch (s)	Switch (s)
$\vdash^0_{(\mathcal{A} \vee \mathcal{B}) \wedge \mathcal{C}} \mathcal{A} \vee (\mathcal{B} \wedge \mathcal{C})$	(self-dual)
Retract (r↓)	Retract (r↑)
$\vdash^0_{\forall x \mathcal{A} \wedge \mathcal{B}} \mathcal{A} \wedge \forall x \mathcal{B}$ $\vdash^0_{\forall x \mathcal{A} \vee \mathcal{B}} \mathcal{A} \vee \forall x \mathcal{B}$ where x is not free in $\mathcal{A}$	$\vdash^0_{\mathcal{A} \wedge \exists x \mathcal{B}} \exists x \mathcal{A} \wedge \mathcal{B}$ $\vdash^0_{\mathcal{A} \vee \exists x \mathcal{B}} \exists x \mathcal{A} \vee \mathcal{B}$ where x is not free in $\mathcal{A}$
Instantiate (n↓)	Instantiate (n↑)
$\vdash^0_{\mathcal{A}(t/x)} \exists x \mathcal{A}$	$\vdash^0_{\forall x \mathcal{A}} \mathcal{A}(t/x)$

(c) The syntactic equivalence classes

(a) $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C}) \equiv (\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}$	(a) $\mathcal{A} \vee (\mathcal{B} \vee \mathcal{C}) \equiv (\mathcal{A} \vee \mathcal{B}) \vee \mathcal{C}$
(b) $\mathcal{A} \wedge \mathcal{B} \equiv \mathcal{B} \wedge \mathcal{A}$	(b) $\mathcal{A} \vee \mathcal{B} \equiv \mathcal{B} \vee \mathcal{A}$
(c) $\mathcal{A} \wedge \top \equiv \mathcal{A}$	(c) $\mathcal{A} \vee \bot \equiv \mathcal{A}$
(d) $\forall x \mathcal{A} \equiv \forall y \mathcal{A}(y/x)$ where y is free for x in $\mathcal{A}$	(d) $\exists x \mathcal{A} \equiv \exists y \mathcal{A}(y/x)$ where y is free for x in $\mathcal{A}$
(e) $\forall x \top \equiv \exists x \top \equiv \top$	(e) $\forall x \bot \equiv \exists x \bot \equiv \bot$

the left and the right column are called the down- and the up-rules, respectively. The rules are adapted to the DDS format, for two reasons. First, we don't want to overburden the reader with a new notation. To accommodate CoS rules, we need a single new notational convention: let the superscript 0 on the turnstile symbol ($\vdash^0$) indicate that the CoS rules apply when there are no negations above them—which is, given the CoS treatment of negation, of course always the case. Second, we want to show that the systems are similar enough that presenting CoS in DDS format is easily doable.[16]

Indeed, with several slight amendments, all the rules in Table 8.1b make perfect sense as DDS rules. (Perhaps the most striking difference is that no CoS rule requires a non-target premise. We return to this fact later on.) All we need to do is relax the polarity requirement (i.e. replace $\vdash^0$ by $\vdash^\tau$), interpret the negation in the standard way, and superscript each connective and quantifier of a rule with the target polarity. We provide three examples below. The polarity-sensitive variant of i$\uparrow$ is very close to our Prune. Both rules have the same basic job, eliminating a contradiction (or a *conflict*, in our terminology). In fact, if we replaced Prune by the polarity-sensitive i$\uparrow$, DDS would remain complete. The effect of the polarity-sensitive n$\uparrow$ can be achieved by Copying an effectively universal quantifier node over itself. The CoS rule, Switch, is also nothing but Copy restricted to a particular relation between the premise ($\mathcal{C}$) and the target ($\mathcal{B}$), followed by the Deletion of the premise (if we had a rule called Move, Switch would be closer to Move than to Copy).

Identity (polarity-sensitive) (i$\uparrow$)

$$\vdash^\tau_{\mathcal{A}\wedge^\tau\neg\mathcal{A}} \bot^\tau$$

Instantiate (polarity-sensitive) (n$\uparrow$)

$$\vdash^\tau_{\forall^\tau x\,\mathcal{A}} \mathcal{A}(t/x)$$

Switch (polarity-sensitive) (s)

$$\vdash^\tau_{(\mathcal{A}\vee^\tau\mathcal{B})\wedge^\tau\mathcal{C}} \mathcal{A}\vee^\tau(\mathcal{B}\wedge^\tau\mathcal{C})$$

Moving on to locality, a rule is said to be LOCAL if "determining whether an application of the rule is correct we do not need to inspect arbitrarily big formulas" (Tubella and Straßburger 2019, p. 20). Consider Switch (s)—the official CoS variant, defined in Table 8.1b. Its validity should be clear from the example below, and it is also obvious that it is local in the above sense. To apply it, we only need to find a disjunction (below, $B \vee P$) whose parent is a conjunction. Then the other conjunct (F) can be moved (by conjoining) next to one of the disjunctions (below, P). The structure of the original disjuncts (B and P) and the other conjunct (F) is completely irrelevant.

[16] In particular, the CoS notion of *context* corresponds to our target.

(18) a. $\underbrace{\text{I'm reading a book or a paper, and I'm reading fast.}}_{\text{s}}$ $\underbrace{(B \vee P) \wedge F}_{\text{s}}$

b. $\therefore$ $\overline{\text{I'm reading a book, or I'm reading a paper fast.}}$ $\overline{B \vee (P \wedge F)}$

Now a question: are there any local rules in DDS? For example, is Delete local? After all, we only need to find a conjunction and replace it with one of its children, right? The identity of the children is irrelevant.

(19) a. I'm reading a book, or $\underbrace{\text{I'm reading a paper } \underbrace{\text{fast.}}_{+}}_{\text{DELETE}}$ $B \vee (\underbrace{P \wedge \underbrace{F}_{+}}_{\text{DELETE}})$

b. $\therefore$ I'm reading a book or $\overline{\text{a paper}}$. $B \vee \overline{P}$

We're afraid the answer is still "no." Delete is not local. Add is not local. There are no local rules in DDS because all of them are *sensitive to polarity*!

In order to check whether we may Delete a child of a binary node, we need to see whether the node is an *effective* conjunction, which implies that we need to count the number of negations above it. This is certainly not a local procedure. While we need to inspect only a very well-defined region of the dynamic formula—the ancestors of the target—there is no bound on the depth of formulas, so there is no bound on how many nodes we have to inspect. The same argument applies for all other inline rules of DDS, as they are all sensitive to the polarity of the target, either introducing or eliminating *effective* conjunctions or disjunctions.

We have thus demonstrated the essence of the antagonism between polarity-sensitivity and locality. These properties are mutually exclusive: we can aim for one or the other, but we can't have both. If we want our deductive system to apply to formulas "as they are," our inference rules will be non-local. For the rules to be local, the effect of polarity must be abstracted away, and this is precisely what is achieved by the auto-normalizing negation of CoS.

So, Cos goes for locality, and DDS goes for polarity. Is there anything wrong with one or the other path? Of course not! It all depends on the intended application of the system.

CoS is geared toward applications to computer science and the development of proof theory in mathematics. We cannot go into any detail of why locality is of paramount importance in these contexts, so let us simply say that in math, locality is central for the development of proof normalization methods (Brünnler and McKinley 2008; Gundersen 2009; Das 2014; Straßburger 2011), while in computer science, it "implies a bounded computational cost of applying an inference rule and seems useful for distributed implementation" (Brünnler 2004, p. 5).

However, we do want to go into some detail on why the absence of locality, as understood in CoS, is not detrimental for the application of DDS to natural language. Quite the opposite! As we shall see, enforcing the CoS-type locality upon DDS (or any mechanism intended to work with natural language) would yield unwieldy deductions and make us propose logical forms incongruent with linguistic data.

In linguistics, one cannot work on semantics without at least stealing a glance at syntax, and vice versa. The two disciplines inform each other, and even more importantly, the structures they propose are usually expected to be compatible with one another. In Chapter 12, we will try to convince you that the structures should in fact be more than just compatible—that they should be isomorphic—but right here and now we are only leaning on the idea that the syntax and semantics of an expression should at least be structurally similar (the semantic rules have to accept the syntactic forms as input, after all).

Take the logical form of the negative determiner 'no' as an example. It is variously analyzed as either (20a) or (20b), but we have never seen anybody argue for (20c), which is of course precisely the NNF that strict adherence to locality would force upon us. Why is (20c) never an option? Well, given the guideline of syntax and semantics going hand in hand, one would propose such representation if the language offered us something like (21)—but we haven't seen a language doing that yet, and we highly doubt we ever will.

(20) No angry dog is barking.
a. $\neg\exists x\colon (A(x) \wedge D(x)) \wedge B(x)$
b. $\forall x\colon \neg(A(x) \wedge D(x)) \vee \neg B(x)$
c. ${}^{*}\forall x\colon (\neg A(x) \vee \neg D(x)) \vee \neg B(x)$

(21) *Not-angry no-dog is not barking.

Of course, it would be technically possible that a logical form like (20a) or (20b) could be transformed into NNF (20c) prior to deduction—computers often do precisely that—but this possibility seems quite far-fetched when it comes to natural language. (If you remember, in section 4.4 we had actually criticized L*, the predecessor to L**, for operating only on normal forms.)

A similar argument—that form matters—can be raised against taking the following widespread convention from CoS and importing it into DDS. In CoS, it is usual to abstract away from details such as commutativity and associativity of conjunction and disjunction. In other words, the formulas are grouped into equivalence classes based on the basic properties of various connectives. The equations generating the equivalence classes are presented in Table 8.1c. This practice is perfectly innocuous from the mathematical viewpoint, and it is clear how abstracting away from some of them might simplify a computer implementation of a deductive system, but adopting the practice to natural language analysis seems like a bad move. Staying with commutativity and associativity, we would not be able to distinguish between the subject and the predicate in (22), and similarly, we would not be able to know which noun the adjective in (23) belongs to.

(22) a. Some alien is a bird. $\exists x\colon A(x) \wedge B(x)$
b. Some bird is an alien. $\exists x\colon B(x) \wedge A(x)$

(23) a. Some green alien is a bird. $\exists x\colon (G(x) \wedge A(x)) \wedge B(x)$
b. Some alien is a green bird. $\exists x\colon A(x) \wedge (G(x) \wedge B(x))$

Summing up the discussion in this subsection, we have seen that locality and sensitivity to polarity are opposing forces in the design of a deductive system, and we have put forth the idea that the choice of one over the other is primarily influenced by the intended application of the system. Mathematics and computers prefer locality, linguistics prefers polarity. However, we have not yet presented all sources of non-locality in DDS. Are they all innocuous or even welcome properties of the system, given its application to natural language? This is the question we turn to in the next subsection.

8.3.2 Atomicity and non-local premises

The first source of non-locality in DDS we want to turn our attention to is actually shared with the particular variant of CoS presented in Table 8.1b. Consider the down-rule of Contraction (c↓), repeated below: a disjunction of two identical formulas is reduced to one of the disjuncts. In order to apply the rule, one needs to check whether the disjuncts of the target are in fact identical. However, these disjuncts can be of any size, which directly contradicts locality. The way to deal with this is to assume that the disjuncts of the target must be atoms (either an atomic formula or its negation). The resulting rule (usually called ac↓) is said to be ATOMIC and is clearly local. For example, c↓ can perform both inferences below, while ac↓ applies only in the second example.

Contraction (c↓)	Contraction (atomic) (ac↓)
$\vdash^{0}_{\mathcal{A}\vee\mathcal{A}} \mathcal{A}$	$\vdash^{0}_{a\vee a} a$, where a is an atom

(24) a. It is sunny and warm, or it is sunny and warm. $\underbrace{(S \wedge W) \vee (S \wedge W)}_{\text{c}\downarrow\ /\ {}^*\text{ac}\downarrow}$
b. ∴ It is sunny and warm. $\overline{S \wedge W}$

(25) a. It is sunny, or it is sunny. $\underbrace{S \vee S}_{\text{c}\downarrow\ /\ \text{ac}\downarrow}$
b. ∴ It is sunny. $\overline{S}$

Of course, atomizing a rule might impact the deductive power of the system. It is therefore necessary to prove, for each atomized rule, that the original, general rule is derivable from the atomic rule and the rest of the system. And indeed, this seems to be generally achievable for many logics, including the classical propositional and predicate logic. A fully local CoS system for these logics is presented in Brünnler (2006c).[17]

Now, what does linguistics have to say about atomic rules? Are operations in natural language atomic? Certainly not. If there is anything the linguists of various persuasions might agree upon, it is that language works with *constituents*. For example, in many languages, forming questions involves (overt) movement of the wh-phrase, and there is certainly no evidence that the phrase 'which big black dog' in (26) is moved word by word (or even more implausibly, morpheme by morpheme). Another example—perhaps even better, as it presumably involves checking for structural identity—is ellipsis in (27), which again operates on constituents. In a nutshell, if our inference rules are in any way similar to syntactic operations, we expect them to be non-atomic.

(26) [Which big black dog] is he afraid of ~~[which big black dog]~~?

(27) John likes to play the guitar, and so does Mary ~~[like to play the guitar]~~.

We mentioned syntactic movement in the example above, and it is instructive to consider another property of this operation. What can be moved where? Within the generative tradition, the answer is that movement is subject to a relation called c-command, in the sense that the landing site must c-command the origin of the moved material. C-command, which we mentioned earlier in the book, is defined like this: α C-COMMANDS β iff α does not dominate β and the first branching node properly dominating α also properly dominates β.

Clearly, c-command can relate nodes at arbitrary distance, and is thus not local in the CoS sense. (But note that linguists have their own notion of locality!) Furthermore, note that c-command is not just *some* syntactic relation: think of it rather as the very heart of syntax. Most likely it is involved, in some form or another, in virtually every syntactic phenomenon—it was first proposed as a crucial ingredient of so-called Binding Theory, which tells us when nominal expressions can corefer. For example, the unacceptability of (28a) (under the reading where 'he' and 'John' are coreferential) is argued to be due to the fact that

[17] There is a residue of non-locality in predicate logic: any rule which inspects or manipulates the free variables of a subformula—in Tables 8.1b and 8.1c, instantiations, variable renaming and vacuous quantifier eliminations—is non-local, as it needs to inspect a formula of arbitrary size, and cannot be (easily) made atomic, as the entire point of these rules is to apply to all the free occurrences of a particular variable at the same time. Luckily, the issue does not critically affect the mathematics: for example, the system nevertheless yields to normalization. (Cf. Brünnler 2006c.)

'John' is c-commanded by a coreferential expression; (28b) shows that the relevant condition is not simply precedence, but must be structural.

(28) a. *He$_i$ read all the books on John's$_i$ shelf.
b. [His$_i$ friend] read all the books on John's$_i$ shelf.

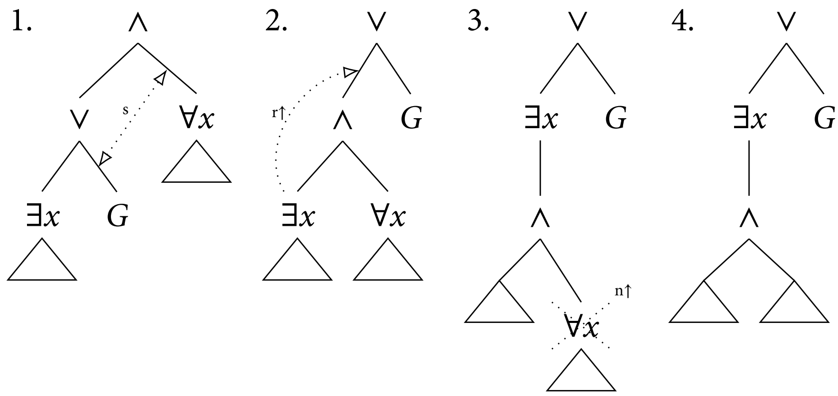
Given how ubiquitous c-command is, we guess no linguist would ever object to another long-distance relation—premise scope—playing a foundational role in a natural deductive system. It might even turn out that c-command and (relative) p-scope are closely related, but such speculation is outside the purview of this particular book.

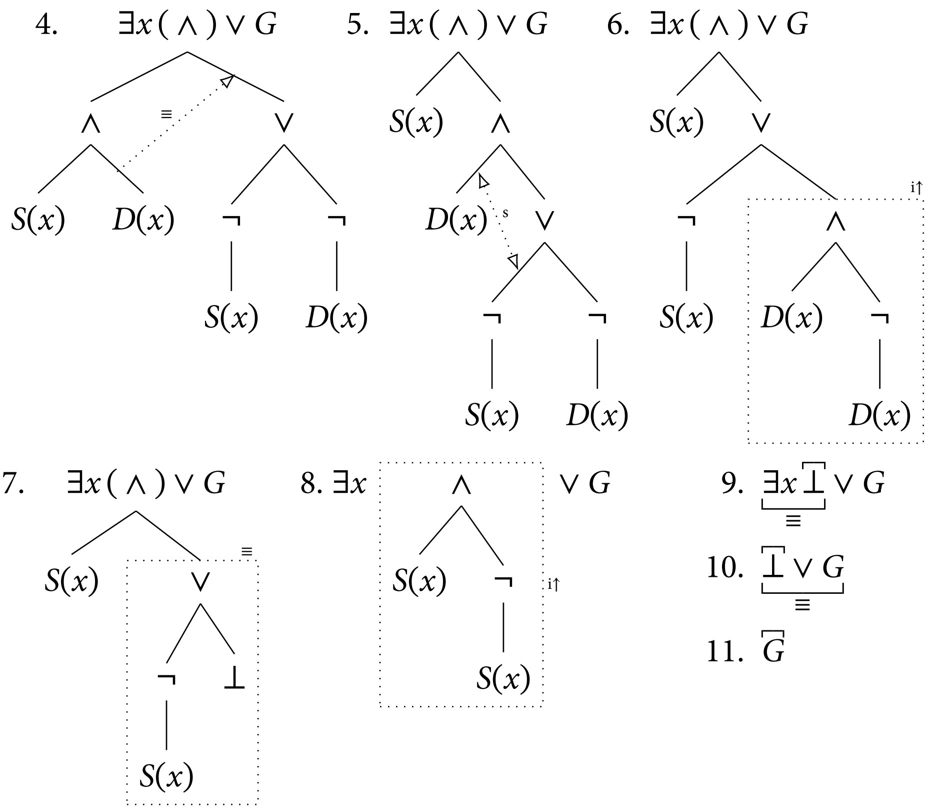
As the final argument against "localizing" DDS, let us assume we made the rules of DDS local. Would we find the deductions of the modified DDS *natural*, i.e. simple enough for the system to be part of a cognitively plausible theory of human reasoning? We believe we would not, but you can judge for yourself. As we have seen, CoS and DDS really differ only in matters of locality, so CoS can easily serve as an approximation of a local variant of DDS. We therefore present a CoS deduction (30) of a simple inference and invite you to compare it to the DDS deduction (31) of the same inference. (For clarity, we present the CoS deduction in the tree format.)

For the full effect of locality, we assume the atomicity of Identity and Contraction. The first effect of this that the CoS deduction cannot be performed in merely three steps. Remember that in CoS, $\overline{\exists x\colon S(x) \wedge D(x)}$ is identical to $\forall x\colon \overline{S(x)} \vee \overline{D(x)}$. If up-Identity was non-atomic, it could be applied to (30.2), cutting directly to (30.10). In fact, a similar remark goes for DDS: if the second hypothesis was given as $\neg\exists x\colon S(x) \wedge D(x)$, we could derive the conclusion by a single application of Prune!

(29) a. Either some sign is dirty, or I need new glasses. $(\exists x\colon S(x) \wedge D(x)) \vee G$
b. No sign is dirty. $\forall x\colon \neg S(x) \vee \neg D(x) \;\sim\; \forall x\colon \overline{S(x)} \vee \overline{D(x)}$
c. ∴ I need new glasses. G

(30)





9. $\exists x \underbrace{\overline{\bot}}_{\equiv} \vee G$

10. $\underbrace{\overline{\bot} \vee G}_{\equiv}$

11. $\overline{G}$

(31)

1. $\Big(\big(\exists x\colon \underbrace{S(x) \wedge D(x)}_{\text{COPY}}\big) \vee G\Big) \wedge \forall x\colon \neg S(x) \vee \neg D(x)$

2. $\Big(\big(\exists x\colon \overline{(S(x) \wedge D(x)) \wedge (\neg S(x) \vee \neg D(x))}\big) \vee G\Big) \wedge \ldots$ PRUNE

3. $\Big(\big(\exists x\colon (S(x) \wedge D(x)) \wedge \overline{\neg S(x)}\big) \vee G\Big) \wedge \ldots$ PRUNE

4. $\overline{G} \wedge \underbrace{\ldots}_{\text{DELETE}}$

5. $\overline{G}$

As matters stand, however, both systems need to take the long way and perform i↑/Prune twice: once on $D(x)$ and its negation, and once on $S(x)$ and its negation. In this respect, the systems behave alike—not, however, in how they set up the situation where these operations apply. A single Copy of DDS (31.1) must be performed by a series of rules (Switch, Retract, Instantiate) in CoS (30.1–4). The conciseness of DDS is of course due to p-scope. (Note that (30.4) is shown twice, with different levels of "zoom.")

P-scope also makes it possible to apply Prune directly to (31.2): the positive polarity $D(x)$ p-scopes over the negative polarity $D(x)$, over some distance. In CoS, on the other hand, we need to move $D(x)$ by Switching (30.5).

Prune contains yet another source of non-locality: in (31.3), it eliminates not just the target of p-scope, but the entire e-disjunct. Yet again, in CoS this effect must be achieved manually—by (30.7) after the first i$\uparrow$ in (30.6), and by all the subsequent steps for the second i$\uparrow$ in (30.8).

In a nutshell, the above deductions of (29) show that the absence of p-scope in CoS makes the deduction much more fine-grained. CoS needs to work node-by-node, while in DDS we work over long distances.

Summing up, locality (as understood in CoS) is not a desirable feature of a deductive system intended for application to natural language. All this is not us trying to say that locality is a bad property of a system per se, as it is not! It is simply suited for *certain* applications, and natural language is not one of them. Of course, the reverse holds as well: anti-locality is perfect for natural deductions and well-aligned with properties of syntactic operations, but presents unnecessary complications for mathematical analysis and computational implementation.

In this section, we have compared two closely related deductive systems, our DDS and CoS—a formalism within the deep inference paradigm. The full list of properties discussed in this section can be found in Table 8.2. Crucially, we have illustrated the principal theoretical conflict between those approaches—between polarity and locality—and we argued that absence of CoS-type locality in DDS is far from detrimental and in fact aligns better with the intended usage of DDS as the deductive system for natural language. In other words, *Dynamic Deductive System is a deep inference system with polarity*, and it is precisely its sensitivity to polarity which makes DDS *natural*.

Of course our comparison with the CoS deductive system was part of our more general comparison of DDS to other deductive systems, which included comparison to Hilbert and Gentzen natural deduction systems. If there is a thread running through these comparative discussions, it is that we believe the DDS is a

Table 8.2 Dynamic Deductive System vs. Deep Inference

	DDS	CoS
inline rules	✓	✓
representationally minimal	✓	✓
stateless	✓	✓
linear	✓	✓
no language-extensions	✓	✓
locality	✗	✓
atomicity	✗	✓
transparent handling of subderivations	✓	✗
polarity-sensitive rules	✓	✗
non-NNF	✓	✗
non-classical logics	✗	✓

more apt way to model deduction in natural language, which on our view is the home of Natural Logic.

Thus our work on DDS in the last three chapters is bringing us closer to the question of just how well our account of L** does as an analysis of natural language. We are almost there. We have one more bit of prepatory work before we examine this project within the context of contemporary linguistic theory. We want to explore some of the insights into the theory of quantification that we can now glean thanks to p-scope and DDS. As we will see, our foray into the logics of polarity provides us a new way of cashing out an idea initially stumbled upon by the Medieval logicians—the idea that polarity and quantification are intimately interrelated. And as we will also see, our work developing the theory of p-scope and rules like Copy will have application here as well. Indeed they will play a key role in cashing out one of the deepest and most important properties explored in generalized quantifier theory—conservativity.

9
Conservativity, Restrictedness, and Quantification

In the previous eight chapters, we explored and updated the Medieval logicians' idea that there are two fundamental principles or rules underlying logical inference. The Medievals called these rules the *dicta de omni et nullo* and we have been describing the environments in which they apply in terms of positive and negative polarity. Our aim thus far has been to show how, at least in select formal languages, positive and negative polarity environments can be syntactically identified, and how those environments can be used in a system of natural deduction. Specifically, we showed that deductions could take place inline with the help of the syntactically simple rules Copy, Prune, Add, and Delete, and a polarity-sensitive relation that we called p-scope. Along the way, however, we discussed two other topics. One of the topics involved the Medieval idea, discussed in section 2.3, that there is a deep relation between polarity and quantification. The second topic, discussed in section 3.3, is the notion of conservativity as applied to determiners in generalized quantifier theory.

Let's consider the polarity/quantification topic first: Is there really some deep relation between polarity and quantification in natural language? We believe that the Medieval logicians were on the right track, although it is clear that they did not have the tools to successfully sharpen up the nature of this relation. As we will see, getting clear on that relation requires that we visit our second topic, conservativity. *Conservativity*, generalized to apply to language *tout cours* and not just determiners, turns out to be the key that unlocks the secret ingredient we need for successfully executing the Medieval insight about quantification.

The secret ingredient itself is *restrictedness*. You are probably familiar with this notion from the concept of a restricted quantifier (also called a bounded quantifier). In restricted quantification, the range of the quantified variable is limited to a certain class of objects—for example, the variable bound in the logical forms of 'some dog is barking' and 'every dog is barking' would be restricted to range over dogs.

It has been noted—if underappreciated and certainly not a point of emphasis by logicians and semanticists—that restrictedness is the syntactic counterpart of the semantic notion of conservativity. We don't claim to have discovered this equivalence, but we can provide the long-overdue explicit proof of the equivalence. More importantly, we will make the case that restrictedness can and should play

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović, Oxford University Press. DOI: 10.1093/oso/9780199591534.003.0009

an important role in natural language semantics—a role that has not been properly considered, we believe, due to the wide-spread and erroneous conviction that no version of first-order predicate logic can be as insightful a tool as the (model-theoretic) semantic analysis of natural language.

What comes as a real surprise, however, is that while restrictedness can be formulated in various ways, the option that seems best suited when dealing with natural language relies on *p-scope*, the relation that powers our Dynamic Deductive System!

In other words, we are going to make the case for an account of quantification that crucially involves polarity, conservativity/restrictedness, and p-scope. How could all of these ideas be related? To understand that, we need to say a bit about a bright idea that turned out to be a failed idea, precisely because it lacked the secret ingredient of restrictedness.

Ludlow (2002) observed, by looking at formulas of L*, that certain negations that appear in the analysis of determiners seem to be excellent predictors that universal quantification is in play. And it likewise appeared to be the case that the lack of these negations is a good predictor of some form of existential quantification. The prototypical example of this correlation between the absence vs. presence of a negation and the existential vs. universal quantification is exhibited by the pair of basic determiners 'a' and 'every', as shown below: in (1a), $D(x)$ has positive polarity and the quantification over x is existential; in (1b), the polarity of $D(x)$ is negative and we get a universal quantifier.

(1) a. Some dogs are barking. $\exists x\colon D(x) \wedge B(x)$
b. Every dog is barking. $\forall x\colon \neg D(x) \vee B(x)$

Ludlow's idea was that if that we can predict whether the quantifier turns out existential or universal just by inspecting the polarity of the variable, then we don't need the quantifiers at all! We can let all the variables be formally free and introduce a *closure rule* which tells us how to interpret them. If we could successfully execute this idea, then we could simplify the logical forms in (1) to those we find below. (In (2) we show, separated by the squiggly arrow, both the simplified *quantifierless* formula, and the result of variable closure in the standard format, which contains what we call the *implicit quantifier*.)

(2) a. Some dogs are barking. $D(x) \wedge B(x) \rightsquigarrow \exists x\colon D(x) \wedge B(x)$
b. Every dog is barking. $\neg D(x) \vee B(x) \rightsquigarrow \forall x\colon \neg D(x) \vee B(x)$

Anticipating the discussion in Chapter 12, where we propose a rudimentary implementation of the quantifierless format in linguistics (specifically, in the Minimalist Program), on this picture determiners have no meanings in and of themselves, and for that matter don't even appear in the syntax—there is a rule for the interpretation of free variables, and everything else is structure. Among other things, this allows us to generalize the idea of Kamp (1981), Heim (1982), and Diesing (1992) that indefinite descriptions like 'a man' in natural language are not quantifiers, but rather contain free variables that get unselectively bound. While this idea was already extended to other phenomena such as sentential negation (e.g. Acquaviva 1994; Zeijlstra 2004) our proposal might be unique in claiming that unselective binding applies across the board. There are no quantificational operators in the syntax of natural language.

Ludlow's (2002) idea is very appealing, but it doesn't work because it is missing our secret ingredient: restrictedness! While he gets the position of the implicit quantifier right—proposing that it should have scope just wide enough to bind all occurrences of the variable—Ludlow tries to pin down its type by the polarity of the variable, the problem being that the variable can have more than one occurrence, and these occurrences might (and often do) have different polarities. How can we know which occurrence is relevant? The answer is to be found by paying attention to conservativity.[1]

All of this may sound interesting, and perhaps even surprising. But as we will see, there is a lot more to conservativity than its current use as a tool to define natural language determiners. In fact, conservativity applies not just to determiners, but to all natural language expressions! In other words, it is not only determiners that are conservative (or conservative in their first position, as the thesis is generally presented), but rather all of natural language is conservative.

That's probably a lot to absorb, but that's ok, because we are going to step through the relevant elements as carefully as possible. We will first take up the question of what conservativity is, and show how the notion can and should be generalized. We will then explore its traditional role in an account of determiners and show how that story, too can be generalized. We will then move on to a discussion of restrictedness and show how it cashes out the notion of conservativity and how notions like restriction and nuclear scope are not local notions, but sentential level notions. And finally we will apply all of this to an account of quantification, with some reflections on the deep relation running between quantification and polarity, albeit with the help of our understanding of conservativity, deduction, and p-scope.

[1] Sánchez Valencia (1991, chapter III) notes a related connection between conservativity and restrictedness in Peirce's *Beta System of existential graphs*. Peirce's formal language and deductive system appear to resemble our quantifierless L** and DDS in many respects: his formal language deploys implicit quantification, and the inference rules are sensitive to polarity and include a copy rule. However, the notion of premise scope seems to be unique to our proposal. See also Zeman (1967) for an implementation and development of Peirce's idea in the familiar notation of contemporary predicate logic. Further research is needed for a detailed comparison of these systems, in particular with respect to the question to what extent conservativity can be said to actually play a role in Peirce's system.

9.1 Conservativity and Aboutness

9.1.1 What are sentences about?

What is sentence (3) about?

(3) Joan bought every red apple at the market.

A very good answer might be that it is about Joan. And indeed, this particular idea of aboutness, in some form or other, appears in a wide range of linguistic frameworks. Put simply, languages employ various means of catching our attention. Sometimes what a sentence is about is syntactically characterized as the *subject*, sometimes as the *theme*, sometimes as the *topic* (the rest of the sentence is then often called the predicate, the rheme, and the comment, respectively). Often, it can be seen that this element is subject to some form of syntactic movement. Some languages mark it by morphological means, like the nominative case of the subject, or a topic marker such as suffix '-wa' in Japanese. However, this is not the answer we are aiming for.

The notion of aboutness we are after is at the same time much more general and (actually, for that very reason) much more mundane. Sentence (3) is about Joan, yes, but also about apples and markets, red things, and even about buying. It is not about Mary, pears, and blue things, simply because the words 'Mary', 'pear', and 'blue' do not feature in the sentence. Every sentence is about whatever the content words of the sentence denote or refer to.[2] And while this notion of aboutness might at first seem trivial to the point of our asking, "why even mention it?" we shall see that, in fact, it expresses a very deep insight into the nature of human language.

It might be illuminating to consider the practical value of aboutness in the linguistic context. You see, aboutness allows us to narrow our attention to only those parts of reality that are mentioned in a sentence by some content word. For example, to evaluate whether (3) is true or false, we don't need to consider what Mary might be doing, and it is irrelevant whether Joan is buying pears, or whether Mary is even around or there are any pears sold at the market. Just imagine needing to consider every blueberry at the market to judge whether you agree with someone stating (3)! This would be a daunting task, even with the context heavily limiting the objects relevant in a discourse. Aboutness allows us to keep the focus of the semantic evaluation on the things we are talking about, and ignore everything else.

It didn't have to be this way. From a logical point of view, there is nothing that says our sentences should be about the things we explicitly talk about in a sentence. It is easy to define operators that don't have this property (we will look at some examples in a bit). Of course, from the perspective of pragmatics, we clearly

[2] The only real exceptions seem to be connected to focus, which we discuss in section 10.1.

can and do communicate about things that we aren't explicitly talking about. One might ask someone to shut the door as a way to communicate something about the noise outside, or alternatively they might comment on the noise outside to communicate that they would like the door shut. But that is pragmatics, which in turn relies on a clear notion of aboutness and other semantics properties. When we restrict our attention to the semantics itself, it is usually transparent what we are literally talking about.

As we noted, the fact that sentences are about whatever is expressed by the words they contain is so obvious we typically don't think it worth remarking, but sometimes it pays to think about the obvious and wonder why things are that way. It may seem completely obvious that apples should fall from trees, but when you start thinking about why this obvious thing should happen, you can open the door to some very deep understanding of nature (as Newton showed). Accordingly, we will devote a fair bit of time to discussing aboutness (and its formal expression) in the sections to follow, and we will even return to the topic in section 10.4.

9.1.2 The traditional definition of conservativity

The traditional notion of conservativity can be thought of as a formal statement of aboutness defined for a small but important set of natural language expressions: determiners. It was first extensively discussed by Barwise and Cooper (1981), van Benthem (1983), and Keenan and Stavi (1986) in the framework of then nascent generalized quantifier (GQ) theory, where it functioned as a semantic constraint on possible denotations of natural language determiners.

On the GQ view, noun phrases and (unsaturated) verb phrases denote properties, i.e. subsets of the domain of interpretation E, and determiners can be seen as denoting binary relations between those properties. We provide GQ-style definitions of some determiners below; you can think of the arguments A and B as being the denotations of the NP and VP in a simple sentence of a form [[D NP] VP].

(4) a. $[\![\text{some}]\!](A, B)$ iff $A \cap B \neq \emptyset$
b. $[\![\text{every}]\!](A, B)$ iff $A \subset B$
c. $[\![\text{at least 2}]\!](A, B)$ iff $|A \cap B| \geq 2$
d. $[\![\text{most}]\!](A, B)$ iff $|A \cap B| > |A - B|$

A determiner D is said to be conservative, or more precisely WEAKLY CONSERVATIVE iff

$$[\![\mathrm{D}]\!](A, B) = [\![\mathrm{D}]\!](A, A \cap B).$$

Given a particular determiner, this requirement is easy enough to check. We need to test whether a sentence of form [[D NP] VP] is logically equivalent to (a suitable paraphrase of) [[D NP] [NP and VP]]. For the determiners defined above, this amounts to checking if the pairs of sentences below entail each other, which they clearly do.

(5) a. Some dog is barking. $\Leftrightarrow$ Some dog is a dog which is barking.
 b. Every dog is barking. $\Leftrightarrow$ Every dog is a dog which is barking.
 c. At least two dogs are barking. $\Leftrightarrow$ At least two dogs are dogs which are barking.
 d. Most dogs are barking. $\Leftrightarrow$ Most dogs are dogs which are barking.

To see what a non-conservative determiner would look like, it is actually necessary to invent one. Consider the hypothetical determiner 'smome' below, which is not weakly conservative. To see this, observe that (6a) but not (6b) is true if there is no cat and a dog barks—(6a) isn't really about cats, nor about things that bark, but at best is about the union of those two things.

(6) $[\![\text{smome}]\!](A, B)$ iff $A \cup B \neq \varnothing$
 a. Smome cat is barking. "There is a cat or a barker."
 b. $\not\Leftrightarrow$ Smome cat is a barking cat. "There is a cat or a barking cat."

Semanticists sometimes distinguish weak conservativity (defined above) from STRONG CONSERVATIVITY. Strongly conservative determiners are those which are not only weakly conservative but also "stable under growth of the universe" (van Benthem 1983, p. 453). What is stability? The definition of a determiner in GQ can in principle depend on the size of the domain of interpretation. Saying that $[\![\text{D}]\!]$ is the denotation of D is open ended as it does not tell us which domain we have in mind. To be more precise, we need to define $[\![\text{D}]\!]_E$ for each domain E, but this is often omitted because natural language really only tolerates determiners that are STABLE:

$$A, B \subset E \subset E' \Rightarrow ([\![\text{D}]\!]_E(A, B) = [\![\text{D}]\!]_{E'}(A, B)).$$

To get an intuitive grasp on stability, we consider the truth value of a sentence of a form [[D NP] VP] in different domains. Stability of the determiner guarantees that the truth value of the sentence does not change if the domain grows or shrinks, as long as the changes do not impact the denotations of NP and VP. For example, if some dog (or every dog) is barking in a given situation, then it does not matter if a cat shows up or if we take the dog house away; as long as the set of dogs (and the set of barkers)[3] remains the same, the truth value of the sentence remains intact.

[3] In this particular case, we could have left out "and the set of barkers," i.e. the requirement that the denotation of VP should not change when we test for stability. However, this only works because the example contains a simple intransitive verb, where VP = V. This detail, in tandem with weak

Stability of a determiner guarantees that the determiner itself is never responsible for any cross-domain changes in the truth value of a sentence. Given a sentence of form [[D NP] VP], the truth values it receives on two domains can only differ if the denotation of either NP or VP has changed. Hypothetically, there could be determiners that are weakly conservative but not stable: behold 'smevery', which, we promise, exists in no natural language.

(7) $[\![\text{smevery}]\!]_E(A, B)$ iff $\begin{cases} A \cap B \neq \varnothing & \text{if } |E| = 2 \\ A \subset B & \text{otherwise} \end{cases}$

This hypothetical determiner is like 'every', except when the domain contains exactly two individuals, in which case it takes on the meaning of 'some'. As it is composed of two weakly conservative determiners, 'smevery' is weakly conservative itself. However, it is certainly not stable. Given a domain of two individuals, both dogs but only one of them barking, 'smevery dog is barking' is true (because some dog is barking); add an elephant (which is neither a dog nor a barker), and the sentence becomes false (because it is now relevant that not every dog is barking).

For an example of a hypothetical (and also non-existent) deteminer which violates both weak conservativity and stability, consider Larson and Segal's (1995) example, 'nall', which we introduced in section 3.3.

(8) a. $[\![\text{nall}]\!](A, B)$ iff $(E - A) - B = \varnothing$
b. Nall squares are striped. "Everything except squares is striped."

It is common practice (one which we follow as well) that strong conservativity is referred to simply as CONSERVATIVITY, and it is also common to express its two components (weak conservativity and stability) by the following single requirement: a determiner D is (STRONGLY) CONSERVATIVE iff for every domain E and every $A, B \subset E$,

$$[\![\text{D}]\!]_E(A, B) = [\![\text{D}]\!]_A(A, A \cap B).$$

To test a determiner for weak conservativity, we compared (in (5)) the truth values of sentences [[D NP] VP] and [[D NP] [NP and VP]] *on a single domain*. It is tempting to see the above condition as inviting us to compare the truth value of [[D NP] VP] on domain E and the truth value of [[D NP] [NP and VP]] on domain A, where A is the denotation of NP on E, i.e. $A = [\![\text{NP}]\!]_E$. But that would

conservativity of the determiner, guarantees that the denotation of VP cannot change "in a bad way." We will discuss complex VPs (containing another determiner) in subsection 9.1.3.

be wrong, unless we limit the tests to cases where VP = V, i.e. to sentences with intransitive verbs.

The issue is intimately connected to stability; in fact, footnote 3 refers to the flip side of the same coin. The gist of the problem is that in general, the denotation of a verb phrase on a subdomain ($[\![\mathrm{VP}]\!]_A$) is not necessarily a restriction of its denotation on the original domain ($[\![\mathrm{VP}]\!]_E$) to A, i.e. we sometimes have $[\![\mathrm{VP}]\!]_A \neq A \cap [\![\mathrm{VP}]\!]_E$. This is precisely what happens in (9): whatever domain we construct for (9a) to be true, once we restrict the domain to dogs, there are no bones left for dogs to gnaw, so (9b) is false on that domain. If this was a relevant test, we would need to conclude that 'some' is not conservative! However, the test is not relevant, as (9b) does not equal $[\![\mathrm{D}]\!]_A(A, A \cap B)$ from the right side of the condition on strong conservativity. Clearly, the culprit is the transitive verb, with another determiner in the object position.[4] Limiting attention to sentences with intransitive verbs avoids the problem, as $[\![\mathrm{V}]\!]_A = A \cap [\![\mathrm{V}]\!]_E$ always holds.

(9) a. $[\![\text{Some dog is gnawing a bone.}]\!]_E = \top$
b. $[\![\text{Some dog is a dog gnawing a bone.}]\!]_A =$
$= [\![\text{Some dog is gnawing a bone.}]\!]_A = \bot$, where $A = [\![\text{dog}]\!]_E$

The moral of the story is that if we wish to test a determiner for strong conservativity (or stability), we should *not* use sentences with a transitive verb, but instances of what we call a DETERMINER SCHEMA—a sentence of form [[D NP] V]. This works because on one hand, keeping to intransitive verbs avoids the above problem, while on the other hand, the situation is still general enough to fully investigate the properties of the determiner. The condition on strong conservativity given above requires us to test the determiner for any $B \subset E$, and indeed, this can be achieved by plugging in the denotation of an intransitive verb ($[\![\mathrm{V}]\!]_E$) for B, as an intransitive verb can in principle denote any property.

We now know precisely what conservativity, at least as traditionally understood, amounts to. But why is it important?

Keenan and Stavi (1986, p. 260) advance a very general hypothesis called CONSERVATIVITY UNIVERSAL, which states that all natural language determiners are conservative. To support their hypothesis, Keenan and Stavi grind through a long list of English determiners, both simple and complex, and show that all of them are conservative. Subsequent work on the subject has identified several potential violators of the Conservativity Universal—see Westerståhl (1985) for an early and influential exposition of a potential counterexample, the relative proportional reading of 'many'—but all of them were eventually (more or less satisfactorily) reanalyzed as conservative. All in all, the generalized quantifier

[4] This is also noted by van Benthem (1991, pp. 129–131).

literature seems to indicate that Conservativity Universal should be taken as a valid empirical generalization.

In our opinion, the work on conservativity in the framework of generalized quantifier theory has been enormously fruitful. Much of our own work would be impossible without it. But we have reservations about Conservativity Universal. While we are convinced of its empirical validity—indeed, we shall argue it can be construed even more generally—we don't consider it to be explanatory, and we believe that a more explanatory account can be given through a syntactic characterization in terms of restrictedness. But before we get to restrictedness, we need to engage the following question: Is conservativity a property pertaining only to determiners, or is it more general? For example, might it be a property of the logical forms that determiners give rise to? This is the question that we turn to next.

9.1.3 Generalized Conservativity

Is 'only' conservative?

There is a famous alleged counterexample to the idea that all determiners must be conservative: 'only'. Given the standard definition of conservativity, it is easy to show that 'only' is not conservative—the second sentence of (10) is a tautology, while the truth of the first one is contingent (for example, it can be false if some seals are barking).

(10) a. Only dogs are barking.
b. $\not\Leftrightarrow$ Only dogs are dogs which are barking.

Some authors have argued that 'only' is not a determiner at all, while others have attempted to rescue it by changing the definition of conservativity itself. We believe that both parties to this dispute are partly correct. In the first place we agree with the former group that the distribution of 'only' is simply not what one would expect of a deteminer—even the small sample in (11) illustrates that Herburger (2000) is quite right in categorizing 'only' as "admanythings."

(11) a. Only the dog is gnawing bones.
b. The dog is only gnawing bones.
c. The dog is gnawing only bones.

As for the latter group (the people that want to change the definition of conservativity), we would say that they do not go far enough, for the definition of conservativity needs more than a tweak, it needs to be rethought on a grander scale. The mistake has been in thinking that conservativity is a property of denotations (of determiners). We will see that we arrive at much cleaner picture if we take it to be a property of logical forms instead—and for sure, determiners will give rise to conservative logical forms. But this brings us back to 'only', for it is not merely the exception to the standing theory, it is also our clue as to what the correct picture will look like.

Given the facts in (10), how could anybody claim that 'only' is conservative, or in other words, that (10a) is about dogs? Well, nobody does. What they observe is that (10a) is about barkers, i.e. to evaluate it, we may only consider invididuals who bark. In general (but restricting our attention to 'only' complemented by a focused NP), 'only' would be conservative if we reversed the order of its arguments. Or, to put it differently, while determiners are conservative in their first argument, 'only' is conservative in the second one (12). This is far from surprising, as 'only' is in fact just like 'every', but with the reversed order of arguments (13).[5]

(12) a. Only dogs are barking. $\Leftrightarrow$ Only barking dogs are barking.
b. $[\![\text{only}]\!]_E(A, B) \Leftrightarrow [\![\text{only}]\!]_B(A \cap B, B)$

(13) a. $[\![\text{every}]\!](A, B)$ iff $A \subset B$
b. $[\![\text{only}]\!](A, B)$ iff $B \subset A$

The conservativity with respect to the second argument is called neoconservativity by de Mey (1991); Zuber and Keenan (2019) define 'weak conservativity' as conservativity in either the first or the second argument and propose that Conservativity Universal should in fact refer to this, weakened notion. But in our opinion, this is just playing with definitions. An idea that we find more insightful comes from Herburger (2000), who argues that 'only' only seems non-conservative because of our erroneous assumptions about how it gets its arguments, and even what its arguments are.

We are going to pursue Herburger's line of reasoning in an attempt to show that the natural home of conservativity is not in determiners but in sentential level (and, as we shall argue, sometimes in discourse level) structures. But to get there, we need to examine the behavior of 'only' in finer grain of detail.

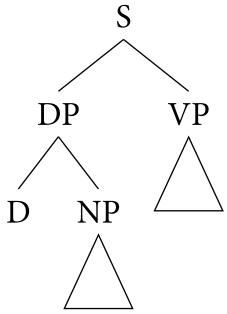
Why do we assume that in a sentence of form [D NP VP], $[\![\text{NP}]\!]$ forms the first argument of $[\![\text{D}]\!]$ and $[\![\text{VP}]\!]$ the second one? Because the order of argument saturation in semantics is assumed to reflect the syntactic structure,[6] and in syntax,

[5] In (13), we ignore the intricate questions about the existential import of both 'every' and 'only'. For discussion, see e.g. Herburger (2000).

[6] We are simplifying the semantics of deteminers when we say they are binary relations. In fact, whenever compositionality is important, the denotations of determiners are assumed to be functions from properties to generalized quantifiers (these are denotations of DPs), which are in turn functions from properties to truth values. Simply put, we should in fact write $[\![\text{D}]\!]([\![\text{NP}]\!])([\![\text{VP}]\!])$ instead of $[\![\text{D}]\!]([\![\text{NP}]\!], [\![\text{VP}]\!])$.

determiners first combine with a noun phrase to form a determiner phrase, which is then merged with the verb (phrase).

(14)



Of course, the null hypothesis is that the syntax–semantics interface is uniform. If determiners are assigned arguments in some way, the same procedure should work for 'only' as well. Given the syntactic structure '[[only dogs] are barking]', we must thus apparently conclude that in (13b), *A* stands for dogs and *B* stands for barking, leading to non-conservativity. Or so it seems.

Here's the catch: at which level of analysis does the syntax–semantics correspondence described above hold? So far, we have naively assumed that semantics matches the *surface* syntactic structure. This is definitely not a safe assumption to make in contemporary syntactic and semantic theories. Within most versions of generative linguistics (including the Minimalist Program), one would normally assume that the input to semantics is the so-called Logical Form (LF).[7] As a first approximation, we can say that LF is produced by taking the surface form (SF) of the sentence and performing various operations on it, like Quantifier Raising.

In the case of determiners, LF is (usually) faithful to the surface syntax. At the very least, LF and SF have the same argument structure. However, matters are different when it comes to 'only', which is one of the operators associated with focus. Assuming that focusing is a syntactic operation (but note that there is certainly no consensus on that), it is clear that it can radically alter the covert syntax of a sentence. For example, Herburger (2000) would argue that in (15), the surface structure (the focused element is underlined) is modified by what she calls *focal mapping*, producing an LF radically different from SF.[8] She also notes that, viewed from the LF perspective, 'only' is then perfectly conservative: given (15b), we correctly predict that the sentence is about barking dogs.

(15) a. [Only [angry dogs]] bark. (surface)
b. [Only=every [dogs bark]] angry dogs bark. (LF)
$\forall x\colon \neg(D(x) \wedge B(x)) \vee ((A(x) \wedge D(x)) \wedge B(x))$

[7] Not to be confused, conceptually, from what we have been calling logical form, although we will in fact argue (in Chapter 12) that, at the end of the day, they are one and the same.
[8] We simplify Herburger's logical forms by disregarding event semantics.

The key idea, expressed in (15b), is that 'only' is really nothing but 'every' in disguise. What is funky about it is that it applies to a structure that has been shuffled by focal mapping. We will have more to say about focus in section 10.1, when we put forth the hypothesis that if we fully integrate the effect of focusing into logical forms, all natural language expressions are conservative. But in this section, we want to further develop the idea that 'only' only seemed problematic because of its peculiar argument structure—we will pursue this idea to its logical conclusion that argument structure of lexical items should play no role in considerations of conservativity.

Liberating conservativity from interface assumptions

The GQ approach makes the erroneous assumption that the argument structure of 'only' is dictated by the surface syntax. Herburger (2000) shows that it is better to assume that the argument structure is read off of LF. However, implicit in her proposal is still the idea that we are dealing with the argument structure of the denotation of some lexical item, and that this denotation is conservative (in the usual, first argument). Can we define conservativity without any reference to argument structure? Of course!

Let's switch to basic predicate logic for a moment, taking 'some' from (16) as an example. The question we want to ask is the following: is (16) about parrots? Obviously, it is. So there is some property of basic predicate logic by virtue of which (16) is conservative with respect to P. Whatever our generalized definition of conservativity turns out to be, we want it to cover not just environments that appear within generalized quantifier theory, but across formal languages. In other words, we want a definition of conservativity that frees it from syntax–semantics interface assumptions. Notice that there is no reference to argument structure in (16), no mention of determiners even, yet we can say that the sentence is about parrots. Our definition of conservativity needs to track that.

(16) Some parrots quack. $\exists x : P(x) \wedge Q(x)$

The following generalized conservativity condition was already implied in the discussion of aboutness: a sentence is P-conservative iff for any context, the truth value of a sentence in that context matches its truth value on the narrowed context consisting solely of Ps. The formal definition is as follows:[9] Given a sentence (closed formula) φ and a (single-argument) predicate P, φ is P-CONSERVATIVE iff

$$[\![\varphi]\!]_E = [\![\varphi]\!]_{[\![P]\!]_E}$$

[9] The definition could be generalized to any formal language, including generalized quantifiers. Later on, we shall extend it to conservativity with respect to several predicates simultaneously.

for any domain E. $[\![P]\!]_E$ is the interpretation of predicate P on domain E. As it is a subset of E, it can serve as a subdomain of E: $[\![\varphi]\!]_{[\![P]\!]_E}$ is thus the interpretation of φ on the subdomain generated by P, i.e. the domain "consisting solely of Ps." $[\![\varphi]\!]_E$ is of course the interpretation of φ on the original domain. Conservativity thus requires that the truth values of φ on the two domains match.[10]

How does this definition connect to the traditional, GQ-based definition $A, B \subset E \Rightarrow [\![\mathrm{D}]\!]_E(A, B) = [\![\mathrm{D}]\!]_A(A, A \cap B)$? In particular, what happened to the intersection $A \cap B$, and even B itself? Let us show you that traditional conservativity is a special case of our generalized conservativity—it is generalized conservativity applied to the determiner schema.

Below, we take the determiner schema S = [[D NP] V] and transform its denotation in the original domain (E) into its denotation on the domain restricted to the denotation of the noun phrase ($[\![\mathrm{NP}]\!]_E$); to easily see how the GQ-based conservativity applies, we define A and B to stand for the denotation (on the original domain) of the noun phrase and the verb, respectively: $A := [\![\mathrm{NP}]\!]_E$, $B := [\![\mathrm{V}]\!]_E$. The crucial steps are the application of GQ conservativity and the subsequent realization that the denotation of the verb on the subdomain ($[\![\mathrm{V}]\!]_A$) is the intersection of the denotation of the noun phrase (A) and the denotation of the verb on the original domain (B). The latter equality ($[\![\mathrm{V}]\!]_A = A \cap B$) holds because the verb is intransitive—we have discussed why the determiner schema must be based on an intransitive verb in subsection 9.1.2—and it is this equality which "hides" the explicit intersection found in the GQ-based definition.

$$
\begin{aligned}
[\![\mathrm{S}]\!]_E &= [\![[[\mathrm{D\ NP}]\ \mathrm{V}]]\!]_E && (\mathrm{S} = [[\mathrm{D\ NP}]\ \mathrm{V}]) \\
&= [\![\mathrm{D}]\!]_E([\![\mathrm{NP}]\!]_E, [\![\mathrm{V}]\!]_E) && (\text{compositionality}) \\
&= [\![\mathrm{D}]\!]_E(A, B) && (A := [\![\mathrm{NP}]\!]_E,\ B := [\![\mathrm{V}]\!]_E) \\
&= [\![\mathrm{D}]\!]_A(A, A \cap B) && (\text{GQ conservativity}) \\
&= [\![\mathrm{D}]\!]_A([\![\mathrm{NP}]\!]_A, [\![\mathrm{V}]\!]_A) && ([\![\mathrm{NP}]\!]_A = A,\ [\![\mathrm{V}]\!]_A = A \cap [\![\mathrm{V}]\!]_E = A \cap B) \\
&= [\![[[\mathrm{D\ NP}]\ \mathrm{V}]]\!]_A && (\text{compositionality}) \\
&= [\![\mathrm{S}]\!]_A && (\mathrm{S} = [[\mathrm{D\ NP}]\ \mathrm{V}]) \\
&= [\![\mathrm{S}]\!]_{[\![\mathrm{NP}]\!]_E} && (A = [\![\mathrm{NP}]\!]_E)
\end{aligned}
$$

We see that $[\![\mathrm{S}]\!]_E = [\![\mathrm{S}]\!]_{[\![\mathrm{NP}]\!]_E}$ is clearly a special case of our definition of P-conservativity, $[\![\varphi]\!]_E = [\![\varphi]\!]_{[\![P]\!]_E}$. Crucially, we can conclude that mixing the argument structure and sets into the GQ definition of conservativity makes this

[10] We should really be talking about models here, not domains. The precise formulation of conservativity can be found in section A.4.

definition unnecessarily complicated. Conservativity is a much simpler and also a much more general phenomenon.

Observe, however, that while conservativity in GQ theory is a special case of our generalized conservativity applied to determiner schema, the two definitions are not completely equivalent even in this special case. For one, our definition is more robust, as we can consider not just the conservativity with respect to the entire first argument (NP) of the determiner, but also with respect to the components of the complementing noun phrase—in particular, the noun itself. For example, in (17) below, we can ask whether the sentence is 'dog'-conservative (yes, it is) and 'angry'-conservative (again, yes). These questions have no analogue under the traditional understanding of conservativity, which can only consider the NP as a whole.

(17) Some angry dog is barking. $\exists x\colon (A(x) \wedge D(x)) \wedge B(x)$

On the other hand, we can consider the generalized conservativity with respect to the entire complement of the determiner in two ways. One, already mentioned above, is by testing a sentence of form [[D N] V], where NP = N so that the "first argument" of D corresponds to an atomic formula. This is what we did for (16). Second, we can combine the tests of the components. As (17) is both 'angry'-conservative and 'dog'-conservative, any domain may be restricted to either angry things or dogs. So we may take the original domain and restrict it to dogs, and then further restrict the resulting subdomain to angry things (remember, *P*-conservativity implies that *any* domain may be restricted to Ps), in effect restricting the original domain to the intersection of both predicates' interpretations, the angry dogs of the original domain.

We believe it is fair to say that our definition of conservativity is free of syntax–semantics interface assumptions. The notion is not defined with respect to argument structure (either implicitly or explicitly) as is the case with the GQ definition, but relative to a certain predicate—or, as we shall see shortly, with respect to a set of predicates. Conservativity does not care how we arrived at a particular logical form—assuming that the sentences in (18) receive the same logical form, this logical form is characterized as conservative for both of them.

(18) Every A is a B. / Only Bs are As. $\forall x\colon \neg A(x) \vee B(x)$

Another pleasing property of generalized conservativity is *independence of form*. Generalized conservativity cannot distinguish between logically equivalent formulas—and this is as it should be, because conservativity is a semantic notion, or a model-theoretic notion if you wish. For example, both formulas in (19) are characterized as conservative (with respect to *A*), regardless of the different order of disjuncts—such differences will only become important once we start thinking about how natural language syntax might feed semantics (we will talk a bit about this in subsection 9.3.2, and then extensively in Chapter 12).

(19) Every A is a B.
  a. $\forall x\colon \neg A(x) \vee B(x)$
  b. $\forall x\colon B(x) \vee \neg A(x)$

Summing up, our generalized definition of conservativity liberates this notion from syntax–semantics interface assumptions. We do not define conservativity with respect to argument structure (either implicitly or explicitly), as is the case with the GQ definition, but relative to a certain predicate—or, as we shall see next, with respect to a collection of predicates.

Conservativity with respect to multiple predicates

So far, we have been dealing with determiner schemata—simple sentences containing an intransitive verb. What happens once we move to a larger fragment of the language allowing transitive verbs? For example, it is easy to see that (20) is neither D-conservative (i.e. 'dog'-conservative) nor B-conservative (i.e. 'bone'-conservative).

(20) Some dog is gnawing a bone. $\exists x\colon D(x) \wedge \exists y\colon B(x) \wedge G(x, y)$

Suppose this sentence is true given some domain. As we already mentioned in the previous section, if we narrow down the domain to consist only of dogs, there are no bones left for dogs to gnaw! Similarly, restricting the domain to bones leaves no one to gnaw them. However, if we retain both dogs and bones, the sentence remains true. Therefore, even though (20) is conservative with respect to neither dogs nor bones, it is *jointly* conservative with respect to dogs and bones. In short, we will say it is {'dog', 'bone'}-conservative.

How can we formally capture the intuition behind joint conservativity? We can use the set-theoretic concept of *union*. When we have a $\{D, B\}$-conservative sentence such as (20) above, we can "safely" restrict the domain to $D \cup B$, the union of D and B. In other words, {'dog', 'bone'}-conservativity of a sentence tells us that its truth value on any domain will match its truth value on the subdomain consisting of the union of the denotations of nouns 'dog' and 'bone'.

In general, if we have some set of predicates κ ("kappa") and we know that a certain sentence is κ-conservative (for example, $\{D, B\}$-conservative), we may safely restrict the domain to the union of the denotations of predicates we find in κ. Formally, we shall say that, given a set of predicates κ, φ is κ-CONSERVATIVE iff

$$[\![\varphi]\!]_E = [\![\varphi]\!]_{\bigcup_{P\in\kappa}[\![P]\!]_E}$$

for any domain E, where $\bigcup_{P\in\kappa}[\![P]\!]_E$ is the union of interpretations (on E) of all predicates in κ.

Earlier in the section, we had an example of a sentence which was both A-conservative and D-conservative; we repeat it in (21) below. Is this sentence $\{A, D\}$-conservative? Well, yes, but that does not mean that a sentence being A-conservative and D-conservative is *equivalent* to it being $\{A, D\}$-conservative. Example (20) taught us that the right-to-left direction of this hypothetical equivalence does not necessarily hold.

(21) Some angry dog is barking. $\exists x\colon (A(x) \wedge D(x)) \wedge B(x)$

The left-to-right direction, however, always holds. In fact, an even stronger implication is valid: if a sentence is P-conservative (for some predicate P), it is $\{P, Q\}$-conservative as well, for any predicate Q. This is a trivial consequence of stability, which is a component of (strong) conservativity. For example, (21) is D-conservative: restricting the (original) domain (E) to dogs (D) retains the truth value. What about restricting the original domain to the union of dogs and parrots ($D \cup P$)? Well, this restriction is at the same time an extension of the subdomain consisting solely of dogs (D). By stability, the truth value is therefore preserved when moving from D to $D \cup P$. In plain English, when a sentence is 'dog'-conservative, parrots are irrelevant anyway, so they may or may not be present. In general, if a sentence is κ-conservative (remember that κ is a set of predicates), it is also conservative with respect to any *super*set of κ. Formally:

$$\kappa \subset \kappa' \Rightarrow (\kappa\text{-conservative} \Rightarrow \kappa'\text{-conservative})$$

Once we start considering logical forms of sentences over and above determiner schemata, joint conservativity becomes the norm rather than an exception. Actually, this is where it becomes patently obvious that conservativity is in fact the formal statement of aboutness.[11]

This section was dedicated to generalizing the notion of conservativity familiar from generalized quantifier theory. We got rid of the dependency on determiners and their argument structure. This resulted in a simpler, and at the same time more general statement of conservativity, relativized to a predicate. As the resulting

[11] The fact that conservativity is preserved when "going to a superset" (but not to a subset!) of κ actually implies that conservativity is not *precisely* the same notion as aboutness. Aboutness, as a pre-theoretical notion, has a sort of an exactness built-in, at least in our opinion. While we have no quarrel with the statement that (21) is conservative with respect to dogs and whales (after all, who are we to fight the math), we would not want to say that this sentence is *about* dogs and whales. To truly formalize aboutness, we would therefore need to define minimal conservativity along the following lines: a sentence is MINIMALLY κ-CONSERVATIVE iff it is κ-conservative and it is not κ'-conservative for any $\kappa' \subset \kappa$.

notion is a property of sentences rather than determiners, it was natural to further extend it to joint (or multi-predicate) conservativity.[12]

We will later explore the empirical consequences of generalizing conservativity. In particular, we will argue that Conservativity Universal—the claim that all natural language determiners are conservative—might be strengthened into an even stronger empirical hypothesis that all logical forms are conservative, if we only assume that logical forms integrate some aspects of discourse representations. But that discussion is really about natural language, so we postpone it to Chapter 10. In the next section, we will see how our notion of generalized conservativity has a direct counterpart in syntax—the notion of restrictedness. And as in the case of the semantics of determiners, we will see that restrictedness will not be a local property of determiners, but will be a more general property of logical forms.

9.2 Restrictedness: The Syntactic Counterpart of Conservativity

In this section we turn to the task of developing a theory of the syntactic counterpart of conservativity—restrictedness. We will see that there are several such syntactic characterizations available (this is possible because the characterizations are all modulo logical equivalence), but we will argue that in the context of natural language, the most appropriate notion of restrictedness is (surprisingly) based on p-scope! In other words, our friend Perceval and p-scope are going to give an important encore here.

So far throughout this book, we have primarily used UNRESTRICTED QUANTIFIERS. They are unrestricted in the sense that they quantify over *all* the objects in the domain of interpretation: $\exists x\, \varphi$ is true iff some object in the domain satisfies φ; $\forall x\, \varphi$ is true iff every object in the domain satisfies φ. Formula φ is called the SCOPE of the quantifier.

RESTRICTED QUANTIFIERS, on the other hand, restrict the range of quantification to a subset of the domain of interpretation. These quantifiers are binary: the RESTRICTION specifies which subset of the domain of interpretation to quantify over; the NUCLEAR SCOPE expresses the proposition that holds for some/every object in the range of quantification. Although it is possible to define the semantics of restricted quantifiers directly, they are most often defined using the following abbreviations:[13]

[12] It is also possible to generalize the notion of conservativity so that it allows for non-monadic predicates as members of κ—more precisely, conservativity must then be defined with respect to argument positions of predicates—but we will not explore this option in the book.

[13] We define the restricted quantifiers using the wide-scope variant of the unrestricted quantifiers (note the colon), as this variant is more useful in logical forms.

(22) Restricted quantifiers: Qx [restriction] nuclear scope

a. $\exists x\,[\varphi]\,\psi \;:\equiv \exists x\colon \varphi \wedge \psi$ (existential)
"ψ holds for some element of the domain for which φ holds."

b. $\forall x\,[\varphi]\,\psi \;:\equiv \forall x\colon \neg\varphi \vee \psi$ (universal)
"ψ holds for every element of the domain for which φ holds."

Let's take a look at how the above abbreviations work. For the $\exists x\,[\varphi]\,\psi$ to be true, some object in the domain must satisfy both φ and ψ. In particular, it must satisfy φ, so if it doesn't, we can simply disregard ψ. In effect, we are restricting our attention to the objects satisfying φ. Similarly for $\forall x\,[\varphi]\,\psi$, ψ is only relevant when $\neg\varphi$ is false, otherwise $\neg\varphi$ itself guarantees the truth of the disjunction. In effect, our attention may be restricted to objects satisfying φ.

This elementary description already hints that restrictedness and conservativity are related. After all, restrictedness says we can safely restrict our attention to objects satisfying a certain condition, while conservativity says that the truth conditions will not change if the domain changes in a way which does not affect objects satisfying a certain condition.

Restricted quantification is ubiquitous in work on natural language semantics that employs some form of predicate logic. In such literature, we find restricted quantification not only over individuals, but also over events, times, possible worlds, etc. Many authors use the restricted notation itself, but even when the unrestricted format is used, it is most often clear that the proposed logical forms could be translated into the restricted format. We of course believe we have a very good idea as to why this state of affairs obtains. Restrictedness and conservativity are respectively the syntactic and the semantic side of the same coin. In fact, the widespread use of restricted quantification indirectly argues in favor of our idea from the previous section that Conservativity Universal should be assumed to be generally true, and not limited to determiners as in generalized quantifier theory.

Defining a RESTRICTED FORMULA as a formula which is built using only restricted quantification—regardless of whether it is actually written down using only the restricted quantifiers from (22) or is simply of the form which could be abbreviated into that format—our claim in this section is that all restricted formulas are conservative, and all conservative formulas are restricted. This statement, however, is only the first approximation. On the one hand, we will see that for the restricted-to-conservative direction to hold, the restriction of a quantifier cannot be of any form; using only the abbreviations above is a necessary but not a sufficient condition for the resulting formula to be conservative. For example, given a quantifier over x, the restricted quantification only makes sense if x is a free variable of the restriction. On the other hand, we will see that a formula can be conservative although it uses unrestricted quantification: here, the catch is that we can always find a restricted formula logically equivalent to the original formula.

In set theory, restricted quantification takes the forms shown below. Consider (23): A in $Qx \in A$ should be understood as a set, and the formulas read as "for some/every member of the set A, ψ holds." The pure predicate logic counterpart, shown on the right-hand side in (23), arises if we take A to stand for a monadic (i.e. one-place) predicate. We call this ATOMIC RESTRICTEDNESS: a restricted quantifier over x is atomically A-restricted iff its restriction is an atomic formula built of the monadic predicate A with argument x.[14]

(23) Restrictedness in set theory—atomic restrictedness
a. $\exists x \in A : \psi(x)$—$\exists x\,[A(x)]\,\psi(x)$
b. $\forall x \in A : \psi(x)$—$\forall x\,[A(x)]\,\psi(x)$

The above definition is aimed toward occurrences of quantifiers, but we really want to define restrictedness for formulas. These will be restricted with respect to a set of predicates rather than a single predicate: given a formula φ and a set of predicates κ, φ is κ-restricted iff every quantifier in φ is P-restricted for some $P \in \kappa$. The definition is meant for both the atomic restrictedness defined above and the other variants of the notion to be defined later. We furthermore omit the "κ-" prefix when the identity of the restricting predicates is irrelevant: a formula is *restricted* iff there is a set of predicates κ such that the formula is κ-restricted.

It is fairly easy to prove that any atomically κ-restricted formula is κ-conservative. The other direction of the statement—every κ-conservative formula is atomically κ-restricted—is much harder to prove, and furthermore requires an important qualification. Formula (24a) is A-conservative but not A-restricted, as the topmost connective in the scope of $\exists x$ is a disjunction rather that the conjunction required by (22). However, it is clear what is wrong with (24a). The second disjunct, being a contradiction (and thus always false), has no effect on truth-conditions. Deleting it produces the logically equivalent (24b), which *is* A-restricted.

(24) a. $\exists x : (A(x) \wedge B(x)) \vee (A(x) \wedge \neg A(x))$
b. $\sim \exists x : A(x) \wedge B(x) \equiv \exists x\,[A(x)]\,B(x)$

In more general terms, conservativity is a semantic property and is as such shared by logically equivalent formulas. Restrictedness, on the other hand, is a syntactic property and, as the above example shows, it is trivial to transform a

[14] Notice that strictly speaking, the nuclear scope ψ does not need to contain any free occurrence of x for the quantifier to count as restricted, i.e. the formula is conservative even in this case. However, this is quite a pathological situation, which we believe never obtains in natural language—in fact, the rule of Restricted Closure we will propose in section 9.3 will have a variant ruling out nuclear scopes without an occurrence of the bound variable.

restricted formula into a logically equivalent non-restricted formula. The claim that restrictedness equals conservativity must therefore be understood modulo logical equivalence, as expressed in the following theorem.

(25) Restrictedness Theorem:
Let κ be a set of predicate symbols and φ an arbitrary sentence. Then φ is κ-conservative if and only if it is logically equivalent to some κ-restricted sentence.

You may have noticed that we don't mention *atomic* restrictedness in the theorem. Why is that? Later on, we will define two further species of restrictedness, strong and weak restrictedness, and perhaps surprisingly, Restrictedness Theorem holds for all of them. It is worth spending a minute to understand how this can be, given that restrictedness is supposed to be the syntactic *characterization* of conservativity.

We have actually already discussed the key feature of the restrictedness–conservativity correspondence that makes this state of affairs possible: a random conservative sentence is not necessarily restricted, but it is *logically equivalent* to an atomically restricted sentence. The other piece of the answer to the above puzzle is that our variants of restrictedness characterize progressively larger sets of restricted formulas: atomically restricted formulas are a subset of strongly restricted formulas, which are in turn a subset of weakly restricted formulas, as illustrated in Figure 9.1.

There are two directions to the Restrictedness Theorem. To prove the syntax-to-semantics direction, it is enough to concentrate on the largest class of restricted formulas: if we prove that weakly restricted formulas are conservative, we thereby prove the same statement for atomically and strongly restricted formulas as well. On the other hand, proving that for any conservative formula, we can find a logically equivalent atomically restricted formula (the arrow in Figure 9.1), automatically proves the semantics-to-syntax direction for the larger classes of strongly and weakly restricted formulas as well. A way to interpret this is to say

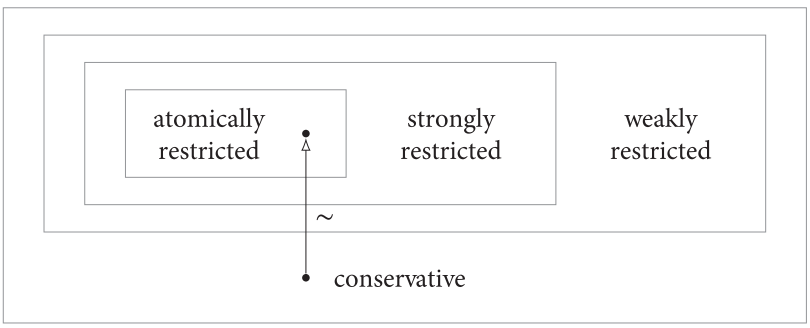


Figure 9.1 Variants of restrictedness

that atomically restricted formulas alone have all the expressive power required by conservativity. If we were only interested in mathematics, we would perhaps need no further classes of restrictedness. As we shall see, what drives us to invent strong restrictedness is that we want to automatically recognize various classes of linguistically important expressions as restricted and thus conservative; weak restrictedness, defined in the Appendix, is more of an exercise in mathematics—an attempt to mechanically recognize as conservative as large a class of formulas as possible.

We present the proof of Restrictedness Theorem in section A.4 of the Appendix. We will prove the simpler, syntax-to-semantics direction directly by induction of complexity. The (much) harder semantics-to-syntax direction will follow from some hard-core mathematical results of Feferman (1968). These results have been around for a while, and while we might be the first to explicitly show how they imply the restrictedness = conservativity equation, we are not the first to notice they do. Van Benthem (1984, p. 462, 1986, p. 38) attributes the observation to Kit Fine.

Enough technical details. What we are really interested in is whether the restricted format offers enough freedom to build the logical forms of natural language expressions—in particular the logical forms of simple sentences containing various determiners. From this perspective, it is clear that atomic restrictedness is inadequate for the job. It only works for the square of opposition (26). In (27), we provide the logical forms of some other determiners from section 5.3 in the restricted format. The restrictions of quantifiers in these formulas clearly include complex, non-atomic formulas. In fact, the same problem occurs with the square of opposition when the subject is a complex expression (28).

(26) a. An alien is a bird. $\exists x\,[A(x)]\,B(x)$
b. Every alien is a bird. $\forall x\,[A(x)]\,B(x)$
c. An alien is not a bird. $\exists x\,[A(x)]\,\neg B(x)$
d. No alien is a bird. $\forall x\,[A(x)]\,\neg B(x)$

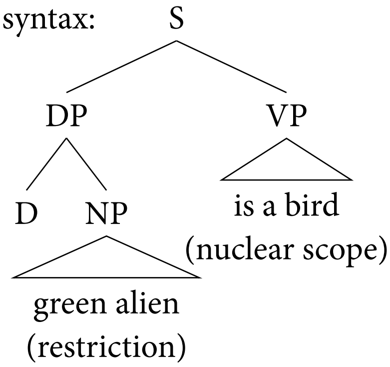
(27) a. The alien is a bird. / The aliens are birds. $\exists x\,\big[D(x) \wedge \forall y\,\big[D(y)\big]\,y \circ x\big]\,B(x)$
b. Five aliens are birds. $\exists x\,[D(x) \wedge \exists n\,[5(n)]\,\#(n,x)]\,B(x)$
c. Most aliens are birds.
$$\exists x\,\big[D(x) \wedge \exists n\,[\#(n,x)]\,\neg\exists y\,\big[D(y) \wedge \#(n,y)\big]\,y \wr x\big]\,B(x)$$

(28) a. A green alien is a bird. $\exists x\,[G(x) \wedge A(x)]\,B(x)$
b. Every green alien is a bird. $\forall x\,[G(x) \wedge A(x)]\,B(x)$

We emphasize that the problem is not that we cannot provide atomically restricted logical forms for sentences in (27) and (28). The semantics-to-syntax direction of the Restrictedness Theorem guarantees that we can. For example, we could ascribe the A-restricted logical forms in (29) to the sentences with complex subjects. This would suffice if our only goal was to get the truth conditions right, but we have higher aims. We want our logical forms to reflect, at least roughly, the syntax of natural language expressions. At the very least, we want the content of a quantifier's restriction to correspond to the content of the DP, as indicated in (30). (28) is a better match to the syntax than (29) because both 'green' $G(x)$ and 'alien' $A(x)$ occur in the restriction.

(29) a. A green alien is a bird. $\exists x\,[A(x)]\,G(x) \wedge B(x)$
b. Every green alien is a bird. $\forall x\,[A(x)]\,G(x) \wedge B(x)$

(30) a. syntax:



b. semantics: $Qx\,[\text{restriction}]$ nuclear scope

The question is, how can we characterize restrictedness while at the same time respect the structure of natural language constructions. Here (finally!) we revisit p-scope and our old friend Perceval.

9.2.1 Restrictedness based on p-scope

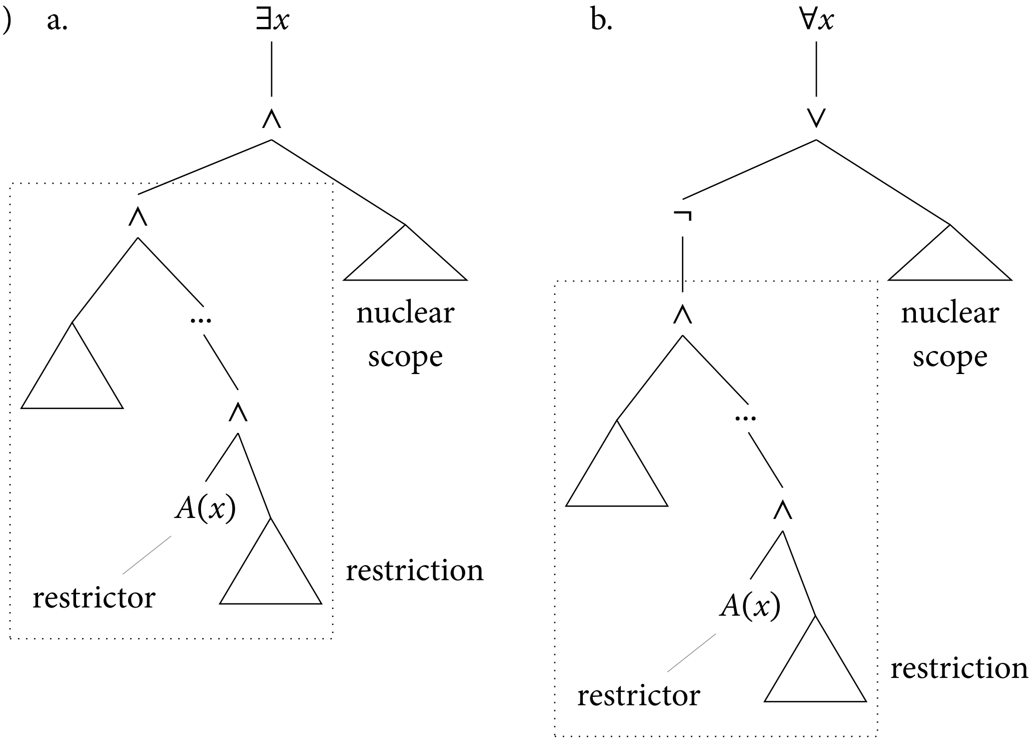
With our goal of fidelity to natural language in mind we need the right account of restrictedness. Perhaps surprisingly (even to us), the ideal characterization falls out immediately from the technology we developed for our Dynamic Deductive System. Using the notion of p-scope we can define restrictedness as follows.

(31) A quantifier over x is STRONGLY A-RESTRICTED (where A is a monadic predicate) iff
   (a) it is of form $\exists x\colon \varphi \wedge \psi$ or $\forall x\colon \neg\varphi \vee \psi$ for some φ (the restriction) and ψ (the nuclear scope), and
   (b) an occurrence of $A(x)$ in the restriction *positively* p-scopes over the nuclear scope. We will call such occurence of $A(x)$ a RESTRICTOR of the quantifier.

As illustrated in (32) below, a restrict*or* (an atomic formula, above: $A(x)$) should not be confused with the restrict*ion* (above: φ). A restrictor is always a constituent of the restriction.[15] The atomically restricted quantifiers can be characterized as the cases where the restriction consist only of the restrictor. Also note that in general, a quantifier may have more than one restrictor—it can happen that several atomic formulas p-scope over the nuclear scope.

The characterization of restrictedness in terms of p-scope covers not only the examples in (27) and (28), but all the determiners discussed in section 5.3. The quantifiers occurring in their logical forms are either of the existential or of the universal form shown in (32) (the dots may be only conjunctions)—this includes even the quantifiers embedded in the restriction of another quantifier, such as quantifiers over y and n in (27),[16] and quantifiers introducing the comparison class of 'more'.

(32) a. b.



To see that a structure of form (32a) or (32b) is A-restricted, we need to consider the p-scoping properties of $A(x)$ in these trees. It is easy to see that $A(x)$ positively p-

[15] In principle, we could have also allowed restrictors occurring within the nuclear scope and positively p-scoping over the restriction, effectively switching the roles of φ and ψ in the unabbreviations $\exists x\colon \varphi \wedge \psi$ and $\forall x\colon \neg\varphi \vee \psi$. After all, $\exists x\colon A(x) \wedge B(x)$ is B-conservative just as well as A-conservative, for example. We decide to limit the restrictors to φ (i.e. to the left side of the junction) to keep the matters closer to the traditional definition of restricted quantifiers. Furthermore, this limitation will be important in subsection 9.3.2.

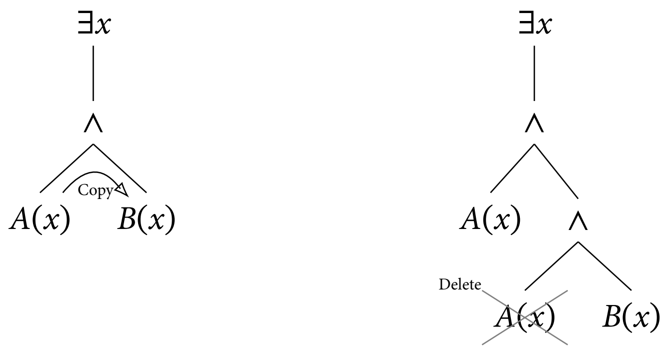
[16] Strictly speaking, the definition in (31) does not cover all the examples. While $\#(n, x)$ in (27c) restricts n, it does so only indirectly via x, because # is a dyadic predicate. As mentioned in footnote 12, we leave the issue of non-monadic restrictors to further research. Furthermore, numeric variables could easily turn out to be restricted to (natural) numbers anyway.

scopes over the nuclear scope of the restricted quantifier (the triangle on the right). If you remember our allegory, this will be the case if Perceval, hitting the road at $A(x)$ donning the golden armor, can reach the topmost junction, i.e. the node just below the quantifier at the root. In (32a), this is clearly the case as Perceval only enters Angel towns (conjunctions); in (32b), the golden armor gets him up to the negation, where the armor is transmuted to silver, so he may enter the disjunction just below the quantifier.

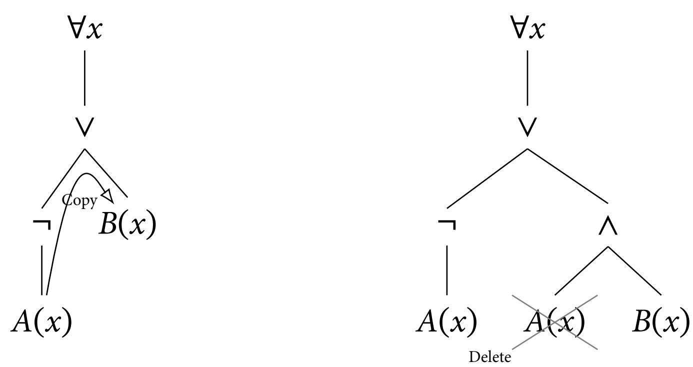
We emphasize that starting with the golden armor works for both existential and universal restricted quantifiers, i.e. the p-scope of the restrictor $A(x)$ over the nuclear scope is positive in both cases. Furthermore note that the restrictor p-scopes into the nuclear scope even if the structures in (32) are embedded in some formula, and even irrespectively of the polarity of the embedding—the p-scope relation between the restrictor and the nuclear scope is one of relative p-scope, which is computed locally.

The p-scope-based characterization of restrictedness immediately explains the utility of the [[D NP] VP] ~ [[D NP] [NP and VP]] test. As the restrictor always positively p-scopes over the nuclear scope, we may Copy it there to get from (33a) to (33b); for the reverse direction, we only need to Delete it. The same procedure works for (34), even if the structure is different.[17]

(33) a. Some alien is a bird. b. Some alien is an alien and a bird.



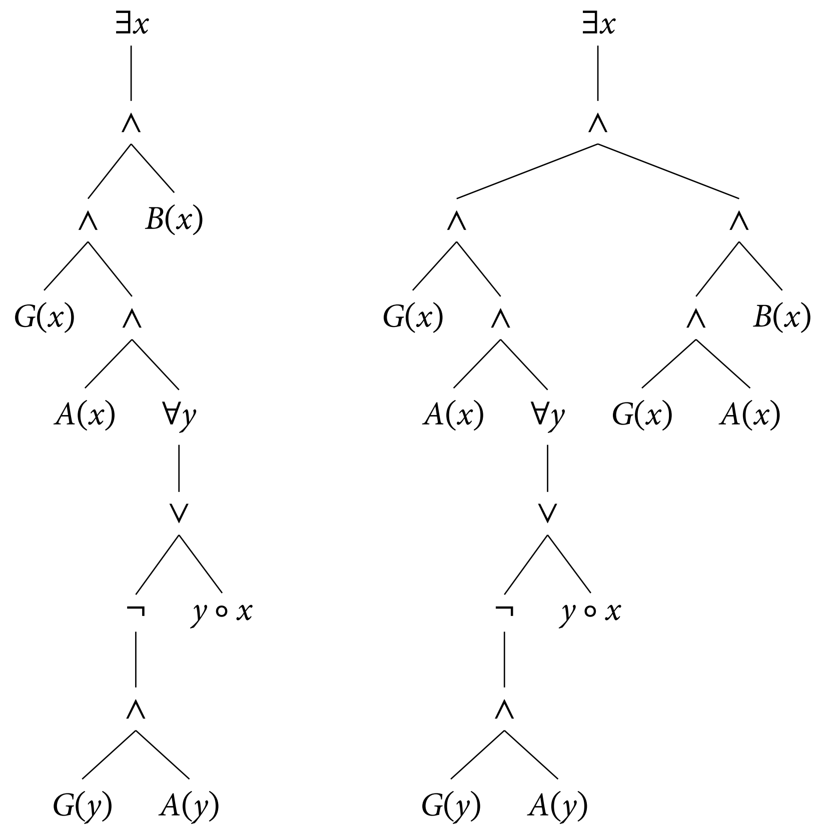
(34) a. Every alien is a bird. b. Every alien is an alien and a bird.



[17] When the restricted quantifier has negative polarity, these equivalences are derived using Add and Prune.

For determiners with complex restrictions, the left-to-right procedure is a little bit more involved, and depends on what precisely we want to Copy. In the example below, we decide to Copy 'green alien', but we could perform the test by Copying only 'alien' as well. To get $G(x) \wedge A(x)$ into the nuclear scope in (35), we have at least two options. We can either Copy the entire restriction, as before, and Delete whatever we don't need (below, $\forall y\,[D(y)]\, y \circ x$); or we can Copy $G(x)$ and $A(x)$ independently.

(35) a. The green alien is a bird. b. … is a green alien and a bird.



Summing up, strong restrictedness is a syntactic characterization of conservativity, and it is defined in terms of the relation of p-scope, a key ingredient of our Dynamic Deductive System. This variant of restrictedness successfully recognizes all determiners as restricted and thus conservative, and transparently justifies the applicability of the standard test for conservativity. In the next subsection, we will explore the restrictedness of logical forms arising from lexical items other than determiners. But before we do that, let's illuminate some further correlations between p-scope and the structure of strongly restricted formulas.

Perhaps the most striking connection between p-scope and restrictedness is that almost nothing can p-scope out of a restricted quantifier. This is of course due to the affinity of $\exists$ to $\wedge$ and $\forall$ to $\vee$ in restricted quantification. Restricted quantifiers are "composed" of an Angel quantifier and a Demon junction, or vice versa. No formula below the restricted quantifier's junction can ever p-scope out of the quantifier, as Perceval cannot get over the sequence of an Angel and a Demon node (or the reverse). One could say that restricted quantifiers are "islands" for p-scope.

There is a single exception to the p-scope island property of restricted quantifiers. The only node below the quantifier which can "escape" from the p-scope island is the restricted quantifier's junction itself. The type of the junction is irrelevant when Perceval sets out on his journey from that very node. This exception is extremely important and we have seen it at work in many DDS deductions in the preceding chapters.

Look at the tree (38), illustrating the first step of deduction (37). The tree illustrates both that a restricted quantifier is a p-scope island and that the quantifier's junction is an exception. To see that $C(y)$ does not p-scope over $B(x)$, remember that Perceval should start with the silver armor if he is to get over the disjunction above $C(y)$, but this silver armor will then prevent him from continuing to the universal quantifier $\forall y$. On the other hand, $\neg B(x) \lor C(y)$ can "escape" the island: Perceval must start with golden armor, which gets him both through $\forall y$ and the conjunction above it.

(36) a. Some alien is a bird. $\exists x\colon A(x) \land B(x)$
b. Every bird is chirping. $\forall y\colon \neg B(y) \lor C(y)$
c. ∴ Some alien is chirping. $\exists x\colon A(x) \land C(x)$

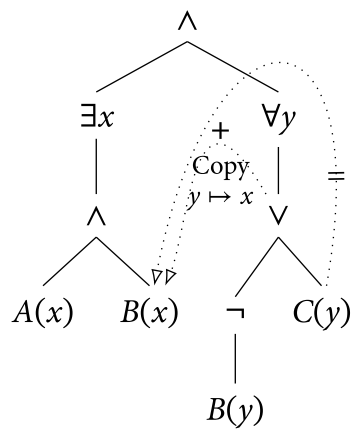
(37) 1. $\exists x\,(A(x) \land \underbrace{B(x)}_{\text{COPY}}) \land \forall y\,(\underbrace{\neg B(y) \lor C(y)}_{y \mapsto x})$

2. $\exists x\,(A(x) \land (B(x) \land (\neg B(x) \lor C(x)))) \land \ldots$ PRUNE

3. $\exists x\,(A(x) \land \underbrace{(B(x) \land C(x))}_{\text{DELETE}}) \land \ldots$

4. $\exists x\,(A(x) \land C(x)) \land \ldots$

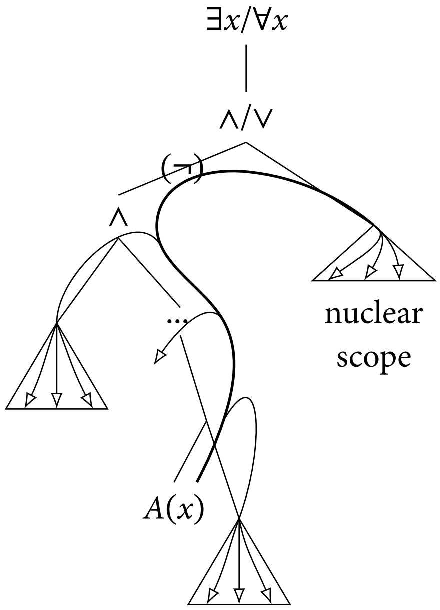
(38)



Incidentally, our Dynamic Deductive System can be deployed on a restricted language (i.e. a language which only allows restricted formulas) just as well as on a general language—in fact, most of our examples in Chapters 6 and 7 were restricted!—as long as we only apply a rule when it produces a restricted formula.

The following property will be very important in section 9.3: a restrictor $A(x)$ p-scopes over all occurrences of the variable x bound by the restricted quantifier. Let us see how it comes about. For one, the restrictor p-scopes over itself, as always.[18,19] But more interestingly, as strong restrictedness requires the restrictor to p-scope over the nuclear scope of the restricted quantifier, and as Perceval can freely travel South, $A(x)$ will p-scope over any variable in the nuclear scope. Furthermore, note that Perceval can decide to turn South not only at the junction just below the restricted quantifier, but anywhere on his way North. As shown below, the p-scope relation between $A(x)$ and any other atomic formula containing x (bound by the restricted quantifier) can "break off" the "outermost" p-scope relation, as shown below. Finally, as the p-scope polarity within the relative p-scope domain is uniform, all these p-scope relations (except maybe the p-scope over the restrictor itself) have positive polarity.

(39)



Finally, let us discuss a property which turns out to be crucial in subsection 9.3.2. The requirement that a restrictor *positively* p-scopes over the nuclear scope of

[18] The polarity of this p-scope can be either positive or negative. As p-scope over oneself is an instance of descendant p-scope, it depends not only on the structure of the subtree rooted in the restricted quantifier, but also on the number of negations above it.

[19] We will discuss the somewhat special case of a quantifier binding a single occurrence of its variable in section 9.3. Note that the prototypical example of such situation, $\exists x\, A(x)$, although conservative, is not subsumed under our definition of strong restrictedness. (But the logically equivalent $\exists x\colon A(x) \wedge A(x)$ is, so this does not constitite a counterexample to the Restrictedness Theorem.)

the restricted quantifier forces it to have positive polarity within an existential restricted quantifier, and negative polarity within a universal restricted quantifier. (We emphasize that we are talking about constituent polarity *within the quantifier node* here, not the absolute constituent polarity within the entire formula.) This property is not surprising at all if we consider that strong restrictedness is a generalization of atomic restrictedness. Let us see how it comes about. If Perceval can start out with the golden armor at the restrictor and enter the (Angel) conjunction below the existential quantifier, then there must have been an even number of Alchemists between the restrictor and that conjuncion—see the path from $A(x)$ to $\wedge$ below $\exists x$ in (38). Similarly, if he can start out with the golden armor at the restrictor and enter the (Demon) disjunction below the universal quantifier, then there must have been an odd number of Alchemists between the restrictor and that disjunction—see the path from $B(y)$ to $\vee$ below $\forall y$ in (38).

Our next step will be to extend these ideas beyond basic determiner positions and think about restrictedness in more global terms, but before we go there, it is already possible to make an important observation. The Medieval logicians noted that you can get a lot of mileage out of paying attention to the polarity of environments in which a term occurs. We are in the process of generalizing this idea, not just to the notion of inference, where we saw the role of polarity in p-scope, which in turn drove our Dynamic Deductive System, but now we can see that p-scope is going to play a foundational role in our understanding of restrictedness, and hence conservativity. But of course this is also the ingredient that is going help us to show how quantification is intertwined with polarity (an idea we will get to shortly). But before we get there, we want to show one of the more amazing features of restrictedness—its applicability to structures that are much larger than simple determiner positions. That is, it can apply over and above determiners.

9.2.2 Restrictedness over and above determiners

So far in this section, we have been focusing on the traditional locus of conservativity, the argument positions of determiner phrases. Our p-scope-based notion of strong restrictedness correctly and naturally gives an account of restrictedness that cashes out the notion of conservativity in these argument positions. However, in subsection 9.1.3, where we introduced generalized conservativity, we saw that, given proper syntactic and semantic analysis, some logical forms arising in the presence of ‘only’ can be seen as conservative as well, *if* we test the logical form for conservativity after it is affected by focusing. The question naturally arises as to whether our account of restrictedness extends to ‘only’ as well. We will see that it does.

In subsection 9.1.3, we followed Herburger's (2000) assumption that 'only' is really nothing but a universal (restricted) quantifier whose restriction and nuclear scope were scrambled by the process of focal mapping. By taking the argument structure of 'only' out of the equation, we were able to characterize (40c) as B-conservative. The question now is, can we recognize its conservativity by formal means, i.e. by seeing that it is (strongly) restricted? Definitely: (40c) is B-restricted, because $B(x)$ positively p-scopes into the nuclear scope of $\forall x$.

(40) a. [Only [a dog]] is barking. (surface)
b. [Only=every [is barking]] a dog is barking. (LF)
c. $\forall x\colon \neg B(x) \vee (D(x) \wedge B(x))$

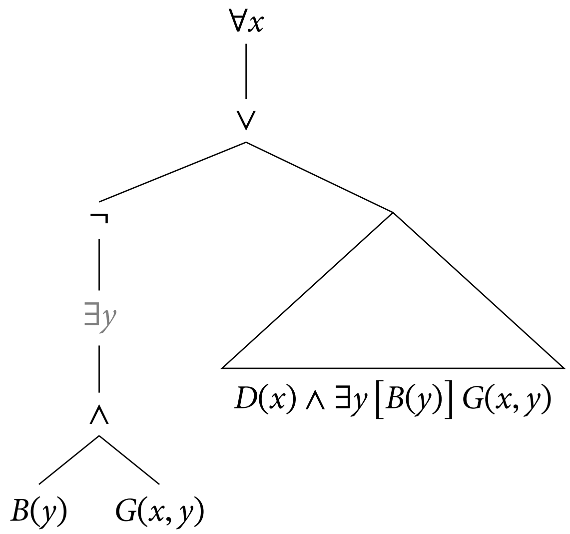
In (41), we progress to an example where only a part of the complement of 'only' is focused (such examples are often not even considered in generalized quantifier literature). In subsection 9.1.3, we showed that this sentence is both D- and B-conservative. If everything goes well, we should now see that it is strongly D- and B-restricted as well, and this is precisely what we see: both $D(x)$ and $B(x)$ positively p-scope over the nuclear scope. (For example, if Perceval starts out with golden armor at $D(x)$, this armor will take him through the conjunction between $D(x)$ and $B(x)$. The negation will then change his armor to silver and this will get him through the disjunction between the restriction and the nuclear scope, where he finally turns South.)

(41) a. [Only [an angry dog]] is barking. (surface)
b. [Only=every [a dog is barking]] an angry dog is barking. (LF)
c. $\forall x\colon \neg(D(x) \wedge B(x)) \vee ((A(x) \wedge D(x)) \wedge B(x))$

As noted by Herburger, 'only' can in general attach to anything, and it can also associate with focus targeting any constituent. This can lead to interesting cases where a variable doesn't occur as an argument of any monadic predicate in the restriction of 'only' (after focal mapping). Below, x only occurs in a dyadic atomic formula $G(x, y)$ in the restriction. Given that we have only defined restrictedness with respect to monadic predicates, don't we then characterize this example as non-restricted?

(42) a. Only a dog is gnawing a bone. (SF)
b. [Only [is gnawing a bone]] a dog is gnawing a bone. (LF)
c. $\forall x \left[\exists y \left[B(y)\right] G(x, y)\right] D(x) \wedge \exists y \left[B(y)\right] G(x, y)$

d.



We will discuss focus again in subsection 10.3.2, and in that section we will adopt a standard proposal that focusing deploys a predicate introducing contextual alternatives. Under that view, the restriction of 'only' in (42) turns out as $\mathbb{A}(x) \wedge \neg\exists y\colon B(y) \wedge G(x,y)$ and the apparent problem disappears, because x is restricted by $\mathbb{A}(x)$. But it might be worthwhile to think about what we could say in the absence of such an analysis, i.e. if the logical form was as shown in (42). In other words, what would happen if we allowed for dyadic restrictors?

The idea is that one could say that (42) is about bone-gnawers (and that it asserts that they are all dogs). In other words, this sentence can be thought of as jointly conservative with respect to B and the first argument of G.[20] Assuming that we allow for dyadic restrictors, does our p-scope-based definition of restrictedness, as it stands, characterize this example as restricted? It turns out that it doesn't. Quantifier $\exists y$ is clearly restricted by $B(y)$ (the restriction is even atomic). However, it is easy to see that under the current definition of strong restrictedness, $\forall x$ is not strongly G-restricted. Obviously, the culprit is the existential quantifer; donning the golden armor at (the left) $G(x,y)$, Perceval can enter $\wedge$ but cannot continue up to $\exists\, y$.

Interestingly, x would turn out to be strongly G-restricted if $\exists y$ was not there. Let us observe Perceval's travel from $G(x,y)$ to the nuclear scope under this assumption—we have grayed out $\exists y$ in (42d) so it is easy to pretend it is not there. Perceval starts in $G(x,y)$ with the golden armor, which gets him into $\wedge$. The grayed out node is not there, so Perceval proceeds to $\neg$, where his armor is transmuted to silver. Having silver armor, he can enter $\vee$ and then finally descend into the nuclear scope of $\forall x$.

[20] As we already noted, we don't define conservativity with respect to non-monadic predicates in the book (see also footnotes 12 and 16). But let us assume for a moment that we did.

The question, of course, is whether the "irrelevance" of the quantifier node (of another variable) exhibited by this example is accidental, or a general property of restrictedness, and it appears that the latter might be the case. In other words, if we remove the quantifier nodes in a restriction of a quantifier, and then test the quantifier for strong restrictedness, we get the result congruent with the conservativity of the quantifier.

As we noted above, this is a hypothetical situation as we will upgrade our analysis of focus to deploy the alternatives predicate. But what we find interesting in our rudimentary exploration of this hypothetical situation is that things work out well if we assume that the embedded quantifiers are "not there." The reason we find this interesting is possibly quite surprising—in the following section, we will see that the quantifier nodes are going to be eliminated for independent reasons!

9.3 Quantification without Quantifiers

We began this chapter by noting that the Medieval logicians had seen a deep connection between quantification and polarity. We followed this with a suggestion from Ludlow (2002) that quantification could be reconstructed simply by paying attention to the local polarity of free variables. The claim was that we could deduce (based solely on the structure of the formula) whether the variable should be existentially closed or universally closed. But, as we also saw, that claim was a bit premature; we needed to lay more groundwork to get things right, and in particular we needed to develop a notion of restrictedness to pair with Ludlow's suggestion. We are now at the point where we can tie things together.

We say we can tie things together, but there are a lot of threads to keep track of, so let's review those threads, see how they are interwoven, and ultimately come to better understand the resulting tapestry.

We have seen that conservativity (or more informally, aboutness) applies not only to determiner argument positions, but also more globally—for example, it also applies to sentence level logical forms. And we have seen that there is a syntactic counterpart to conservativity—restrictedness. Furthermore, we have seen that restrictedness, too, does not just apply to local determiner positions but to logical forms, and that the secret ingredient to restrictedness is p-scope. Natural language is restricted with respect to a predicate A, just in case $A(x)$ successfully positively p-scopes into "the other side" of the quantifier over x. It is in this way that we cash out the idea that 'Det As are Bs' is equivalent to 'Det As are As that are Bs'. It is p-scope, and the rule Copy, that executes the copying of the predicate A into the nuclear scope. And of course p-scope is nothing if not a polarity-sensitive relation. So, in this way, the thread of polarity runs through p-scope and Copy, thus restrictedness, thus providing a syntactic execution of conservativity, and ultimately aboutness.

As noted, we plan to dispense with quantifiers altogether. This is not to say that we want to dispense with *quantification*. The point is rather that there will be no quantificational *operators* in the syntax of (natural) language. Existential and universal closure over free variables will take their place. But closure rules can only do this work thanks to some of the very interesting features of natural language that we just reviewed. And the idea will be that it is the local polarity of the restrictor that determines whether a free variable is universally closed or existentially closed.

This claim might sound surprising, and it probably should. Quantifiers, or more accurately, quantifier symbols, are regarded as an indispensable ingredient of predicate logic—after all, predicate logic is propositional logic extended by quantification. But if we limit our attention to restricted (and thus conservative) sentences, it turns out that it is possible to set up a quantifierless system which retains both the full expressive power of the formal language and the full deductive power of the deductive system, in particular the Dynamic Deductive System.

Of course, limiting our attention to restricted/conservative sentences might be insufficient for various applications (perhaps some mathematical or computer science problems might require the full fledged predicate logic). Here, however, we are concerned with natural language, where, as discussed in the previous sections, restrictedness/conservativity is the norm. Thus restrictedness is not only an option, but even preferred on conceptual grounds, as it avoids the overgeneration of full predicate logic and generalized quantifier theory.

The main idea underlying our quantifierless system is that if we take a restricted formula and remove all the quantifier symbols (quantifier nodes, if we imagine formulas as trees), the resulting quantifierless formula contains enough information to recover the original formula. In other words, the quantifier symbols in restricted formulas are redundant.[21]

Looking at the issue from the perspective of the interpretation of the quantifierless formulas, if we wanted to be redundant, we could assign meanings indirectly via translation into a standard formula containing quantifiers. Below, we present the two canonical examples based on the restricted quantifier templates. This helps illustrate that the only difference between a quantifierless formula and its standard counterpart is the overt presence of the quantifier in the latter.

(43) a. $A(x) \wedge B(x) \rightsquigarrow \exists x\colon A(x) \wedge B(x)$
b. $\neg A(x) \vee B(x) \rightsquigarrow \forall x\colon \neg A(x) \vee B(x)$

[21] As we aim for the application of the system to natural language, we obviously take L** from Chapter 5 as the base language, i.e. the language with quantifiers—remember that the meaning of some determiners like 'most' is simply inexpressible in standard first-order logic. However, nothing we say in this section hinges on this assumption. Quantification without quantifiers is possible for the standard first-order language as well.

Of course, we don't want to put quantifiers back, what we really need is a closure rule that can tell us whether to interpret a free variable as being existentially bound, or universally bound.

In more detail, for each variable (say x) occurring in a quantifierless formula, RESTRICTED CLOSURE must perform two tasks: determine (i) the syntactic position where closure occurs and (ii) the type of the quantificational closure (existential or universal).

We will develop the Restricted Closure rule in two steps. In subsection 9.3.1 we develop the idea illustrated by (43). We will show that (i) we can predict the syntactic position of the closure from the positions of the occurrences of the variable, and that (ii) we can predict the type of the closure from the type of the junction where the closure occurs. In subsection 9.3.2, we will take the idea even further by bringing polarity into the picture. We will show that (iii) the type of the closure junction, and thus the type of quantificational closure, can be predicted from the polarity of the restrictors.

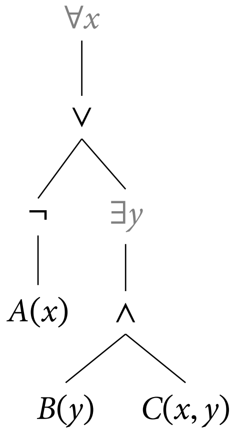
9.3.1 Farewell to quantifiers

Because the quantificational closure must subsume all occurrences of the variable (or, if you prefer, the entire variable chain), it is clear that closure must hold at or above the lowest common ancestor of all these variable occurrences, LCA(x). In fact, we will see that it makes the most sense to apply it precisely at LCA(x)—we will call this node the CLOSURE JUNCTION.

As for the type of closure (existential or universal), it turns out we will always produce a restricted formula if we take a hint from the definition of restricted quantifiers. Remember that a restricted existential quantifier combines an (unrestricted) existential quantifier and a conjunction, while a restricted universal quantifier combines an (unrestricted) universal quantifier and a disjunction. We can therefore set the type of closure to existential if the closure junction is a conjunction, and to universal if the closure junction is a disjunction.

To see how this works, let us look at the example in (44). It is best to consider the tree: the quantifierless formula consists of the black nodes only; the grayed-out quantifiers indicate the location and type of quantification introduced by the Restricted Closure. (The grayed-out quantifiers just above closure junctions are not actual operators in the syntax, but visual aids. It is helpful to imagine Restricted Closure as recovering the position and type of quantifiers. On this view Restricted Closure would be a—as we will see, partial—mapping from quantifierless formulas to standard formulas containing quantifiers.)

(44) $\neg A(x) \vee (B(y) \wedge C(x,y)) \rightsquigarrow \forall x\colon \neg A(x) \vee \exists y\colon B(y) \wedge C(x,y)$
"Every alien cured some bird."



Above, LCA(x) is the root of the quantifierless formula, so the closure over x applies at the root; as the closure junction is thus a disjunction, Restricted Closure dictates universal quantification over x. On the other hand, LCA(y) is a conjunction, so the quantificational closure over y is existential.

If we choose the wrong *type* of quantification, the result of the closure corresponds to a non-conservative standard formula—and if the Conservativity Universal discussed in section 9.1 is on the right track, such logical forms have no place in natural language!

(45) $A(x) \wedge B(x)$
a. $\rightsquigarrow \exists x\colon A(x) \wedge B(x)$ "Some alien is a bird."
b. $\not\rightsquigarrow \forall x\colon A(x) \wedge B(x)$ (non-conservative)

Now consider (46), where we test what happens if we get the closure *position* wrong. The actual result of Restricted Closure is (46a), and it is $\{A, B\}$-conservative. In (46b) we position $\exists y$ one level higher than instructed by Restricted Closure. The result turns out to be $\{A, B\}$-conservative as well, but not only that. In fact, the result turns out to be equivalent to (46a)! The equivalence is an example of a well-known class of equivalences in predicate logic, which allow us to raise or lower a quantifier over a junction, as long as the "crossed" member of the junction does not contain a free occurrence of the variable bound by the quantifier. The moral of (46b) is that while Restricted Closure does not produce all conservative formulas, it produces all conservative formulas modulo logical equivalence.

(46) $\neg A(x) \vee (B(y) \wedge C(x,y))$
a. $\rightsquigarrow \forall x\colon \neg A(x) \vee \exists y\colon B(y) \wedge C(x,y)$ "Every alien cured some bird."
b. $\not\rightsquigarrow \forall x \exists y\colon \neg A(x) \vee (B(y) \wedge C(x,y))$ (equivalent to a)
c. $\not\rightsquigarrow \exists y \forall x\colon \neg A(x) \vee (B(y) \wedge C(x,y))$ (non-conservative)

If we deviate from the instructions of Restricted Closure more than this—as in (46c), where we applied quantifier closure over y above the quantifier closure of x—we risk getting a non-conservative result. In the empty domain, (46c) is clearly false, as it is headed by an existential quantifier; on a domain with a single element, which is neither A nor B, (46c) is true, as $\neg A(x)$ is true.[22] From a deductive point of view, the failure to produce (46c) is related to the well-known fact that quantifiers of different types do not "commute"; in particular, $\forall x \exists y\, \varphi$ does not entail $\exists y \forall x\, \varphi$.

On the other hand, quantifiers of the same type do commute, i.e. $\forall x \forall y\, \varphi$ is equivalent to $\forall y \forall x\, \varphi$, and $\exists x \exists y\, \varphi$ is equivalent to $\exists y \exists x\, \varphi$, and indeed, the only strings of quantifiers that can result from Restricted Closure are strings of quantifiers of the same type. This results in formulas which were proposed as logical forms of the famous donkey anaphora sentences, such as (47). We will discuss these sentences in detail in section 11.1, the only point we want to make here is that there will be occasions when Restricted Closure determines the same closure position for more than one variable. If the occurrences of x and the occurrences of y have the same LCA, the closure positions of x and y will coincide, and furthermore, given that the closure type depends on the type of the closure junction, the quantifiers binding x and y will be of the same type. In such cases, Restricted Closure thus actually results in two standard formulas—(47a) and (47b)—corresponding to the quantifierless formula, but as they are equivalent, the question of which one is the "real" correspondent is irrelevant (and furthermore recall that we provide the translation to standard formulas only for convenience anyway). The quantifierless system simply does not formally distinguish $\forall x \forall y\, \varphi$ and $\forall y \forall x\, \varphi$, and there is also no reason why it should.

(47) Every farmer who owns a donkey likes it.
$\neg(F(x) \wedge D(y) \wedge O(x,y)) \vee L(x,y)$
a. $\rightsquigarrow \forall x \forall y\colon \neg(F(x) \wedge D(y) \wedge O(x,y)) \vee L(x,y)$
b. $\rightsquigarrow \forall y \forall x\colon \neg(F(x) \wedge D(y) \wedge O(x,y)) \vee L(x,y)$

Adopting a quantifierless language with Restricted Closure eliminates certain configurations familiar from standard predicate logic with quantifier symbols. For one, there are no open formulas, in the sense that no quantifierless formula maps to the standard formula such as (48a) containing a free variable. This is so because Restricted Closure applies to all variables of the quantifierless formula. If the quantifierless formula contains x, then the corresponding standard formula contains a quantifier over x, as in (48b).

[22] The non-conservativity of (46c) indicates the tight relation between conservativity and inclusive logic, i.e. logic which allows the empty domain. We intend to explore this relation in further research.

(48) a. $\ldots \not\rightsquigarrow A(x) \land B(x)$
b. $A(x) \land B(x) \rightsquigarrow \exists x\colon A(x) \land B(x)$

On the other side of the spectrum, there is no vacuous quantification either. Quantification results only from Restricted Closure, and Restricted Closure only applies to variables which occur in the formula. For example, no quantifierless formula maps to the standard formula (49a), because this formula contains no occurrence of y. Applying Restricted Closure to the quantifierless part of this formula only results in quantification over x, as shown in (49b).[23]

(49) a. $\ldots \not\rightsquigarrow \exists y \exists x\colon A(x) \land B(x)$
b. $A(x) \land B(x) \rightsquigarrow \exists x\colon A(x) \land B(x)$

As the third and final example of an eliminated standard configuration, note that no quantifierless formula maps to a standard formula containing two quantifiers binding the same variable. For example, while we may propose that logical forms in (50) and (51) both contain (and quantify over) variable x, a mindless conjunction of these logical forms does not yield the intended result—the logical form of the conjunction of the sentences. (52a) does not map to a standard formula containing two occurrences of $\exists x$, but to a formula containing a single occurrence of that quantifier (the closure occurs at the LCA of all occurrences of x, i.e. at the root). The correct logical form of the conjoined sentence (52b) deploys two variables, x and y, which results in two applications of Restricted Closure, as intended.

(50) Some alien is a bird. $A(x) \land B(x) \rightsquigarrow \exists x\colon A(x) \land B(x)$

(51) Some cat is a diplomat. $C(x) \land D(x) \rightsquigarrow \exists x\colon C(x) \land D(x)$

(52) Some alien is a bird, and some cat is a diplomat.
a. $^*(A(x) \land B(x)) \land (C(x) \land D(x)) \not\rightsquigarrow (\exists x\colon A(x) \land B(x)) \land (\exists x\colon C(x) \land D(x))$
$\rightsquigarrow \exists x\colon (A(x) \land B(x)) \land (C(x) \land D(x))$
b. $(A(x) \land B(x)) \land (C(y) \land D(y)) \rightsquigarrow (\exists x\colon A(x) \land B(x)) \land (\exists y\colon C(y) \land D(y))$

All situations of standard formulas without a corresponding quantifierless formula can be subsumed under the slogan "one variable—one quantifier," and it is clear that none of the quantifierless formulas are "missing" at the expense of reduced expressive power. First, open formulas are—strictly speaking—not

[23] In subsection 7.2.3 we saw that DDS can be naturally adjusted to produce only closed formulas and formulas without vacuous quantification.

interpretable anyway, the usual convention in mathematics being that they be interpreted as implicitly universally closed at the root. Second, vacuous quantification makes no contribution to the meaning.[24] Finally, multiple quantifiers over the same variable can be simply replaced by quantifiers over distinct variables.

Summing up, Restricted Closure over variable x applies at the lowest node containing all occurrences of x. If this node is a conjunction, the closure is existential; if the node is a disjunction, the closure is universal.

The final issue we want to address here concerns the case of a variable (x) having a single occurrence in the formula, or more precisely, when x occurs in a single atomic formula (even if it has multiple occurrences there). Of course, LCA(x) is then this very atomic formula. In particular, LCA(x) is not a junction, so Restricted Closure, which determines the type of quantificational closure by inspecting the type of the closure *junction*, cannot be applied.

There are two ways to resolve this issue. One way is to amend Restricted Closure to apply universal closure over x when LCA(x) is a disjunction, and existential closure *otherwise*. It turns out that Restricted Closure, amended in such a way, still always results in conservative formulas—as illustrated in (53), we get a conservative formula if and only if we assume that quantificational closure is existential.

(53) a. $A(x) \rightsquigarrow \exists x \colon A(x)$
b. $A(x) \not\rightsquigarrow \forall x \colon A(x)$ (non-conservative)

This solution will turn out to be even more natural and elegant once we base Restricted Closure on the polarity of the restrictor rather than on the type of the closure junction (we will do this in the next section), but we believe that for the application to natural language, the better option might be to say that formulas containing a variable occurring in a single atomic formula should not be considered appropriate input to quantificational closure (54). Nothing we say will crucially depend on whether (53) or (54) is adopted, so we leave the issue open.

(54) $A(x) \not\rightsquigarrow \ldots$

We began this chapter by talking about aboutness, and you may well have thought we were engaged in stating the completely obvious. And while there is a sense in which aboutness is indeed obvious, its formal execution in terms of

[24] This statement only holds for standard logic. It might well turn out that in inclusive logic, the quantifierless system fares better than the standard format deploying the quantifier symbols.

conservativity and restrictedness is less so. But there is something that was not obvious at all from the outset: If this idea of aboutness is a fundamental property of language, and its expression is captured via restrictedness and thus p-scope of our Dynamic Deductive System, then it places powerful constraints on possible natural language constructions, and as a consequence of those constraints, universally quantified constructions are going to involve a disjunction and existentially quantified constructions are going to involve a conjunction.

But this insight gives rise to a new question. Our closure rule, as developed so far, dictates that quantified expressions are either instances of universal quantification over a disjunction, or instances of existential quantification over a conjunction. But we know from section 9.2 that the restrictor of a restricted universal quantifier must have negative local polarity, and that the restrictor of a restricted existential quantifier must have positive local polarity (where by local polarity we mean polarity within the quantifier). This raises a question: Can't we run everything off of polarity? After all, we can predict the nature of the junction from the local scope of the restriction. So doesn't polarity run the whole show here? This is the question we turn to next.

9.3.2 Can polarity rule it all?

In the previous section, we showed that if we limit our attention to restricted/conservative sentences, quantificational structures can be formally represented without actually using quantifiers. The position of a quantifier symbol binding a variable can be recovered from the distribution of the occurrences of the variable in the formula: the quantificational closure applies at the lowest syntactic position dominating all occurrences of the variable. The type of the closure is determined by the type of the node where it applies: it is universal if this node is a disjunction, and existential otherwise (if the node is either a conjunction or an atomic formula).

However, there is another property covarying with the type of the closure: the polarity of the restrictors, more precisely their polarity within the position of the closure. While we can in general have several restrictors of a single variable[25] (simply because several atomic formulas can positively p-scope into the nuclear scope), we have seen in subsection 9.2.1 that they must all agree in polarity; any restrictor of an existential/universal restricted quantifier has positive/negative polarity within the quantifier. The triple correspondence between the type of quantificational closure, the type of the closure junction (or more precisely, to

[25] In section 9.2 we defined the term 'restrictor' as relative to an occurrence of a quantifier, but as our closure rule from the previous subsection always "produces" at most one quantifier per variable, we can now safely see restrictors as relative to variables—in fact, as we have moved into a language without explicit quantifiers, this view now makes more sense than the original definition.

Table 9.1 The covarying properties of Restricted Closure

the type of closure	existential	universal
the type of the closure position	conjunction/atomic	disjunction
the polarity of the restrictors	positive	negative

cover the cases of a variable occurring in a single atomic formula, the type of the position where the closure applies), and the polarity of the restrictor(s) is summed up in Table 9.1.

In the previous section, we "encoded" the type of quantificational closure by the type of the closure junction. Can we go further and encode the type of the closure junction (and thus the closure type as well) by the polarity of the restrictor(s)?[26]

As we shall see, the idea is sound, but a bit trickier to execute than the reduction in the previous section. Here is the apparent problem: We want to determine the junction types from the polarity of the restrictors. But the notion of a restrictor is based on p-scope, which depends on some aspects of the structure of the formula, including the types of the junctions. Are we engaged in circular reasoning here? No, we are not.

We need to be mindful of two things. First, we know the type of many junctions in the formula. The only unknowns are the junctions where the quantificational closure applies. Second, the identity of the restrictors can be figured out as soon as we know the types of all junctions in the restriction. In particular, we don't need to know the type of the LCA(x) to figure out who the restrictors of x are. What this implies is that we can compute the junction types by a bottom-up recursion.

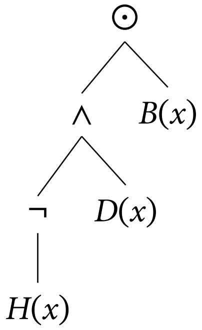
Let's first take a look at the simple case of (55), which contains a single variable. We use the symbol $\odot$ to mark the junction of the unknown type. The point here is that we don't need to know whether $\odot$ will turn out as a conjunction or as a disjunction to realize that only $D(x)$ could possibly be the restrictor. The (relative) p-scope of $H(x)$ over the nuclear scope is negative if anything—if Perceval does not start out from $H(x)$ wearing silver armor, he cannot even get as far as to $\wedge$, let alone $\odot$—so it cannot be a restrictor. On the other hand, $D(x)$ can have positive (relative) p-scope over the nuclear scope, as it is immediately dominated by a conjunction, and furthermore, if Perceval starts out at $D(x)$, he can reach the *checkpoint* of $\odot$, whatever this node turns out to be. In other words, $D(x)$ is a POTENTIAL RESTRICTOR, and will turn out to be a restrictor for real if only we choose the type of $\odot$ correctly!

[26] The reverse possibility of encoding the polarity of the restrictor by the type of junctions does not seem very appealing. In fact, it is hard to imagine how one would even start such an enterprise. For a linguistic argument against dropping negations, see footnote 30.

(55) $(\neg H(x) \wedge D(x)) \odot B(x)$

a. $\rightsquigarrow \exists x\colon (\neg H(x) \wedge D(x)) \wedge B(x)$ "Some unhappy dog is barking."

b. $\not\rightsquigarrow \forall x\colon (\neg H(x) \wedge \overline{D(x)}) \vee B(x)$ (non-conservative)

c. 

But we know how to choose the closure junction type given a potential restrictor such as $D(x)$ above even without sending Perceval on his dangerous journey—and we know how to do this not only in (55) in particular, but in general! We only need to look at the polarity of the potential restrictor within the quantifier. If it is positive/negative, there was an even/odd number of negations between the restrictor and $\odot$, so to let Perceval (who started out wearing golden armor) through the final checkpoint, $\odot$ must resolve to a conjunction/disjunction. In (55), $D(x)$ has positive polarity within $\odot$, so $\odot$ must be a conjunction, as shown in (55a). (Of course, once we know $\odot$ is a conjunction, we also know that the closure must be existential.)

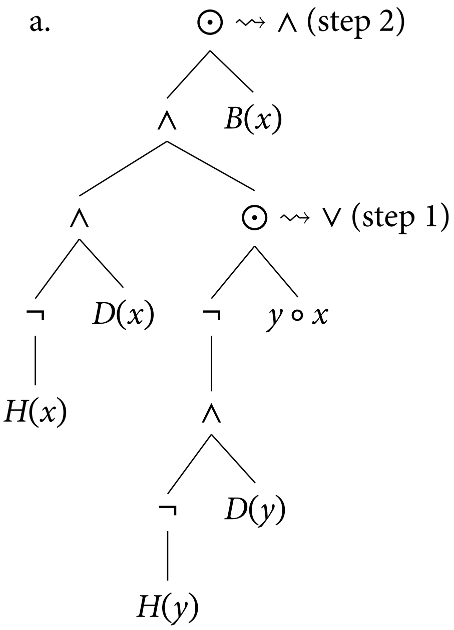
There is a deep point at work here—one that is missed in the proposal in Ludlow (2002). The polarity of the restrictor is not merely an indicator of what the nature of the quantification must be. To choose the wrong form of quantificational closure (or the wrong node at which closure happens) would actually lead to structures that violate restrictedness and thus conservativity. To put this in more dramatic terms, it would undermine the conservativity of natural language if these two notions (quantification and polarity) were not connected as they are. For example, in (55b), we see that if we choose the wrong type of quantification, the result is a non-conservative formula. The connection between polarity and closure is not accidental nor is it a stipulation. Natural language breaks if they are not coordinated as they are.

Let's move on to a more complex situation involving multiple variables. In the logical form of a definite determiner shown below, the restriction of the outer quantifier ($\exists x$) contains another quantifier ($\forall y$) (in the standard formal language with quantifiers). In these cases, we have to work out the unknown junction types recursively (the junctions of unknown type are shown as $\odot$). Consider (56a). The first step is to attack the lower $\odot$. By the same logic as for (55) above, we figure out that $H(y)$ cannot possibly be a restrictor, but that $D(y)$ can be one if only we interpret the lower $\odot$ as a disjunction—observe that there is a negation between

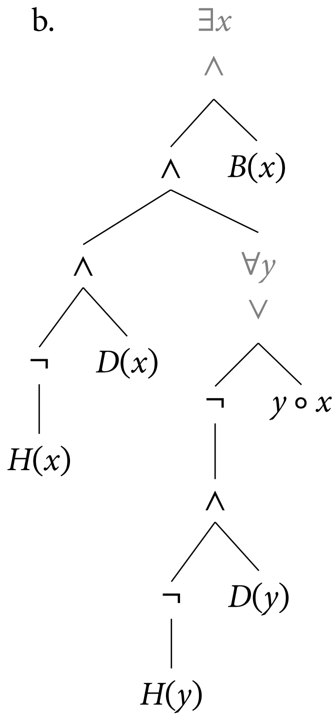
$D(y)$ and $\odot$. Once we have figured out the identity of the lower $\odot$, we move on to the upper one. Again, $H(x)$ is not a potential restrictor, but $D(x)$ is, and it will be an actual restrictor if we interpret the upper $\odot$ as a conjunction.

(56) The unhappy dog is barking. / The unhappy dogs are barking.
$((\neg H(x) \wedge D(x)) \wedge (\neg(\neg H(y) \wedge D(y)) \odot y \circ x)) \odot B(x)$
$\rightsquigarrow \exists x\colon ((\neg H(x) \wedge D(x)) \wedge \forall y\colon \neg(\neg H(y) \wedge D(y)) \vee y \circ x) \wedge B(x)$
$\equiv \exists x \big[(\neg H(x) \wedge D(x)) \wedge \forall y \big[\neg H(y) \wedge D(y)\big]\, y \circ x\big]\, B(x)$

a.



b.



The tree in (56b) shows the standard formula corresponding to the quantifierless version in (56a). In this tree, we follow two conventions, introduced to facilitate a compact representation of both the formula to which Restricted Closure applies, and of its interpretation yielded by our closure rule.[27] The first convention was introduced in the previous section: we gray out all the information computed by the closure rule. The second convention is new: we show the "implicit" quantifier and (the type of) its corresponding junction "in a single node," so that once you take away the gray parts, you are left with a faithful representation of the quantifierless formula. (For now, you will want to imagine the gray matter replaced by $\odot$, but not for long.)

[27] We retain the term 'Restricted Closure', even if this rule now determines not only the position and type of quantificational closure, but the type of the closure junction as well. But in a way, the name makes perfect sense: Restricted Closure recovers the position and type of a *restricted* quantifier, and a restricted quantifier is an abbreviation of a structure containing a certain kind of junction.

Finding a (potential) restrictor is a crucial step in an application of Restricted Closure, so let us make sure we have a complete grasp of this notion. Above, we have repeatedly used the fact that a restrictor positively p-scopes across the junction where the quantificational closure applies, but we didn't much emphasize *where* a restrictor may occur. In principle, one can have an atomic formula p-scoping from the restriction (the left member of the closure junction) into the nuclear scope (the right member of the closure junction), or vice versa. As we have mentioned in subsection 9.2.1, both directions result in a conservative formula. However, once we want to deploy restrictors in Restricted Closure, it becomes crucial that we limit their source to a single member of the closure junction. If we don't do that, the reasoning which lead us to conclude that all potential restrictors have the same polarity within the closure position does not apply. For example, in (57b) and (57c) below, the polarity of the potential restrictors is not uniform, so choosing one above the other results in a different type of closure, both leading to a conservative interpretation.

(57) a. $A(x) \odot B(x)$

i. $\rightsquigarrow \exists x\colon A(x) \wedge B(x)$

ii. $\rightsquigarrow \exists x\colon \overline{A(x)} \wedge \underline{B(x)}$

b. $A(x) \odot \neg B(x)$

i. $\rightsquigarrow \exists x\colon A(x) \wedge \neg B(x)$

ii. $\not\rightsquigarrow \forall x\colon \overline{A(x)} \vee \neg \underline{B(x)}$

c. $\neg A(x) \odot B(x)$

i. $\rightsquigarrow \forall x\colon \neg A(x) \vee B(x)$

ii. $\not\rightsquigarrow \exists x\colon \neg \overline{A(x)} \wedge \underline{B(x)}$

d. $\neg A(x) \odot \neg B(x)$

i. $\rightsquigarrow \forall x\colon \neg A(x) \vee \neg B(x)$

ii. $\rightsquigarrow \forall x\colon \neg \overline{A(x)} \vee \neg \underline{B(x)}$

In effect, allowing both members of the closure junction as the source of restrictors can make the result of Restricted Closure indeterminate. It is only by designating one member of the closure junction as special, i.e. as the restriction which may contain the restrictor(s), that we can guarantee that Restricted Closure yields the same result irrespectively of which restrictor is inspected for polarity. Designating the left member of the closure junction as the restriction—in other words, requiring that restrictors p-scope "from left to right"—is of course a decision mediated by natural language considerations.[28]

[28] Having mentioned natural language, let us speculate on a possible relation between Restricted Closure and endocentricity of natural language. While linguists agree on hardly any aspect of natural

Our Restricted Closure has evolved quite a bit from its humble beginnings in the previous section (remember that we at first only predicted the closure type from the type of the closure junction), but concluding its development at this point would be stopping half-way toward our goal. Observe that while on one hand, our quantifierless format is certainly simpler than standard formulas, it is on the other hand also more complex, as it introduces another propositional connective, $\odot$. Could we simplify the situation by somehow reducing the number of binary propositional connectives? Now we certainly don't want to get rid of our new $\odot$—let us call it the GENERALIZED JUNCTION—but could we perhaps do without $\wedge$ or $\vee$ (or both)?

Sure we can. In fact, we can do without either. First, we can dismiss $\vee$ by defining it as not a connective in its own right, but rather as the abbreviation $\mathcal{A} \vee \mathcal{B} :\equiv \neg(\neg\mathcal{A} \wedge \neg\mathcal{B})$, based on the De Morgan's laws.[29] With $\vee$ out of the way, the idea is to discard $\wedge$ as well, using $\odot$ in its place, and to assume that the identity of $\odot$ is only variable in positions where quantificational closure applies: in closure positions, the "real type" of the generalized junction will be determined from the polarity of the restrictors, as detailed above; elsewhere, the generalized junction will always resolve to a conjunction.

Let us write down the final statement of our Restricted Closure, and illustrate it using (59). (We emphasize yet again that the translations into standard predicate logic are merely aids which help us understand the interpretive effect of Restricted Closure. No actual translation needs to happen.)

(58) RESTRICTED CLOSURE applies to formulas built from atomic formulas of standard *predicate* logic using only connectives $\neg$ and $\odot$.

(RC1) Find the closure positions: for each variable x, the quantificational closure applies at the lowest common ancestor of all occurrences of x in the formula. (This is how we get Qx and Qy in (59b).)

language structure, one thing they seem to agree on, at least within the generative tradition, is that complex natural language expressions are binary (composed of two constituents) and asymmetric (one of these constituents is the head, in the sense that the syntactic properties of the phrase are inherited from the syntactic properties of the head). Above, we have seen that the mathematics of Restricted Closure imposes an asymmetry between the two branches of a closure junction. Could it be that this asymmetry is the ultimate source of headedness in natural language?

[29] Of course, this complicates the representation of any formulas containing $\vee$, but who cares? Mathematicians certainly won't—the above abbreviation is a standard move in propositional logic—and in fact, even linguists might not, if for a very different reason: in subsection 10.3.3 we shall see that there is something funky about 'or' in natural language, and recent research suggests that it might in fact correspond to structures much more complicated than the simple Boolean $\vee$.

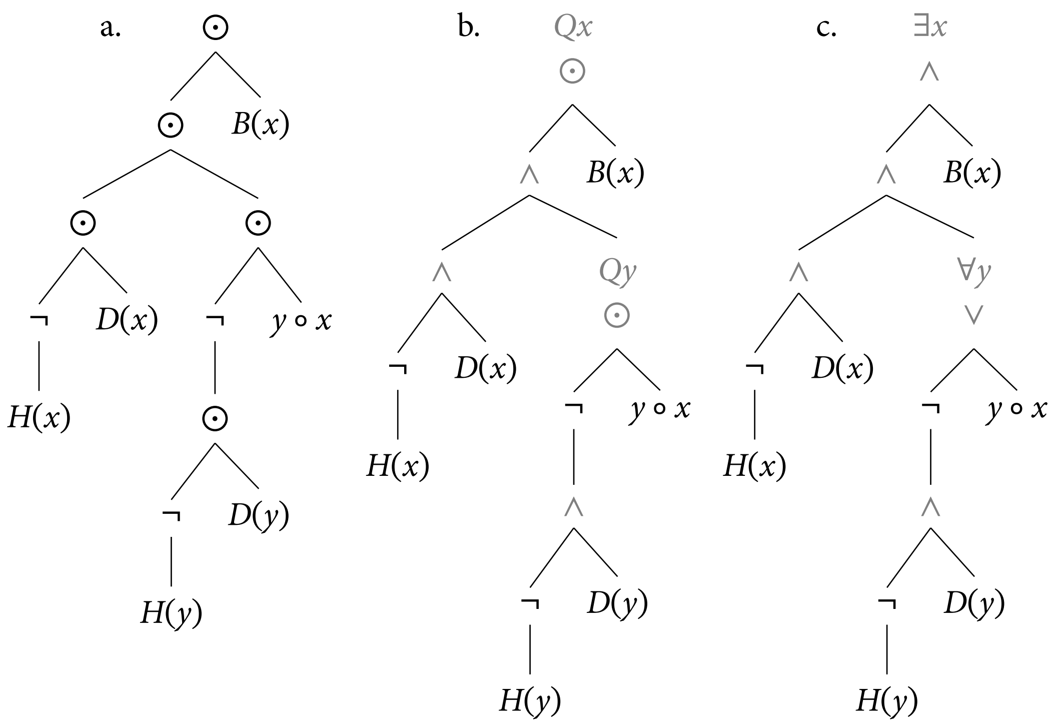
While $\vee$ seems to be a rare animal in logical forms of natural language expressions, to be found mainly as a part of a universal quantifier, $\wedge$, on the other hand, seems to be all over the place. In Chapter 12, we argue that—precisely as in (59)—every branching node in syntax corresponds to $\wedge$ by default. It is based on this linguistic argument that we choose to abbreviate $\vee$ using $\neg$ and $\wedge$, rather than $\wedge$ by $\neg$ and $\vee$.

All generalized junctions ($\odot$) where the closure does *not* apply are interpreted as conjunctions. (We have arrived at (59b).)

(RC2) Figure out the type of each closure using bottom-up recursion: to figure out the type of the quantificational closure over x, find a potential restrictor of x and set the closure type to existential/universal (and the type of the closure junction to a conjunction/disjunction) if the polarity of the restrictor within the closure position (Qx) is positive/negative.
(By proceeding bottom-up in (59b), we first resolve the identity of Qy: we find a potential restrictor $D(y)$; it is of negative local polarity, so Qy is resolved to $\forall y$. We then consider Qx: $D(x)$ is a potential restrictor; it is of positive local polarity, so we resolve Qx to $\exists x$.)

(59) The unhappy dog is barking. / The unhappy dogs are barking.

$$((\neg H(x) \odot D(x)) \odot (\neg(\neg H(y) \odot D(y)) \odot y \circ x)) \odot B(x)$$
$$\rightsquigarrow \exists x\colon ((\neg H(x) \wedge D(x)) \wedge \forall y\colon \neg(\neg H(y) \wedge D(y)) \vee y \circ x) \wedge B(x)$$
$$\equiv \exists x \left[(\neg H(x) \wedge D(x)) \wedge \forall y \left[\neg H(y) \wedge D(y)\right] y \circ x\right] B(x)$$

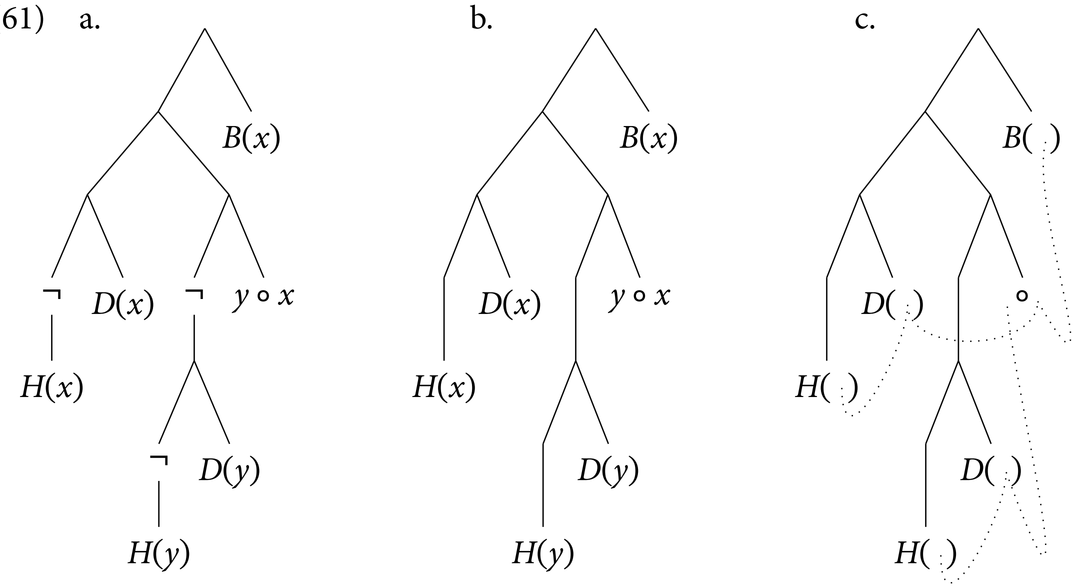


Note that there is a residue of quantifierless formulas which Restricted Closure cannot be applied to. This is perfectly fine, because these are precisely the formulas which turn out non-conservative regardless of how we close their variables. Syntactically, these formulas are characterized by the property of containing at least one variable without a potential restrictor. For example, because the generalized junction between $A(x)$ and $B(x)$ in (60) resolves to a conjunction, the relative p-scope of both $A(x)$ and $B(x)$ negative, so neither of them is a potential restrictor. (Remember that $C(x)$ can never function as a restrictor, as it occurs within the wrong member of the closure junction.) On the semantic side, the failure to apply Restricted Closure is reflected in the fact that neither (60a) nor (60b) is conservative (with respect to A or B).

(60) $(\neg A(x) \odot \neg B(x)) \odot C(x)$
a. $\not\leftrightarrow \exists x\colon (\neg A(x) \wedge \neg B(x)) \wedge C(x)$ (not conservative)
b. $\not\leftrightarrow \forall x\colon (\neg A(x) \wedge \neg B(x)) \vee C(x)$ (not conservative)

If we wish, we can push the simplification even further. In (61a), we drop the $\odot$ labels on binary branching nodes—we don't need them, as $\odot$ is now the only binary connective anyway. In fact, we can drop the negation labels in the tree as well, since all unary nodes are negations—this yields (61b).[30] And while we are at it, who needs variable symbols? In (61c), we use the *bonds* of Quine (1981, p. 70). Following this line of investigation is beyond the scope of this book, but it is certainly interesting—a potential minimalist program for logic.

(61) a. b. c.



[30] Actually, dropping the negations is the one move we would not consider if we were carrying out the simplification project with the application to natural language in mind. Negation is certainly the logical operator with the clearest expression in natural language. In contrast to junctions and quantifiers, it is indisputable that negation, or rather a negation feature or head, is an active ingredient of the (morpho)syntax of natural languages. For example, there is no language without sentential negation; there are negative concord languages; most if not all languages display negative polarity items.

Rather than follow that reductive path let's stay on mission, because our results regarding restrictedness and quantification have set us up to move from talk of formal languages like L** to natural language. And, as we will see, the tools that we acquired from our study of conservativity and restrictedness in formal languages are going to serve us very well, as they are going to help us crack open some longstanding puzzles in the semantics of natural language. In a bit, we will begin by looking at one such famous puzzle—donkey anaphora—and we will use that to springboard into a closer look at the logical form of natural language, and ultimately how these insights about the nature of language can be cashed out within the Minimalist Program in generative linguistics. But before we get there we will find it useful to understand just how important restrictedness is to natural language. As we will see, it is really a fundamental property of natural language, and thus lies at the very foundation of everything that will follow.

10
Restrictedness as a Fundamental Property of Natural Language

In the previous chapter, we claimed that the property of conservativity, as applied to determiners in generalized quantifier theory, can be extended to apply to much more than traditional determiner positions, and indeed it can be applied to larger linguistic structures like sentences. More importantly, we showed how conservativity has a reflex in the syntactic notion of restrictedness. That connection is so deep and so fundamental that we need to devote this chapter to developing the idea. As we have suggested, restrictedness is not merely something that attaches to determiners; it is something that attaches to every well-formed truth-evaluable natural language structure. It is a property that governs the interpretability of natural language sentences, and (as we will see in section 11.1) discourse structures. It is a property that applies to levels of representation like LF, which is the level of linguistic representation that is visible to the semantics. If a natural language sentence failed to be restricted it would be simply uninterpretable.

When we say that the role of restrictedness is not merely fundamental, but also deep, we are not simply generating hype for the idea. The depth stems from the properties that are folded into our understanding of restrictedness. As we saw in the previous chapter, restrictedness is closely related to our understanding of quantification, in that the type of quantificational closure (existential or universal) depends upon the polarity of the restrictor. And restrictedness itself is intimately tied to the notion of polarity, since it is defined in terms of our polarity-sensitive relation p-scope. So the depth of restrictedness is tied to its many connections to polarity, and thus ultimately to what we have called the Holy Grail of Natural Logic.

So, the idea is important and it needs to be well-anchored. Providing that anchoring is the goal of this chapter. In the next chapter, we will turn to a specific example of discourse anaphora to illustrate this idea and its importance to the study of natural language. Restrictedness will become particularly important in Chapter 12, in which we begin exploring specific strategies for embedding our proposal in contemporary linguistic theory.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović, Oxford University Press. DOI: 10.1093/oso/9780199591534.003.0010

10.1 Generalized Conservativity Universal and Restrictedness Universal

In the generalized quantifier framework, conservativity is a property of the denotations of determiners. A determiner D is conservative if and only if (1) holds for every domain E and every $A, B \subset E$, where $[\![\mathrm{D}]\!]_A$ and $[\![\mathrm{D}]\!]_E$ are the denotations of the determiner on domains A and E. This denotation-based understanding of conservativity is reflected in the Conservativity Universal (CU) hypothesis (2) postulated by Keenan and Stavi (1986), following an idea in Barwise and Cooper (1981). Conservativity Universal does not directly assert something about the denotation of a natural language sentence. It makes a statement about particular building blocks of that denotation—the denotations of determiners.

(1) $[\![\mathrm{D}]\!]_E(A, B) = [\![\mathrm{D}]\!]_A(A, A \cap B)$

(2) Conservativity Universal (CU)
Extensional determiners in all languages are always interpreted by conservative functions.

In the framework based on generalized conservativity, on the other hand, conservativity is a property of sentences, or more precisely, their logical forms. The logical form of a sentence is conservative with respect to some predicates iff its interpretation on any domain is the same as its interpretation on the reduced domain containing only objects satisfying the given predicates. Formally, given a closed formula φ and a set of predicates κ, φ is κ-conservative iff (3) holds for any domain E; see section 9.1 for the derivation of this condition. (When we omit the qualifier κ, we refer to some unspecified set of predicates.)

(3) $[\![\varphi]\!]_E = [\![\varphi]\!]_{E'}$, where $E' := \bigcup_{P \in \kappa} [\![P]\!]_E$

In this section, we will sometimes refer to the GQ conservativity as LOCAL, and to generalized conservativity as GLOBAL. What we propose is as simple as it gets:

(4) Generalized Conservativity Universal (GCU)
Logical forms of natural language expressions are conservative.

Is GCU a rehash of the original CU, or do they differ in some important way? If they differ, what are the differences? In particular, is GCU perhaps a stronger hypothesis than CU?

In this section, we will not only show that the answer to the final question is affirmative—GCU is thus not only a [[generalized conservativity] universal], but a [generalized [conservativity universal]] as well—but we will argue that this stronger hypothesis is in fact empirically supported. In section 10.2 we will demonstrate that extra strength by showing that a number of additional empirical predictions are generated when we expand the domain of quantification to include entities other than people and things (our attention will be focused on events). As we will see, these predictions will be born out by the data. Of course, with a rich flow of new predictions come apparent counterexamples. In section 10.3, we will take a look at three apparent counterexamples to GCU (non-intersective adjectives, focus, and disjunction) and we will argue that they dissolve upon a more detailed semantic and syntactic analysis. Then, in section 10.4, we will take another look at aboutness. We will first show how our notions of conservativity and restrictedness allow us to formalize the various levels of detail at which aboutness can be understood, and then argue that the existence of aboutness at any level of detail implies that GCU will follow directly if CU holds.

Before we explore GCU, however, let us strengthen it even further by weaving in the role of restrictedness. While the core idea behind Restrictedness Universal (RU) below is largely the same as that for GCU, RU pushes matters further—it is a stronger empirical hypothesis. In the previous chapter we saw that while every restricted formula is conservative, not every conservative formula is restricted—we can only guarantee that it is logically equivalent to a restricted formula. Thus, expressing the universal in syntactic terms (restrictedness) rather than in semantic terms (conservativity) adds some empirical bite to our generalization of CU.

(5) RESTRICTEDNESS UNIVERSAL (RU)
Logical forms of natural language expressions are restricted.

It is worth emphasizing that we see GCU and RU as empirical generalizations, not as part of human cognitive endowment. It is the particular implementation of quantification—the Restrictive Closure on a quantifierless formal language—that we envision as a part of linguistic faculty, and Restrictive Closure is what makes natural language conform to RU.

RU will become particularly important in Chapter 12, where we will talk about the relation between logical form as a semantic representation and Logical Form (LF) in linguistic theory. In this chapter, the contrast between GCU and RU will play a role in subsection 10.3.3, as the objection presented in that section is in fact an objection against RU and not GCU. We thus begin our deeper dive into RU.

10.2 Conservativity on Abstract Domains

The quantificational domain in the natural language examples in the previous chapters most often consisted of tangible individuals (cats and dogs, birds and apples, Norwegians and diplomats). This is not surprising. Quantification over people and common things is the cornerstone of every theory of quantification. However, many other sorts of entities have been proposed to belong to linguistic ontology. Alongside familiar individuals, we encounter exotica like events, kinds, sums, properties, degrees or intervals of degrees, moments or intervals of time, and possible worlds or situations (Szabolcsi 2010, p. 33). Even more, there is research implying that natural language employs the same devices to quantify over different domains; Schlenker (2006) argues that there is "a pervasive symmetry between the linguistic means with which we refer to [individuals, times and possible worlds]" (Schlenker 2006, p. 504). In light of such research, one would expect Conservativity Universal, which is really a universal about possible quantificational structures, to apply (and hold) across quantificational domains, and GCU does (it applies by definition and we will argue that the generalization holds up).

Let's illustrate quantification over such a domain by looking at events—in particular at the *event semantics* approach toward the relation between a verb and its arguments. The traditional way to encode the relation between a verb and its nominal arguments is to assume that in logical form, the terms corresponding to the nominal arguments of the verb occur as arguments of the verbal predicate. This is shown in (6a), where the transitivity of the verb 'hunt' is reflected in the assumption that the predicate H takes two arguments.

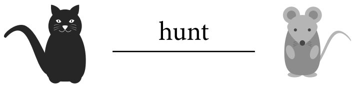
(6) A cat is hunting a mouse.
  a. $\exists x\colon C(x) \wedge \exists y\colon M(y) \wedge H(x, y)$
  b. $\exists x\colon C(x) \wedge \exists y\colon M(y) \wedge \exists e\colon H(e) \wedge \text{Agent}(x, e) \wedge \text{Theme}(y, e)$

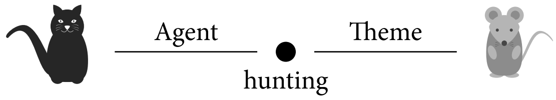
Davidson (1967) pointed out that seeing logical forms of action sentences as involving events and quantification over events avoids the problem of variable polyadicity of verbs, which typically arises in the case of adverbial modification.[1] In the neo-Davidsonian rendition of this idea (T. Parsons 1990, 1995), the sentence is given the logical form shown in (6b). The crucial new component of this logical form is the (existentially quantified) event variable e. In neo-Davidsonian event semantics, verbs correspond to monadic predicates—$H(e)$ asserts that e is an event of hunting—while the nominal arguments of the verb are associated with the event by theta-role predicates—for example, $\text{Agent}(x, e)$ asserts that the individual x is the agent of the event e. The (particular kind of) transitivity of verb 'hunt' is reflected in the fact that the logical form above contains two theta-role predicates: Agent and Theme.

We are interested in the two different renditions of (6) because they differ in conservativity properties. As we will see, they both turn out to be conservative, but with respect to different sets of predicates. The first logical form (6a) is

[1] Since then, many other arguments in favor of event semantics have been put forth. For an overview, see Ramchand (2007).

$\{C, M\}$-conservative, implying that the sentence is about cats and mice. However, this is not the case for (6b), which turns out to be $\{C, M, H\}$-conservative, implying that the sentence is about cats, mice, and (events of) hunting. To intuitively grasp how this comes about, we depict the relevant part of the model for the two logical forms.

(7) a. 

b. 

Picture (7a) shows the relation between the two individuals that makes (6a) true. The cat and the mouse are directly related by the dyadic predicate H. Restricting the quantificational domain to cats and mice does not disturb this relation, which implies that (6a) is $\{C, M\}$-conservative. In (7b), however, the relation between the cat and mouse is indirect. The two individuals are directly related to another entity, an event of hunting, represented by the dot, the former by the relation of being the agent of the event, and the latter by the relation of being the theme of the event. If (6b) was $\{C, M\}$-conservative, we could cut out the middle man without changing the interpretation of the sentence. However, this is not the case. If the event is removed from the domain, the indirect relation between the mouse and the cat is lost. Thus, (6b) is not $\{C, M\}$-conservative. However, it turns out that it is $\{C, M, H\}$-conservative. If we restrict the domain in such a way that the events of hunting (and also cats and mice) are retained, then we can be sure that the interpretation of the logical form does not change.

We can also investigate the difference between the two logical forms of (6) by considering restrictedness. For a formula to be restricted (and thus conservative), all its variables must be restricted. So (6a) needs two restrictors—these turn out to be $C(x)$ and $M(y)$—while (6b) can only be restricted if the event variable e is restricted as well, and indeed, e turns out to be restricted by $H(e)$.

We have merely glimpsed at event semantics here. Discussing the many phenomena it is relevant for, like distributive vs. collective verbs, is out of the scope of the book. But the headline idea here is that the event variable turns out to be restricted (by the verbal predicate) in all logical forms based on event semantics, and furthermore, restrictedness seems to obtain even if we cast our net wider by considering other non-individual domains like times (restricted by temporal adverbs or tense morphemes) and possible worlds (restricted by conditional clauses or accessibility relations). If our GCU is on the right track, any variable of any type in any logical form will end up restricted (with an important caveat about focus, which we will discuss in subsection 10.3.2; but as we shall see, this caveat equally applies to the domain of standard individuals).

In a nutshell, GCU is a very strong empirical hypothesis, apparently interacting with every type of quantificational structure in natural language, playing a role in each and every semantic analysis of any natural language phenomenon. To exhaustively test GCU, one would need to show that each and every (successful) proposal involving quantification (over non-individuals) employs only conservative logical forms, or that it may be recast in such a form. Again, carrying out such an extensive taxonomic investigation is outside the scope of this book, but we welcome any such empirical investigations. Our conjecture is that GCU would survive the test. As in the case of quantification over individuals, quantifiers over other domains are typically seen as binary in the literature, the two arguments being the restriction and the nuclear scope—see e.g. von Fintel (1994) for an analysis of adverbial quantifiers (over situations)—a situation which would make no sense without underlying conservativity.

Summing up: As a consequence of generalized conservativity, which applies globally (to logical forms) rather than locally (to determiners), our generalization (GCU) of Conservativity Universal automatically applies across the entire ontology, whatever that ontology might turn out to be. Adding events or other non-standard individual types to the ontology changes the particulars of the requirements for conservativity and restrictedness in the sense that the non-standard individual variables occurring in logical forms must be restricted as well. And this lands us precisely on the central idea: The logical forms of natural language sentences are always conservative/restricted with respect to *some* set of predicates.

This gives us a clean and quite strong working hypothesis. But despite its clarity, it is not at all trivial, and in point of fact we immediately encounter several apparent counterexamples. In the next section, we will discuss three potential counterexamples to this claim. As we will see, they arguably dissolve upon a more careful syntactic and/or semantic analysis.

10.3 Apparent Counterexamples to GCU and RU

In the previous section we proposed that a very simple generalization applied to all well-formed natural language constructions. Or more precisely, there are two generalizations, one semantic and the other syntactic. The semantic generalization extended the earlier claim by Keenan and Stavi (1986) that natural language determiners are conservative into the claim that conservativity is a property of logical forms, and that all natural language logical forms are conservative. We called this Generalized Conservativity Universal (GCU). The syntactic generalization holds that all well-formed natural language constructions obey restrictedness. We called this Restrictedness Universal (RU). The generalizations are equally elegant, as they should be, since they are two sides (semantic side and syntactic side) of the same coin. However, there are some apparent counterexamples. In this section, we will take a look at three potential counterexamples to Generalized Conservativity Universal and Restrictedness Universal. The first two (non-intersective adjectives and focus) target the weaker GCU; the third one (disjunction) applies only to the stronger RU. We aren't claiming that these exhaust all apparent counterexamples, but showing that they are not in fact counterexamples should give us an insight into how we might respond to any additional related objections.

10.3.1 Non-intersective adjectives

The first apparent problem with GCU stems from the narrower "zoom level" of generalized conservativity. As pointed out in subsection 9.1.3, GQ conservativity applies to arguments of determiners, which are noun phrases (wider zoom level), while generalized conservativity applies to nouns themselves (narrower zoom level). This is so because we have defined logical forms to be conservative with respect to individual logical predicates (and collections thereof).

The apparent problem involves non-intersective adjectives, so let us first look at a case of a complex noun phrase involving a non-problematic intersective adjective. GQ conservativity can only zoom down to 'black cat' in (8), as that is the complement (and thus the inner argument) of determiner 'a'. Generalized conservativity must zoom further in, for example down to 'cat' itself. Consequently, the standard analysis tells us that (8) is about black cats, whereas for generalized conservativity, it is about cats.

(8) John owns a black cat. $\exists x \colon (B(x) \wedge C(x)) \wedge O(j, x)$

At first sight, it might seem that standard conservativity tells us more here, but this is not the case. True, generalized conservativity must zoom in to a constituent smaller than the noun phrase, but it can zoom in to either the noun 'cat' or the adjective 'black'. Using generalized conservativity, we can therefore see that the sentence is both about cats and about black things, but this also tells us that the sentence is about black cats. This conclusion rests on the intersectivity of the adjective 'black', i.e. the fact that black cats are cats which are black.

When it comes to non-intersective adjectives, the detailed statement yielded by generalized conservativity seems wrong at first sight. A former president is not necessarily a president and an alleged murderer is not necessarily a murderer. Thus, while the sentences in (9) are about (John, knowing and) former presidents and alleged murderers (as predicted by the GQ account), they are not about presidents and murderers (as predicted by our account at first sight). Does generalized conservativity have it wrong?

(9) a. John knows a former president.
    b. John knows an alleged murderer.

No, these facts simply indicate that there must be more to noun phrases 'former president' and 'alleged murderer' in (9) than to the noun phrase 'black cat' in (8), and indeed, adjectives such as 'former' and 'alleged' are recognized as non-intersective in the literature (and most often excluded from considerations of conservativity by stipulation). It is simply incorrect to analyze the sentences in (9) in the same way as (8), for such an analysis yields incorrect truth conditions. A truth-conditionally suitable analysis of the problematic noun phrases along the lines of "a person who used to be a president" and "a person who was alleged to be a murderer" would reveal that their individual variable (x) is restricted (by 'person') and that they contain quantification over other domains as well. For example, in (9a), we find quantification over times ('use*d* to') and states ('be a president'), and in (9b), we quantify over times ('was') and events ('alleged'), etc. The point here is that the logical forms of these noun phrases clearly resemble the logical forms of full sentences 'This person used to be a president' and 'This person was alleged to be a murderer', which we have no trouble recognizing as conservative.

A similar argument could be made for 'non-smoker', which is non-conservative if we analyze it simply as $\neg S(x)$, but turns out conservative if we analyze it along the lines of "a person who does not habitually smoke." In all probability and in line with much contemporary (semantic and syntactic) research,[2] even deverbal nouns 'murderer' and 'smoker' themselves should be analyzed as "a person who murdered someone" and "a person who habitually smokes."

The apparent problem of non-intersective adjectives (and negated nouns) can thus be resolved with a sufficiently detailed semantic and syntactic analysis involving non-individual domains. However, there is still a glitch. The above paraphrases of sentences including non-intersective adjectives and negated nouns are all of the form 'a person who', leading to the conclusion that these noun phrases are conservative with respect to 'person'. We have a strong intuition that (8) is about cats (and John and perhaps owning), but do we have an equally strong intuition that the sentences in (9) are about people (and John, etc.)? We will discuss these intuitions in section 10.4; right now, we turn to what is surely the most interesting apparent counterexample to our Generalized Conservativity Universal: focus.

10.3.2 Focus

We already touched on focus in the previous chapter, when we talked about 'only'. At first sight, 'only' is like 'every' but with reversed arguments, as shown in (10);

[2] One can see such research as developing the main idea of generative semantics—when supported by syntactic data.

sentence (10b) can be paraphrased by 'every barker is a dog'. However, it is well-known that 'only' is an operator associated with focus. The argument reversal is just a special case which obtains when 'only' is complemented by a noun phrase and the entire complement is focused. In general, any constituent of its complement can be focused. In subsection 9.1.3, we followed Herburger (2000) in proposing that focal mapping rearranges the quantificational structure, leading to more complex logical forms such as (11).[3]

(10) a. Every dog barks. $\forall x\colon \neg D(x) \vee B(x)$
b. Only dogs bark. $\forall x\colon \neg B(x) \vee D(x)$

(11) Only angry dogs bark. $\forall x\colon \neg(D(x) \wedge B(x)) \vee ((A(x) \wedge D(x)) \wedge B(x))$
$\sim \forall x\colon \neg(D(x) \wedge B(x)) \vee A(x)$

Both above examples with 'only' are conservative: (10b) is *B*-conservative (it is about barkers) and (11) is both *D*-conservative and *B*-conservative (it is about dogs, and it is also about barkers). However, these are yet again special cases. The following logical form, where the verbal predicate is negated, is not conservative.

(12) Only penguins don't fly. $\forall x\colon \neg\neg F(x) \vee P(x)$

It is easy to see that (12) is not *F*-restricted. Because $F(x)$ is negated twice and then connected to the nuclear scope $P(x)$ by a disjunction, it negatively p-scopes over the nuclear scope. Not being *F*-restricted is a strong hint for not being *F*-conservative, and indeed this turns out to be the case. Restricting attention to fliers can result in a changed truth value. The sentence can only be true if every object in the domain either flies or is a penguin; if we add an elephant (a non-flier), the sentence becomes false.

We believe that the road to solution is through the discourse. In a neutral context, (12) is most probably about birds (and in a particular context, it might be about several specific species of birds), asserting that non-flying birds are penguins. We will formalize this idea presently, but let us first provide another example of a seemingly non-conservative sentence that will involve focus but not the use of 'only'. This will make clear that it is focus rather than 'only' itself that is making things interesting here.

The example is from Herburger (2000, pp. 29–34, 50–58). 'Sascha didn't visit Montmartre' is a negative sentence containing a focused element, 'Montmartre'. Among the several possible readings of this sentence, we are most interested in the contrast between what Herburger calls the *bound* (13) and the *free reading* (14). Their paraphrases are given in (13a) and (14a), and to help you get a grip on the two readings, it is worth mentioning that they differ in the intonational pattern on

[3] When analyzed using focal mapping, (10b) also receives a more complicated (but logically equivalent) logical form: $\forall x\colon \neg B(x) \vee (D(x) \wedge B(x))$.

the focused element: in the bound reading, the focus receives a fall-rise (˜) contour, while the free reading is associated to a fall (`) contour.

(13) Sascha didn't visit ˜Montmartre. (bound reading)
  a. "What Sascha visited wasn't Montmartre."
  b. … but the Louvre.
  c. $\exists x\,[V(s,x)]\,\neg M(x) \wedge V(s,x)$

(14) Sascha didn't visit `Montmartre. (free reading)
  a. "What Sascha didn't visit was Montmartre."
  b. … and not the Louvre.
  c. $\exists x\,[\neg V(s,x)]\,M(x) \wedge \neg V(s,x)$

The continuations in (13b) and (14b) can serve as a further aid toward disambiguation. Accordingly, the two readings are felicitous in different pragmatic contexts. As aptly described by Herburger (2000, pp. 29–30),

> the bound reading appears when we are assuming that Sascha engaged in some sight-seeing and we are wondering what in particular he saw. The free reading becomes readily available if we imagine uttering [(14)] in a different context, one where it counts as established that Sascha is shunning the main tourist sites of Paris and we are wondering which one he managed to avoid on a particular, contextually salient occasion.

Moving on to the logical forms of the two readings (13c) and (14c), we follow Herburger's suggestion that they differ with respect to the scope of negation (as reflected in the terms "bound" and "free" reading). In the bound reading (13c), the negation scopes over the focus, while in the free reading (14c), it is the verb phrase which is negated.[4,5]

[4] Remember that the background (the non-focused part of the sentence) is duplicated in Herburger's analysis. This is why $V(s,x)$ occurs both in the restrictor and in the nuclear scope of $\exists x$ in (13c) and (14c).

[5] We depart from Herburger's logical form for (14) in several respects. A minor issue is that we simplify it by disregarding event semantics. More importantly, we believe that her logical form (i) states incorrect truth conditions. (Predicate C should be interpreted as a restriction to contextually relevant individuals.)

(i) $\exists e\,[C(e) \wedge \neg V(e) \wedge \text{Agent}(e,s) \wedge \text{Past}(e)]\,\text{Theme}(e,m) \wedge \neg V(e) \wedge \text{Agent}(e,s) \wedge \text{Past}(e)$
"Some (relevant) past event of not-visiting by Sascha was a past event of Sascha not-visiting Montmartre."

This logical form comes out as true only when there is a (contextually relevant) event involving Sascha as the agent and Montmartre as the theme which is not an event of visiting; for example, an event of Sascha writing home about Montmartre. But (14) can be true even when there is no such event—Sascha might have had nothing to do with Montmartre. The correct logical form (ii) states that Montmartre is among the locations (alternatives to Montmartre, perhaps the tourist sites of Paris) not visited by Sascha. (This logical form includes the alternatives predicate $\mathbb{A}$ we will introduce a bit later in the main text.)

Here comes the crucial observation. While we could in principle see (13c) as restricted by $V(s, x)$ (but remember that strictly speaking, we have only defined conservativity and restrictedness with respect to monadic predicates), (14c) cannot possibly be restricted by this atomic formula, because $V(s, x)$ has negative polarity in the free reading. So if we are on the right track when we claim that every natural language expression *is* conservative/restricted, (14c) cannot be the correct logical form of (14). But what could we add to (14c) to make it (specifically, $\exists x$), restricted?

The idea is that the primary function of focus is to introduce a *set of alternatives* into a discourse. Above, the alternatives to Montmartre might be the other tourist attractions of Paris. This idea is familiar in the semantics literature, and has received an influential formalization in Rooth (1985); the framework is usually called *alternative semantics*. To bring the idea to life within our approach, we assume that focusing introduces a context-dependent predicate $\mathbb{A}$, for "alternatives."[6]

(15) Sascha didn't visit <u>Montmartre</u>.
a. $\exists x\, [\mathbb{A}(x) \wedge V(s, x)]\, \neg M(x) \wedge V(s, x)$ (bound reading)
b. $\exists x\, [\mathbb{A}(x) \wedge \neg V(s, x)]\, M(x) \wedge \neg V(s, x)$ (free reading)

Introducing $\mathbb{A}(x)$ preserves the restrictedness (and thus conservativity) of the free reading—variable x is now restricted by $\mathbb{A}(x)$. For consistency, we clearly want to introduce $\mathbb{A}$ into the logical forms of both readings—we don't see the addition of $\mathbb{A}$ as some sort of last resort repair strategy, but as an independently motivated and quite standard account of focusing a constituent. So the question is, does the added $\mathbb{A}(x)$ play well with the bound reading? It does. The bound reading evokes the alternatives to Montmartre as well, it is just that the free reading seems more

(ii) $\exists x\, [\mathbb{A}(x) \wedge \neg\exists e\colon V(e) \wedge \mathrm{Agent}(e, s) \wedge \mathrm{Theme}(e, x) \wedge \mathrm{Past}(e)]$
$M(x) \wedge \neg\exists e\colon V(e) \wedge \mathrm{Agent}(e, s) \wedge \mathrm{Theme}(e, x) \wedge \mathrm{Past}(e)$
"Some individual which is not a theme of Sascha's past visits is such that it is Montmartre (and it is not a theme of Sascha's past visits)."

In general, we disagree with Herburger's assumption that focusing always modifies the structure of the quantifier over events. We rather assume that it modifies the structure of the quantifier binding the variable introduced by the focused element, which may or may not be the event variable.

Finally, note that we do not include Herburger's contextual restrictor (C) in our logical forms. First, if we included C in the restriction of every quantifier, every quantifier would be automatically restricted, which would deprive us of the explanatory power of restrictedness. Second, we believe that the fact that every instance of quantification in natural language is limited by the context is in fact an argument against this restriction being expressed in the logical form. We assume that in principle, every element of a logical form can be either present or not, so having C occur in some logical forms, we'd expect there to be other logical forms without C. In other words, we'd expect context-insensitive sentences to exist, but this is surely not the case.

[6] The set of alternatives depends on the focused constituent. In our example, it is the alternatives to Montmartre that are relevant, so we should have really used $\mathbb{A}_M$.

forceful in this respect. By and large, we agree with Herburger (2000, p. 34) as to why this might be the case:

> The free reading also seems quite marked, I think because it involves a negative description of an event. [. . .] unless such descriptions are interpreted relative to a rich context, they run the danger of being vacuously true of all sorts of irrelevant things.

Both readings are $\mathbb{A}$-restricted. However, as $V(s, x)$ of the bound reading will generally filter out more alternatives than $\neg V(s, x)$ of the free reading, the context (which determines $\mathbb{A}$) may be weaker for the bound reading. The free reading, on the other hand, will prefer a more constrained set of alternatives provided by a rich context.

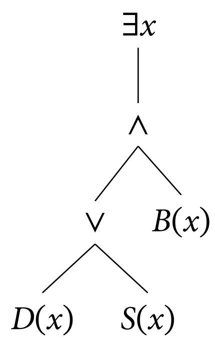
The above proposal is not original (it is standard stuff, in fact), but it does help us illustrate a very important idea. The rearrangement of quantificational structure (for example as implemented by Herburger's focal mapping) is not the only effect of focus. Focus has another task. Focus ensures conservativity by restricting the rearranged quantifier with the alternatives predicate. The most important job of focus is to ensure the conservativity of the expression in which it occurs, and it accomplishes this by making sure that the resulting logical form satisfies Restrictedness Universal.

10.3.3 Disjunction

The joint or multi-predicate conservativity of (16), defined in subsection 9.1.3, comes about because the verb has more than one argument position, which makes the logical form contain more than one variable. However, this is not the only possible source of joint conservativity. Another source are disjunctively coordinated noun phrases as in (17) below.

(16) Some dog is gnawing a bone. $\exists x \colon D(x) \wedge \exists y \colon B(x) \wedge G(x, y)$

(17) Some dog or seal is barking. $\exists x \colon (D(x) \vee S(x)) \wedge B(x)$



The latter sentence is neither exclusively about dogs nor seals—it is about dogs and seals, jointly. We may restrict the domain to the union of dogs and seals—and clearly retain the truth value—but no further. For example, if we have a single barking individual and this individual is a dog, (17) is true; however, if we restrict the domain to seals, it becomes false. Therefore, the sentence is neither 'dog'-conservative nor 'seal'-conservative. It is only {'dog', 'seal'}-conservative.

The logical form in (17) is conservative and thus not a counterexample to Generalized Conservativity Universal. However, it does raise some interesting questions for Restrictedness Universal, if restrictedness referred to in RU is understood as the p-scope-based strong restrictedness, defined in subsection 9.2.1.

The issue is this: strong restrictedness is limited to a single predicate per quantifier/variable, in the sense that it cannot characterize a quantifier as jointly restricted by two predicates—remember that a restrictor of a quantifier is an *atomic* formula which occurs in the restriction and positively p-scopes into the nuclear scope. Strong restrictedness has no chance of characterizing the joint conservativity of disjunctive structures by its very design—by definition, the source of the positive p-scope must be an atomic formula, and an atomic formula contains a single predicate.

As far as the negative characterization goes, however, strong restrictedness yields the correct result for (17). Looking at the arboreal representation of its logical form, it is easy to see that neither $D(x)$ nor $S(x)$ p-scope over $B(x)$. Starting from either $D(x)$ or $S(x)$, Perceval cannot get over the sequence of $\vee$ and $\wedge$. Quantifier $\exists x$ thus has no restrictor. This is a nice result, as restrictedness with respect to some predicate, say D, would imply conservativity with respect to that predicate, and we have seen that such single-predicate conservativity does not hold.

We should also emphasize the following: The fact that the logical form in (17) can be characterized as conservative but not restricted does not contradict our Restrictedness Theorem from section 9.2. That theorem only states that every conservative formula is logically equivalent to some restricted formula, and indeed, as we shall see, the logical form in (17) is logically equivalent to (18b) below, which *is* strongly restricted. The real issue with (17) is that it appears to violate RU, which states that the logical *form* of any natural language sentence will be restricted. So once again, there is more here than meets the eye, and we will be well served by a deeper dive into syntax. We submit that RU can help to illuminate the logical form of natural language constructions like these, and in the process help us to unravel some long-standing linguistic puzzles.

There are innumerable proposals regarding the syntactic and semantic status of 'or' and its sibling 'and' (most often focusing on the latter). The research on these items includes their relation to quantifiers, negation, plurals, collective vs. distributive verbs, etc., and spans not decades but millennia. Lasersohn (1995, p. 12) attributes the first such observations regarding 'and' to Aristotle, and consequently, 'and' also caught the attention of Medieval logicians. The debate as to which approach (or a combination thereof) to their meaning might be best continues to the present day.

One idea is that natural language conjunctive and disjunctive noun phrase coordinators 'and' and 'or' correspond to predicate logic connectives $\wedge$ and $\vee$, but that the logical connectives connect sentential rather than nominal logical forms. To make this work, it had been proposed that (17) is syntactically derived (in a meaning-preserving way) from (18).[7] The first such proposals employed the so-called *conjunction reduction* transformation rule, originating from Chomsky (1957, p. 36). A contemporary approach was developed in detail for conjunctions in Schein (2017).

(18) a. Some dog is barking or some seal is barking.
b. $\exists x_1\,(D(x_1) \wedge B(x_1)) \vee \exists x_2\,(S(x_2) \wedge B(x_2))$

By containing two instances of quantification, a logical form along the lines of (18) would clearly resolve the difficulties we encountered for RU and its demand that the source of the positive p-scope be an atomic formula.

There are, of course, other approaches here, and one very popular idea builds on the thought that 'and' is an operator creating plural (or groups of) individuals, see e.g. Link (1983) and Lasersohn (1995). We can then say that 'and' and 'or' denote meet and join operating (facilitated by type shifting) on any boolean type (Partee and Rooth 1983; Keenan and Faltz 1985).

We are inclined to be attracted to Schein's approach, as it offloads the issue into the syntax in a way that we find compelling. However it is worth noting that there is no obvious barrier to executing the meet and join operations within our framework. This would require that we say some things about how a single predicate can be syntactically constructed from two predicates using the 'and' and 'or' operators. The semantics would proceed as in the standard story.

From the morphosyntactic perspective, one additional observation about 'and' and 'or' stands out: they might not be simplex items at all. For example, Mitrović (2021, pp. 164–178) proposes that exclusive disjunction is composed of no less

[7] Of course, the derivation of nominal disjunction from sentential disjunction cannot be purely mechanical if we are to preserve the meaning. For example, if we substitute 'every' for 'some', the nominal disjunction is not logically equivalent to sentential disjunction but to sentential conjunction, and an analogous remark holds for nominal conjunction. Having absorbed the lessons from our Dynamic Deductive System, it is clear that the transformation is highly sensitive to the polarity of the nominal disjunct, which is positive in the case of 'some' and negative in the case of 'every'; see also the discussion on (anti-)additivity and (anti)-multiplicativity in section 11.3.

(i) Every dog or seal is barking.
a. $\not\Leftrightarrow$ Every dog is barking or every seal is barking.
b. $\Leftrightarrow$ Every dog is barking and every seal is barking.

than five distinct syntactic heads (corresponding to five semantic operators). A common thread in research decomposing these connectives is that conjunction and disjunction are related to universal and existential quantification, respectively. For example, here is how Szabolcsi (2015, p. 162) summarizes the data using the Japanese particles 'mo' and 'ka':

> Regarding the question whether the roles of each particle form a natural class with a stable semantics, a beautiful generalization caught the eyes of many linguists working with data of this sort [...] In one way or another, the roles of 'ka' involve existential quantification or disjunction, and the roles of 'mo' involve universal quantification or conjunction.

This generalization shouldn't be too surprising if you remember the Medieval theory of Supposition on which universal quantification was a long conjunction and existential quantification a long disjunction (even if not described in those terms).

The moral is that 'and' and 'or' might not correspond to Boolean connectives $\wedge$ and $\vee$, and even if they do, the location of the Boolean connectives in the logical form might not straightforwardly correspond to the surface location of the natural language connectives. In light of this moral, the objection to RU based on the fact that the formula in (17) is not strongly restricted loses much of its force, as this formula might not be the logical form of the sentence in (17), or indeed any natural language sentence.

But the most interesting observation to draw, in our view, is that the characterization of conservativity by strong restrictedness raises questions precisely with structures that have been receiving conflicting analyses in the linguistic literature for a long time. What this means is that when used as a methodological principle, RU should be able to bring the nature of certain linguistic puzzles into relief, appreciate their depth, and help us to think about them in new ways.[8]

10.4 Aboutness Revisited

We began our exploration of conservativity and restrictedness in the previous chapter by introducing the informal notion of aboutness. We first reminded ourselves that languages employ a variety of syntactic means for catching our

[8] A less appealing, but certainly possible option, would be to relax the notion of restrictedness in play. In the Appendix, we explore this idea and offer our most general variant of restrictedness—we call it WEAK RESTRICTEDNESS—which correctly characterizes disjunctive structures such as (17) as jointly restricted with respect to several predicates, but does not rely on p-scope.

attention. For example, sentence (19) can be understood as being about cats because its subject, or perhaps the topic, is 'a cat'.

(19) A cat is hunting a mouse. $\exists x\colon C(x) \wedge \exists y\colon M(y) \wedge H(x, y)$

It is easy enough to see what the formal reflex of this understanding of aboutness might be, in both our RU account and in the GQ account. In our account, the natural hypothesis seems to be that it is the restrictor of the highest quantifier in the logical form of the sentence that characterizes aboutness in its attention-catching sense. In the GQ account, the highest generalized quantifier is relevant, with the caveat that generalized quantifiers over non-nominal domains count, as 'Yesterday, a cat was hunting a mouse' is probably about what was happening yesterday.

However, at what we might call a greater level of detail, a sentence is not just about its subject or topic, but about the denotations of all the nouns (or noun phrases; we ignore this detail for a moment) contained in the sentence. For example, (19) is about cats and mice. This understanding of aboutness corresponds to the traditional GQ notion of conservativity, which applies to determiners and therefore to nouns, but it would not be hard to formalize it using generalized conservativity, under the assumption that we can somehow tell which variables are introduced by noun phrases. If we have this information, inspecting for restrictedness while limiting the attention to nominal variables yields the desired result.

Considering restrictedness or generalized conservativity without limitation to nominal domains results in an even more detailed understanding of aboutness. Assuming that the sentence under discussion is analyzed using event semantics, as in (20), an understanding of aboutness is produced where the sentence is about cats, mice, and hunting. And it seems to us that such an understanding resonates with our linguistic judgments as well.

(20) A cat is hunting a mouse.

$$\exists x\colon C(x) \wedge \exists y\colon M(y) \wedge \exists e\colon H(e) \wedge \text{Agent}(x, e) \wedge \text{Theme}(y, e)$$

However, our judgments get uncertain when the logical form includes further details. Let us switch to a simpler example involving tense, as shown in (21) below—the analysis of tense is provisional (here and later in the section), but it will do for our purposes.

(21) A dog barked. $\exists t\colon \text{Past}(t) \wedge \exists x\colon D(x) \wedge \exists e\colon B(e) \wedge \text{At}(t, e) \wedge \text{Agent}(x, e)$

According to generalized conservativity, this sentence is about dogs, barking, and past times (and here we want to concentrate on the last case—past times).

Is that something we judge to be correct? Perhaps, although past times are clearly not as salient as dogs in this case. We more readily judge that (21) is about dogs (and barking) than about time points or intervals preceding the time of utterance.[9]

A similar point could be made for other ontological domains. Even if GCU holds in general, it seems that we do not immediately judge a sentence to be about something from each ontological domain reflected in its logical form. Such judgments seems to arise more readily for variables restricted by a predicate corresponding to a content word (noun, verb, adjective, or adverb). For example, adverbial modification certainly helps in the case of tense.

(22) A dog barked yesterday. $\exists t\colon (\mathrm{Past}(t) \wedge Y(t)) \wedge \exists x\colon D(x) \wedge$
$\wedge\, \exists e\colon B(e) \wedge \mathrm{At}(t, e) \wedge \mathrm{Agent}(x, e)$

Content words seem to be important in the individual domain as well. An occurrence of 'dog' in the sentence induces a strong feeling of the sentence being (also) about dogs. Morphemes '-one' and '-thing', as in 'someone' or 'everything', seem to produce a weaker effect. We're not sure to what extent speakers might be willing to concede that sentences containing them are (also) about people or things.

Of course judgments about aboutness can be heavily influenced by priming effects. In (21) and (22) the word 'dog' is right there, after all, so it is easy to see why a rapid judgment of these sentences being about dogs is forthcoming. Then too, events and times and other entities are somewhat more exotic, and certainly the product of scientific theorizing, so the vocabulary of semantics may not be ideal for prompting the relevant judgments.

So, for example, English speakers might not judge (21) to be about past times, but they may be willing to judge that it is about the past or about something that happened in the past. The thought is that while we might be on the same page in terms of what (21) is about, we may not be on the same page regarding how to describe that aboutness, especially when it comes to technical vocabulary we use to describe it.

We can see generalized conservativity as being manifest in speakers' judgments of aboutness. As we noted, this does not imply that there is complete uniformity—some judgments of aboutness are much more salient (topicality, for example, induces the most robust judgments of aboutness). Furthermore, we need to exercise caution in understanding how those judgments are packaged by speakers; for example, it is easier to affirm that (21) is about "something that happened" than it is to affirm that it is about "past time points" or "past events." These caveats in

[9] Of course, another possibility is that the judgments falling as they do tells against the accounts of tense that involve referents to past and future times and intervals, and tells in favor of operator theories, for example. We set that issue aside for other books.

place, we can highlight the superficially trivial, but in fact very deep observation that speakers judge sentences to be about something at *some* level.

All of this leads to the very interesting question of what happens if a sentence contains a mix of locally conservative and locally non-conservative operators—where here by "locally conservative" we mean conservative in the GQ sense: conservative relative to the arguments of an operator. And what happens is that a locally non-conservative operator can destroy the aboutness effect of the locally conservative operators in the sentence. In other words, a logical form is certain to be globally conservative only if *all* the quantificational operators used to construct it are locally conservative. If a locally non-conservative operator occurs in the sentence, we can get a globally non-conservative logical form. To see how this happens, let us take a look at a sentence involving an imaginary tense morpheme '-smomes' inspired by the non-existent non-conservative determiner 'smome' we have defined in subsection 9.1.2. The meaning of '-smomes' (24a) only slightly differs from the meaning of the present tense morpheme '-s' (23a)—we have replaced the conjunction with the disjunction—but the effect on global conservativity is destructive.[10]

(23) a. $[\![\text{-s}]\!](\varphi) := \exists t\colon \text{Present}(t) \wedge \varphi$, where t is free in φ
b. A dog barks. $\exists t\colon \text{Present}(t) \wedge \exists x\colon D(x) \wedge B(x,t)$

(24) a. $[\![\text{-smomes}]\!](\varphi) := \exists t\colon \text{Present}(t) \vee \varphi$, where t is free in φ
b. A dog bark-smomes. $\exists t\colon \text{Present}(t) \vee \exists x\colon D(x) \wedge B(x,t)$

Sentence (24b) asserts that there is either a present time or a time at which there is a barking dog. The mere existence of the present time makes this sentence true, even if there are no barking dogs in the present, or even if there are no dogs at all; only when the present time does not exist, or perhaps when it is not contextually relevant, can the barking dogs make a difference. We find the intended meaning of (24b) distinctly odd. We would certainly not characterize this sentence as being about dogs (and barking).

Another way of looking at this example is through the lens of context change. Assume that the sentence is true by virtue of there being a single barking dog at some non-present time (and there is no present time in the domain either). If the logical form was conservative with respect to dogs, barking, and the present time, we should be able to take that time point away without changing the interpretation. However, this is clearly not the case: taking away that time point will make $B(x,t)$ and thus the entire logical form false.

Above, $\exists t$ scoped over $\exists x$. While the details change if the scope relation is reversed, as in (25), the conclusion does not. Sure, there being a dog is now a

[10] For simplicity, we don't employ event semantics here. This does not affect the argument.

necessary condition for the truth of the sentence, but yet again it turns out that reducing the domain by taking away an object that is neither a dog nor the present time moment can change the interpretation of the sentence from true to false. As above, we assume that the sentence is true in virtue of there being a single dog barking at some non-present time point, and then take that time point away.

(25) A dog bark-smomes. $\exists x\colon D(x) \wedge \exists t\colon \text{Present}(t) \vee B(x, t)$

Even a Boolean junction of a conservative and a non-conservative logical form can result in non-conservativity. Let us illustrate this using the imaginary non-conservative determiner 'nall' we have met in section 3.3 and subsection 9.1.2. Remember that 'Nall circles are striped' is intended to mean something like "Everything but circles is striped." Conjoining this non-conservative sentence with the conservative 'All dogs bark' results in a non-conservative conjunction. For example, if (26) is true (entailing, incidentally, that all dogs are striped) and we add a non-striped baloon, it becomes false.

(26) All dogs bark and nall circles are striped.

$$(\forall x\colon \neg D(x) \vee B(x)) \wedge (\forall y\colon C(y) \vee S(y))$$

We could replicate these examples *ad nauseam*. The point we're making here is that containing a locally conservative operator is not sufficient to make a logical form conservative with respect to whatever that operator quantifies over. We have to consider the effect of all operators, and a locally non-conservative operator can destroy the local conservativity of the other operators in the sentence. To put it prosaically, it prevents their local (argument-based) conservativity from shining through at the level of logical form.

All this is very easy to see by considering restrictedness instead of conservativity. The definition of restrictedness (which provably corresponds to generalized conservativity) says that a formula is restricted iff *every* quantifier (variable) in the formula is restricted. Having a single non-restricted variable wreaks havoc with restrictedness (and thus conservativity) of the entire formula.

However, our experience with aboutness is different. If a sentence contains (a referential use of) the noun 'dog', we judge the sentence to be about dogs (among other things), regardless of how complex the sentence is and how the determiner complemented by 'dog' interacts with other operators. As the examples above show, this state of affairs could not obtain if sentences contained non-conservative operators.

So what does all this tell us about the validity of GCU? The usual test for the validity of the original Conservativity Universal is the logical equivalence of sentences of form [[D NP] V] and [[D NP] [NP and V]]. However, in our opinion this test is merely a shallow probe. It tests for local nominal aboutness, but, as we have seen, nominal aboutness holds *globally* as well. And we have shown above that this could not be the case if all other, non-nominal quantificational operators were not conservative as well. In other words, CU cannot possibly hold if GCU does not hold as well.

We hope to have convinced you that Generalized Conservativity Universal and Restrictedness Universal play important roles in one of the most basic properties of language—its ability to reliably express thoughts about specific things. But RU is important for more than just being the syntactic expression of this property. It also places powerful constraints on the class of possible natural language logical forms. In the next two chapters we will see just how fruitful these generalization are. We will see that once we take on the constraint it will deepen our understanding into some longstanding puzzles at the syntax–semantics interface. Indeed, wielding this new constraint, we will begin to unravel those puzzles.

11
Linguistic Phenomena through the Lens of L**

In Chapter 9 we saw that the semantic notion of conservativity can be cashed out as the syntactic notion of restrictedness in our system. Or, if you prefer, we can say that the semantics and syntax run in parallel, with conservativity cashing out the phenomenon on the semantic end and restrictedness cashing it out on the syntactic end. In the next chapter, we want to take this talk of syntax and show that it can also be integrated with the notion of syntax employed within generative linguistics. Before we get there, however, it will be worth our while to pause and consider how certain canonical linguistic problems unfold within L** syntax, in particular L** syntax constrained by restrictedness.

What we are going to learn is that our discussion of restrictedness in Chapters 9 and 10 is going to help us unlock some hard problems in discourse anaphora, that our Dynamic Deductive System may be instrumental in understanding presupposition projection, and that the Boolean nature of L** syntax, when wedded to our account of polarity, can offer a new perspective on the puzzle of Negative Polarity Items (NPIs).

In the next chapter we will get into the nitty gritty of how the syntax of L** can have expression in the syntax of natural language, but in the meantime it should be helpful to consider these problems at a certain level of abstraction. First, it will help us to see quite clearly how our new understanding of restrictedness as expressed in Restrictedness Universal in the previous chapter can have immediate implications within linguistic theorizing. Second, this abstract level of analysis should help people to take these ideas into a number of alternative syntactic frameworks. Once we get into the matter of embedding the analysis within a particular linguistic framework, some less ecumenical choices have to be made. So our plan is to be ecumenical for one more chapter, since it will not prohibit us from showing the elegant relation between Restrictedness Universal and the problem of discourse anaphora. Similarly, it will not prevent us from showing how a broad palate of NPI phenomena can be cashed out in our system.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović, Oxford University Press. © Peter Ludlow and Sašo Živanović 2022. DOI: 10.1093/oso/9780199591534.003.0011

11.1 Discourse Anaphora

The theory of discourse anaphora is sometimes considered one of the more technical and, dare we say it, anally retentive topics in the philosophy of language. It certainly doesn't have the cachet that studying Wittgenstein does. The core example, from Geach (1962), remains a central example in the problem space, even though it reads like someone's parody of technical linguistics and philosophy.

(1) Every farmer who owns a donkey likes it.

Nevertheless, there is a reason that philosophers and linguists have obsessed over this example and the problem it exposed over half a century ago. The reason that this example and others like it have garnered so much attention in linguistics and the philosophy of language is that despite their apparent simplicity, the examples seem to defy our deepest and best theories of the logic and semantics of natural language.

This isn't a book about discourse anaphora, but we feel it is worth pointing out that the theory proposed in the previous chapters has a lot to say about this and related cases. And what it has to say is that the problem (or at least much of it) dissolves of its own accord once we realize that Restrictedness Universal places heavy constraints on natural language logical forms, and once we understand how Restricted Closure enforces these constraints.

To set this up, we will engage in a little history of the problem space, albeit a coarse-grained history. Just hitting the key notes will be sufficient for our current purposes.

Geach observed that there are sentences in which indefinite descriptions appear to have the force of universally quantified noun phrases. The most famous example, no doubt, is example (1), produced above. Geach noted that if the indefinite 'a donkey' is treated as an existentially quantified noun phrase and 'it' as a variable, the indefinite will not be able to bind the pronoun, as the pronoun will lie outside the scope of the quantifier. This is so because (in modern terms) a quantificational phrase is only assumed to bind (in the logical form) pronouns it c-commands (in the syntactic representation). The logical form of (1) will come out as (2), and the problem is that (2) leaves the final occurrence of y unbound.

(2) $\forall x\colon (F(x) \land \exists y\colon D(y) \land O(x, y)) \Rightarrow L(x, y)$

On the other hand, if 'a donkey' is given wide scope (let's say by quantifier raising), the result, given in (3), is the less salient reading which might be paraphrased as "There is a donkey (e.g. Eeyore) which is liked by every farmer who owns him." This reading—which we are not interested in, so we will ignore it in our discussion—becomes much more salient in the presence of items such as 'certain'.

For example, it is readily available in sentence 'Every farmer who likes a certain donkey feeds it.'

(3) $\exists y\colon D(y) \wedge \forall x\colon (F(x) \wedge O(x,y)) \Rightarrow L(x,y)$

Geach's solution amounts to treating 'a donkey' in (1) as a *universally* quantified noun phrase with wide scope. More precisely, the quantifier over y has the same (wide) scope as the quantifier over x in (4); in fact, the two quantifiers share the restriction.[1]

(4) $\forall x \forall y\colon (F(x) \wedge D(y) \wedge O(x,y)) \Rightarrow L(x,y)$

But Geach's proposal is considered to be ad hoc, and for several reasons. First, it is only in certain environments that "universalization" of indefinites occurs. The existential reading of indefinites, as in 'a donkey brays', analyzed as $\exists x\colon D(x) \wedge B(x)$, is certainly more common (cf. Kamp 1981, p. 192). Next, there is no independent motivation for the universalization to happen. It seems to be introduced just to fix the donkey anaphora problem.[2] Finally, what determines the scope of a quantificational noun phrase if not c-command?

Numerous ways of addressing these and related questions have been proposed in the literature. However, most of the proposals fall within one of two categories. The proposals from the E-type anaphora family of approaches follow Evans (1977, 1980) in the assumption that the discourse pronouns under discussion do not stand for bound variables after all. The proposals from the Discourse Representation Theory (DRT) family, which follow the seminal work of Kamp (1981) and Heim (1982) and include the contemporary Dynamic Semantics approaches grounded in Groenendijk and Stokhof (1991), bite the bullet and try to make sense of the bound variable approach in some way or other.

We will continue the discussion by briefly outlining Evans' E-type approach in subsection 11.1.1, but we don't intend to dwell on this approach, as our own proposal firmly belongs to the DRT family, which we will present in subsection 11.1.2. We will turn to our own proposal in subsection 11.1.3, where we will show how our restrictedness-based theory derives (and even improves upon) some core

[1] Strictly speaking, the scope of $\forall x$ in (4) is wider than the scope of $\forall y$, as $\forall x$ (immediately) dominates $\forall y$. Obviously, the point here is that $\forall x \forall y\, \varphi$ is logically equivalent to $\forall y \forall x\, \varphi$. It is in this sense that we can see the two quantifiers as having the same scope.

[2] Disregarding our own story to be presented in this section, the closest we come to a motivation is the logical equivalence between $\exists y\, \varphi(y) \Rightarrow \psi$ and $\forall y\, (\varphi(y) \Rightarrow \psi)$, which holds when y is not free in ψ; the equivalence, exemplified in (i), was first noted in this connection by Egli (1979).

(i) If Pedro owns a donkey he is rich.
  a. $(\exists y\colon D(y) \wedge O(p,y)) \Rightarrow R(p)$
  b. $\forall y\colon (D(y) \wedge O(p,y)) \Rightarrow R(p)$

assumptions of DRT. In subsection 11.1.4, we will address a few of the many topics in discourse anaphora above and beyond the core phenomenon.

11.1.1 The E-type approach

One radical idea on how to resolve the problems with Geach's approach is to drop it completely. Specifically, what is dropped is the assumption that pronouns with quantifier antecedents in sentences such as (1), repeated below as (5a), stand for bound variables. One such approach was developed by Evans (1977, 1980) and later advocated in some form or other by T. Parsons (1978); Heim (1990); Neale (1990); Ludlow (1994); and Chierchia (1995); among others. On this theory, known as the *E-type anaphora* approach, the discourse pronouns under discussion stand proxy for descriptions in some sense; for example, (5a) resolves to a sentence along the lines of (5b):

(5) a. Every farmer who owns a donkey likes it.
b. $\Leftrightarrow$ Every farmer who owns a donkey likes the donkey he owns.

We will not go into any details of the E-type anaphora approaches, as our own proposal keeps the assumption that the relevant pronouns stand for bound variables. What we want to do in this subsection is to outline Evans' two major arguments against the bound variable approach, as it turns out that the theory behind our own proposal completely collapses one of his objections, and at least hints at a partial reply to the second objection.

For Evans, one crucial piece of evidence against the bound variable approach is provided by ungrammatical examples such as (6). If the indefinite 'a donkey' could bind the pronoun 'it' in (5a), he argues, then the negative quantifier 'no donkey' should be able to bind it in (6).

(6) *Every farmer who owns [no donkey]$_i$ likes it$_i$.

While this reasoning seems *prima facie* sound, our restrictedness-based proposal can poke a hole in the argument. The thing is, we have a very good reason for allowing (5a) and rejecting (6). The polarities of 'farmer' and 'donkey' match in the former sentence (both are negative) but not in the latter ('farmer' has negative polarity and 'donkey' has positive polarity), and this mismatch leads to failure to perform Restricted Closure and thus to a non-interpretable sentence. We will go through the details of this in subsection 11.1.3; here, we only want to point out that Evans' (1977, p. 494) conviction that "upon the view that these pronouns are bound pronouns, [the distinction between (5a) and (6)] is inexplicable" is simply unwarranted.

Evans' other argument is supported by evidence such as (7). At least in the most natural interpretation, this sentence does not imply that Harry vaccinates only *some* of the sheep John owns, which is an implication we get if we somehow embed the second clause within the scope of 'some sheep', either by putting it in a relative clause, as in (7a), or by assigning wide scope to 'some sheep', as in (7b). Rather, Harry must vaccinate *all* of John's sheep, and it is not clear at all how a bound variable approach could yield this result without further assumptions.

(7) John owns some sheep and Harry vaccinates them.
 a. $\not\Leftrightarrow$ John owns some sheep which are such that Harry vaccinates them.
 b. $\not\Leftrightarrow$ Some sheep are such that John owns them and Harry vaccinates them.
 c. $\Leftrightarrow$ John owns some sheep and Harry vaccinates (all) the sheep John owns.

Our proposal has nothing to contribute toward the resolution of the maximality problem presented above. In other words, we need to make some further assumption on how cross-sentential discourse anaphora works—just as everybody else does. Our impression is that the issue has to do with the sequentiality of sentence processing, and pragmatics in general, but in principle, we could adopt any proposal from the DRT family of approaches.

However, it would be unfair to say that the maximality problem plagues only the bound variable approaches. As far as we see, every theory must make special provisions to address it. For example, Evans' E-type anaphora approach, where discourse anaphors are understood as standing proxy for descriptions, does not resolve the issue unless it is also assumed (as Evans does) that (at least in these cases) E-type pronouns stand proxy for *definite* descriptions. Replacing 'the' in (7c) by 'some' simply wouldn't work, and it is far from obvious where the definite article comes from.

Evans (1977) addresses another example alongside (7). In (8), the maximality problem is compounded by the contribution of the quantifier 'many'. The issue here is that at least in one reading, we want to understand 'many girls' as relative to Mary's (potential) dancing partners, and this interpretation is unavailable both in the attempted relative clause paraphrase (8a), where 'many' is understood as relative to girls who found Mary interesting, and in the attempted wide-scope paraphrase (8b), where 'many girls' gets the absolute interpretation.

(8) Mary danced with many girls and they found her interesting.
 a. $\not\Leftrightarrow$ Mary danced with many girls who found her interesting.
 b. $\not\Leftrightarrow$ Many girls are such that Mary danced with them and they found her interesting.
 c. $\Leftrightarrow$ Mary danced with many girls and the girls Mary danced with found her interesting.

However, we will see that our L**–based approach does not founder on such examples. We will address this example in detail in subsection 11.1.4; the solution will turn on a particular feature of L**—the divorce of cardinality and quantification.

11.1.2 The Discourse Representation Theory approach

Kamp (1981) and Heim (1982) offered (virtually equivalent) theories of donkey sentences which are often thought to redress the problem of mismatched surface syntax and logical form by spelling out the relationship between the superficial syntactic form of a sentence and its *discourse representation structure (DRS)*, and then defining the truth conditions of the sentence (actually, of the discourse) off of the DRS.[3] Their theories draw a sharp distinction between indefinites and other quantifiers. Indefinites are not treated as (existential) quantifiers, but as devices which introduce new *discourse referents*. The free variable introduced by an indefinite is quantified over either by the closest (unselective) quantifier—introduced by a "real" determiner such as 'every', 'no', or 'most', or a quantificational adverb such as 'always', 'never', and 'usually'—or existentially closed in the absence of such an operator. In the *discourse representation (DR)* shown in (9b) below, the indefinite only contributes the discourse referent x and the condition $D(x)$. The quantification over x turns out as existential because no real quantifier is present to bind it. (It is safe enough to imagine all the rows of a discourse representation as conjoined.)

(9) a. A donkey brays.

b.

x
$D(x)$
$B(x)$

A real quantifier such as 'every' creates a pair of DRs (the restriction and the nuclear scope) and embeds it into the matrix DR. The quantificational force of the quantifier is expressed as a meta-language quantification over assignments to discourse referents; every assignment under which the restriction is true must be extendable into an assignment under which the nuclear scope is true as well. Crucially here, the new discourse referents introduced into the restriction are *accessible* from within the nuclear scope (furthermore, a DR can also access discourse referents introduced by its parent, and the accessibility relation is transitive)—below, x of $R(x)$ in the nuclear scope is bound by the same (universal) quantifier as $F(x)$ in the restriction.

[3] We use Kamp's terminology.

(10) a. Every farmer is rich.

b.
$$\begin{array}{|l|}\hline \\ \hline \begin{array}{|c|}\hline x \\ \hline F(x) \\ \hline\end{array}\quad \begin{array}{|c|}\hline \\ \hline R(x) \\ \hline\end{array} \\ \hline\end{array}$$

Returning to donkey anaphora, as a consequence of the setup presented above, the discourse referent y is accessible in the embedded DR $[L(x, y)]$ in (11), even if it was not introduced either in this DR or in any of its ancestors. Its status is no different than the status of x—which is exactly the same as in (10) above—including the type of quantificational force. The quantification in DRT is *unselective* (this is so because it is defined as quantification over assignments), which means that y will be quantified over by the same quantifier as x, which is a universal quantifier, resulting in Geach's truth conditions. In effect, the DRT setup simultaneously "widens" the scope of quantification over y and gives it the universal flavor.

(11) a. Every farmer who owns [a donkey]$_i$ likes it$_i$.

b.
$$\begin{array}{|l|}\hline \\ \hline \begin{array}{|c|}\hline x, y \\ \hline F(x) \\ D(y) \\ O(x, y) \\ \hline\end{array}\quad \begin{array}{|c|}\hline \\ \hline L(x, y) \\ \hline\end{array} \\ \hline\end{array}$$

The DRT account of donkey anaphora is certainly better motivated than Geach's account. The objections toward the latter listed above (where does the universalization occur and why, how is the scope determined) all receive explanation within DRT. However, the question is, to what extent is DRT itself motivated by issues unrelated to donkey anaphora?

One crucial feature of DRT, the assumption that a discourse referent introduced in the restriction is accessible in the nuclear scope, is in fact motivated by a property of natural language quantification central to this book: restrictedness (cf. Kamp 1981, pp. 199–200; Heim 1982, p. 83). The gist of the relation between restrictedness and accessibility is in the fact that if the scope of a quantifier is to be divided into the restriction and the nuclear scope, then both parts must have access to the variable that is quantified over. However, in DRT, restrictedness (or conservativity) is not further deployed to explore the permissible structure of the restriction, as we have done in Chapter 9. As a consequence, DRT must stipulate at least two further crucial assumptions.

One such assumption is the central idea to treat indefinites in a different manner than other noun phrases; more precisely, to treat them as non-quantificational. As we have seen in Chapter 9, and will elaborate in the discussion of our approach to donkey anaphora in the following section, the special treatment of indefinites is unnecessary if we fully exploit the potential of restrictedness. In our approach, *no* noun phrases are assumed to be quantificational, as all quantification results from Restricted Closure.

Another necessary stipulation of DRT is the assumption that negation introduces a discourse representation and thereby acts as an existential closure environment (see Heim 1982, p. 94). This assumption is needed to block coreference in examples such as the following (where the first sentence is intended to receive the reading "It is not the case that John bought a cat," where the negation scopes over the indefinite).

(12) #John didn't buy [a cat]$_i$. It$_i$ was black.

As we will see, these additional stipulations will be unnecessary in our approach, because we investigate the structure of the restriction using our polarity-based tool, p-scope.

We might also be concerned about an additional level of structure employed by DRT, viz. discourse representation structures. In Chapter 8, we made a case for representational minimalism in our Dynamic Deductive System. Should we not have a similar concern about representations that are constructed in DRT? Or at least, should we not try to do without them if we can?

The level of concern actually depends on the version of DRT one has in mind. The original DRT cannot be criticized for not being representationally minimal, as it does not introduce the level of DRSs as an additional level of representation. Rather, the DRSs take over the function of logical forms expressed in predicate logic. At worst, we could argue that this version of DRT employs resources which are not strictly necessary. However, the later, GQ-based versions of DRT might be susceptible to such a criticism, as they work with two layers of representations, DRSs and GQs, which can be further argued to overlap to some extent. Here, our own account can be seen as more representationally minimal. It works with a single level of representation (which we will further argue to be nothing but LF of the Minimalist Program in Chapter 12), a language of restricted plural logic. The plural nature of L** gives us the non-elementary quantifiers, and restrictedness takes care of the discourse anaphora.

11.1.3 Donkey anaphora through the lens of restrictedness

We are now ready to apply our account of conservativity to donkey anaphora. In particular, we will show how the syntactic reflex of this conservativity, restrictedness, and its specific implementation by a quantifierless formal language proposed in Chapter 9, dissolve the key puzzles regarding donkey anaphora.

We concluded Chapter 9 by proposing a "quantifierless" variant of L**. That quantifierless formal language employs no quantifier symbols and also has a single, generalized junction $\odot$, replacing $\wedge$ and $\vee$. All the variables in a quantifierless formula are formally free. We stated a closure rule that tells us in which syntactic position the closure over a variable should apply, and whether the closure should be existential or universal. The closure rule also tells us whether to interpret a particular occurrence of the generalized junction as a conjunction or a disjunction. Let us review how our Restricted Closure works:

(RC1) Find the closure positions: the quantificational closure over x applies at the lowest position dominating all occurrences of x.

Once we have done this step, we have also resolved the identity of all generalized junctions where the closure does *not* apply: they are conjunctions.

(RC2) Determine the type of closure for each variable (and thereby the type of the closure junction): if we can find a potential restrictor of positive local polarity, quantification is existential (and the closure junction is a conjunction); and if we can find a potential restrictor of negative local polarity, the closure is universal (and the closure junction is a disjunction).[4]

To follow this instruction, we need to remember that a restrictor is an atomic formula which positively p-scopes from the restriction into the nuclear scope, and that a potential restrictor is an atomic formula that will be an actual restrictor if we choose the closure type correctly. Also remember that by local polarity we mean the polarity within the position where the closure applies, i.e. when we are computing it, we ignore all negations above this position.

Let's see how this plays out for the universal determiner phrases with a simple noun phrase internal argument. The quantifierless L** logical form of (13a) is (13b). To clearly show the interpretive effect of Restricted Closure, we write down the standard formula (13d) corresponding to the resulting interpretation; to help you follow the application of the closure rule, we also write down the "intermediate stage" (13c) reached after the application of the first step of the rule.

(13) a. $[_{DP}$ Every farmer] is working.
b. $\neg F(x) \odot W(x)$
c. $\rightsquigarrow Qx: \neg F(x) \odot W(x) \rightsquigarrow$
d. $\forall x: \neg F(x) \vee W(x)$

[4] Remember that we can never find two (potential) restrictors having opposite local polarities. On the other hand, there are situations where no (potential) restrictor can be found. In such cases, the formula in unintepretable, as any type of quantifier would result in a non-restricted formula.

So let's apply Restricted Closure to (13b). (RC1) The quantificational closure over our sole variable x will apply at the root. We have a single junction, but this junction is where the closure applies, so its type will be determined later, along with the type of the quantification. We are now at the stage shown in (13c). (RC2) We have a single potential restrictor, $F(x)$ (remember that the restrictor must occur in the restriction, i.e. in the left member of the junction). To have it positively p-scope over $W(x)$, the junction must be a disjunction—of course, this is so because $F(x)$ has negative (local) polarity. The quantificational closure is therefore universal. The result is (13d).

The analysis of a (non-donkey) universal DP containing a restricted relative clause is also straightforward. Consider (14). We have two variables now, x and y. The logic for x is the same as above,[5] and results in a universal closure at the root node, which is interpreted as a disjunction. What about y?[6] (RC1) The quantificational closure over y applies lower than the closure over x; the position of Qy shown in (14c) is high enough to dominate the all occurrences of y. After determining the closure position for both variables, we also know that the junction above $F(x)$ is a conjunction, as neither of the two closures applies there. (RC2) The only potential restrictor for y is $D(y)$. It has positive polarity within Qy (the negation above Qy is irrelevant), so the type of closure over y must be existential, and the closure junction must be a conjunction—note that this goes hand in hand with the fact that $D(y)$ then positively p-scopes over the nuclear scope $O(x, y)$.

(14) a. [$_{\text{DP}}$ Every farmer who owns a donkey] is working.
b. $\neg\big(F(x) \odot (D(y) \odot O(x, y))\big) \odot W(x)$
c. $\rightsquigarrow Qx\colon \neg\big(F(x) \wedge (Qy\colon D(y) \odot O(x, y))\big) \odot W(x) \rightsquigarrow$
d. $\forall x\colon \neg\big(F(x) \wedge (\exists y\colon D(y) \wedge O(x, y))\big) \vee W(x)$

Now let's consider the donkey sentence in (15). This time y occurs not just in the relative clause, but in the main clause as well, as an argument of 'likes'. Again, there is nothing special about x here—it is universally closed at the root and the closure junction is a disjunction—so let us go straight to y. (RC1) This time, the closure over y applies at the same position as the closure over x—in (15b), no position but the root dominates all occurrences of y. As both closures apply at the root

[5] While the logic behind determining the type of closure over x in (13) and (14) is the same, these examples differ in a small detail, orthogonal to the quantificational closure. In (13), $F(x)$ could in principle p-scope over $W(x)$ either positively or negatively, depending on how we interpret the junction. But in (14), the interpretation of the junction does not correlate with positive vs. negative p-scope of $F(x)$; it correlates with positive p-scope vs. no p-scope. This is so because, in (14), the junction directly above $F(x)$ is surely a conjunction (no closure applies there), which forces $F(x)$ to have positive relative p-scope.

[6] For clarity, we discuss variables x and y separately, but strictly speaking, we first perform (RC1) for both variables and then (RC2) for both variables.

junction, both remaining junctions must be conjunctions.[7] (RC2) To figure out the type of closure over y, we must find a potential restrictor for y and check its polarity. $D(y)$ will do: our friend Perceval can reach $L(x, y)$ if he starts out with the golden armor at $D(y)$—he will pass through two conjunctions, then get the armor transmuted to silver at the negation and finally pass through the disjunction at the root. The polarity of $D(y)$ within Qy is negative, so the quantificational closure must be universal and the closure junction must be a disjunction.[8]

(15) a. $[_{\mathrm{DP}}$ Every farmer who owns [a donkey]$_i$] likes it$_i$.
b. $\neg\big(F(x) \odot (D(y) \odot O(x, y))\big) \odot L(x, y)$
c. $\rightsquigarrow Qx Qy\colon \neg\big(F(x) \wedge (D(y) \wedge O(x, y))\big) \odot L(x, y) \rightsquigarrow$
d. $\forall x \forall y\colon \neg\big(F(x) \wedge (D(y) \wedge O(x, y))\big) \vee L(x, y)$

Notice that the Geach/Kamp/Heim analysis came for free. There was no need to convert existentials into universals or to map onto Discourse Representation Structures. The analysis (and truth conditions) stemmed directly from the account of quantification we gave in the previous chapter, and which was inspired by the Medieval accounts of quantification. The initial appearance of a problem stemmed from the belief that there were actual quantificational operators in the syntax of natural language. The reality is that the expressions which we take to be quantifiers are in fact simply bundles of features that control structure building (we will come back to this idea in Chapter 12). Once we understand how we quantify over the variables in linguistic structures, the correct truth conditions (in this particular case, the Geach/Kamp/Heim truth conditions) fall out automatically.

Our analysis works the same as above in donkey anaphora environments other than clauses with a universal subject. For example, if we rewrite the implication in the naive analysis of a conditional using negation and disjunction, the entire antecedent ends up with negative polarity, so all indefinites in the antecedent coindexed with some pronoun in the consequent will receive universal closure at the root.[9]

[7] Compare this to the previous example, where the type of the junction between $D(y)$ and $O(x, y)$ was still unknown at this step.

[8] If you are wondering what comes first: Is it x that forces the root junction to be interpreted as a disjunction, or is it y? The answer is, neither, they do it together. The math from section 9.3 guarantees that whenever there is more than one potential restrictor (i.e. an atomic formula in the restriction which could positively p-scope into the nuclear scope if only the junction was interpreted "correctly"), they will agree on the type of junction they enforce. This holds even in our situation, where we have restrictors for two different variables (closed in the same syntactic position).

[9] The same would hold for a more realistic predicate logic analysis of a conditional. Such an analysis would presumably contain a universal quantifier (over possible worlds), but in our account, this universal quantifier would in fact be triggered by the negated restriction, once again leading to universal closure of all indefinites whose variable chains cross into the consequent.

(16) a. If [a farmer]$_i$ owns [a donkey]$_j$, he$_i$ likes it$_j$.
b. $\neg(F(x) \odot D(y) \odot O(x,y)) \odot L(x,y)$
c. $\forall x \forall y \colon \neg(F(x) \wedge D(y) \wedge O(x,y)) \vee L(x,y)$

Note that the DRT assumption that indefinites are not quantificational is a special case of our assumption that *no* determiner phrases are quantificational. This is why in the above case, the DRT analysis works exactly the same as our account. However, the two theories provide a different reason for the ungrammaticality of examples where the relative clause contains some other determiner. For DRT, 'no donkey' in (17) cannot bind 'it' because it is a quantifier and thus forces narrow scope of the variable it binds (by creating a pair of DRs). In our account, the ungrammaticality is due to the wrong polarity of 'donkey'.

(17) a. *[$_{\mathrm{DP}}$ Every farmer who owns [no donkey]$_\mathrm{i}$] likes it$_\mathrm{i}$.
b. $\neg\big(F(x) \odot (\neg D(y) \odot \neg O(x,y))\big) \odot L(x,y)$
c. $\rightsquigarrow QxQy \colon \neg\big(F(x) \wedge (\neg D(y) \wedge \neg O(x,y))\big) \odot L(x,y) \rightsquigarrow$
d. $\forall x ?y \colon \neg\big(F(x) \wedge (\neg D(y) \wedge \neg O(x,y))\big) \vee L(x,y)$

Let us go through our analysis step by step. The quantifierless L** logical form of (17a) is given in (17b). Everything is just as in the previous example—in particular, the closure over x and y applies at the same node, the root node—until we try to find a restrictor for y in (17c). In this example, $D(y)$ will not do as a (potential) restrictor. Because we find it negated under a conjunction, it has negative relative p-scope and therefore cannot positively p-scope over $L(x,y)$ (regardless of the interpretation of the root junction as a conjunction or a disjunction).

You might have wondered why haven't we considered whether $O(x,y)$ could be a potential restrictor of y in (17). While it is clear that it would fail to be one, for the same reason as $D(y)$, the interesting question is whether a verbal predicate could ever play this role. To address this question, we will turn to an example of donkey anaphora with a universal 'donkey' phrase. In (18), the negative polarity of the verbal predicate will allow it to positively p-scope into the nuclear scope. At first sight, this should predict that (18) will be grammatical, but (18) is ungrammatical just as (17) is. We can only arrive at the correct prediction if we explain why the verbal predicate cannot act as a restrictor. To do that, we will have to remember a tentative conclusion from section 10.4 that only lexical predicates, i.e. predicates corresponding to content words, may act as restrictors. But to see how this saves the day, we will have to switch to the event semantics for the next example.

Below, we explode the traditional analysis $L(x,y)$ of 'like' into the event semantics analysis $L(e) \wedge \mathrm{Ex}(x,e) \wedge \mathrm{Th}(y,e)$, where predicates Ex and Th stand for the thematic roles of experiencer and theme. (We don't analyze 'feed' using event semantics, as that is not important for our discussion.) As shown in (18di), $D(y)$ has the wrong polarity to be a potential restrictor of y. But if $\mathrm{Th}(y,e)$ could function as a potential restrictor of that variable, it would be a potential restrictor indeed, as it positively p-scopes over the nuclear scope, and Restricted Closure would then interpret y as universally closed, because $\mathrm{Th}(y, e)$ has negative polarity. This would lead to the incorrect conclusion that the sentence can be interpreted as in (18dii), which can be paraphrased as "every farmer who likes something which is not a donkey feeds it."

(18) a. *[$_{\mathrm{DP}}$ Every farmer who likes [every donkey]$_{\mathrm{i}}$] feeds it$_{\mathrm{i}}$.

b. $\neg\big(F(x) \odot (\neg D(y) \odot (L(e) \odot \mathrm{Ex}(x, e) \odot \mathrm{Th}(y, e)))\big) \odot F'(x, y)$

c. $\rightsquigarrow QxQy\colon \neg\big(F(x) \wedge (\neg D(y) \wedge (Qe\colon L(e) \wedge \mathrm{Ex}(x, e) \wedge \mathrm{Th}(y, e)))\big) \odot F'(x, y) \rightsquigarrow$

d. i. $\forall x?y\colon \neg\big(\underline{F(\underset{-}{x})} \wedge (\underline{\neg D(\underset{+}{y})} \wedge (\exists e\colon L(e) \wedge \mathrm{Ex}(x, e) \wedge \mathrm{Th}(y, e)))\big) \vee F'(x, y)$

ii. $^{*}\forall x\forall y\colon \neg\big(\underline{F(\underset{-}{x})} \wedge (\neg D(y) \wedge (\exists e\colon L(e) \wedge \mathrm{Ex}(x, e) \wedge \underline{\mathrm{Th}(\underset{-}{y}, e)})))\big) \vee F'(x, y)$

Is there an independent reason to assume that thematic predicates cannot function as restrictors? Well, yes. We saw in section 10.4 that the intuitive judgments of aboutness seem stronger for content words (nouns, verbs, adjectives, and adverbs) than grammatical (or, in the Minimalist parlance, functional) categories such as the past tense. The idea is then that predicates corresponding to functional categories (which include thematic roles) are not candidates for (potential) restrictors at all.[10] If we assume this, then only $D(y)$ could ever be a potential restrictor of y in (18). But as it does not positively p-scope into the nuclear scope, it is not one, which leaves Restricted Closure without a potential restrictor and thereby yields an uninterpretable sentence.

For the examples above, the DRT approach and our approach yield the same results. Are there any situations where they disagree? Consider the following, often discussed example (see e.g. Roberts 1989, p. 702; Chierchia 1995, p. 8), attributed to Barbara Partee. In a DRT setting, the grammaticality of (19) is unexpected—'no', being a quantifier, should make its variable inaccessible to a pronoun outside its syntactic scope—and can only be accounted for if we introduce some additional assumption. For example, Chierchia (1995, pp. 702–703) deals with the example using accommodation: the (un)negation of the first disjunct, 'there's a bathroom in this house', is copied into the second disjunct, which makes the indefinite 'a bathroom' in the copy accessible to 'it'.[11]

(19) Either there's [no bathroom]$_i$ [in this house]$_H$ or it$_i$'s [in a funny place]$_F$.

a. $\neg(B(x) \wedge H(x)) \vee F(x)$

b. $\rightsquigarrow Qx\colon \neg(B(x) \wedge H(x)) \vee F(x) \rightsquigarrow$

c. $\forall x\colon \neg(B(x) \wedge H(x)) \vee F(x)$

[10] It might be even more parsimonious to assume that the relevant distinction is not between lexical and functional categories, but between monadic and non-monadic predicates.

[11] Chierchia assumes that accommodation takes place at the level of DRSs, not in syntax, but this is irrelevant here. All we are saying is that there is an additional mechanism at work.

Our own account has no trouble with this example. Let us consider a simplified analysis first: We will assume that the Boolean connectives are given as $\wedge$ and $\vee$ and that the Restricted Closure only checks if they are of the correct type. In (19a), the closure must apply at the root, as x occurs in both disjuncts; we arrive at (19b). There, $B(x)$ is a potential restrictor. To see that it positively p-scopes over $F(x)$, consider Perceval's journey from $B(x)$ to $F(x)$. Donning golden armor, he gets to the conjunction; his armor is then transmuted to silver, and this gets him through the disjunction. The polarity of $B(x)$ (within Qx, which is the root) is negative, so Qx is a universal quantifier. Being a universal quantifier, it must be a parent of a disjunction—and it is.

Intuitively, the bathroom example works because as far as the logical form is concerned, it has a structure no different than the structure of a universal. The difference lies in the source of the disjunction and the negation over the first disjunct. In the case of a universal, both come from the universal itself. Above, the disjunction is given explicitely, and the negation is contributed by the negative determiner.[12]

A more precise analysis of this example does not deploy an explicit disjunction, but expresses it through generalized junction $\odot$—remember De Morgan's law $\varphi \vee \psi \sim \neg(\neg\varphi \wedge \neg\psi)$. The quantifier turns out to be a (negated) existential, as shown in (20c). To see that the truth conditions are correct, consider the paraphrase "There is no object which is a bathroom in this house but not in a funny place."

(20) Either there's no bathroom [in this house]$_H$ or it's [in a funny place]$_F$.

a. $\neg\big(\neg\neg(B(x) \odot H(x)) \odot \neg F(x)\big)$

b. $\rightsquigarrow \neg Qx\colon \neg\neg(B(x) \wedge H(x)) \odot \neg F(x) \rightsquigarrow$

c. $\neg\exists x\colon \neg\neg(B(x) \wedge H(x)) \wedge \neg F(x)$

Under our analysis, the variable introduced by a negative noun phrase (or any other quantificational phrase, for that matter) is sometimes "accessible" outside the c-commanding domain of the phrase, and sometimes not (and we correctly predict which option obtains in a particular case). This contrasts with the DRT analysis, where 'no' is considered a quantifier and is thus always opaque for accessibility, an analysis which causes problems in certain environments such as (19) above.

Furthermore, as our account is based on polarity rather than on special treatment of indefinites, we can easily explain what is going on in the case of a negated indefinite. A donkey sentence such as (21) containing a negated restricted clause, where the negation is intended to scope over the indefinite, is ungrammatical for

[12] In (19), we have analyzed the negative determiner as a negated existential, but the logically equivalent analysis as a universal over a negation would work as well. In (i), Perceval starts out with golden armor, which is immediately transmuted to silver, and this gets him across both disjunctions.

(i) $(\neg B(x) \vee \neg H(x)) \vee F(x)$

the same reason as (17) with the embedded 'no'. To account for the ungrammaticality of (21), DRT needs to stipulate that negation acts as an unselective existential closure operator (see example (12) in subsection 11.1.2). In contrast to DRT, we do not have to assign any special properties to negation to deal with (21)—or rather, we have already done that, but on a more general and principled basis, in the definition of p-scope (recall that p-scope was developed for our deductive system, and its applicability to donkey anaphora comes as a discovery here).

(21) a. *[$_{\text{DP}}$ Every farmer who doesn't own [a donkey]$_{\text{i}}$] likes it$_{\text{i}}$.
b. $\neg\big(F(x) \odot \neg(D(y) \odot O(x,y))\big) \odot L(x,y)$
c. $\rightsquigarrow QxQy\colon \neg\big(F(x) \wedge \neg(D(y) \wedge O(x,y))\big) \odot L(x,y) \rightsquigarrow$
d. $\forall x?y\colon \neg\big(F(x) \wedge \neg(D(y) \wedge O(x,y))\big) \vee L(x,y)$

There is an immediate objection to our analysis that you can raise if you are familiar with the data used in DRT to motivate the (sentential) negation as preventing access to variables within its scope. The opening sentences of the discourses in (22) and (23) are logically equivalent but differ in what anaphoric possibilities they offer. We have not yet talked about cross-sentential anaphora (we will do that in the next subsection), but no matter. Here, we can get by with the naive assumption that consecutive sentences are simply conjoined. The DRT account straightforwardly predicts the failure of 'a dog' to antecede the pronoun in (23). A negation introduces a DR box, and thus acts as an unselective existential closure operator (see Heim 1982, p. 94). For our analysis, the apparent problem is that $D(x)$ positively p-scopes over $H(x)$ in both (22) and (23). The double negation in (23) is irrelevant for Perceval's journey, as transmuting the armor twice has no external effect. By being sensitive to polarity, which double negations do not affect, instead of assigning some special property to the negation operator, we seem to allow access to variables that in reality cannot be accessed.

(22) John bought [a dog]$_i$. It$_i$ was huge.
a. $(D(x) \odot B(j,x)) \odot H(x) \rightsquigarrow$
b. $\exists x\colon (D(x) \wedge B(j,x)) \wedge H(x)$

(23) #It is not the case that John didn't buy [a dog]$_i$. It$_i$ was huge.
a. $\neg\neg(D(x) \odot B(j,x)) \odot H(x) \rightsquigarrow$
b. $\exists x\colon \neg\neg(D(x) \wedge B(j,x)) \wedge H(x)$

But do we? The objection relies on the assumption that the first conjunct of (23a) is in fact the logical form of the first sentence. But is it? In other words, is it sensible to assume that all the matrix clause 'it is not the case' brings to the logical form is a single negation, ¬? We think not. It is one thing to have 'it is not the case' as the "logical talk" for ¬, but to assume that ¬ is in fact the *logical form* of 'it is not the case' is not even remotely a trivial assumption. At the very least, the logical form should contain the contribution of the noun 'case', i.e. some predicate C, as sketched below.

(24) It is not the case that φ. $\neg\exists y\colon C(y) \wedge \ldots \wedge \varphi$

Our aim here is not to propose a realistic logical form for (24). All we want to say is that a realistic logical form must surely be more complex than a simple $\neg$. In all probability, it will contain (con)junctions, as shown above. As a result (and simplifying the structure to a bare minimum required to illustrate the logic), the two negations of the opening sentence in (23) will not be adjacent in the logical form. As shown in (25), there will be a conjunction between them, and it is this conjunction that Perceval cannot pass through. He starts out with golden armor, gets through the first conjunction (linking $D(x)$ and $B(j,x)$), but then his armor gets transmuted to silver, so his journey ends at the conjunction[13] between 'case' and the embedded clause. Perceval can never reach $H(x)$, and this is why x is inaccessible from the second sentence.

(25) #It is not the case that John didn't buy [a dog]$_i$. It$_i$ was huge.
a. $\neg(C(y) \odot \neg(D(x) \odot B(j,x))) \odot H(x)$
b. $\rightsquigarrow Qx\colon \neg(Qy\colon C(y) \odot \neg(D(x) \wedge B(j,x))) \odot H(x) \rightsquigarrow$
c. $?x\colon \neg(\exists y\colon C(y) \wedge \neg(D(x) \wedge B(j,x))) \odot H(x)$

We believe it is fair to say that the treatment of negation is where our account surpasses the DRT account. In DRT, negation is assumed to act as an unselective binder, implementing the idea that no quantifier should ever bind out of negation. This includes double negation among the binding barriers (and causes problems with the bathroom sentences), but as we argued above, the idea rests on a flawed analysis of 'it is not the case that'. We have argued that this construction does not introduce a double negation after all, but that the two negations it contains are separated by a conjunction (at least one). Under such analysis, our approach then correctly predicts both that 'it is not the case that S' will block binding from S, and that the bathroom sentences will contain no binding barrier.

Summing up, our approach to donkey anaphora is similar to the DRT approach, but it achieves the same core goal—motivating the "universalization" of indefinites in this environment—without the introduction of a new level of analysis (discourse representation structures) or special assumptions about indefinites and negation.

[13] The idea is that y will have another occurrence which relates the noun 'case' and the relative clause. This will position Qy as shown in (25b), and thereby force the junction above $C(y)$—which has positive polarity within Qy—to be interpreted as a conjunction. And if Qy somehow ends up higher up in the tree, the junction above $C(y)$ still turns out to be a conjunction, as it is then not a closure site.

The key feature of our approach is conservativity, syntactically reflected as restrictedness. By fully exploiting the consequences of restrictedness, our approach can explain the core facts about donkey anaphora without any further assumptions. But of course, there is more to say about donkey anaphora, and discourse anaphora in general, than we have said in this section. We will discuss some of these details in the following section.

11.1.4 Beyond the essentials

In the previous subsection, we laid out our theory of discourse anaphora based on restrictedness, and we showed that many essential assumptions of Discourse Representation Theory (DRT) either follow from our theory, or are rendered obsolete. But of course we cannot claim that this closes off the topic of discourse anaphora. The decades of work on this topic produced a multitude of empirical details that any successful theory must address. Covering all this detail is out of scope of this book, both in terms of addressing each and every discovered phenomenon, and in terms of fully explaining how they work. We will take a look at three of the more prominent issues—the proportion problem, the existential readings of donkey sentences, and the cross-sentential anaphora—and hint at how our theory might deal with them.

The proportion problem

One crucial idea behind DRT is that quantification is unselective, i.e. that a quantifier will bind all free variables in its scope. Technically, this is implemented by quantifying over assignments rather than individuals, but in effect, it is safe enough to imagine that quantifiers bind tuples of free variables. The examples below contain two variables each, bound (Geach-style) at the same location, so the quantifier binds a pair of variables, as shown in the logical forms below. (Note that DRT can only deal with these examples if it adopts some resource allowing non-elementary quantification. Most often this is the GQ theory.)

(26) a. At least two farmers who own a donkey like it.
b. *at least two $(x, y) \left[F(x) \wedge D(y) \wedge O(x, y)\right] L(x, y)$

(27) a. Most farmers who own a donkey like it.
b. *most $(x, y) \left[F(x) \wedge D(y) \wedge O(x, y)\right] L(x, y)$

The so-called *proportion problem* refers to the fact that these logical forms do not capture the correct truth conditions. In (26), the logical form states that there are (at least) two farmer–donkey pairs such that the farmer owns and likes the donkey.

Thus, the formula is true in a situation where there is a single farmer owning and liking two donkeys. However, the sentence itself is false in such a setting. For the sentence to be true, we need two farmers. A similar problem arises for (27). Say we have ten farmers who own a single donkey but don't like it, and one farmer who owns ninety donkeys which he does like. There are a hundred farmer–donkey pairs in this situation, and most of these pairs (ninety out of a hundred) are such that the farmer likes the donkey. According to the logical form, the sentence should be true, but it is not. For the sentence to be true, the farmers liking (at least one of) their donkeys should form a majority of the farmers owning a donkey. But in the situation described above, we only have one such farmer out of ten.

Various solutions to the problem have been proposed—see Chierchia (1995, §2.1.2) and references therein—but it is not our aim to describe or evaluate them. Rather, we want to show that our L**–based account does not suffer the proportion problem at all.

The proportion problem does not arise in L** because cardinality is divorced from quantification in L**.[14] Let us consider the logical forms below to see how this works—we will focus on the simpler (28), but the same reasoning applies to (29) as well. As the variable chains in the logical form span the entire formula, Restricted Closure requires quantificational closure (for both x and y) at the root. But L** is a first-order language, so the quantifier binding x and y cannot be the generalized quantifier "at least two" as in the DRT analysis (26), not even in principle. The quantifier binding x and y is the existential quantifier $\exists$ (the type is determined by Restricted Closure given the positive polarity of $F(x)$ and $D(y)$). Crucially, Restricted Closure does not and can not touch the part of the formula encoding cardinality. The cardinality condition on the farmers, $\underline{2}(x)$—remember that this is an abbreviation for $2(n) \wedge \#(n, x)$—stays next to at $F(x)$ and constrains only the value of x. In effect, the numeral 'two' puts the requirement on the number of farmers—as it should—and not on the number of farmer–donkey pairs; these pairs do not even enter the equation.

(28) a. At least two farmers who own a donkey like it.
b. $\big((F(x) \wedge \underline{2}(x)) \odot (D(y) \odot O(x,y))\big) \odot L(x,y)$
c. $\rightsquigarrow \exists x \exists y\colon \big((F(x) \wedge \exists n\colon 2(n) \wedge \#(n,x)) \wedge (D(y) \wedge O(x,y))\big) \wedge L(x,y)$

(29) a. Most farmers who own a donkey like it.
b. $\big((F(x) \odot (\#(n,x) \odot \neg(\#(n,x') \odot F(x') \odot x' \wr x)) \odot (D(y) \odot O(x,y)))\big) \odot L(x,y)$
c. $\rightsquigarrow \exists x \exists y\colon \big((F(x) \wedge \exists n\colon \#(n,x) \wedge \neg(\exists x'\colon \#(n,x') \wedge F(x') \wedge x' \wr x)) \wedge (D(y) \wedge O(x,y))\big) \wedge L(x,y)$

[14] The predecessor of L**, L*, is susceptible to the proportion problem, because cardinality conditions in this language are (at least partially) encoded on quantifiers, as quantifier subscripts.

So far so good, but there is nevertheless a problem with the L** logical forms. Let us again focus on (28). Imagine a situation where there are two farmers, each owning exactly one donkey, but further assume that neither farmer likes his own donkey—they each like the other farmer's donkey. In such a situation, the sentence is presumably false, but our analysis pegs it as true. This is so because when x is assigned the two farmers and y is assigned the two donkeys, the two dyadic atomic formulas, $O(x, y)$ and $L(x, y)$, are satisfied regardless of which of x likes which of y, as long as all xs like some y and all ys are liked by some x. In other words, the sentence should come out as true only if $O(x, y)$ and $L(x, y)$ are satisfied "in parallel," but our system, as it stands, does not enforce this parallelism.

Existential readings

So far we have assumed that donkey sentences have a single, *universal* reading, presented in (30a). However, some speakers claim that these sentences can also be understood in their *existential* reading, presented in (30b). The existential reading is not very salient in canonical donkey sentences such as (30), but it comes to the fore (for some speakers) in examples such as (31).[15] Imagine that some (perhaps all) people who have submitted a paper have submitted more than one paper. On one attested interpretation, (31) is true even if each of these people had only a single paper rejected. The truth of the sentence does not hinge on each of these people having *all* their papers rejected (once).

(30) Every farmer who owns a donkey likes it.
  a. Every farmer who owns a donkey likes *every* donkey he owns.
  b. Every farmer who owns a donkey likes *some* donkey he owns.

(31) Every person who submitted a paper had it rejected once.

Is there a way for our approach to accommodate those who assign the existential reading to donkey sentences? There is. In fact, no adjustment of our interpretation procedure is required, what we need to look at, and adjust, is our understanding of how pronoun resolution works. So far we have assumed that a pronoun stands for a bound variable. However, this assumption fails in a class of examples known as *lazy pronouns* or *paycheck pronouns* (Geach 1962; Karttunen 1969). In the example below, 'it' cannot be semantically bound by (or coreferential with) 'his paycheck', as that would imply that the second man's bookie received the first man's paycheck.

(32) The man who gave his paycheck to his spouse was wiser than the man who gave it to his bookie.

[15] Example (31) is taken from Chierchia (1995), which contains many further examples of donkey sentences with a salient existential reading, including an ingenious context (attributed to Paolo Casalegno), where the existential reading becomes salient for the canonical donkey sentence (30).

Such examples are usually taken to illustrate that some pronouns function by implicitly *repeating* the antecedent noun phrase, i.e. (32) is interpreted as if it contained another instance of 'his paycheck' in place of the pronoun 'it'. Let us assume that this analysis is essentially correct, but that what is repeated is in fact a part of the LF structure rather than the words themselves, and let us furthermore assume that pronoun 'it' in a donkey sentence can function not only as a bound variable but also as a lazy pronoun. As we already know, the first option yields the universal reading of a donkey sentence. It turns out that in our quantifierless formal language armed with Restricted Closure, the second option yields the existential reading.

The exact idea behind the construction of (33a) is that if a lazy pronoun is coindexed with a noun phrase, it stands for the noun phrase plus the entire scope of that noun phrase; in other words, if that noun phrase introduces variable y, the lazy pronoun will stand for the copy of the smallest constituent containing all occurrences of y, but with y' substituted for y. Once we have constructed (33a), it is easy to see that Restricted Closure produces (33b), where both y and y' receive existential closure, producing the meaning that every farmer who owns a donkey likes some donkey he owns.

(33) Every farmer who owns [a donkey]$_i$ likes it$_i$. (∃ reading)

a. $\neg(F(x) \wedge \underline{(D(y) \wedge O(x,y))}) \vee \underbrace{(D(y') \wedge O(x,y'))}_{\text{THE COPY}} \wedge L(x,y')$

b. $\rightsquigarrow \forall x\colon \neg(F(x) \wedge \exists y\colon D(y) \wedge O(x,y)) \vee \exists y'\colon (D(y') \wedge O(x,y')) \wedge L(x,y')$

How plausible is this "copying" account? For one, why is the variable renamed? (If we don't rename it, we will end up with the total of two variables and thus two quantifiers in the formula. As we will see, this actually leads to the universal reading.) Next, the two roles a pronoun can play (it can either stand for a variable, or trigger the copying mechanism) are suspiciously different. And perhaps most importantly, doesn't this copying approach overgenerate? For example, what prevents the copying in sentences with 'no donkey' in place of 'a donkey' above? After all, those sentences are just plain ungrammatical—is it not the case that our copying story provides them with an unattested interpretation?

Slightly adapting our perspective on pronoun resolution (of both bound and lazy pronouns), we can address all these objections in one fell swoop. The key observation is that the formulas in (34) are logically equivalent. Our Dynamic Deductive System developed in Chapters 6 and 7 can help us understand why this is the case. (34a) is the original logical form of the universal reading, and we get (34b) by Copying the smallest constituent containing all occurrences of y in the restrictor into the nuclear scope (but in contrast to the lazy pronoun procedure above, keeping the variable y intact); conversely, (34a) can be derived from (34b) by Delete. Given that the copied constituent always contains the restrictor of y, and that a restrictor always positively p-scopes over the nuclear scope, this copying will yield a logically equivalent formula for any donkey anaphora sentence.

(34) Every farmer who owns a donkey likes it. (∀ reading)

a. $\neg(F(x) \wedge (\underline{D(y) \wedge_{+} O(x, y)})) \vee \underbrace{L(x, y)}_{+\ \text{COPY}}$

b. $\neg(F(x) \wedge (D(y) \wedge O(x, y))) \vee ((\overline{D(y) \wedge O(x, y)}) \wedge L(x, y))$

Comparison of (33a) and (34b) reveals that the universal and the existential reading of a donkey sentence have almost the same logical forms, the only difference being in having one "donkey variable" (y) in the universal logical form, and two of them (y and y') in the existential logical form. And even this difference can be minimized if we eliminate variables altogether in favor of Quine's (1981) bonds, as hinted at the end of Chapter 9. The logical forms below differ in the presence of a single bond.

(35) Every farmer who owns a donkey likes it.

a. $\neg(F(\ ) \wedge (D(\ ) \wedge O(\ ,\ ))) \vee ((D(\ ) \wedge O(\ ,\ )) \odot L(\ ,\ ))$ (∀)

b. $\neg(F(\ ) \wedge (D(\ ) \wedge O(\ ,\ ))) \vee ((D(\ ) \wedge O(\ ,\ )) \odot L(\ ,\ ))$ (∃)

Pronoun resolution works almost the same for bound variable pronouns as for lazy pronouns. First of all, the two types of pronouns share the grammaticality condition—the restrictor must positively p-scope over the pronoun. Also, in both cases the smallest constituent containing all the occurrences of the antecedent variable is copied to the position of the pronoun. But there is a decision to make while copying. If the copying creates the variable bond between the original and the copy, the resulting logical form contains a single variable for both the original and the copy; we get the bound variable pronoun interpretation, which results in the universal reading in the case of donkey anaphora. If the variable bond is not created, the variables of the original and the copy are distinct; we get the lazy pronoun interpretation, which results in the existential reading in the case of donkey anaphora.

Cross-sentential anaphora

So far we have been talking about intra-sentential donkey anaphora, but we can extend our story to cross-discourse anaphora, where things usually start getting especially hairy for Geach-style accounts. The basic idea is to assume that the formal language of discourse representations is the same as the formal language deployed for the analysis of single sentences, and that sentences in a discourse are joined using the same connective as the constituents of a sentence are: the generalized junction $\odot$.

While this might seem to constitute a departure from the zero hypothesis that sentences are joined by a conjunction, this is arguably not the case. First of all, if we really stand by our idea of the quantifierless format, which recognizes the generalized junction $\odot$ as the only binary connective, then sentences cannot be joined by a conjunction as a matter of principle. Next, conjunction is the default interpretation of the generalized junction anyway; it is what we will get when no variable chain crosses the sentential boundary. Furthermore, assuming that there is no formal and interpretive difference between sentence-internal and cross-sentential logical forms, we *predict* that sentences will not be always interpreted as conjoined—the "identity" of the generalized junction must be resolved by Restricted Closure, and it may not be always resolved into a conjunction. More precisely, we predict that cross-sentential variable chains will lead to cross-sentential binding, and we furthermore predict that the closure of those variable chains might be either existential or universal. The closure type will of course depend on the polarity of the restrictor of the sentence-crossing variable: a positive polarity restrictor will yield conjoined sentences (and an existentially quantified variable), while a negative polarity restrictor will yield disjoined sentences (and a universally quantified variable). We will argue that this approach eventually leads to correct truth conditions, even if our rudimentary analysis will leave several issues open for further research.

Starting with the simplest situation where no variable chain crosses the sentences, it is clear that Restricted Closure (RC) interprets the junction between the sentences in (36) as a conjunction and that this yields the correct truth conditions.

(36) a. A dog is barking. A cat is sleeping.
b. $(D(x) \odot B(x)) \odot (C(y) \odot S(y))$
c. $\rightsquigarrow (\exists x\colon D(x) \wedge B(x)) \wedge (\exists y\colon C(y) \wedge S(y))$

Let us move on to a simple case of a variable chain crossing the sentences. Assuming that the two sentences below are joined into a single logical form, and that we have a variable chain for x crossing the sentences, the closure of x must apply at the junction of the two sentences. The restriction of x is the left member of this junction, i.e. the entire first sentence. The restrictor of x is $D(x)$, and since it has positive polarity, the quantificational closure over x is existential and the generalized junction connecting the two sentences comes out as a conjunction.

(37) a. Pat owns a donkey. It is brown.
b. $(D(x) \odot O(p, x)) \odot B(x)$
c. $\rightsquigarrow \exists x\colon (D(x) \wedge O(p, x)) \wedge B(x)$

So far so good, but Evans (1977) offered an important argument that this strategy will not work in the case of quantifiers like 'some', as in (38). Evans' point is that this sentence cannot mean that there are some sheep that Pat owns and that Francis shears. The sentence means that there are some sheep that Pat owns, and Francis shears *all* of the sheep that Pat owns. But this is not the result that extrudes from our theory. As shown below, our theory yields the wrong meaning.

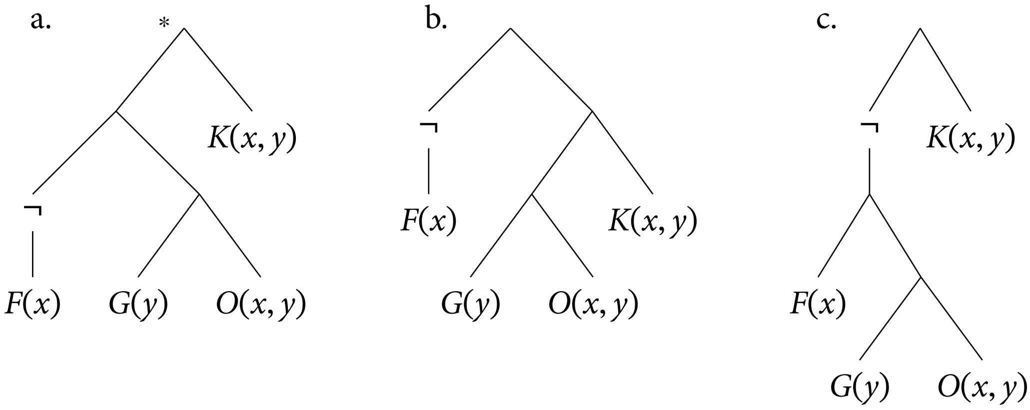
(38) a. Pat owns some sheep. Francis shears them.
b. $(S_1(x) \odot O(p,x)) \odot S_2(f,x)$
c. $\rightsquigarrow \exists x\colon (S_1(x) \wedge O(p,x)) \wedge S_2(f,x)$

However, we believe that the failure to account for this *maximality* problem is not a fatal blow for our theory. For one, our approach gets many aspects of the logical form right. We predict that the sentences are joined by a conjunction, that the variable chain can be successfully bound by a restricted quantifier and that the type of this quantifier is existential. The maximality of the pronoun's reference seems to be an issue above and beyond these correct results, and it is an issue that plagues all accounts of discourse anaphora, in the sense that everybody must make some additional stipulation to address it. For example, the approach taken by Evans himself is to assume that the discourse pronoun 'them' (an E-type anaphora in his approach) stands for a description, but note that this does not solve the maximality issue unless it is also assumed (as Evans does) that it is a Russelian *definite* description.

(39) Pat owns some sheep. Francis shears them.
a. $\Leftrightarrow$ Pat owns some sheep, and Francis shears (all) the sheep Pat owns.
b. $\not\Leftrightarrow$ Pat owns some sheep, and Francis shears some sheep Pat owns.

Further complications arise with the examples where the variable chain crossing the sentences is connected to a universal noun phrase—this is a case of what Sells (1985) and Roberts (1989) called *telescoping*. Here, the naive joining of sentences does not produce the correct result. In (40a), the potential restrictors of x ($F(x)$) and of y ($G(y)$) have opposing polarities, so the join of the sentences should be uninterpretable, as shown in (41a). (The first formula in (41a) is the linear representation of the tree (40a); we get the second formula by applying Restricted Closure to the first one. The same holds for the other two trees.)

(40) Every farmer owns a gun. He keeps it under his bed.



(41) a. i. $\big(\neg F(x) \odot (G(y) \odot O(x, y))\big) \odot K(x, y)$

ii. $\rightsquigarrow QxQy\colon \big(\neg F(x) \wedge (G(y) \wedge O(x, y))\big) \odot K(x, y)$

b. i. $\neg F(x) \odot \big((G(y) \odot O(x, y)) \odot K(x, y)\big)$

ii. $\rightsquigarrow \forall x\colon \neg F(x) \vee \big(\exists y\colon (G(y) \wedge O(x, y)) \wedge K(x, y)\big)$

c. i. $\neg\big(F(x) \odot (G(y) \odot O(x, y))\big) \odot K(x, y)$

ii. $\rightsquigarrow \forall x \forall y\colon \neg\big(F(x) \wedge (G(y) \wedge O(x, y))\big) \vee K(x, y)$

There are two ways we could modify the tree in (40a) to arrive at the correct truth conditions. The idea behind (40b) is that the interpretability of the structure is saved by moving the original restriction of x ($\neg F(x)$), i.e. the universal noun phrase, on top of the joined structure (this could even turn out to be an instance of regular syntactic movement of universals, see subsection 12.2.3). Or—and this is perhaps the most attractive option—the second sentence ($K(x, y)$) could be joined to the nuclear scope of x rather than to the root of the first sentence; this would again result in (41b). The other approach is to somehow extend the scope of the negation above $F(x)$ to produce (41c). Originally, this negation dominates only the restrictor of x; the "raised" negation needs to dominate the restriction of x in the logical form of joined sentences, i.e. the entire first sentence. This approach produces the correct truth conditions only if we assume that the first sentence is first processed in isolation; thereby we learn that every farmer owns a gun, and only then does (41c), which is really the logical form of "every farmer who owns a gun keeps it under his bed," give us the same information as (41b).

We are not sure which (if any) of the options listed above is best (although the idea to simply join the second sentence at a non-root position of the first sentence, producing (41b), seems the simplest and thus the most attractive to us). We believe that the road to the answer leads through the investigation of *modal subordination* (and related) environments (Roberts 1989), but due to the multitude and complexity of the empirical details, we must leave the issue for further research. What is important for us with regard to our initial view on the telescoping examples is that our theory certainly wants to "lift" the quantification closure position of variables occuring in the individual sentences, and that there is ample indication that this closure should be universal.

Our approach, as it stands, does not pretend to resolve all the problems facing a theory of cross-sentential discourse anaphora. However, we believe that it gets many essential things right, and even more importantly, we believe that our toolbox consisting of the quantifierless first-order logic and the interpretive device of Restricted Closure provides a new and exciting way of exploring the details of the phenomenon.

As noted above, the discussion of donkey anaphora and other forms of discourse anaphora have endured over fifty years, and the literature on the topic is vast, so it would be foolish to say that we are ready to close the book on this class of phenomena. However, on the one hand, it is not a trivial result that certain aspects of standard theories fall out for free on this proposal, and whether the glosses of anaphora offered in this section are successful or not, we believe that our general analysis of polarity and quantification provides a new framework for investigating these questions. And of course the big point here is that once again we can make use of p-scope, which in turn served as the foundation of our Dynamic Deductive System and which played a critical role in our understanding of restrictedness and conservativity in natural languange. And of course, coming back to the organizing thread of the book, p-scope itself is interwoven with the phenomenon of polarity and the ways in which polarity interacts with inference in natural language.

11.2 Presupposition Projection

It is worth pausing to examine the phenomenon of *presupposition projection*, since it is known to track the phenomenon of discourse anaphora in interesting ways (see in particular van der Sandt 1992). The study of presupposition projection, crudely, involves determining when a root sentence inherits a presupposition of an embedded clause. For example, consider the contrast between the following conditionals.

(42) John's dog is ferocious. > John has a dog.
   a. If dogs take after their owners, then John's dog is ferocious.
      > John has a dog.
   b. If John has a dog, then John's dog is ferocious.
      $\not>$ John has a dog.

When standing alone, the consequent clause of these sentences ('John's dog is ferocious') presupposes (>) that John has a dog. However that presupposition does not "project" to the entire sentence in both examples. In (42a), the presupposition projects to the root sentential level, in the sense that the compound sentence presupposes that John has a dog, but not so in (42b). The standard explanations offer either pragmatic or semantic accounts of the phenomenon, but the question arises as to whether we cannot also offer a syntactic account. Can we shed light on why these facts fall as they do?

Let's bring Perceval back into our story. He has done a lot of work for us already, but he is going to be particularly helpful here. Rather than think of presuppositions as being things that project upwards in a sentences, let's think of them instead as things that need to be *satisfied*, or if you prefer, *confirmed*. So, in the example above the consequent is waiting for confirmation that John has a dog. But where does this confirmation come from?

Well, on the story we envision, the confirmation must be delivered by Perceval and the question is where he will deliver the confirmation from. If the confirmation cannot be delivered from within the sentence, then Perceval will have to deliver the confirmation from outside of the sentence, i.e. from the discourse common ground; perhaps accommodation will be necessary. These are the cases in which linguists typically talk about "projection," but to our way of thinking, nothing is projected. It is merely a question of whether confirmation is found within the sentence or in the discourse. Sentence internal confirmation yields the illusion that the presupposition does not project. Sentence external confirmation yields the illusion that the presupposition has projected.

There are two crucial features of (42b) which allow for the sentence internal confirmation of the presupposition. First, the presupposition 'John has a dog' is contained in the conditional; more precisely, it forms the antecedent of the conditional. Second, Perceval can freely move from the antecedent 'John has a dog' to the position in the consequent that is awaiting confirmation that the presupposition is satisfied. In other words, the presupposition 'John has a dog' is contained in the compound sentence and positively p-scopes over the position awaiting the confirmation.

In (42a), Perceval cannot deliver the confirmation from within the sentence (creating the illusion that the presupposition projects) simply because the sentence does not contain the relevant information in any shape or form. But sometimes the presupposition projects even if it can be found in the sentence. These are the cases when Perceval cannot make the trip from the presupposition to the position awaiting confirmation. That's what happens in (43a). Perceval starts out with the golden armor from the presupposition 'Jane smoked', but encounters a silver disjunction at the root, and is unable to proceed into the consequent. Compare this sentence, where presupposition projects, to (43b), where it does not project, because Perceval can make the trip. Once again, he starts out with the golden armor, but his armor is transmuted to silver by the alchemist sitting in the negation triggered by 'never'. When Perceval now arrives to the silver disjunction at the root, he is wearing the correct armor and may proceed into the consequent to deliver the confirmation.

(43) Jane stopped smoking. $>$ Jane smoked.

a. Either Jane smoked, or she stopped smoking.
$>$ Jane smoked.

b. Either Jane never smoked, or she stopped smoking.
$\not>$ Jane smoked.

We believe that the phenomenon of presupposition projection is another case where a deeper syntactic explanation trades on the relation of (positive) p-scope. As we know, the antecedent of a conditional positively p-scopes over the consequent, and this is why conditionals constitute an environment where the presupposition can be confirmed by sentence-internal information. It need not be confirmed by the common ground (hence the presupposition "does not project").

We take this story to be compatible with much of the semantics literature on presupposition projection and we think it is worth noting that our story answers the worry that the account in Heim (2002) is ad hoc on the difference in behavior between binary logical connectives and quantifiers—see Rothschild (2011) for a survey of literature attempting to ameliorate the problem. From a syntactic perspective, we can motivate the difference in these connections as extruding from our standard story about p-scope and Perceval's travels. If you understand why Perceval can travel where within a sentence, then you will understand why the facts about presupposition break as they do with respect to the logical connectives. They break as they do because of the properties of p-scope!

11.3 Negative Polarity Items

In the previous two sections we looked at cases where our theory provided a new framework for investigating topic areas in linguistics (discourse anaphora and presupposition projection) that are quite vast. In this section, we look at an area of investigation that is even more vast; again, not with the intent to solve every problem, but with the intent to provide a new framework and a fresh perspective from which we can investigate these matters.

Negative Polarity Items, or *NPIs*, are expressions like 'anyone' and 'ever' that appear in negative contexts, as in the following cases.

(44) a. *John saw anything/anyone.
b. John didn't see anything/anyone.

(45) a. *Max said that he had ever been there.
b. Max never said he had ever been there.

The inventory of these expressions is extensive (including expressions like 'budge an inch' and 'give a damn' in English), and they are ubiquitous across human languages. The question is, why are they there, and what triggers their occurrence? The "why are they there" question has a number of interesting answers, not least of which is a proposal by Dowty (1994) that they play a role in identifying downward entailing environments to facilitate inferences in Natural Logic. So for example, if someone says 'The landlord doesn't allow any dog' the 'any' is a sign that you can plug in any subset of dogs. Definitely no collies either! Whether or not this story is correct,[16] it is natural to wonder whether we, having written a book on polarity and logical inference, have anything helpful to say on the subject.

The literature on NPIs is dense and astoundingly complicated. Not all NPIs behave the same way, even in the same language. What we have to say is thus not the final word, but the introduction of some new tools for investigating the subject matter.

Our point of departure for this topic is Ladusaw (1979), who argued that negative polarity items (again, expressions such as 'any' and 'ever') are licensed (or triggered) by downward entailing environments. For instance, examples (44a) and (45a) above are cases where the NPIs are placed (unhappily) in upward entailing environments. The corresponding (44b) and (45b), in which the NPIs occur in downward entailing environments, are much more acceptable.

Care is necessary to distinguish these instances of 'any' from so-called "free-choice" 'any', which need not appear within the scope of negation, as in the example below.

(46) a. I might have said anything. I was furious.
     b. I would have punched anyone who said that to me.

It's clear that 'any' in these examples has a different meaning than it does in (44). 'Any' in (44) means something or someone. Not so in (46), where there is an additional suggestion that everything is sayable and that everyone is a possible target of my wrath. A standard analysis is that free-choice 'any' is a universal quantifier with wide scope over the modal. Not only is there an apparent difference in meaning, but as a general rule, free-choice 'any' needs to be licensed by modals; see Carlson (1981) for a discussion of the distribution of free-choice 'any'. It is interesting to note that in certain natural languages, 'any' and free-choice 'any' are not homophones. Serbo-Croatian, for example, has 'iko' ("any") and 'bilo' (free-choice "any"); see Progovac (1990) for discussion.

[16] Recent research by Barker (2018) suggests perhaps not.

As we saw in section 5.4, there are many other possible downward entailing environments, including the first or second position of certain determiners. Thus we have the following distribution of facts:[17]

(47) a. Every [person who has ever been to NY] [has returned to it].
b. *Every [person who has been to NY] [has ever returned to it].

(48) a. *Some [person who has ever been to NY] [has returned to it].
b. *Some [person who has been to NY] [has ever returned to it].

(49) a. No [person who has ever been to NY] [has returned to it].
b. No [person who has been to NY] [has ever returned to it].

One of the consequences drawn by Ladusaw was that apparently syntactic well-formedness conditions must in fact appeal to the semantics of the expression. Syntax alone can not get the job done, or so he suggested.

> We have seen that the property of [unacceptable sentences with NPIs] which renders them unacceptable is to be defined in terms of the entailments licensed by certain lexical items, rather than by simply marking certain morphemes with a semantic feature. It seems to follow directly that no grammar can in principle distinguish [between acceptable and unacceptable sentences with NPIs] unless its semantic component aims higher than at simply disambiguating sentences by deriving 'logical forms' for them to the goal of providing a theory of entailment for the language it generates. (Ladusaw 1979, pp. 14–15)

Clearly, if L** translations of natural language sentences are construed as giving the logical forms of those sentences, then there is an immediate answer to Ladusaw—specifically, that the property which renders some sentences with NPIs unacceptable can be defined syntactically in terms of negative and positive occurrences.

This suggestion would also require that the general treatment of directional entailingness can also be extended to certain intuitively negative predicates like 'doubts', 'forgets', and 'difficult' below.

(50) I doubt that he ever speaks to her.

(51) John forgot to bring anything to dinner.

(52) It is difficult to find any squid at Safeway.

Following C. L. Baker (1970) and Linebarger (1987), we might decompose these lexical items into a more primitive predicate and a negation. The suggestion is

17 These specific examples are drawn from Larson (1995).

natural, given that all of the verbs which induce downward entailing environments appear to have a negative connotation to them. For example, in the pairs 'doubt/ believe', 'forgot/remembered', and 'difficult/easy' any informant would judge that 'doubt', 'forgot', and 'difficult' were the pair members which have a negative element. The idea would be that (50)–(52) might be rendered as in (53)–(55).

(53) I [not-believe] that he ever speaks to her.

(54) John [[not-remember]-ed] to bring anything to dinner.

(55) It is [not-easy] to find any squid at Safeway.

Ladusaw (1979) objected to this proposal, observing that the analysis predicts that sentences like (53)–(55), when conjoined with the negation of their corresponding sentence in (50)–(52) should form a contradiction. In some cases, this prediction is born out (56), but in others it is not (57).

(56) John didn't remember to bring anything, but he didn't forget to either.

(57) It isn't hard to find squid at Safeway, but it isn't easy either.

However, Ladusaw's objection does not falsify the hypothesis that 'difficult' and similar items should receive a complex logical form involving a negation. It only dispels the naive idea that all negatively flavored items will be a simple negation of their basic, positive counterpart. And indeed, the predicates which do not invoke contradictions admit a middle range of objects or events which have neither of the contrastive properties (e.g. which are neither hard nor easy), which is a certain indicator of a structure more complex than a structure yielded by a simple non-negated/negated dichotomy.

While this line of investigation breaks with Ladusaw on the question of whether a syntactic account of directional entailingness is possible, it still assumes that Ladusaw was correct in insisting that there is a close relation between entailment relations and the licensing of NPIs. Our point would simply be that, following the central thesis of this book, entailment relations can be syntactically defined as well.

Theoretical questions about the nature of entailment relations aside, it has been known for some time that, despite its appeal, there are a number of apparent counterexamples to Ladusaw's generalization. Specifically, there are a number of cases in which NPIs are licensed, though they do not appear in a downward entailing environment.

(58) a. Most [people who know anything about politics]$_P$ [hate it]$_H$.
b. $\exists x\colon (\exists n\colon \#(n,x) \wedge \neg(\exists y\colon \#(n,y) \wedge \underline{P(y)} \wedge y \wr x)) \wedge P(x) \wedge H(x)$

(59) a. Exactly two [people that know anything about logic]$_P$ [read the book]$_R$.
b. $(\exists x\colon \underline{2}(x) \wedge P(x) \wedge R(x)) \wedge \neg(\exists x\colon \underline{3}(x) \wedge \underline{P(x)} \wedge R(x))$

(60) a. The [philosopher that knows anything about logic]$_P$ [can snow anyone]$_S$.
b. $\exists x \colon \forall y\,(\neg \underline{P(y)} \vee y \circ x) \wedge P(x) \wedge S(x)$

(61) a. More [cats] than [dogs] [have ever eaten a mouse]$_M$.
b. $\exists n \colon (\exists x \colon \#(n, x) \wedge C(x) \wedge M(x)) \wedge \neg(\exists y \colon \#(n, y) \wedge D(y) \wedge \underline{M(y)})$

Considered in their L** logical form, one observation regarding these sentences stands out. These are all sentences which have translations into L** in which the NPI has at least one negative occurrence (underlined). It might be, then, that a better generalization is available. Specifically, it is not directional entailingness which is key, but rather whether the NPI has at least one negative occurrence when in L** form.

This revised generalization, if correct, would also shed light on certain facts about conditionals discussed in Heim (1984). Heim notes that while the antecedents of conditionals in natural language routinely license NPIs—consider (62)—they are not downward entailing environments. If the antecedents of these sentences are "strengthened," as in (63), truth is not necessarily preserved.

(62) a. If you ever ate worm cheese, you know what I'm talking about.
b. If anyone sees you eat worm cheese, they will never talk to you again.

(63) a. #If you ever ate worm cheese but don't remember doing it, you know what I'm talking about.
b. #If anyone sees you eat worm cheese (and don't know what it is), they will never talk to you again.

This anomaly can be accounted for if, following several theories (e.g. Lycan 1984; Kratzer 1989) conditionals are thought of as having an implicit quantification over events or situations. Specifically, if conditionals like the above are thought of as holding generally (that is, for most cases of worm cheese eating), then they will receive an analysis like we gave for 'most' in section 5.3, so the predicates occuring in the antecedent of these conditionals will have both positive and negative occurences. So we predict that these predicates will be in neither upward nor downward entailing environments, but that the environment will license NPIs.

This revised generalization also opens the door to a possible explanation of why questions license NPIs, as in (64), when it is unclear whether the notion of directional entailingness is even applicable.

(64) a. Did you see anyone?
b. Who saw anything?

If a question such as (65a) is thought of as having the underlying form of a disjunction (65b), then we would have a natural explanation of why yes/no questions license NPIs when they do not seem to support either downward entailing or upward entailing inferences. In fact, there are languages, like Chinese, where such a logical form of (some) yes/no questions is even suggested by the surface syntax.[18]

(65) a. Is the Earth round?
b. $\neg R(e) \vee R(e)$

(66) Nǐ chī bú chī yángròu?
you eat not eat lamb?
"Do you eat lamb?"

If this idea is correct, the logical form of a question has a very special logical property of being a tautology, which might very well form an important contribution to its interrogative force. Can we generalize the idea to wh-questions?

Yes, we can. The logical form below is a synthesis of two ideas. The first is a basic intuition about focus, shared by many linguists: focus is the answer to a question (Kadmon 2001, pp. 261–253); the question itself is therefore the background to the focus. Second, we follow Herburger's (2000) analysis of focus, which we have already outlined in section 9.1, to arrive at (67) (P in that logical form stands for 'person'). Let us explain how this logical form arises. Under Herburger's analysis, the background forms the restriction of the wide-scope quantifier (below, $\forall x$), while the scope consists of both the background and the focused element. Now, in a wh-question, the focus is the interrogative element itself and therefore makes no contribution to the logical form.[19] The restriction and the scope of $\forall x$ are thus the same. (67b) is therefore a tautology.

(67) a. Who did you see?
b. $\forall x : \neg(P(x) \wedge S(\text{you}, x)) \vee (P(x) \wedge S(\text{you}, x))$

Summing up, if we adopt the idea that the logical form of questions is a tautology in the form of a universally quantified structure, arising from the tight association of question-formation and focusing, we can straightforwardly explain why questions license NPIs. As the restrictor and the scope of the universal quantifier are the same, every word in a question occurs both with negative and positive polarity. The NPI will therefore always have an occurrence of negative polarity.

All of this leads us to posit the following working hypothesis (emphasis on *working*).

[18] Example (66) is from Po-Ching and Rimmington (2004, p. 351).

[19] Two remarks are in order here. First, we take the restriction of 'who' to people to be a part of the background. Second, while the syntactic form of the sentence surely contains a functional head carrying the interrogative force, we assume that this functional head has no truth-conditional impact.

(68) An NPI is licensed in a sentence iff at least one occurrence of the subformula corresponding to the NPI lies within the scope of at least one negation in the L** rendering of the sentence.

Does this mean that syntactic accounts of NPIs have the edge over semantic accounts? That seems unlikely to us, given that in our L** system, the syntax and semantics are supposed to run in parallel, and that should lead us to suppose that our syntactic story would have an analogous story in the semantics. Take our idea regarding 'most', for example. As we noted, the first position of the determiner will have two occurrences, one positive and one negative. Does the traditional semantics for determiners have something similar going on? As we noted in section 3.3, it would seem so. Consider again a standard set-theoretic definition for 'most' in generalized quantifier theory.

(69) $[\![\text{most}]\!](A, B)$ iff $|A \cap B| > |A - B|$

If we are looking for something parallel to the syntactic case, then we want to look at the different occurrences of A in this definition and see if something different is going on, and of course there is. In the first occurrence of A, by basic principles of set theory we know that you can keep plugging in supersets for A and preserve truth value. With respect to the second occurrence of A we know that we can plug in subsets of A and preserve truth value. Thus, the A position has both an upward and downward entailing environment, in the full definition of the determiner.

We assume that this will be true for most if not all cases. If the semantics and syntax are in lock step, as we intend them to be, any syntactic account of NPIs should break precisely where the semantic accounts do (and vice versa), and they should both succeed in parallel as well.

Still, it is important that syntactic accounts exist, for NPIs are elements that occur in the syntax of natural language, after all, and we would like our account to integrate with the syntax of natural language.

Keeping in mind this ecumenical approach, there is still a question to be asked: Just how good are polarity logic accounts of NPIs (whether semantic or syntactic)? For now, let's consider this as a question at a kind of abstract level of syntax. Does the hypothesis above hold up?

As we noted, the literature on NPIs is quite vast, not least because NPIs appear to show up in every language and they also appear to show up in multiple forms. We can't possibly work through all of this literature, but we can close with a discussion of some observations about NPIs as an illustration of how our framework can be useful as a tool for investigating the variety of NPIs that have been discovered.

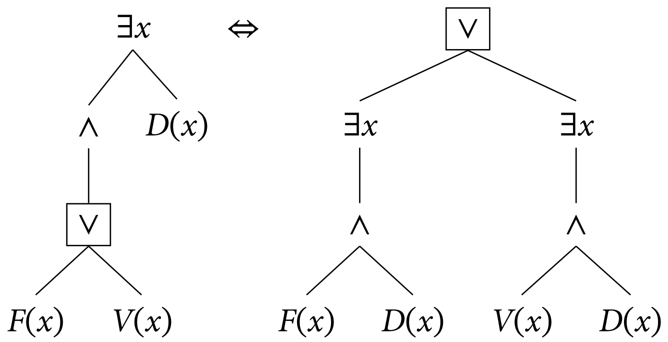
Zwarts (1981) has noted that NPIs come in different strengths in languages like English and Dutch (and many other languages). So, for example, there are weak NPIs like 'any' that only seem to need one negative occurrence to be licensed. But strong NPIs like 'in years' must satisfy a stronger condition, and super-strong NPIs like 'one bit' require something more like an explicit negation to appear. Zwarts suggests that the strength of the NPI that is licensed depends on certain monotonicity properties of the environment in which the NPI appears. Specifically, it depends upon whether the NPI appears in an environment that is anti-additive, or anti-multiplicative, or both. What are those notions?

The properties are defined in (70), where the function f is the environment created by the determiner or operator in question. So, for example, if 'some' is additive in its first position (it is) then (71) should hold. This is illustrated in (72): if we plug a disjunction into the first argument (the left tree), we can "move it up" (the right tree) without changing its type (the type changes with the anti-properties).

(70) a. f is ADDITIVE iff $f(x \vee y) = f(x) \vee f(y)$
b. f is MULTIPLICATIVE iff $f(x \wedge y) = f(x) \wedge f(y)$
c. f is ANTI-ADDITIVE iff $f(x \vee y) = f(x) \wedge f(y)$
d. f is ANTI-MULTIPLICATIVE iff $f(x \wedge y) = f(x) \vee f(y)$

(71) Some fruits or vegetables are delicious iff some fruits are delicious or some vegetables are delicious.

(72)



So the idea is that different NPIs are triggered by the environments defined in (70). That might seem like a win for semantic accounts of NPIs, but it is not necessarily so. The interesting question is whether there are obvious syntactic features of these environments that can be readily identified by inspection (just as we showed that upward and downward entailing environments can be identified by inspection). It turns out that it isn't difficult to identify these environments in our system.

To see how the monotonicity properties in (70) are reflected in the syntax of our system, consider Table 11.1—the first part for now. In the third column, we represent the structural environment determined by the determiners. First of

Table 11.1 Monotonicity properties of determiners

operator	first position	L** analysis	second position
some	additive	$\exists[+\wedge+]$	additive
some are not	additive	$\exists[+\wedge-]$	anti-multiplicative
every	anti-additive	$\forall[-\vee+]$	multiplicative
no	anti-additive	$\forall[-\vee-]$	anti-additive
(only	multiplicative	$\forall[+\vee-]$	anti-additive)
is	add. & mult.	$++$	add. & multi.
is not	anti-a. & anti-m.	$--$	anti-a. & anti-m.

all, all of the items introduce a quantifier and a junction.[20] For example—and look at (72)—'some' is built of an existential and a conjunction, with none of the conjuncts being a negation (thus two pluses). Flanking those representations are the monotonicity properties of the first and the second position of each operator.

So our first hypothesis would be that these monotone environments are as in (73), where by governed we mean being in the immediate scope of a restricted quantifier.[21]

(73) a. An environment is additive if it has positive polarity and is governed by an existential quantifier.
b. An environment is multiplicative if it has positive polarity and is governed by a universal quantifier.
c. An environment is anti-additive if it has negative polarity and is governed by a universal quantifier.
d. An environment is anti-multiplicative if it has negative polarity and is governed by an existential quantifier.

As Zwarts notes, the super-strong NPIs like 'one bit' have a preference for environments that are both anti-additive an anti-multiplicative. What would trigger such an environment in our system? A pure negation, i.e. a negation not governed by a quantifier, as in a negative copular sentence. And similarly a positive polarity copular sentence would be both additive and multiplicative.[22] Both environments are shown in the second part of Table 11.1. So we can flesh out the rest of our paradigm as in (74).

[20] The "missing" entries correspond to non-conservative combinations of the quantifier and the junction. But the monotonicity properties hold for non-conservative combinations as well, and this is why we include 'only'—see section 9.1 and subsection 9.2.2 for discussion about its conservativity and details about its logical form.

[21] (73) and (74) provide sufficient but not necessary conditions.

[22] It is easy to see that junctions have no effect on the monotonicity properties. This follows from basic properies of Boolean connectives: associativity, idempotence, distributivity, and De Morgan's laws.

Table 11.2 Syntactic characterization of monotonicity properties

	polarity	**governing quantifier**
additive	positive	~~universal~~
multiplicative	positive	~~existential~~
anti-additive	negative	~~existential~~
anti-multiplicative	negative	~~universal~~

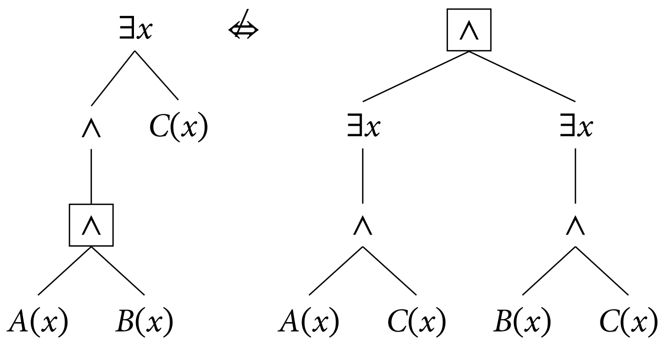
(74) a. An environment is additive and multiplicative if it has positive polarity and is not governed by a quantifier.

b. An environment is anti-additive and anti-multiplicative if it has negative polarity and is not governed by a quantifier.

Notice something? What we have really discovered here is not that a certain type of quantifier allows a certain monotonicity property, but rather that it *blocks* it! As shown in Table 11.2, existentials block multiplicativity and anti-additivity, while universals block additivity and anti-multiplicativity.

But now look at the definitions in (70) again. Additive and anti-multiplicative functions have a conjunction on the right-hand side; multiplicative and anti-additive functions have a disjunction there. Therefore, existentials block a conjunction "above" them—the marked node on right side in (75), which shows the failure of multiplicativity of 'some'—and universals block a disjunction. Generalizing further, we arrive at (76). Not only is it possible to syntactically characterize the monotonicity environments in (70) (and two more), we can do it in a single sentence!

(75)



(76) A quantifier of a certain Angel/Demon type blocks a junction of the opposite type.

This hypothesis obviously needs to be sharpened up, and a proof is ultimately called for, but our interest here and now is not to deliver a technical result, but rather to convey something about the excitement and utility of this approach to the logic of determiners and polarity. Nor is it our desire to "solve" the problem of NPIs (whatever that might mean) but rather to provide a new platform for investigating the syntactic reflexes of NPIs in their rich variety and complexity.

Of course in saying all of this we have so far been discussing linguistic structure at a fairly abstract level, and we have often helped ourselves to the structures on offer in our language L^{**}. We promised at the beginning of the chapter that we would ultimately show how those properties can be reflected directly in contemporary grammatical theory and in particular in the syntax of the Minimalist Program. In the next chapter we deliver on that promise.

12
L**, LF, and Logical Form

In earlier chapters we showed that for formal languages like L* and L**, directional entailingness is a direct reflection of logical polarity. We also showed that a very deep and dynamic deductive system is available for those formal languages. But along the way you may have wondered what any of this had to do with natural language, and hence what any of it had to do with Natural Logic. For example, the syntax of L** is full of disjunctions and conjunctions that are not evident in spoken and written language, so one might ask whether we aren't talking about some sort of formal language rather than natural language.

In this chapter we will make the case that L** actually provides a very good initial picture of the syntax of natural language. To show this, we have to tackle two issues. The first issue has to do with fact that L** is a version of first-order predicate logic, and we have all been told over and over again, in many semantics textbooks, that predicate logics are not viable candidates for accounts of natural language. For example we are told that first-order predicate logic lacks sufficient expressive power, that it isn't compositional, and that it isn't remotely isomorphic to the apparent structure of natural language. Let us call these concerns *the semantic worry*, *the interface worry*, and *the syntactic worry*. But as we will see, we have already addressed the semantic worry, and the interface worry disappears if we can answer the syntactic worry—which we will, as we will argue that there is actually a strong isomorphism between L** and natural language syntax. And after we show you that L** is actually a plausible candidate for the syntax of natural language, there will remain the nitty gritty business of showing, in detail, how our account would embed within contemporary syntactic theory.

As we will see, the last quarter century of work in generative linguistics has revealed the fine-grained structure of natural language, in particular in the structure of determiner phrases and in appearances of abstract negations. While it may have been difficult to embed L** within linguistic theory as it stood forty or fifty years ago, that is no longer the case. We will see that there is a very natural way to incorporate the insights of L** syntax into the technical machinery of contemporary generative linguistics. But to really have a viable story we expect some payout from the attempt to integrate L** with generative linguistics. We expect the effort to be illuminating of some important features and/or problems within generative linguistics. As we will see, L** actually has a lot of useful things to say about the structure of Determiner Phases, about movement, and even some things about a class of phenomena that linguists have called *Relativized Minimality*.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović,
Oxford University Press. © Peter Ludlow and Sašo Živanović 2022. DOI: 10.1093/oso/9780199591534.003.0012

In other words, we won't claim that L** solves all outstanding linguistic problems, but we can say that it communicates with and contributes nicely to the enterprise of contemporary generative linguistics.

We obviously have a lot of ground to cover in this chapter. We begin with the three worries mentioned above, and show that the usual reservations about predicate logics don't present problems—at least not for the L** version of predicate logic. We then turn to the issue of embedding the L** syntactic analysis within generative linguistics. We have chosen to do this within the Minimalist Program, associated with Chomsky (1995) and much subsequent work. We don't do this to endorse a particular framework so much as to show how the integration can be carried out in one popular contemporary framework. As we will see, there will be syntactic payoffs from the integration.

12.1 Natural Language as Predicate Logic?

On our view, the quantifierless L**, a "plural" language of first-order predicate logic, is an excellent candidate for the formal logic of natural language. However, predicate logics are not the choice of mainstream contemporary semantic theories, which prefer some variant of Generalized Quantifier theory. What are the main objections against deploying predicate logics in this role? We will discuss three objections typically encountered in semantic textbooks, and we will argue that our quantifierless L** founders on none of them.

One objection is that predicate logic allegedly does not have the expressive power that could do justice to natural language. To respond to this objection, we will return to the results from Chapter 5 to show that while the objection applies to the standard predicate calculus, it does not apply to predicate logics in general, and in particular, it doesn't apply to L**. In fact, we will argue that L** resides in the Goldilocks zone between the unsupplemented predicate calculus (which does not have sufficient expressive power) and generalized quantifiers (which have too much expressive power).

The second objection is that predicate logic approaches are sometimes said to be non-compositional, although this objection is usually immediately qualified to explain that a compositional treatment is possible after all. In subsection 12.1.2, we will thus consider our proposal (and predicate logic proposals in general) through the lens of compositionality.

Finally, it is frequently argued that the overall structure of logical forms based on predicate logic differs from the structure of natural language sentences. In subsection 12.1.3, we will show that at least for L**, this objection only holds if we are stubbornly inflexible in the encoding of predicate logic expressions. We will see that if we take advantage of our observations concerning restrictedness and the quantifierless language that this allows us, the resulting formalism appears very much isomorphic to the syntax of natural language.

12.1.1 The expressive power worry

The most straightforward—and, if valid, also the most powerful—objection against predicate logic as a candidate for the logic of natural language is that it simply does not have sufficient expressive power to encode the truth conditions for each and every (declarative) sentence of natural language.[1] The prototypical example of the alleged ineffability—sentences containing the superlative determiner 'most'—is due to the often-cited work of Barwise and Cooper (1981).

We readily concede that 'most' cannot be analyzed using predicate logic *calculus*, i.e. predicate logic (with or without identity) without any additional axioms. And this is precisely what Barwise and Cooper (1981) prove. However, this does not mean that a particular predicate logic *theory*, i.e. predicate calculus armed with certain predicates and axioms governing their interpretation, cannot do the job. In fact, the opposite is patently true. The mainstream analysis of natural language quantification, i.e. Generalized Quantifier Theory, is based on set theory, and set theory is typically axiomatized within first-order predicate logic, using the well-known Zermelo-Fraenkel axioms.

This was of course clear to Barwise and Cooper, who wrote that in order to define 'most', for simplicity interpreted as 'more than half', "one has to leave traditional first-order logic in one of two ways. One possibility [...] one might mirror the high-order set-theoretic definition of 'more than half' in the semantics by forcing every domain E to contain all of the abstract apparatus of modern set-theory" (p. 161). As is well known, they advocated for the other possibility—now known as Generalized Quantifier theory—which is to "keep the formal definition as part of the metalanguage, and treat generalized quantifiers without bringing all the problems of set theory into the syntax and semantics of the logic *per se*" (p. 161).

Now there is one detail where we don't agree with Barwise and Cooper, and it turns out to be crucial. We don't believe it is necessary to force the logic of natural language to contain *all* the abstract apparatus of modern set theory—for example, we certainly don't need the axiom of infinity or the axiom of choice to make our theory tick. As illustrated by our development of L^* and L^{**} in Chapters 4 and 5, a more modest part of that apparatus will suffice—or more precisely, a more modest

[1] See e.g. Cann (1993, p. 188); Chierchia and McConnell-Ginet (1993, p. 114); Heim and Kratzer (1998, pp. 191–193); Saeed (2009, p. 325); Kroeger (2019, p. 258); and Coppock and Champollion (2021, p. 237).

apparatus, full stop, will suffice, as the axioms of L** theory proposed in section 5.2 do not involve set theory in any shape or form.[2]

We don't want to rehearse the development of (L* and) L** here, so let us just drop in a quick reminder of what the final language looks like. We have characterized L** as a first-order mereology supplemented with a counting mechanism. As such, the language relies on two groups of non-logical predicates. The first group comprises the subcollection predicate $\subseteq$ and predicates overlap $\circ$ and disjointness $\wr$ definable off of $\subseteq$. These are the basic mereological predicates, and their linguistic utility was argued for in the context of the semantics of plurals (Link 1983) and analyses of various part–whole relations in natural language (e.g. Wągiel 2018). The other group consists of the quantity predicate # and the number predicates (but note that the latter don't actually play a role in the L** analysis of 'most'). Here, it seems even more obvious that natural languages must make use of some such predicates—after all, we do use numerical expressions in our speech, and we do count and we do refer to numbers.[3]

The moral of L** is that complex quantification can be implemented within a first-order theory of meager resources. After all, it is well known that mereology is much weaker than set theory—for example, Hellman (2017, p. 413) notes that "all by itself, mereology is too weak to provide a framework for even very elementary mathematics"—and the rudimentary theory of counting supplementing the mereological aspect of L** does not add much to the theory—just enough to implement non-elementary quantifiers such as 'most'.

The intuition that we are dealing with a relatively weak theory is reaffirmed under the alternative interpretations of L**. As we remarked in section 5.2, we can see L** as a kind of plural logic—which, remember, Boolos ([1984] 1998) shows to be equi-interpretable with monadic second-order logic—or as a fragment of the simplest infinitary logic, $L_{\omega_1\omega}$. While these three systems are not exactly equi-interpretable—for example, the $L_{\omega_1\omega}$ interpretation only works for countably infinite collections of individuals, a property that was argued to be a feature, not a bug, by Law and Ludlow (1985)—their expressive power is certainly "of the same magnitude." They transcend the power of first-order calculus, but not by much, and can thereby retain several desirable first-order properties. For example, $L_{\omega_1\omega}$ is complete, and so is the variant of plural logic developed by Oliver and Smiley (2006).

We don't claim that L**, as it currently stands, suffices for the analysis of the entirety of natural language. For example, there is the question of whether

[2] But see Lewis (1991) for a mereological treatment of the set theory.

[3] The language Pirahã is famous for allegedly lacking numbers or a concept of counting (Everett 2005). However, the existence of such a language does not invalidate the need for number predicates in other languages. Even more, subsequent research on Pirahã (Frank et al. 2008) seems to support our distinction between the quantity predicate (perhaps better understood as one-to-one correspondence predicate, see footnote 14 on page 74) and the number predicates.

something special needs to be said about quantification over times and possible worlds in the analysis of tense and modality. In our view, L^{**} is more of a platform for future investigation than it is some sort of finished science. But we believe that even in its current state of development, it gives us a strong reply to the main objection to deploying predicate logic in the analysis of natural language. That objection derives from the issue of non-elementary quantification, and we believe that L^{**} handles that issue cleanly. The existence of L^{**} also shows that it is careless to reject all incarnations of predicate logic because of the lack of expressive power in predicate calculus. Such an objection is ungrounded in the case of L^{**} and (most probably) many other first-order theories.

12.1.2 The compositionality worry

Compositionality is often seen as a central methodological principle guiding the formalization of theories in the semantics of natural language; a non-compositional theory is automatically considered suspect. Some introductory texts include non-compositionality among the arguments against using predicate logic in the analysis of natural language.[4] We will rehearse both the argument and the standard reply, and then move to a more general discussion on compositionality, with three goals in mind. First, we want to emphasize that if our project can be fully executed, the mapping from LF to L^{**} is trivially compositional. Second, we want to anticipate an objection that the interpretation of L^{**} itself is non-compositional. Third, we want to address the question of whether compositionality is all that different from recursiveness—a property that predicate logics are known to have. Another way to put this: At the end of the day, we may well wonder what all the fuss is about.

In its most general form, the principle of compositionality can be stated as follows.

(1) COMPOSITIONALITY. The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.
(Partee 2004, p. 153)

Given this definition, we need to understand compositionality in terms of syntactic expressions (whatever they are), and meanings (whatever they are). As we will see, the exact understanding of 'meaning' is fluid. Atomic expressions have some given (lexical) meaning, while the meaning of complex expressions is computed. Simplifying somewhat, whatever 'Every dog barks' means depends on

[4] They include Cann (1993, p. 159); Larson and Segal (1995, §8.1); and Partee (2005, p. 2).

what 'every', 'dog', and 'barks' mean. And clearly, how the words are composed into a sentence matters. For example, switching the subject and the object of a sentence with a transitive verb will change the meaning; 'A dog saw a cat' does not mean the same as 'A cat saw a dog'. But ideally,[5] nothing else matters; the meanings of the words composing a sentence and the manner in which these words are combined jointly *determine* the meaning a sentence—or, as this is commonly phrased, the meaning of a complex expression is a *function* of the meanings of its constituents and the manner of their composition.

The principle stated above can be strengthened by requiring that the meaning of an expression must not only be a function of its parts, but the meaning of each node must be a function of the meanings of its *immediate* daughter nodes. In fact, it is this STRICT or STRONG version of the compositionality principle which is usually taken on in the semantic circles (see e.g. Larson and Segal 1995; Partee 2004). We adopt the formulation from Hodges (1998), as he clearly states (2a), a presupposition of the principle which remains implicit in most other formulations.

(2) STRONG COMPOSITIONALITY (Hodges 1998, p. 17)
a. DOMAIN RULE. If an expression is meaningful then so are its constituents.[6]
b. FUNCTIONALITY. There is a function ϕ such that for every meaningful complex expression e of the form $\alpha(e_1, \ldots, e_n)$,

$$\mu(e) = \phi(\alpha, \mu(e_1), \ldots, \mu(e_n)).$$

We will make use of the somewhat more abstract formulation of the functionality component of the principle toward the end of the section. To connect (2b) to the more familiar plain English statement of compositionality, let us embellish (1) with Hodges' symbols: The meaning ($\mu(e)$) of an expression (e) is a function (ϕ) of the meanings ($\mu(e_i)$) of its parts (e_i) and of the way they are syntactically combined (α).

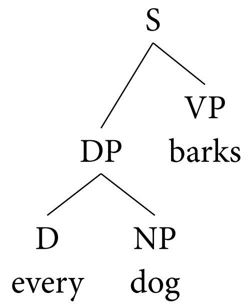
Let us go through the symbols in (2b) once again to make it more concrete. First, e is a syntactic expression such as the word 'dog' or the sentence 'every dog barks'. More precisely, as the latter expression is complex, e in fact refers to the syntactic structure of this sentence, (3a). Second, μ is the meaning function, so μ('dog') and μ((3a)) refer to the meanings of 'dog' and (3a). Finally, α is a syntactic formation rule, for example a rule which composes DP 'every dog' out of determiner (D) 'every' and noun phrase (NP) 'dog', as depicted in (3a). In this case, the immediate parts e_1 and e_2 refer to 'every' and 'dog'. However—and this is the essence of *strict* compositionality—'every' and 'dog' are not among the immediate parts of

[5] We abstract away from the issues such as the role of context and non-compositionality of idioms.

[6] Janssen (2011, p. 530) provides a nice example of a semantics which is non-compositional despite the fact that it does not violate the domain rule.

the entire (3a). The entire sentence (S) is composed by a subject–predicate rule with e_1 being DP 'every dog' and e_2 being VP 'barks'.

(3) a.



b. $\forall x\colon D(x) \Rightarrow B(x)$

To reiterate, in our illustration of the compositionality of the sentence 'Every dog barks' above, we imagined a sentence as a template which gets filled by words, and the meaning of the sentence was determined from the structure of the template and the meanings of the words. But in reality, words in a sentence form a hierarchical structure such as (3a), and the idea behind strict compositionality is that every node (actually, subtree) of that structure has meaning, and that the computation of the meaning of a node may only involve the meanings of its children (rather than arbitrary descendants). For example, to arrive at the meaning of (3a) in a strictly compositional way, we need to compute it from the meanings of 'every dog' and 'barks', rather than directly from the meanings of 'every', 'dog' and 'barks'.

This is where predicate logic is supposed to give rise to compositionality problems. We are supposed to have the logical form (3b) as the meaning for (3a), but what is the logical form of the intermediate constituent 'every dog'? As the determiner 'every' is intuitively responsible for both $\forall$ and $\Rightarrow$ in (3b), one might be tempted to say that the logical form of 'every dog' is $\forall x\colon D(x) \Rightarrow$, but that won't work as this component is not a well-formed formula. An alternative might be to take $\forall x\colon D(x)$ as the logical form of the DP, but this runs into trouble as well. Even assuming we could somehow cook up a rule to combine $\forall x\colon D(x)$ and $B(x)$ into (3b) in a principled way—and this might be an impossible task—'every dog' certainly does not mean $\forall x\colon D(x)$, which can at best be paraphrased as "everything is a dog."

In a nutshell, predicate logic allegedly founders on the domain rule part of strong compositionality. The issue can be resolved by borrowing an idea from formal semantics and extending predicate logic with lambda calculus (cf. e.g. Cann 1993). We will not go into any details on lambda calculus—for a thorough recent introduction, see Coppock and Champollion (2021)—but the following example should make the gist of the idea clear. The meaning of 'every' is taken to be $\lambda P \lambda Q \forall x\colon P(x) \Rightarrow Q(x)$, which can be imagined as a quantificational template $\forall x\colon P(x) \Rightarrow Q(x)$ with slots P and Q, which the meanings of NP and VP will be plugged into. Determiner 'every' first combines with NP 'dog', whose meaning is D.[7] This strips away the outer λP and replaces P by D. The result of this step is the meaning of 'every dog': $\lambda Q\,\forall x\colon D(x) \Rightarrow Q(x)$. Combining this expression with the meaning of 'barks' (B) then strips away λQ and substitutes B for Q, resulting in (3b) as the logical form of (3a).

Can predicate logic theorists help themselves to this maneuver? Is it not cheating to introduce the lambda calculus in this way, and does it not leave us with a system that is more powerful than a simple predicate calculus? The answers to these question are not entirely straightforward. One could argue that the appeal to the lambda calculus is benign here because the lambda conversion machinery is only deployed to build predicate logic structures, while the real semantic action is found in the resulting predicate calculus.

Before moving on to a more general discussion, let's extract the moral of the story developed so far: The price of demanding a strongly (strictly) compositional semantics is an enriched concept of meaning. While the weakly compositional account can equate (an important aspect of) the meaning with predicate logic formulas, a strongly compositional account has to take meanings to be lambda abstractions over such expressions. Such enrichment of meaning is a recurrent theme in "compositionalization." For example, Dynamic Predicate Logic (Groenendijk and Stokhof 1991) is essentially a compositional version of Discourse Representation Theory (Kamp 1981), and it sees meanings as context change potentials rather than truth conditions. In fact, Hodges (2001) shows that any semantics can be made strongly compositional if the notion of meaning is suitably enriched—a result which we will keep in mind throughout this section.

We will elaborate the above moral later on, but it is now time to take a broader view on the applicability of the principle of compositionality in linguistics, with the goal of evaluating the overall compositionality of the L** approach. The applicability of the principle depends on how exactly one envisions the architecture of the language faculty. We will talk about this architecture in more detail in subsection 12.2.1; the discussion below focuses on a particular issue within the topic, the nature of the interface between the syntax and semantics of natural language.

It is well known that the principle of compositionality is in fact vague in the sense that it specifies neither what meaning is nor which syntax we should use, i.e. what precisely the term 'meaning' on one side, and the terms 'expression' and 'syntactic rule' on the other side of the principle refer to (Hodges 1998). On the syntactic side, the question might be whether to take Surface Structure (SS) or Logical Form (LF) representations as the relevant syntactic expressions. Montagovian approaches such as Direct Compositionality (Jacobson 2014) opt for the former;[8]

[7] The denotation of 'dog' should really be $\lambda x\, D(x)$, but we ignore this and several other otherwise important details for the sake of clarity.

[8] In fact, the position of Direct Compositionality approaches is a bit more involved, in that they assume that syntactic structure building and semantic interpretation proceed in parallel. It remains

the Chomskyan camp, including our own approach, opts for the latter. On the meaning side, the possible choices are delineated by the direct and the indirect interpretation approaches. The direct approaches such as Montague (1970) will see meaning as a model-theoretical construct, so they will apply the principle of compositionality to the transformation of the syntactic structure into this model-theoretical construct. On the other hand, the indirect approaches such as Coppock and Champollion (2021) will introduce an intermediate representation language, the language of logical form if you will, which mediates between the syntax and the model-theoretic semantics of natural language. In the latter approaches, two pieces of the road from syntax to semantics of natural language can be evaluated with respect to compositionality. First, we can investigate whether the mapping from the syntactic structure into the representation language is compositional (this is an evaluation that is common in linguistic literature). Second, we can evaluate the compositionality of the model-theoretic interpretation of the representational language (cf. Partee 2004, p. 156).

You might think that we belong among the indirect interpretation approaches, with L^{**} as the representation language, but as we have already announced, our ultimate goal is to integrate L^{**} into the Minimalist Program, an issue which we will take on in section 12.2. So let us say, for the sake of argument, that we can fully execute our project. Let us say that our idea that the Minimalist LF and L^{**} are just notational variants of each other turns out to be correct. Then LF and L^{**} are one and the same level of representation. There is no transformation from one to the other, and the issue of compositionality (in the sense of mapping to an intermediate interpretation language) does not even arise. Or alternatively, if we take LF and L^{**} to be distinct levels after all, they are isomorphic and the transformation of the former into the latter is trivially compositional in this sense of compositionality.[9] The bottom line, if our ideas are correct, the mapping from LF to L^{**} is as compositional as it gets.

Moving on to the compositionality of the model-theoretic interpretation of L^{**}, we anticipate an objection that the quantifierless L^{**}—or in fact, the quantifierless version of any predicate logic—is not compositional in this sense. The gist of

true, however, that they don't assume that LF feeds the semantic interpretation (or that it even exists, for that matter), therefore firmly belonging to the anti-LF camp.

[9] It might be worth emphasizing that compositionality (of the mapping from the syntactic structure to the logical form) is a different notion than the isomorphism (between the syntactic structure and the logical form). We want to make this clear as isomorphism belongs to the "range of other notions [that] have become attached to the Principle of Compositionality, although there is no way of reading them into the literal statement of it" (Hodges 1998, p. 31); for a clear example of confusing the two notions, see Saeed (2009, p. 326). The question of isomorphism is about the similarity of the syntactic structure and the logical form; compositionality, on the other hand, is about *how* we get from the former to the latter (cf. Hodges 1998, p. 29).

This said, it is clear that even if isomorphism and compositionality are different notions, they are nevertheless related. Isomorphism implies the existence of a compositional transformation; it is the reverse that does not hold.

the objection is that the meaning of a quantifierless formula depends on the syntactic environment it occurs in. Take formula $D(x)$. On its own, it means the same as the standard formula $\exists x\, D(x)$ "something is a dog." And it has the same meaning within the (quantifierless) conjunction $D(x) \wedge B(y)$, which means the same as the standard $\exists x\, D(x) \wedge \exists y\, B(y)$ "something is a dog and something barks." But conjunction $D(x) \wedge B(x)$, where $D(x)$ is conjoined with another formula containing an occurrence of x, means the same as the standard $\exists x\colon D(x) \wedge B(x)$ "a dog barks," paraphrasable as "something is both a dog and barks." In contrast to its stand-alone interpretation, $D(x)$ as a part of the latter conjunction has the same meaning as in the standard interpretation, i.e. $D(x)$ "x is a dog."

To see that we are dealing with a failure of compositionality, let us look at the same example from the top-down perspective. The meaning of the "well-behaved" conjunction $D(x) \wedge B(y)$ "something is a dog and something barks" is computed from the meanings of its two constituents: $D(x)$ "something is a dog" and $B(y)$ "something barks"; in particular, it is a conjunction of these meanings. This is not the case with "ill-behaved" conjunction $D(x) \wedge B(x)$ "a dog barks," whose meaning is not a conjunction of $D(x)$ "something is a dog" and $B(x)$ "something barks," but rather "something is both a dog and barks" or simply "a dog barks."

To see what is going on here, remember that we can imagine the interpretation of quantifierless formulas as a two-step process. In the first step, Restricted Closure transforms a quantifierless formula into a standard formula containing quantifiers, and the second step is the standard interpretation of the latter; for details, see section 9.3. The source of sensitivity to syntactic environment and the corresponding failure of compositionality is the first step. Restricted Closure needs to inspect the entire formula to find the closure position; remember that quantificational closure over variable x takes place at the lowest node containing all occurrences of x.

Let us take the final look at the same example, this time through a prism of synonymy. In a compositional semantics, replacement of a subexpression by a synonymous expression preserves the meaning; for details, see Hodges (1998). However, even though the quantifierless formulas $B(x)$ and $B(y)$ are allegedly synonymous, both meaning "something barks," substituting $B(x)$ for $B(y)$ in $D(x) \wedge B(y)$ "something is a dog and something barks" yields a formula with a different meaning: $D(x) \wedge B(x)$ "a dog barks." Conclusion? In a compositional semantics, the quantifierless $B(x)$ and $B(y)$ cannot be synonymous. But how can we achieve this?

To see how we can overcome the problem described above, we would do well to remember that even standard Tarskian semantics for first-order logic is not compositional. It is not hard to see why. For predicate logic, truth cannot be defined directly. The truth value of $\exists x\, D(x)$ "something is a dog" cannot depend on the truth value of $D(x)$ "x is a dog"; the latter has no truth value, as the reference of x is not fixed. Tarski defined truth through the auxiliary notion of satisfaction; an assignment is a function from variables to individuals, and an assignment may satisfy a formula or not. For example, an assignment g which maps variable x to Fido (and other variables to whatever) satisfies formula $D(x)$ if and only if Fido (the value of g at x) is a dog. To define truth via satisfaction, we then say that a formula is true/false if all/no assignments satisfy it; closed formulas thus have a truth value because they are clearly satisfied either by all or by no assignments.

Combining formulas with sentential connectives is unproblematic for compositionality. An assignment g satisfies $B(x) \wedge D(x)$ "x is both black and a dog" iff it satisfies both $B(x)$ "x is black" and $D(x)$ "x is a dog." Crucially, we only need to check whether $B(x)$ and $D(x)$ are satisfied by g itself. Quantification, however, leads to a problem. Under standard semantics, an assignment g satisfies $\exists x\, D(x)$ "something is a dog" iff there is an assignment g', differing from g at most in its value for x, which satisfies $D(x)$ "x is a dog." Crucially, to compute satisfaction of a quantified formula by assignment g, we need to generally check the satisfaction properties of other assignments as well—and this is what leads to non-compositionality.

However, the interpretation of first-order logic can be made compositional—if (remember the moral) the concept of meaning is properly enriched. The idea is to take the meaning of a formula to be a function from assignments to truth values (rather than a truth value simpliciter). We will not go into any details here—the endpoint of the compositionalization of first-order logic are the so-called cylindrical algebras (Henkin et al. 1971); for an accessible presentation, see Janssen (2011, §10.2.4)—but we note that such an enrichment of meaning consists of including some syntactic information into the notion of meaning. Under this view, meanings are functions from assignments to truth values, but assignments are functions from variables to individuals, and variables are syntactic objects.

In our first example, the compositional mapping from LF to predicate logic, lambda calculus played the same role as cylindrification played above—it made it possible to refer to certain syntactic information from within the meaning. Cylindrification gives us access to some information about variables; lambda calculus gives us access to information about syntactic positions of certain elements. The updated moral is thus that a semantics can be made compositional if we enrich the concept of meaning to include certain syntactic information.

Having reaffirmed and updated the moral by inspecting the compositionalization of standard first-order predicate logic, can we deploy it against the objection that the quantifierless format (of L**) cannot be model-theoretically interpreted in a compositional way? Yes, we can. In fact, it should be clear what we will do. We will take the relevant syntactic information and plug it into the meaning. (We will not provide a full-blown model-theoretic intepretation of the quantifierless L** here. We merely wish to illustrate the gist of the idea behind the compositional interpretation of quantifierless formulas.)

What is the relevant syntactic information? Which formal feature sets apart $D(x)$ and $D(x) \wedge B(y)$ from $D(x) \wedge B(x)$ with respect to the interpretation of subformula $D(x)$ within these formulas? Clearly, it is the information which variables contained in the subformula occur outside the subformula as well; or alternatively, which variables contained in the subformula will get closed above the subformula by Restricted Closure. In our case, x is the only variable of $D(x)$, so the answer is "none" for $D(x)$ as a subformula of $D(x)$ or $D(x) \wedge B(y)$, as RC of x will apply at $D(x)$ itself in both cases. On the other hand, the answer is "x" for $D(x)$ embedded in $D(x) \wedge B(x)$, as RC of x will apply at the root, i.e. above the subformula.

To implement a compositional model-theoretic interpretation of the quantifierless L** (or the quantifierless version of any first-order predicate logic), we would take the above-mentioned cylindrification approach, where meaning is a function from assignments to truth values, but with a twist. To encode the relevant syntactic information in the assignments, we would assume they are partial rather than total functions (from variables to individuals). Partial functions need not be defined on the entire domain (the variables), so the domain of a particular assignment can encode which syntactic environments it will be applicable to. For example, an assignment defined on $\{x_1, \ldots, x_n\}$ will only be applicable when $x_1, \ldots, x_n$ receive quantificational closure above the subformula. To get the truth value of a formula (as a standalone expression), we only need to look at the value of the assignment with the empty domain.

Having outlined the replies to several compositionality-based objections against deploying L** (and predicate logic in general) in the analysis of natural language, let us summarize the insights into compositionality gained along the way. Strong compositionality is the idea that the composition of the meaning parallels the composition of syntactic structure; this is sometimes called the rule-by-rule hypothesis. However, it often turns out that some proposed semantics is not compositional; above, we have mentioned Tarskian interpretation of first-order logic, Discourse Representation Theory and the two-step interpretation of the quantifierless L**. We have presented responses to the former two cases, and outlined a solution for L**. All the replies have a common denominator. They enrich the notion of the meaning. In particular, the meaning is enriched by reference to syntactic information.

In (2), we presented one of the ways in which Hodges (1998) expresses the strong compositionality principle. The central point of this principle is the functionality requirement, encoded by equation $\mu(e) = \varphi(\alpha, \mu(e_1), \ldots, \mu(e_n))$. Moreover, Hodges warns us that compositionality is not the same as "recursion on the syntax," symbolized in (4).

(4) $\mu(e) = \varphi(\alpha, \mu(e_1), \ldots, \mu(e_n), e_1, \ldots, e_n)$

The difference between the functionality component of compositionality and recursion on the syntax is in the dependence of meaning on $e_1, \ldots, e_n$ in the latter. In plain English, recursion on the syntax has full access to the syntactic structure of the constituents; a strongly compositional function has no such access (it can only refer to the type of the parent constituent α).

With this in mind, it is easy to see that while the presented "compositionalizations" of various semantics adhere to the letter of compositionality, they actually violate its spirit. The whole point of compositionality is to compute the meaning without reference to the syntactic structure of constituents, but the often-used compositionalization strategy is a way to do just that—it "syntactifies" the meaning.

To drive the point home, consider the extreme case of syntactification—including all the syntax in the meaning. Given a meaning function μ which satisfies (2a) and (4), but possibly not (2b), let's define a new meaning function μ' such that the new meaning of an expression is composed of the old meaning plus the expression itself:

(5) $\mu'(e) := (\mu(e), e)$

We can then easily show that μ' satisfies (2b). If φ is the recursion on the syntax for μ from (4), then φ' defined as in (6) is the strong compositionality function from (2b) for μ', as proven in (7) (remember that $e = \alpha(e_1, \ldots, e_n)$).

(6) $\phi'(\alpha, \mu'(e_1), \ldots, \mu'(e_n)) := (\phi(\alpha, \mu(e_1), \ldots, \mu(e_n), e_1, \ldots, e_n), \alpha(e_1, \ldots, e_n))$

(7)
$$
\begin{aligned}
\mu'(e) &= (\mu(e), e) \\
&= (\phi(\alpha, \mu(e_1), \ldots, \mu(e_n), e_1, \ldots, e_n), \alpha(e_1, \ldots, e_n)) \\
&= \phi'(\alpha, (\mu(e_1), e_1), \ldots, (\mu(e_n), e_n)) \\
&= \phi'(\alpha, \mu'(e_1), \ldots \mu'(e_n))
\end{aligned}
$$

What this shows is that as soon as we allow syntactification of meaning—which is, indeed, the standard practice—the principle of compositionality becomes indistinguishable from recursion on the syntax. Or to put the point rhetorically: Compositionality, as it appears in linguistic theorizing, is just recursiveness in disguise. But if this is the case, why was recursiveness not sufficient to begin with?[10] And of course there was never any suggestion that predicate logics could not be recursive.[11]

[10] In fact, recursiveness on its own is truly not sufficient, but for a reason independent of compositionality. As pointed out by Pelletier (2004), a recursive procedure must also be *grounded*, in the sense that a recursive definition must "bottom out" in independently defined base clauses. For example, the recursive (more precisely, inductive) definition of a factorial, $(n+1)! := (n+1)n!$, is grounded by the base clause $1! := 1$. The standard recursive interpretation of first-order logic is of course grounded by clauses determining the interpretation of atomic formulas.

[11] For further discussion on the principle of compositionality, we recommend Janssen (1983); Larson and Segal (1995); Hodges (1998); Szabó (2000); Hodges (2001); Fodor and Lepore (2002); Partee (2004); Pelletier (2004); Dever (2008); and Janssen (2011).

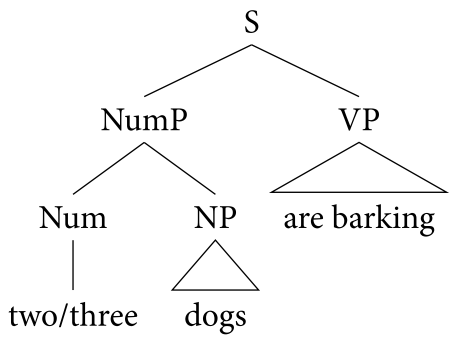
12.1.3 The isomorphism worry

The third objection against deploying a predicate logic in an account of language is that it leads to a mismatch between the syntactic and the logical form of a sentence. Here we start to get into objections about the plausibility of L** as a candidate for the syntax of natural language. In the next section we will get into the fine details of how our proposal dovetails with the machinery of contemporary linguistic theory, but before we get into those details we can make a lot of progress by viewing the question at a relatively abstract level.

Viewing the matter at this abstract level, we will see that much of the syntactic complexity we see in predicate logic will be swept away once we start taking restrictedness seriously, for as we saw in the previous few chapters, if we assume restrictedness as a global picture it allows us to dispense with explicit quantifiers and Boolean connectives, leaving us with structures that encode binary branching trees and not much else. To put matters in more sweeping terms, the hard work we did in deploying the relation of p-scope (and polarity marking) allowed us to cash out conservativity via the syntactic notion of restrictedness, and restrictedness is now going to be the secret ingredient that allows us to prepare our version of predicate logic (as developed in L**) for its integration with linguistic theory.

First, however, it will be useful for us to articulate the alleged problem, and show precisely how it is resolved when we begin paying attention to restrictedness. An important example of the alleged mismatch problem for predicate logic is exhibited by sentences containing numerals.[12] In predicate calculus, these receive logical forms bearing little resemblance to their syntactic counterparts. Consider the following examples.[13]

(8) Two/three dogs are barking.



(9) Two dogs are barking.

$$\exists x_1 \exists x_2 : D(x_1) \wedge D(x_2) \wedge x_1 \neq x_2 \wedge B(x_1) \wedge B(x_2)$$

[12] For instances of this objection, see Cann (cf. 1993, pp. 187–188); Kroeger (2019, p. 257); and Coppock and Champollion (2021, p. 237).

[13] For clarity, we omit the determiner phrase (DP) layer above the NumP in (8).

(10) Three dogs are barking. $\exists x_1 \exists x_2 \exists x_3 : D(x_1) \wedge D(x_2) \wedge D(x_3) \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \wedge B(x_1) \wedge B(x_2) \wedge B(x_3)$

In (singular) predicate logic, a variable (x_i) can only stand for a single individual, so we need to introduce several variables (and thus quantifiers) and predicate the noun and the verb phrase of each of them; but in natural language syntax, we find a single subject–predicate structure. Furthermore, to keep the reference of the introduced variables distinct, predicate logic renderings of numerals contain a number of occurrences of the (in)equality predicate, of which there is no trace in the natural language sentences. Perhaps most importantly, while in natural language syntax higher numerals yield the same syntactic structure as lower numerals,[14] in predicate logic the complexity of the logical form grows as we increase the numerals, linearly for the predication part and even quadratically for the distinction part.

The first problem with this objection is that it only applies to *singular* predicate logics. Plural logics such as our L^* and L^{**} are immune to this mismatch objection. In fact, formulas such as (9) and (10) are familiar from Chapter 4; they exhibit precisely the complexity swept away by L^*'s subscripted quantifiers, shown in (11a). In L^{**}, we went a step further by divorcing cardinality from quantification and treating numerals as predicates. As illustrated in (11b), the contribution of a numeral to the logical form in L^{**} is compact and localized—$\exists n : 3(n) \wedge \#(n, x)$, which we have abbreviated to $\underline{3}(x)$—and the logical form reflex of the numeral connects with the objectual variable x at a single point ($\#(n, x)$). In a nutshell, the representation of a numeric nominal modifier in L^{**} matches the natural language LF cleanly, particularly if we understand that each branching node tacitly encodes a logical junction (in this case a conjunction) and that the existential quantifier is eliminable—these are the points we turn to next.

(11) Three dogs are barking.
a. $\exists_{\geq 3} x : D(x) \wedge B(x)$ (L^*)
b. $\exists x : (\underline{3}(x) \wedge D(x)) \wedge B(x)$ (L^{**})

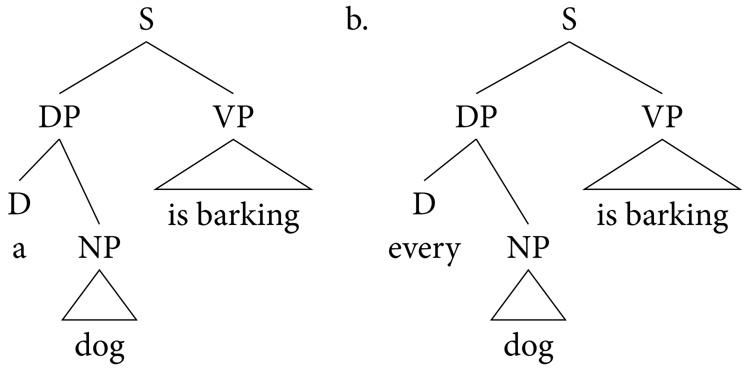
The other alleged mismatch problem has to do with the overall form of the basic quantifier constructions 'some' and 'every', which receive the same logical form in both singular predicate logic and L^{**}. Partee (2005, pp. 2–3) proposes the trees in (13) as the predicate logical representations of natural language structures in (12),[15] and then objects: "Predicate logic helps us express the truth-conditional

[14] The higher numerals are (morpho)syntactically more complex than the lower numerals (see e.g. Ionin and Matushansky 2006), but this is orthogonal to the issue, as the higher complexity pertains to their internal structure, whereas we are discussing the interaction of a numeral with the noun phrase.

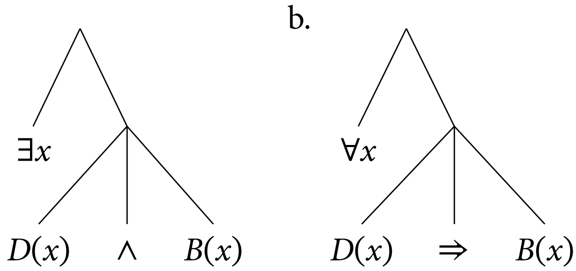
[15] We adapt Partee's (2005)'s trees in inconsequential details.

content of these English sentences, but it does not capture the structure of such English sentences."[16] Let us compare (12) and (13) in detail to see what the mismatches are.

(12) a.



(13) a.



Indeed, these pairs of trees look nothing alike. For one, the natural language syntactic trees are consistently binary branching, while the predicate logic forms contain ternary branches for sentential connectives—and we would get further ternary branches if we followed D'Alfonso (2011, p. 89), who splits a quantifier node ($\exists x/\forall x$) into a quantifier node ($\exists/\forall$) and a variable node (x). Next, the predicate logic forms contain a conjunction and disjunction, but there is no trace of 'and' and (say) 'if ... then' in the natural language representations. But worst of all, the effect of determiners 'a' and 'every' is "smeared" across the entire logical form. For one, the type of the determiner is reflected both in the type of the quantifier ($\exists$ as opposed to $\forall$) and in the type of the binary connective ($\wedge$ vs. $\Rightarrow$), but even worse, if the logical form looked anything like the natural language syntactic tree, we would find these reflexes close to $D(x)$—or more precisely, they would need to

[16] Other sources containing similar statements include Chierchia and McConnell-Ginet (1993, p. 113); Heim and Kratzer (1998, p. 191); and Saeed (2009, p. 326). Furthermore, Partee (2005) provides one other instance of non-isomorphism. As shown below, the logical form of a sentence with a quantified subject radically differs from the logical form of a sentence with a proper name subject, whereas their syntactic structures are essentially the same. We will disregard this instance of the objection, as it rests on the assumption that proper names must be translated to individual constants—see e.g. Pietroski (2003, pp. 235–236, 247–253) and references therein for an argument against such an assumption.

(i) Every dog barks. $\forall x : D(x) \Rightarrow B(x)$

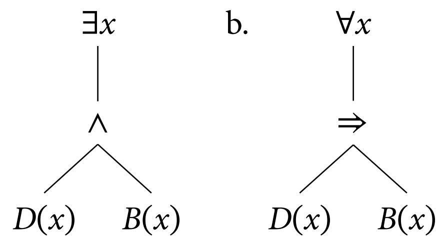
(ii) John sleeps. $S(j)$

be closer to $D(x)$ than to $B(x)$, as 'every' is closer to 'dog' than to 'is barking'—which is not the case in (13). We find the connective deep inside the formula, between $D(x)$ and $B(x)$, and we find the quantifier at the top of the logical form—not only do the connective and the quantifier occur in a "wrong" place, the reflex of the determiner is not even localized in the logical form.

The problem with the objection is that this comparison is highly sensitive to the particular encoding of predicate logic formulas. Choosing the format shown in (13) is not very generous toward the idea that the syntactic and logical form might match, and indeed it certainly isn't a fair way to represent the version of predicate logic we have been developing in this book. In fact, there are two problems. First, we will argue that trees such as (13) do not faithfully correspond to the structure of even standard predicate logic formulas. In a nutshell, the trees present quantifiers and connectives as categorematic items, while the standard treatment is syncategorematic. We will address this problem by reverting to the standard, syncategorematic encoding of predicate logic operators. Second, the standard encoding of quantification does not make use of the observation that all quantification in natural language is conservative, and therefore both overgenerates (by yielding non-conservative formulas) and produces logical forms which are non-isomorphic to syntactic structures of natural language expressions. We will address this problem by rehearsing the development of the quantifierless format of predicate logic from section 9.3.

One thing to note is that the predicate logic trees in (13) are quite different from the trees we have been using in this book, even before we begin the project of streamlining the representations of explicit quantifiers and logical connectives. We would represent the formulas signified by those trees as in (14)—if we were using the material implication $\Rightarrow$, which we are not, but let us assume for a moment that we did. What precisely is the difference between (13) and (14) and why have we opted for the latter representation?

(14) a. b.



The treatment of quantifiers and sentential connectives in (13) is *categorematic*. Let us illustrate that on $\exists x\colon D(x) \wedge B(x)$, depicted in (13a). The idea put forth in this tree is that this formula is composed of two (sub)formulas, $\exists x$ and $D(x) \wedge B(x)$. So we first compose both of these formulas (by whatever rules necessary), and then join them into (13a). However, this is not how composition of predicate logic formulas is usually assumed to proceed. Under the standard formation rules, $\exists x$ is never a formula on its own. The standard treatment of quantification (and of connectives as well) is *syncategorematic*. We do first compose $D(x) \wedge B(x)$, but this is also the only formula that needs to be pre-composed to form $\exists x : D(x) \wedge B(x)$. The latter formula is formed from the former by simply prefixing $\exists x$. (In general, the prefix of course consists of either $\exists$ or $\forall$, and of any variable.) It is this procedure that is reflected by (14), a point which we can see even more clearly if we remember that the (14) format is really just a convenient abbreviation of the (15) format.

(15) a. $\exists x\,(D(x) \wedge B(x))$ b. $\forall x\,(D(x) \Rightarrow B(x))$

$$D(x) \wedge B(x) \qquad\qquad D(x) \Rightarrow B(x)$$

$$D(x) \quad B(x) \qquad\qquad D(x) \quad B(x)$$

The categorematic vs. syncategorematic treatment of quantifiers leads to yet another difference, a difference in how we determine the scope of a quantifier. In the usual, syncategorematic approach, scope is determined by domination. The scope of a (unary) quantifier consists of the entire operand, i.e. a quantifier $\exists x$ or $\forall x$ binds all occurences of x which are dominated by the quantifier node (and not bound by a closer quantifier). In the categorematic approach, the situation is more complex. $\exists x$ or $\forall x$ scopes over its sister, i.e. it binds all occurrences of x (not bound by a closer quantifier) which are dominated by the sister of the quantifier node (or occur in the sister itself, if the sister is a simple atomic node; the relevant dominance relation here is thus non-proper dominance). Even though the determination of scope is more complicated in the categorematic treatment, it is most probably what motivates the syntactically minded authors to choose this option. Why? Because the relation that determines scope in the categorematic approach is actually c-command,[17] the mother of all relations in natural language syntax.

The rationale behind having $\exists x$ as a sister to $D(x) \wedge B(x)$ (and $\forall x$ as a sister to $D(x) \Rightarrow B(x)$) is thus to have scope determined as in natural language syntax. However, forcing predicate logic into the categorematic format has the unavoidable consequence of spoiling the correspondence between natural language syntax and logical form. In the categorematic approach, $\exists x$ is a category, by definition, but it is a very unusual category from the natural language perspective, as it can only be composed of a quantifier symbol and a variable. Conversely, the quantificational phrases in natural language can be as heavy as we wish, e.g. 'a big black dog which doesn't like to chase cats'. This long nominal expression is a real syntactic phrase, whereas $\exists x$ is nothing but a fake.

[17] Remember that a constituent c-commands its sister and all the descendants of the sister.

We find it unfair to first force predicate logic into an unnatural format and then complain about the lack of isomorphism to natural language syntax. And we see no way out of the predicament caused by the categorematic treatment. So let us rather see if the syncategorematic approach—even if it appears to be at odds with natural language syntax at first sight—might lead to a better correspondence. We will see that it ultimately does.

What sort of mismatches do we get between the syntactic structures in (12) and the syncategorematically encoded logical forms in (14)? The situation is better than in the categorematic case of (13), but several issues remain. First, the logical forms contain unary branching nodes (the quantifiers). Second, $\wedge$ and $\Rightarrow$ have no overt corresponding natural language connectives. Third, the quantifier and the connective node are not in the same position as the determiner they reflect—however, note that unlike in (13), they are adjacent to each other. In other words, the logical form reflex of a determiner is localized, and this is what makes it possible to make the next step in the process of producing logical forms isomorphic to syntactic structures.

The idea here is to "compress" the adjacent pairs of a quantifier and the immediately dominated connective into a single node, which transforms (14) into (16). This move eliminates the first two of the mismatches listed above. First, the quantifiers in the new tree are binary. The new logical forms therefore contain no unary nodes (we will address negation below). Indeed, note that the new logical forms neatly correspond to the division of the syntactic S node into DP and VP. Second, the problem of $\wedge$ and $\Rightarrow$ corresponding to no overt natural language connectives disappears, as $\wedge$ and $\Rightarrow$ themselves do not appear in the new logical form.

(16) a. $[_{\exists x}\ D(x)\ \ B(x)]$ b. $[_{\forall x}\ D(x)\ \ B(x)]$

Of course, the question is how can we justify the compression of a unary quantifier and the immediately dominated connective into a binary quantifier, but we already know the answer, and it relies on the well-known notion of conservativity and its formal counterpart, restrictedness. Let us refresh our memory. The Conservativity Universal proposed by Barwise and Cooper (1981), stating that determiners are interpreted as conservative functions, has firmly stood the test of time. In Chapter 9, we armed ourselves by generalizing the notion of conservativity and showing that restrictedness is a formal characterization of generalized conservativity. In section 10.1, we proceeded to first argue that not only determiners, but all natural language expressions are conservative (Generalized Conservativity Universal), and ended up proposing our strongest hypothesis, Restrictedness Universal, which states that all logical forms are restricted. The binary quantifiers we introduced above are thus the familiar restricted quantifiers (modulo some restrictions imposed by the theory from Chapter 9).

It is worth making clear the exact status of restricted quantifiers in our system. First note that restricted quantifiers are usually introduced as abbreviations; for example, a semantics paper might introduce them as in (17). However, the idea behind (16) is for restricted quantifiers to be bona fide elements of the formal language. That is, if we were to stop the development of our project at this point, we would abolish the formation rules for the (unary) unrestricted quantifiers and introduce the formation rules for (binary) restricted quantifiers, and set up the semantics to directly interpret the restricted quantifiers (but of course, in accord with the indirect interpretation via abbreviations).

(17) a. $\exists x\,[\varphi]\,\psi \;:\equiv \exists x\colon \varphi \wedge \psi$
b. $\forall x\,[\varphi]\,\psi \;:\equiv \forall x\colon \neg\varphi \vee \psi$

Second, note that we see the binarity of restricted quantifiers somewhat differently than most, which is also reflected in how we (would have) defined the above abbreviations. In our view, a restricted quantifier takes two operands, the restriction and the nuclear scope, to directly produce the quantificational structure $\exists x\,[\varphi]\,\psi$ or $\forall x\,[\varphi]\,\psi$; in the abbreviations, this is reflected in having the quantifier symbol outside the square brackets. In contrast, the abbreviations (18) commonly found in the semantic literature have the quantifier symbol inside the square brackets (cf. e.g. Chierchia and McConnell-Ginet 1993, p. 113; Saeed 2009, p. 327). Once again, the rationale here is to have a closer correspondence to syntactic structures, in the sense that the quantifier symbol and the restriction are close to each other, just as the determiner and the noun phrase are.

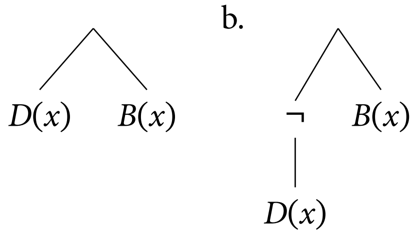
(18) a. $[\exists x\colon\varphi]\,\psi \;:\equiv \exists x\colon \varphi \wedge \psi$
b. $[\forall x\colon\varphi]\,\psi \;:\equiv \forall x\colon \neg\varphi \vee \psi$

If we consider restricted quantifiers to be mere abbreviations, this notational difference is irrelevant. But as soon as we want restricted quantifiers to be more than just that (presumably to eliminate the unrestricted quantifiers), the notation from (18) leads to complications in the definition of the formal language. The notation implies a mixture of syncategorematic and categorematic view of quantifiers (which in turn leads to two conflicting ways of scope taking, as discussed above). The quantifier symbols are syncategorematic; they combine with (a variable and) a formula (the restriction) into a restricted quantifier. A restricted quantifier must not be given the status of a formula—otherwise, we could merge two of them to produce unintelligible formulas such as $[\exists x\colon\varphi]\,[\forall y\colon\psi]$—it must rather form an independent (and non-interpretive) category, which, when combined with a formula (the nuclear scope), yields the familiar (and interpretive) formula such as $[\exists x : \varphi]\,\psi$.

This setup makes sense in enterprises leading to some form of generalized quantification—once we extend the formal language with additional quantifiers, we can have expressions such as $[\text{most}\, x : \varphi]\,\psi$, with the meaning of quantifier "most" defined in the meta-language, most often set-theoretically—but it is not fruitful in our approach, which aims to capture the meaning of complex quantifiers syntactically. As foreshadowed by the discussion in Chapter 9, our idea is not to stick the quantifiers close to their restrictions, but to eliminate them altogether.

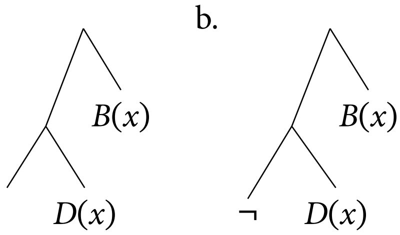
The logical forms in (16) are what one might expect of an author sympathetic to the idea of employing predicate logic but without the results presented in Chapter 9. One thing we learned in that chapter is that the restricted format does us no good unless supplemented by limitations on the polarity of the restriction. For example, despite its appearance, $\exists x\,[\neg D(x)]\,B(x)$ is not an instance of restricted quantification—the existential quantifier and a negative polarity restrictor ($D(x)$) simply do not mix. However, we have also learned how we can turn this seeming problem to our advantage. As long as we live in a conservative/restricted world, quantification is possible without quantifier symbols, and the feat is made possible by paying attention to polarity. Even more, we have seen that in the quantifierless format, we can do with a single binary connective—we have called it the generalized junction—which "resolves" to either a conjunction or a disjunction, again depending mainly on polarity. The net effect of the quantifierless format is that we can rewrite (16) into (19) (of course, to make the transformation into the quantifierless format possible, we need to bring negation into play, i.e. rewrite $\Rightarrow$ using $\neg$ and $\vee$).

(19) a. b.

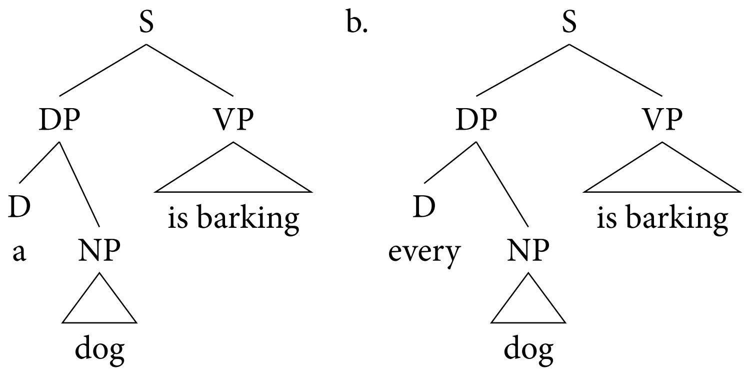


How precisely polarity determines the locus and the type of restricted quantification, and what other factors play a role, was shown in detail in section 9.3. As a quick reminder, Restricted Closure requires that we quantify over x at the root, because that is the lowest node dominating all occurrences of this variable. The closure will be existential in (19a) because the restrictor $D(x)$ has positive polarity within the closure node, and the closure will be universal in (19b) because the restrictor $D(x)$ has negative polarity within the closure node.

As the final adjustment to our encoding, let's ignore the syncategorematic status of negation in predicate logic for the moment. Assume that a negation gets its own terminal node and takes scope via c-command, and let us furthermore allow empty nodes, assuming that an empty node is semantically inert (one could translate it either as the identity operator or as the constant true, to be conjoined with the sister node). With these assumptions, we can rewrite (19) into (20).

(20) a.


These trees are clearly isomorphic to the syntactic trees in (12), which we repeat here for convenience as (21). The overall shape of the trees is the same, including the distribution of lexical material, i.e. we find the reflexes of 'dog' ($D(x)$) and 'is barking' ($B(x)$) in positions corresponding to the positions of the NP and the VP. The logical forms contain no elements not found in the syntactic representations; in particular, we find no quantifier symbols or stray logical connectives. The only unfamiliar aspect of the logical form trees in (20) is the reflex of determiners: 'every' corresponds to a negation operator and 'some' to a semantically inert empty node. But the point is that on this view, the determiners *do* have corresponding nodes in the logical form, and they have them where expected, as sisters to $D(x)$, the reflex of the noun phrase 'dog'.

(21) a.


As we will see, our minimal trees in (20) are tailor made for the integration with contemporary linguistic theory, and in particular with the Minimalist Program. For as we will see, one idea that has extruded from our analysis of quantification is that determiners are not quantificational operators at all. They certainly signal the presence of such operators, but following a suggestion in Ludlow (2002), it is more accurate to think of them as bundles of features that control structure building, while their quantificational force arises only through the application of Restricted Closure.

We are almost home. We have shown how the analysis of polarity in natural language inference can lead us to a better understanding of conservativity and restrictedness, and how restrictedness can help us to very minimal predicate logic structures. This in turn has prepared our predicate logic for integration into the syntax of natural language. It is time to complete this integration, and we do so by taking a dive into the Minimalist program. We want to see how logical forms will be understood within the Minimalist Program, how the theory itself works, how our L** dovetails with the theory, and finally we will look at the linguistic payoff from the integration.

12.2 L** as LF

We hope that in the previous section we were successful in persuading you that the usual worries about predicate logic as an account of natural language can be answered. But once we can be reasonably sure that natural language has a structure like that of our predicate logic, we encounter the final problem: How can we integrate this idea into actual linguistic theory? The principal aim of this section is to further explore the isomorphism argument down to the level of linguistic details. To do so, we have to look at things from the standpoint of natural language syntax. We want to see whether syntax actually produces the structures that are predicted by our system, and investigate how it produces them.

As we promised in the beginning of this chapter, we want to do more than shoe-horn in our analysis in a brute strength way; we want to show how the L** analysis dovetails with contemporary syntax in a very organic way. But before we get to the discussion of integration with syntactic theory, we need to first address a very important technical question. What exactly are we saying when we say that our theory is isomorphic to the syntax of natural language? You have probably already picked up on the idea that we plan to talk about a level of linguistic representation called LF, but what is that exactly, what does it have to do with natural language, and why do we want to understand our theory in terms of LF representations?

12.2.1 LF and L** in linguistic theory

The syntactic proposals we are making in this book should have application within any framework that admits of a level of representation LF, which can be understood as the level of representation that is visible to the semantic theory. Fundamentally, our claim does not even rest on there being a level of representation LF, so much as on an understanding that the structure of natural language is not something that is visible to the naked eye (ear), and that any structure that we posit will of necessity be abstract. What matters is that these abstract structures be describable using our formal mathematical tools, and that they be understood as the interface to semantic theory, or scientific theory of meaning.

In the early days of generative syntax (Chomsky 1965), two levels of syntactic representation were proposed: *Deep Structure* (DS), containing expressions generated by some context-free grammar; and *Surface Structure* (SS), generated from DS by rewrite rules, aka transformations. One idea put forth by Katz and Fodor (1963) and Katz and Postal (1964)—and briefly adopted by Chomsky—was that the meaning of a sentence is determined by DS (transformations were assumed to preserve meaning).

But this early generative view of Deep Structure as the syntax–semantics interface was soon turned on its head. Rather than assuming that syntax and semantics interface *prior* to transformations, most generative syntacticians nowadays believe that this happens *after* the transformations take place (and since the sixties, the transformations themselves have transformed a lot, as well). This contemporary, post-transformational interface level is called Logical Form (LF). It originates from Chomsky (1973) and is developed in detail in May (1977, 1985), who proposed that quantified noun phrases undergo *Quantifier Raising (QR)* from their surface positions to their LF positions and that it is the latter positions which determine their scope.

What May did was shift the border between syntax and semantics a bit toward the syntactic side. May showed that QR is subject to the same restrictions (like subjacency and coordinate structure constraint) as other types of syntactic movement, and Hornstein (1995) argued that it can even be reduced to independently motivated syntactic operations. The point is that these efforts did not merely posit a level of linguistic representation that encoded scope relations, they showed that such a level of representation also had linguistic payoff—it helped illuminate the syntactic theory.

Such considerations have lead to the Minimalist view on the architecture of the language faculty, schematized in Figure 12.1a. There is a uniform syntactic derivation from the selected subset of lexical items (called *numeration*) to LF, and the semantic interpretation depends exclusively upon LF. According to Chomsky (1995), LF is where syntax interfaces with *the conceptual–intentional cognitive system (C-I)*. At some point in the syntactic derivation, the so-called *Spell-Out* occurs, shipping off the syntactic representation for phonological interpretation; *Phonological Form (PF)* is where syntax interfaces with *the articulatory–perceptual cognitive system (A-P)*.[18] Because the shipout to phonology occurs before the LF is

[18] The numeration and the Spell-Out point roughly correspond to the notions of Deep Structure and Surface Structure, respectively, but note that Spell-Out is usually assumed to be cyclic, so that strictly speaking, syntax does not interface with A-P at a single point. Principles and Parameters framework, the predecessor of the Minimalist Program, is the last incarnation of generative syntax which still recognizes DS and SS as independent representational levels.

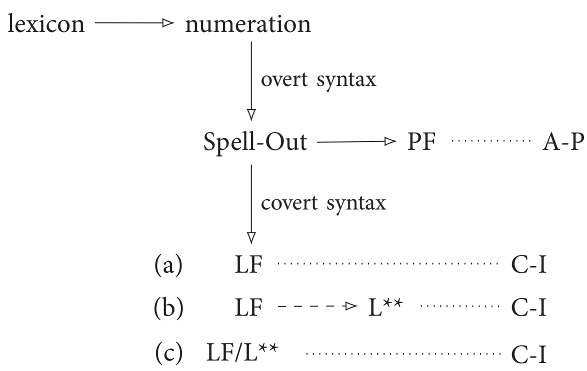


Figure 12.1 The Minimalist architecture of the language faculty: (a) the standard generative view, (b) with L** as a separate level of representation, and (c) with LF = L**

finalized, only the initial part of syntactic operations is visible, or *overt*; anything that happens after Spell-Out is invisible, or *covert*.[19]

For our purposes, the crucial aspects of the Minimalist architecture of the language faculty are as follows. First, it is LF which is the point of interface between syntax and semantics.[20] Second, LF is built by syntactic operations but at the same time disambiguates some aspects of semantics, notably by using scope.

Let us now bring in the semantic theory to fill in a detail in the relation of LF to C-I.[21] At their core, most mainstream contemporary semantic theories are theories of truth, so their primary goal is to provide an algorithm for computing the truth conditions. However, the truth conditions are most often expressed indirectly, through an intermediary formal language, which Coppock and Champollion (2021) call the *representation language*.[22] If we were to integrate L** into such a indirect intepretation approach, L** would stand between LF and C-I, as shown in Figure 12.1b.[23] In such an organization of our project, our language L** would be

[19] The commonly adopted Principle of Full Interpretation states that every feature of lexical input must be interpreted. It follows that all (phonologically non-empty) lexical items from the numeration must be integrated into the structure before the Spell-Out. Therefore, the bulk of the covert syntax consists of movements.

[20] The view of syntax as an LF-building device is valid not only for the standard minimalism, which we will present in subsection 12.2.2, but also for the various offshoots of this framework. For example, in Distributed Morphology (Halle and Marantz 1993) and Nanosyntax (Starke 2009), the lexicon is integrated into the architecture in a fundamentally different way (lexical access occurs *after* the syntactic derivation; partially so in DM and fully so in NS), but the role of LF as the syntax–semantics interface remains unperturbed. Consequently, our proposal is agnostic with respect to the choice among the various incarnations of the Minimalist Program.

[21] Not all semantic theories adopt the generative idea that semantic interpretation feeds off of LF. The Direct Compositionality approaches (see e.g. Jacobson 2014) subscribe to the alternative view—inherited from Montague (1970, 1973)—that semantic composition reflects the surface syntax.

[22] The canonical reference for the *direct* interpretation semantics—semantics which does *not* deploy a representation language—is Montague (1970). A notable contemporary direct interpretation approach can be found in Larson and Segal (1995).

[23] A generative interpretation of Figure 12.1b is that the level of representation employing L** (or whichever representational language is employed in its stead) is a cognitively real level of

a logical (and truth-conditional) representation level distinct from LF, and there would be rules that mapped from LF to L** and *then* the L** would be handed off to the semantics.

However, the LF–to–L** rules in such an approach would be quite different than the rules found with other representational languages, like the language of generalized quantifiers. With other languages, the rules are designed to manipulate the semantic content of lexical items, typically employing typed lambda calculus to convert LF structures into expressions of the representation language. In the case of L**, such a complication is unwarranted. We have argued that logical forms expressed in L** are isomorphic to LF structures, so we are opting for the more constrained working hypothesis on which LF—the level of linguistic representation—itself directly encodes L** representations. In this way, we circle back to the original generative architecture of the language faculty, but with the twist that LF structures and L** formulas are understood as one and the same thing. This view is illustrated in Figure 12.1c.

This approach certainly minimizes the theoretical slack available to us. We know that the level of representation LF is tightly constrained—it has to be compatible with the rest of the theory of grammar. And obviously the structure of L** is also severely constrained. But the constraints placed on both are what makes the constrained hypothesis more interesting. If we can take LF, and its current constraints, and show that it can be identical to L**, given its constraints, then we have achieved a very interesting and perhaps surprising result. If we can likewise get some empirical benefits from unifying these domains, that will be even better. Of course, the strict unification project might not be executable, in which case we need to think of L** as a downstream level of representation, yielding a model of grammar like that shown in Figure 12.1b. But until we are forced to that move we will assume the identity of LF and L** as our working hypothess.[24]

12.2.2 A crash course on Minimalist syntax and the cartography of syntactic structures

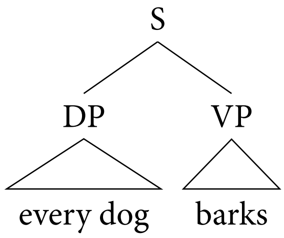
This is primarily a philosophy (of logic) book, and we can't go into *too* much linguistic detail here, but it is an important part of our project to show that L**

representation. There are semanticists who object to such an interpretation and the general view that at the end of the day, semantics is a part of psychology or cognitive science. The cognitive view is certainly not shared by Montague, who regarded "the syntax, semantics, and pragmatics of natural languages [as] branches of mathematics, not of psychology" (Thomason 1974, p. 2). In fact, ever since Lewis (1975), much of the mainstream formal semantics seems to be alienated from cognitive issues (Partee 2016, pp. 3–4). We certainly cannot identify with such a stance, and rather agree with Pietroski (2018, p. 3) who "think[s] this abstraction from psychology has outlived its utility."

[24] Just to be clear, one thing we are not advocating is the idea, advanced in Postal (1972), that collapsing levels of representation would be better because it would involve fewer moving parts. Our point is not that our approach is more simple or more elegant. It is simply that collapsing the levels of representation forces tighter constraints on the project.

offers a genuine story about the structure of natural language. Up until now, we have been fairly non-committal about many details of syntactic structures. We have told you that all the structures are binary branching, and we have (following Abney 1987) consistently analyzed sequences of a determiner and a noun phrase as determiner phrases rather than noun phrases—but we have certainly been presenting syntactic structures as much simpler than currently envisioned within the Minimalist framework. For example, we have been pretending that a sentence (S) is a subject–predicate structure consisting of a DP subject and a VP predicate, as in (22) below. In doing so, we have violated several major assumptions of the Minimalist syntax. But to get clear on what kinds of structures are possible within the Minimalist framework, we first need to get clear on how the Minimalist Program works. We thus offer the following crash course, with the basics that we will need to understand how L^{**} might dovetail with the Minimalist Program.

(22)

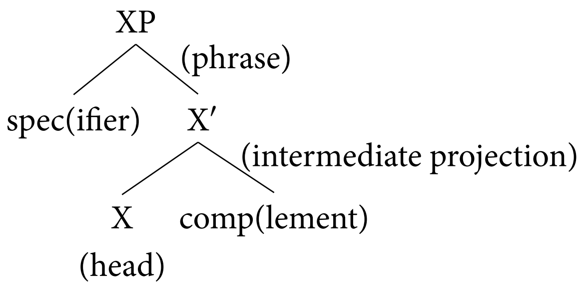


In fact, the crash course will cover not only the Minimalist Program (MP), but also a distinct but closely related research project, the so-called cartography of syntactic structures (CSS). The research topics of the two projects are to some extent independent—while MP is interested in what drives the construction of syntactic structures, CSS explores the *content* of the resulting structures—but as the projects have commonly informed each other, it makes sense to present them in parallel.

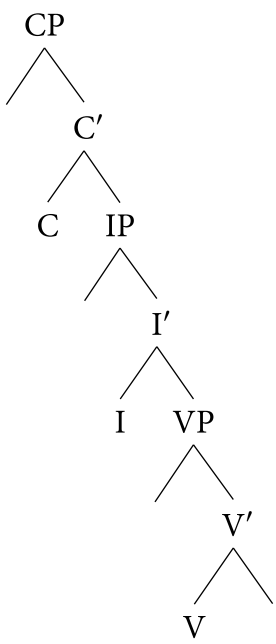
At first sight, the essential insights gained by MP and CSS might seem contradictory. While MP successfully builds upon its basic tenet that a certain simplicity can be found at the core of the language—more precisely, that the language faculty "is an optimal solution to the interface conditions imposed by the systems of the mind–brain in which it is embedded" (Chomsky 2002, p. 90)—CSS is in the business of uncovering the complexity of sentential structures. However, the relation between MP and CSS is one of balance rather than contradiction. The simplicity of the Minimalist syntactic toolbox, exhibited by both the low number of primitive syntactic operations and their highly abstract nature, is balanced by the richness of the syntactic structure that is being discovered by the extensively cross-linguistic cartographic research.

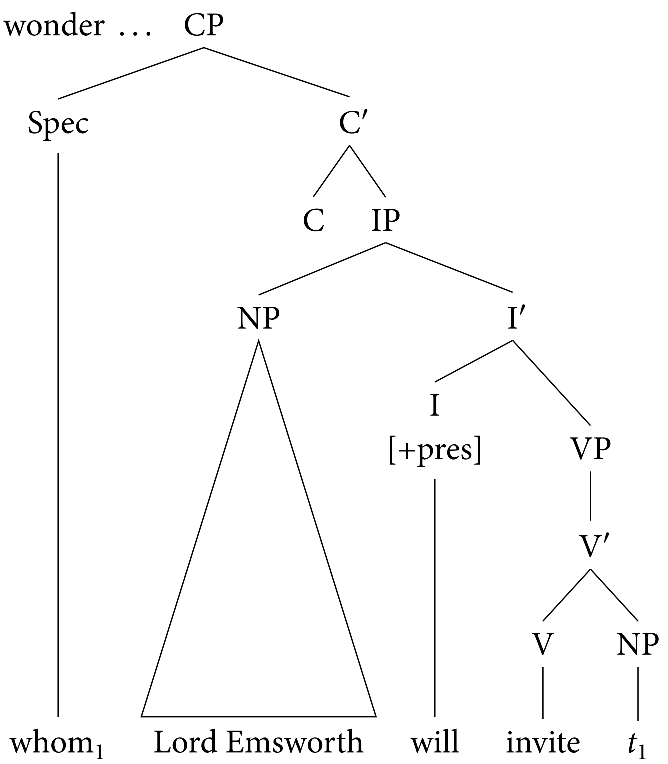
The naive representation in (22) can be seen as a modern paraphrase of the original generative analysis of clause formation by rewrite rule S → NP + VP from Chomsky (1957). In other words, (22) belongs to the fifties. The eighties brought a major reconceptualization of syntactic structures. Chomsky (1986) developed two related ideas. First, the structure of phrases is uniform across the categories;

all phrases conform to the endocentric X′ ("X-bar") format illustrated in (23). Second, the clause gains a "data structure." As illustrated in (24), the verb phrase (VP) is topped by two *functional projections*: an inflectional phrase (IP) and a complementizer phrase (CP). Various constituents are then either *base-generated* in these projections or *move* there. (The ability to move depends on language-specific parameters, but as it is constrained by c-command, movement is always up the tree.) For example, in (25), borrowed from Haegeman (1994, p. 381), the auxiliary verb 'will' is base-generated in the head of IP, and the question word 'whom' moves from its base position in the complement of V into the position of the specifier of CP (movement is signalled by coindexation of 'whom' and *t*).

(23)


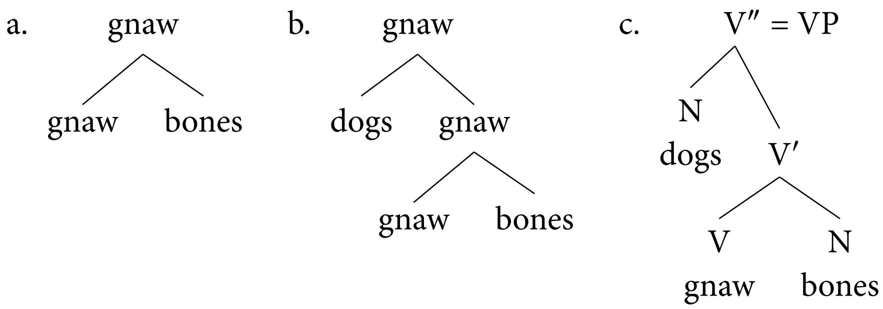
(X = V/I/C/N/D/A/ . . .)

(24)


(25) I wonder . . .


The Minimalist Program (Chomsky 1995) brought further reduction of the theoretical toolbox. In particular, the X′ structures are derived by repeated application of the basic syntactic operations, *Merge* and *Move*,[25] which govern the combination of smaller grammatical elements into larger ones.[26] Let us focus on Merge. Merge is a *binary* operation, meaning that it is always two elements that are merged into the larger element, and it is *endocentric*, which means that the category of the resulting element is inherited from one of the consituents, which is called the *head*; the non-head constituent is called the *complement*. For example, when the verb 'gnaw' and the noun 'bones' are merged, it is the verb which heads the resulting construction, so the structure of the resulting verb phrase 'gnaw bones' is (26a). We often say that the head *projects*, and in fact, a head may project more than once. The constituents other than the complement merged to a head are called *specifiers*.[27] In (26b), 'dogs' is a specifier of 'gnaw'.

(26)



While the Minimalist syntactic structures officially conform to the *bare phrase structure* shown in (26a) and (26b) (and we will see that 'dogs', 'gnaw', and 'bones' actually stand for bundles of features), they are more commonly depicted in the eighties style, in trees labeling nodes with category features (V for verb and N for noun) and indicating projection levels (V′ and V″ are the first and the second projection of V; as V″ does not project further, it is also called the maximal projection, VP). We will follow this convention.

But we have another development in parallel to the Minimalist reduction of the set of basic operations. This precursor to cartographic studies starts with Pollock (1989), who argued that IP should be split into two categories, which became later known as the tense phrase (TP) and AgrS, a category dedicated to subject–verb agreement. In a similar vein, Larson (1988) proposed splitting the VP, resulting (among other things) in a well-known distinction between the functional "little"

[25] Some frameworks admit a third basic operation, *Adjoin*.

[26] These days it is not uncommon to assume that even Move is nothing but a specific manifestation of Merge, sometimes called internal Merge, arising when an object, already a constituent of some larger object, is remerged with the root of its container (see e.g. Hornstein 2001; Starke 2001; Chomsky 2004).

[27] Most versions of the theory allow for a single specifier.

vP (containing the starting position of the agent/causer of an event) and the lexical VP proper. Somewhat later, we get Rizzi's (1997) study of the CP area, introducing TopicP and FocusP. In short, each of the original pieces of structure (CP, IP, and VP) shown in (24) is starting to get decomposed into smaller pieces, a process which gains momentum (and the name cartography) with Cinque (1999).[28]

In its essence, Cinque's method is simple. He first establishes the relative order of various adverbs in many languages, as illustrated in (27), observing that they point to a single, cross-linguistically valid linear order. He then does the same for functional heads, as shown in (28), arriving at a highly compatible result, and he finally refines the result by matching two independently discovered linear orders.[29]

(27) T(ense) > $\text{Mood}_{\text{irrealis}}$ (Cinque 1999, p. 33)
   a. He was then certainly at home.
   b. *He was certainly then at home.

(28) T(ense) > $\text{Asp(ect)}_{\text{perfect}}$ > $\text{Asp(ect)}_{\text{progressive}}$ > Voice (Cinque 1999, p. 57)
   a. These books ha*ve* be*en* be*ing* r*ead* all year.

To give you a taste of the plentitude of functional projections that are being discovered in this cross-linguistic enterprise, we copy Cinque's (1999, p. 106) "second approximation" of the hierarchy of clausal functional projections; each item below presents the name of the head (e.g. $\text{Mood}_{\text{speech act}}$) and the corresponding adverb (e.g. 'frankly'). If you think this is a long list, just wait, there's more: Cinque and Rizzi (2008, p. 71) estimate the number of functional heads at about 400, and according to Shlonsky (2010, p. 424), this "may turn out to be a low estimate of the *magnitude*" (emphasis ours).[30]

(29) [frankly $\text{Mood}_{\text{speech act}}$ [fortunately $\text{Mood}_{\text{evaluative}}$ [allegedly $\text{Mood}_{\text{evidential}}$ [probably $\text{Mod}_{\text{epistemic}}$ [once T_{Past} [then T_{Future} [perhaps $\text{Mood}_{\text{irrealis}}$ [necessarily $\text{Mod}_{\text{necessity}}$ [possibly $\text{Mod}_{\text{possibility}}$ [usually $\text{Asp}_{\text{habitual}}$

[28] Nowadays, most generative linguists see CP, IP, and VP merely as cover terms for (ever larger) collections of functional projections related (respectively) to discourse, tense/modality/aspect and argument structure.

[29] This is of course an oversimplification of Cinque's method. Various processes, such as question formation, focusing, or parenthetic usage—often resulting in special intonations—can disturb the "neutral" relative orders of adverbs. Morphologically fused functional heads exhibit the reverse order—this is usually called M. Baker's (1985) Mirror Principle—and there are mixed cases as well. Even the sheer amount of languages under investigation makes the enterprise formidable; while centered on Romance languages, Cinque (1999) draws data from more than 200 languages.

[30] It is impossible to even begin listing all the works on the cartography of syntactic structures. For extensive, but still far from complete, lists of references, consult the above-mentioned papers.

[again $\text{Asp}_{\text{repetitive(I)}}$ [often $\text{Asp}_{\text{frequentative(I)}}$ [intentionally $\text{Mod}_{\text{volitional}}$ [quickly $\text{Asp}_{\text{celerative(I)}}$ [already $\text{T}_{\text{Anterior}}$ [no longer $\text{Asp}_{\text{terminative}}$ [still $\text{Asp}_{\text{continuative}}$ [always $\text{Asp}_{\text{perfect(?)}}$ [just $\text{Asp}_{\text{retrospective}}$ [soon $\text{Asp}_{\text{proximative}}$ [briefly $\text{Asp}_{\text{durative}}$ [characteristically(?) $\text{Asp}_{\text{generic/progressive}}$ [almost $\text{Asp}_{\text{prospective}}$ [completely $\text{Asp}_{\text{SgCompletive(I)}}$ [tutto $\text{Asp}_{\text{PlCompletive}}$ [well Voice [fast/early $\text{Asp}_{\text{celerative(II)}}$ [again $\text{Asp}_{\text{repetitive(II)}}$ [often $\text{Asp}_{\text{frequentative(II)}}$ [completely $\text{Asp}_{\text{SgCompletive(II)}}$

However, the cartographic explosion is not limited to the verbal domain. There is some excellent work on adjectives and prepositions, but for us, it is the nominal domain that is of primary interest. The noun phrase starts receiving the functional structure with Abney (1987), who proposed that nouns are dominated by D(eterminer)P, the nominal counterpart of the clausal CP (we have been following his proposal throughout the book). Cinque (1994) synthesized various research on the unmarked order of attributive adjectives (for event nominals) into hierarchy (30)—which has since been elaborated by many authors—and proposes that each of the positions corresponds to a functional head.

(30) possesive > cardinal > ordinal > quality > size > shape > color > nationality

Even at this early stage, we can see how the syntactic decomposition of the DP plays well with L^{**}. In L^{**}, the semantic contribution of, say, a universal determiner 'every' (a negated restriction) is much different than the contribution of a numeral modifier (predication, $\#(n, x)$). And indeed, numeral modifiers have their own place within the DP functional hierarchy. In (30), they belong to the category of cardinals, which has been variously called CountP, N(ume)ralP, #P, etc., in the literature (we will use CountP).

In fact, many authors explicitly argue against a unified treatment of determiners, as found in the (early) GQ theory. For example, Giusti (2002) differentiates articles, demonstratives, and possessive adjectives; Brugè (2002) proposes demonstratives to contain [Ref(erential)] and [Deictic]; Ihsane and Puskás (2001) distinguish specificity and definiteness, arguing for the partial hierarchy [DP [Spec(ific)P [Def(inite)P …]]]; and so on. Furthermore, many authors note the parallels between the clausal and the nominal domain; for example, Ihsane and Puskás (2001) argue that SpecP is actually the nominal counterpart of TopicP. But even more importantly, the clausal and the nominal domain seem to interact. Below, we will present Beghelli and Stowell's (1997) influential reanalysis of Quantifier Raising as a syntactic phenomenon driven by feature-checking. But to do that, we have to first say a few words on what features are and how they drive the Minimalist syntactic derivation.

Features are the ultimate (morpho)syntactic primitives. By definition, lexical items are bundles of features; in more detail, they are bundles of syntactic, semantic, and phonological features. For example, the verb 'saw' is taken to consist of the category feature [V] (verb) and tense feature [past], plus semantic features (some kind of links to predicate 'see' and the past tense predicate or operator) and phonological features (however the phonological theory proposes to represent its form, e.g. by /sɔː/). Traditionally, a syntactic category could host several features, but in a syntax with rich functional hierarchy, each feature will correspond to a syntactic category (i.e. functional projection) of its own—these days, many researchers subscribe to Kayne's (2005, p. 15) Principle of Decompositionality: "UG imposes a maximum of one interpretable syntactic feature per lexical or functional element," which more or less amounts to the identification of features and functional heads.

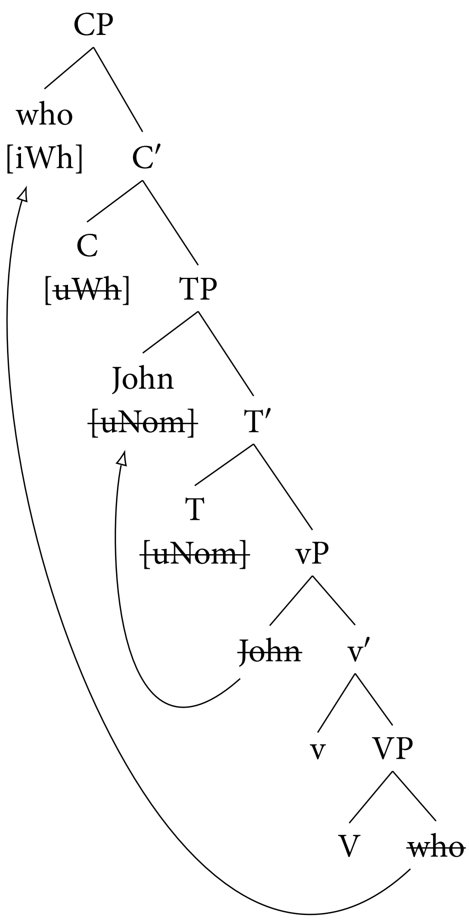
Features/heads drive both merging and movement. One idea, motivated by the contemporary richness of syntactic structures, is to assume that Merge is driven by the requirement to conform to the hierarchies of functional projections such as (29) and (30); see e.g. Adger (2003). Even Move can be seen as driven by the same requirement—see e.g. Starke (2004) and note that we have already hinted that Move is just an internal Merge—but let us stay with the more traditional Minimalist mechanism outlined below.

A (formal, syntactic) feature can be either *interpretable* or *uninterpretable*; for example, subjects and verbs often agree in number, but grammatical number is interpretable on the noun and not on the verb. Uninterpretable features have no place at LF; it is the job of syntactic derivation to eliminate them before reaching LF, lest the derivation *crash*. Uninterpretable features are eliminated in the process of *feature-checking*. Under certain conditions (at the very least, these include c-command and locality), two identical features can enter into the *Agree* relation, which licenses the elimination of the uninterpretable features involved in the relation. However, depending on the parameters of the language (the *strength* of the features involved and the presence of the so-called *EPP* feature), this relationship can have one other consequence: movement of the phrase hosting the c-commanded feature into the specifier of the projection hosting the c-commanding feature. In the example below, 'who' (containing an interpretable [Wh] feature) moves to the top of the clause to check (and eliminate) the uninterpretable [Wh] feature on C,[31] and 'John' moves to the subject position—

[31] In a more refined left periphery, [Wh] would be hosted by head Wh, in line with the above-mentioned Principle of Decompositionality.

the so-called [Spec, TP]—because this is where it can check its [Nom(inative)] feature.[32]

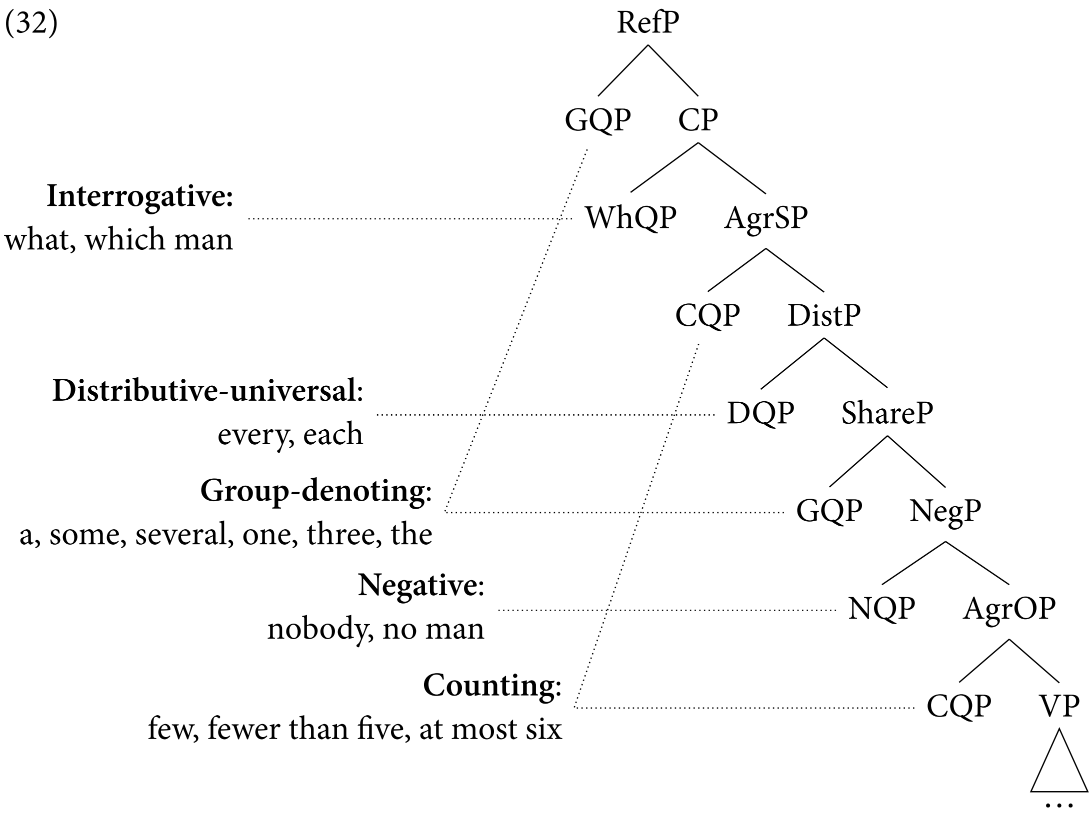
(31)



Having seen how features can drive movement, let us move on to Beghelli and Stowell's (1997) feature-driven reanalysis of Quantifier Raising (QR). Beghelli and Stowell follow and expand on Szabolcsi (1994, 1996) in recognizing that, contrary to the original statement in May (1977, 1985), QR cannot be a uniform phenomenon. The reason is empirical: there seem to be several kinds of quantificational phrases, each with its own set of scope possibilities. The proposed QP-types and their possible scope positions in the clausal hierarchy are shown in (32).

[32] Case features are quite special as they don't seem to be interpretable at all, at least at first sight. There is some research aimed at discovering the semantic contribution of various cases. For example, nominative case turns out to be closely related to finiteness, as illustrated in (31)—this is perhaps a surprising result, but is by now fairly uncontroversial.

(32)



The idea is that different QP types are endowed with different features, which are hosted in functional projections of differing heights. Thus, GQPs can take the widest possible scope because they (can) carry feature [Ref], which is hosted at the very top of the clause. NQPs, on the other hand, are limited to a fairly low position, because their distinguishing feature, [Neg(ation)], is hosted by NegP (the sentential negation), which occurs toward the bottom of the clausal spine.

We now turn to the final ingredient of the Minimalist Program which we will deploy when presenting our proposal: locality. It turns out that two syntactic elements can only interact when they contract a local relation, in the sense that there is no intervening element which could take part in the interaction. This principle, known as *Relativized Minimality (RM)*, first received a *general* formal expression in Rizzi (1990) and was later elaborated in Rizzi (2001, 2004) and Starke (2001), among others. One way to phrase RM, schematized below, is to say that it is illicit to *X*-relate two members of class *X* over an intervening member of the same class.

(33) Relativized Minimality (RM): $^{*}\ \alpha \underbrace{\ \ldots\ \beta\ \ldots\ }_{X} \gamma$, where $\alpha, \beta, \gamma \in X$

A typical syntactic phenomenon explained by RM is the so-called *weak island* effect. For instance, we cannot extract a wh-adjunct out of an indirect question. Here, RM prohibits movement of a phrase over a phrase of the same type, as in

(34), where it is impossible to move a wh-phrase 'how' over another wh-phrase 'who'.

(34) *How do you wonder *who* could solve this problem ⟨how⟩?

Let us fill in some details about Relativized Minimality. We already know that virtually any syntactic process can only take place if the participating positions are in the relation of c-command, where a syntactic node c-commands its sister and all the descendants of its sister. However, c-command also plays a role in the notion of intervener from the definition of RM. Given positions α and γ such that α c-commands γ, position β intervenes between α and γ iff it is c-commanded by α and c-commands γ. Therefore, one aspect of the RM-based expression of locality crucially depends on c-command.

But there is another aspect to RM, independent of c-command, and that aspect is that not just any intervening position is relevant for RM. As a first approximation, only constituents that could in principle take part in whatever relation is being established count as relevant interveners. In (34), it is crucial that 'who' is a wh-word itself and could thus be subject of movement into the specifier of the matrix clause. Curiously enough, however, it is not only wh-elements which prevent movement of wh-elements: focus, negative operators, measure phrases, etc., have this ability as well, as illustrated in (35) for negation.

(35) *How do*n't* you think he could solve this problem ⟨how⟩?

Generally, RM prohibits movement of a phrase over a phrase of the same *type*, where a modern definition of same type is in terms of belonging to the same class of features. Rizzi (2001, 2004) managed to organize a fairly complicated set of data by assuming the classification in (36).

(36) RM classes
   a. Argumental: person, number, gender, case
   b. Quantificational: Wh, Neg, measure, focus
   c. Modifier: evaluative, epistemic, Neg, frequentative, celerative, measure, manner
   d. Topic

Starke (2001) extended the empirical coverage of RM even further by assuming that the above classification is not flat but arboreal. For example, assuming a subclass of the Quantificational class (a subclass that he calls SpecificQ) allowed him to explain the apparent exceptions to the general prohibition against extraction out of weak islands. Given the RM logic outlined above, (37) should be ungrammatical, but it is not. However, Starke observes that extraction is only possible when the question is *specific*—as a first approximation, when "there are reasons to believe that there exists some entity which the interlocutor has in mind as a referent for the wh-phrase" (Starke 2001, p. 13).

(37) What do you wonder *whether* John will cook ⟨what⟩?

The theoretical idea behind introducing subclasses is that a member of some class cannot block a relation belonging to the subclass. Regarding the above example, even though 'what' and 'whether' are both members of Quantificational class—and 'whether' therefore blocks the movement of 'what' in the non-specific reading—only 'what' but not 'whether' is a member of subclass SpecificQ. In the specific reading, where the relation between the source and the target of the movement belongs to the SpecificQ class, 'whether' is therefore unable to block the extraction.

Returning to the content of the broad classes presented in (36), note that calling class (36b) quantificational is perhaps a bit of a misnomer (cf. De Clercq 2013), as it does not cover typical quantificational phrases such as 'every dog', 'some dog', etc. (Rizzi 2001, fn. 10; Starke 2001, p. 6). However, the empirical grounds for the exclusion of these items from (36b) are firm. As shown in (38) below, a universally quantified subject does not prevent the extraction of a wh-phrase (but it does enforce the narrow scope reading of the wh-phrase). The absence of universals from the Quantificational class will actually turn out to be very important in the following section—we will propose to fill out the list of RM classes by adding a class containing universals and negation.

(38) How do you think that *every*body could solve this problem ⟨how⟩?

We said that we were going to give you a crash course, which means fast, and this section was nothing if not fast. If you want to know more you can read an introductory text book on Minimalism (e.g. Adger 2003) or take a course, but we have already laid out everything that we are going to need for our proposal.

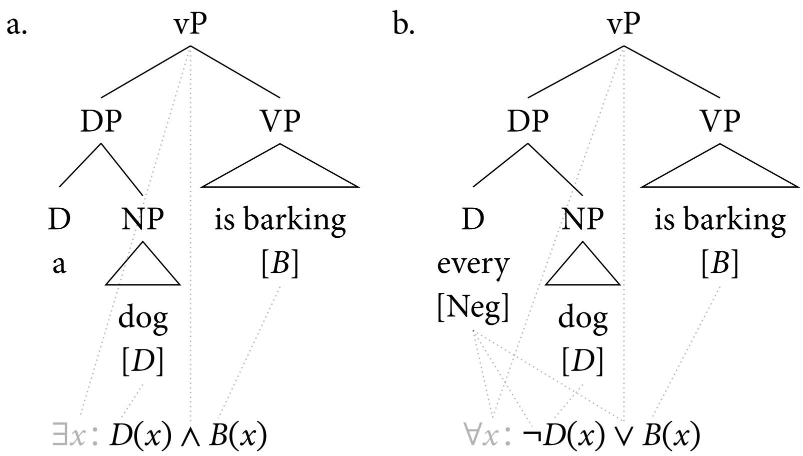
In short, the essential idea we are going to lean on in our attempt to integrate L** with LF representations in minimalism is that the determiners like 'some' and 'every' are just bundles of features. On our view, they are not themselves quantifiers. But because they carry features, and in particular because 'every' carries the negation feature, that will be enough information, in concert with Restrictedness Universal and Restricted Closure, to build the formulas of quantifierless L**.

Of course the abstract negation feature is carrying a heavy load here, and even though the proposal is simple from a technical perspective, it does raise important questions. We attend to those questions next.

12.2.3 The negation feature

The single most important component of our story is certainly polarity—in its essence, our Dynamic Deductive System is a polarity logic, and our quantifierless format for conservative predicate logic hinges on polarity as well. L** allows no binary sentential connectives other than conjunction and disjunction, and this allows it to encode polarity exclusively with negations. Because we want to identify L** logical forms with the LF representations of natural language syntax, we hypothesize that a correct account of natural language syntax will generate these negations precisely where they are needed.

Picking up the story from subsection 12.1.3 on the isomorphism between LF and L**, where we compare the effect of the indefinite and the universal determiner, we see that we need the negation to occur with the latter but not with the former, and it must occur precisely where we find the universal determiner in syntax. We depict the relation between syntactic structure and logical form for these determiners in (39).

(39)


As we noted, this makes the gist of our proposal very simple: the lexical entry of 'every' carries a negation feature [Neg]; the lexical entry of 'a' contains no such feature. Everything else flows from that. The question we now face is whether this is plausible on syntactic grounds.

We will start by arguing that at the very least, the proposal is *not im*plausible, by pointing out that the negation operator is clearly invoked by more than just sentential negation. We will then take a detour into the phenomenon known as *negative concord*, for two reasons. First, to get a handle on what might be the difference between 'every' and 'no', given that they must both contain a negation feature in our account. Second, to familiarize ourselves with a strong syntactic argument for some word ('no') containing a certain feature (the sentential negation [Neg]). The latter will inspire us to search for a syntactic argument for [Neg] in 'every'. We will find it—even if it might be weaker than in the case of 'no'—

in another well-known syntactic phenomenon, the reluctance of universals to scope directly over a negation, and explain how our idea that universals contain a negation feature of their own could contribute toward the understanding of this phenomenon.

We can imagine a criticism of our proposal which says that the negation operator in the logical form in (39b) has no morphosyntactic exponent, in the sense that the corresponding sentences do not contain a voiced 'not'. But such a criticism would be unfounded. While sentential negation (40) is a ubiquitous property of natural languages and is the canonical morphosyntactic exponent of the negation operator, not every negation operator in an LF representation is a reflex of the sentential negation 'not' (or its contracted form). For example, adjectives may be prefixed (with some restrictions) by 'un-' and 'non-' (41), and while the details of the resulting logical forms are not trivial,[33] most semantic analyses of these prefixes involve some form of a negation operator.[34]

(40) Fido does not / doesn't bark.
   a. $\neg B(f)$
   b. $\exists x : F(x) \wedge \neg \exists e : B(e) \wedge \text{Agent}(x, e)$

(41) a. unhappy, unfounded, unreasonable
   b. non-existent, non-linear, non-negative

Given all the forms a negation operator can take in natural language, it is clear that our proposal adds nothing qualitatively new to the morphosyntactic arsenal for expressing the negation operator. It merely updates the list of lexical items carrying a negation feature to include 'every'—or rather the universals in general, as all of their logical forms will presumably contain a negation operator.

However, there are other deteminers to account for. In particular, what can we say about 'no' (as in 'no dogs are barking')? Clearly, it must carry a negation feature of some kind, but how are we to distinguish it from 'every'? A broader question would be whether all words and morphemes which bring the negation operator into semantics contain one and the same syntactic negation feature, or could the negation operator be a semantic reflex of several distinct syntactic features?

We certainly lean toward the latter answer. Even at a glance, the different distributions of the sentential negation ('not'), the negation prefix 'un-' and the

[33] For example, prefixation by 'un-' forms an opposite rather than a complement, i.e. the semantic analysis of 'unhappy' must be construed in a way to leave space for the middle ground of neither happy nor unhappy.

[34] Of course, the examples in (41) hardly exhaust the list of lexical items whose semantic analysis involves negation. Some adjectives take an irregular form of prefix 'un-', so we have 'irregular', 'illogical', and 'irresponsible'; other adjectives have supplementary negative forms: 'bad', 'ugly', etc. Nouns can be prefixed by 'non-' as well—'non-issue', 'non-smoker', etc.—and many other words (the so-called NPI licensers, see section 11.3) presumably involve negation: 'without', 'doubt', 'forget', etc.

negation prefix 'non-' speak in favor of distinct syntactic features—after all, we can't use 'un-' as the sentential negation, for example. Furthermore, detailed research into the syntax of negation is revealing further positions. Whereas the seminal work of Klima (1964) distinguished sentential and constituent negation, De Clercq (2013) proposes no less than four possible negative features contained in negative markers found in adjectival copular clauses alone. She distinguishes negative polarity ($\mathrm{Pol}^{\mathrm{Neg}}$) markers, negative focus ($\mathrm{Foc}^{\mathrm{Neg}}$) markers, negative degree ($\mathrm{Deg}^{\mathrm{Neg}}$) markers, and negative quantity ($\mathrm{Q}^{\mathrm{Neg}}$) markers (with the corresponding features).[35] As they pertain to the adjectival domain, all these negative elements are not directly relevant for our proposal, but the multitude of negative adjectival positions inspires confidence that multiple negative positions should be found in the nominal domain (which 'every' belongs to) as well. Finally, assuming that several functional heads are interpreted as negation does not single out negation as special; having some operator as the semantic reflex of more than one syntactic head is a common occurrence. For example, Cinque's hierarchy of functional projections (see subsection 12.2.2) contains two instances of repetitive aspect, so that an adverb such as 'again' can quantify either over events or over the process (Cinque 1999, p. 92).

The negation zoo outlined above reinforces our conviction that it is syntactically unproblematic to propose that 'every' contains a negation feature and that this negation feature is distinct from the negation feature commonly associated with 'no'. We will see that there is firm evidence that 'no' contains the sentential negation feature [Neg]. For 'every', we propose a novel feature [$\mathrm{Neg}_\forall$], the choice of the name reflecting the broader hypothesis that this feature is found in univerals in general.[36]

Before delving into (even) deeper syntactic waters, let us compare the LF representations of sentences containing 'every' and 'no' with their L** logical forms, with the goal of understanding the fine points of the relation between LF and L** pertaining to negation. The issue, as we will see, is what precisely constitutes the scope of the negation.

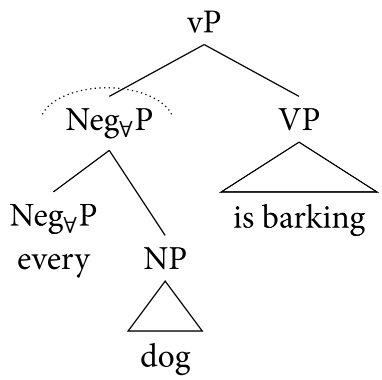
The theory of restrictedness developed in Chapter 9 tells us that the negation feature [$\mathrm{Neg}_\forall$] of 'every' must occur at the very top of the DP in (42), as the formula can only turn out restricted if the entire restriction ($D(x)$, the interpretation of 'dog') is negated. In other words, in the case of a noun phrase headed by 'every', DP *is* $\mathrm{Neg}_\forall$P. The situation is different with 'no', illustrated in (43). Following Beghelli and Stowell (1997), and in fact numerous other authors, we assume that 'no' contains the sentential negation feature [Neg] and that DP 'no dog' is moved

[35] Languages where all these negative markers are different provide the strongest evidence for existence of four distinct features. For example, Greek distinguishes 'dhen', 'oxi', 'mi(-)', and 'a-' (De Clercq 2013, p. 40). In English, 'un-' is the quantity marker, 'non-' is the degree marker, and 'not' is syncretic, functioning both as polarity and as focus marker.

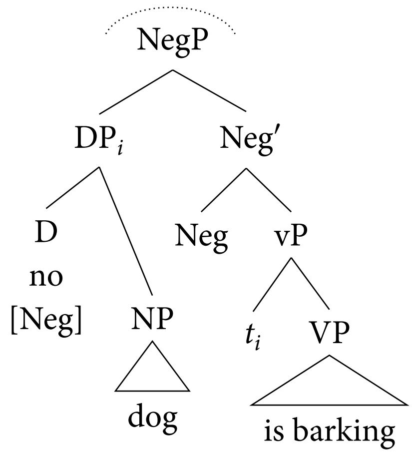
[36] When it comes to 'all', this is perhaps an overstatement.

to the specifier of NegP in the sentential spine, but L^{**} imposes no conditions on the position of the [Neg] feature within the DP.

(42) $\forall x : \neg D(x) \vee B(x)$



(43) $\neg(\exists x : D(x) \wedge B(x))$



Comparing the LF representations to the desired L^{**} formulas, it is clear that we get the correct correspondence if we assume that the negation triggered by head $\text{Neg}_{(\forall)}$ takes scope over the entire $\text{Neg}_{(\forall)}\text{P}$, as indicated by the dotted arches. The difference between (42) and (43) is that in the former, $\text{Neg}_{\forall}$ projects only once while in the latter, Neg projects twice. In other words, $\text{Neg}_{\forall}$ in (42) only takes a complement ('dog') and therefore negates only that constituent. On the other hand, Neg in (43) takes both a complement ('is barking') and a specifier ('dog') and negates both, or more precisely, it negates their conjunction. The conjunction of course arises as the interpretation of the neutral junction by Restricted Closure from Chapter 9. Remember that RC determines the closure site, the closure type and the type of junction. The closure site is just below the negation, as that is the lowest position containing all occurrences of x, and the closure is existential because the *local* polarity of the restrictor ($D(x)$) within the restriction (again, $D(x)$) is positive; the neutral junction is interpreted as a conjunction for the same reason.

We have proposed that 'every' contains a novel negation feature [$\text{Neg}_{\forall}$], but we have also said that 'no' is standardly assumed to contain the sentential negation feature [Neg]. We will now outline an analysis of a well-known phenomenon which strongly argues in favor of the latter assumption. The phenomenon is known as *negative concord*, and we will present an analysis (adapted from Zeijlstra 2004) relying on uninterpretable [Neg] features.

Many languages (like Slovene and some non-prestige dialects of English) feature a special class of so-called *n-words*, whose members contain items translatable by English determiner 'no', negative pronouns 'nobody' and 'nothing', adverbs 'never' and 'nowhere', etc. However, the syntactic and semantic behavior of n-words is unlike the behavior of their (standard/prescriptive) English counterparts. For one, they require the presence of sentential negation (in the same clause), as illustrated by the contrast between (44) and (45). Furthermore, (45) also shows that the n-word and the sentential negation jointly contribute a single negation to the logical form—unlike in standard English, where the negative determiner accompanied by the sentential negation yields a double negation (i.e. it results in an essentially positive sentence). In fact, in negative concord languages, a single negation arises even in the presence of multiple n-words (46).

(44) *Noben pes laja. (Slovene)

*Noben	pes	laja.	
no	dog	barks	

(45) Noben pes ne laja. $\neg\exists x\colon D(x) \wedge B(x)$

Noben	pes	ne	laja.
no	dog	not	bark

"No dog barks."
*"No dog doesn't bark."

(46)

Nikoli	ni	nikjer	nikomur	nič	povedal.
Never	be+not	nowhere	to-nobody	nothing	said.

"He never said anything to anybody anywhere."

Zeijlstra (2004) explains this set of facts by arguing that n-words are not in themselves semantically negative; they only introduce a negation operator by deploying features that require the presence of the sentential negation. He implements this by assigning an *un*interpreted feature named [uNeg] to n-words and an interpreted feature named [iNeg] to the (base-generated) sentential negation. The uninterpreted negation feature on an n-word makes the n-word semantically non-negative; it is not a negation operator by itself. On the other hand, an uninterpreted feature must be checked, which explains the need for the local presence of an item carrying an interpreted negation feature—the sentential negation. Another way to put it is that [uNeg], while not itself a negation, can always be counted on to bring a true negation to the dance.

What we have here is an analysis of a common phenomenon which claims that in some languages, negative determiners contain an uninterpreted sentential negation feature. It is then not a big step to the idea that negative determiners might contain the sentential negation feature in all languages, but that this feature is interpretable in double negation languages. In short, negative determiners carry [uNeg] in negative concord languages and [iNeg] in double negation languages.[37] (Having seen feature [Neg] being realized both as an interpretable and as an

[37] This statement is a simplification. If nothing else, there are several types of negative concord. Furthermore, in the spirit of full disclosure, Zeijlstra (2004) proposes a different story for negative determiners in double negation languages. However, his story relies on the existence of the basic syntactic operation Adjoin (alongside Merge and Move), which we are reluctant to introduce into our syntactic toolbox—in fact, one theoretical offshoot of the cartography research is that given all the space offered by the rich syntactic structures, Adjoin becomes a relic of the past.

uninterpretable feature, you might wonder about the status of [$\mathrm{Neg}_{\forall}$] in 'every'. Here, we clearly need to assume that this feature is interpretable. For one, it must yield a negation operator within the restriction, but it also could not be checked by moving up the verbal spine anyway, as it appears to be absent from the verbal domain.)

One moral of negative concord is that actual negations in the semantics of natural language don't always correlate with the words that appear to be negative in writing and voiced speech. There are, to be sure, clues as to the presence/absence of negation operators, but the actual location of a logical negation in natural language can only be ascertained by constrained theoretical investigation. It cannot be reliably determined by hunches we may have that are grounded in the accidents of morphology.

While this moral is important for our account, as we also see negation in words (like 'every') which are usually thought of as non-negative, there is another lesson to take from the story of negative concord. This phenomenon tells us, without a shadow of doubt, that at least in some languages, negative determiner 'no' is intimately related to sentential negation. The question now is, is there some *syntactic* phenomenon which will give us reason to believe that 'every' contains a negation feature as well? We emphasize that we are searching for a syntactic phenomenon—we hope to have already given you plenty of good semantic and mathematical arguments in favor of the negation in 'every'.

We suspect that for many readers this subsection provided more than enough linguistics. But for other readers (linguists, for example) there are bound to be more questions about how our project works on the linguistic end. And hopefully, we will have piqued the interests of non-linguists so that they want to take a deeper dive into our view of syntax, and perhaps revisit the negation topic in more detail. If that isn't your cup of tea, feel free to jump ahead to the next chapter. But if it *is* your cup of tea, we begin the deeper dive now.

Let us recapitulate the problem. In our quantifierless L**-based logical forms, we can distinguish universals and indefinites by the presence vs. absence of a single logical primitive, negation, as shown in (39) at the start of this subsection. Assuming that this difference maps directly to syntax, we are therefore forced to conclude that universals contain a negation feature. However, search as you will, syntactic literature contains no proposals to this effect. Are we utterly on the wrong track here?

We believe not. In the absence of a theory of quantification based on restrictedness, semantics provides no reason for universals to contain some kind of [Neg]. In absence of such a theory, [Neg] can only correspond to the negation operator $\neg$ itself, which makes proposing such a feature within universals a folly—thus, we believe, the absence of syntactic proposals to this effect. But once semantics provides some principled logico-semantic grounds for having [Neg] within universals, can we find any (morpho)syntactic evidence to support this idea?

The primary data point we have in mind to support the presence of [Neg] in universals is their reluctance to scope directly over a negation, as illustrated by the unacceptability of (47) under the indicated intended reading.[38]

(47) *Every dog doesn't bark. ($\forall > \neg$)
"No dog barks."
a. $\forall x: \neg D(x) \vee \neg B(x)$
b. $\forall x: \neg D(x) \vee \neg \exists e: B(e) \wedge \text{Agent}(x, e)$

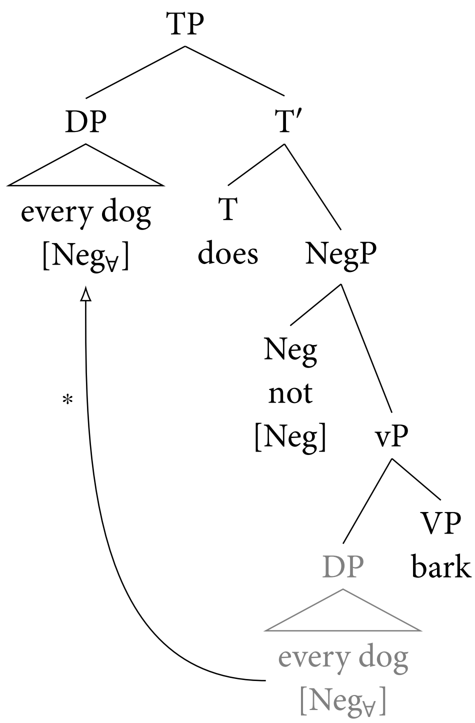
Why would that be the case? There seems to be no straightforward semantic explanation; there is nothing wrong with logical forms that one would assign to the above sentences, neither in the standard format nor in the event semantics rendering. Could the explanation be pragmatic? Perhaps the sentence is blocked by the existence of the paraphrase with 'no', as suggested by Zeijlstra (2004)? This seems unlikely. For one, the unacceptability appears too strong to be due to a pragmatic factor. Furthermore, one would then expect similar blocking effects for other paraphrasable quantified sentences, such as the pair in (48).

(48) a. Not every dog barks. ($\neg > \forall$)
b. Some dog doesn't bark. ($\exists > \neg$)

It seems that the reason for the unacceptability of (47) must be syntactic. We suggest that this is essentially a Relativized Minimality (RM) effect (see subsection 12.2.2 for an introduction to RM). Remember that RM prohibits movement of an item over an item of the same class, and that while several RM classes are recognized, 'every' phrases belong to none of them. We therefore propose to fill in the gap in the list of RM classes—(36) in the previous subsection—by adding the "real" quantificational class. In particular, we submit that this class contains universals and negation—both containing some kind of negation feature—and claim that this is what prevents the movement of 'every dog' across sentential negation in (49).

[38] According to Beghelli and Stowell (1997) and Zeijlstra (2004), sentence (47) is marginally acceptable in the $\neg > \forall$ reading, but note that the contexts where (47) is appropriate all involve particular information structure, which most likely leads to logical forms much different than those presented above.

(49)



Admittedly, however, matters are not that simple, neither theoretically nor empirically, so our account will need some elaboration. For one, RM is usually only taken to apply if the offending feature is the feature driving the movement. This is not the case above, where (at least as depicted) the offending feature is [Neg] but the movement is triggered by the need of the subject to check its uninterpretable [Nom] feature.[39] The other, even more pressing, problem for our rudimentary RM-based analysis is that in the presence of an indefinite object (or an indefinite adverb) scoping over negation, 'every' can in fact scope over both of them. As things stand right now, there is no reason for the indefinite to "fix" the ungrammaticality of the movement.

(50) Every dog didn't notice a cat. $(\forall > \exists > \neg, {}^{*}\forall > \neg > \exists)$
a. $\forall x\colon \neg D(x) \vee \exists y\colon C(y) \wedge \neg N(x, y)$
b. $\forall x\colon \neg D(x) \vee \exists y\colon C(y) \wedge \neg \exists e\colon N(e) \wedge \mathrm{Agent}(x, e) \wedge \mathrm{Theme}(y, e)$

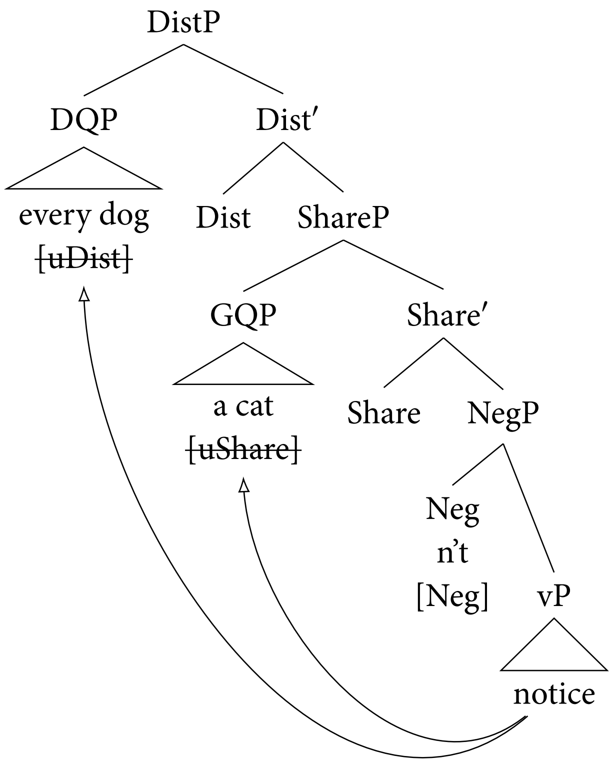
The widely adopted explanation of the pattern exemplified by (47) and (50) is due to Beghelli and Stowell (1997), whose proposal (essentially a feature-checking implementation of QR) was outlined in the previous section. While we are very much in favor of their proposal in general, we do not find it free of problems (to be

[39] Hornstein (1995) proposes that [Spec, NegP] is a case-related position—in terms of (36), this would put Neg into the argumental class as well and thus potentially solve the problem described above—as the presence of negation in Slavic languages shifts the accusative objects to genitive. The problem is that there seems to be no other evidence supporting this idea (Beghelli 1995, pp. 36–38).

described below), and we believe that it must be supplemented by some sort of RM-based account.

On Beghelli and Stowell's account (which does *not* deploy RM), strongly distributive DPs ('each' DPs, and some 'every' DPs) move to the specifier of DistP (to check their [Dist] feature), and this projection must immediately dominate ShareP, a functional category which "requires an existentially quantified indefinite GQP (the distributed share) to occur in its Spec position" (Beghelli and Stowell 1997, p. 92). This explains the grammaticality of (50); as shown below,[40] Dist has the Share complement it requires, and [Spec, Share] is occupied by the indefinite direct object, which has moved there to check its [Share] feature.

(51)

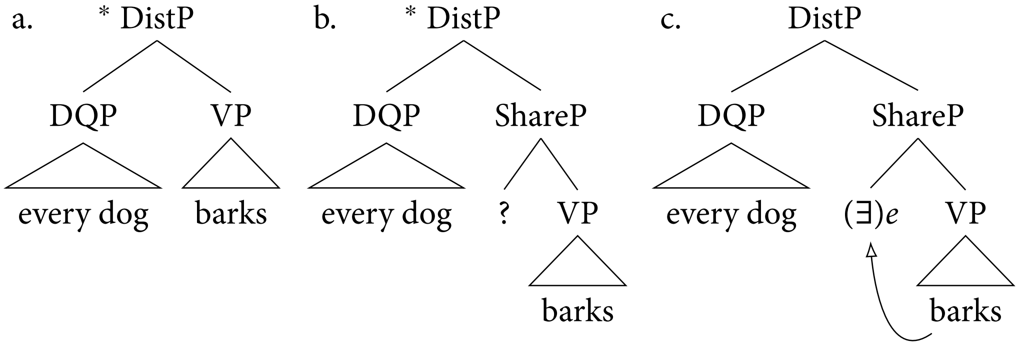


What about universally quantified subjects of intransitive verbs? How can 'every dog' in (the affirmative) (52) occupy [Spec, DistP], given that the sentence contains no indefinite which could fill the [Spec, Share] that the DistP allegedly requires? The structure will be illicit either because DistP does not immediately dominate ShareP (52a), or because [Spec, ShareP] does not contain the distributed share (52b). Beghelli and Stowell claim that in sentences containing no overt indefinite that could move into [Spec, ShareP], this position is occupied by the existential quantifier over events, as shown in (52c). And to explain the ungrammaticality of the negated (47), repeated below as (53), they further assume that this movement is disallowed when "the event variable is bound by [the negative operator in

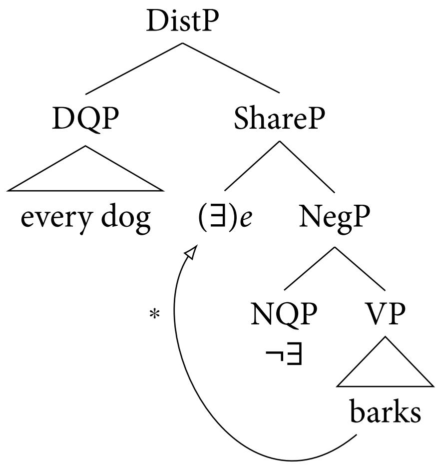
[40] In (51), we omit the structure above DistP; the omission includes TP and thus the auxiliary 'did'.

[Spec, NegP]]" (Beghelli and Stowell 1997, p. 95); as shown in the tree, the latter is an unselective existential quantifier in their account (Beghelli and Stowell 1997, pp. 83, 103).

(52) Every dog barks.



(53) *Every dog doesn't bark.



While we are sympathetic with Beghelli and Stowell's core idea of a DistP requiring a ShareP complement, we cannot subscribe to their analysis outlined above, as we find it internally inconsistent. Their account is essentially syntactic—(52c) and (53) are syntactic trees—but it crucially involves the idea that the resident of [Spec, ShareP] is a semantic entity: a quantifier over events, or perhaps an event variable. Even worse, the syntactic movement of this semantic entity ((∃)*e*) is regulated by its interaction with another semantic element (the unselective existential quantifier in [Spec, NegP]).

It seems to us that attempting to understand the quantifier over events in syntactic terms fails as well. The only feasible syntactic candidate for (∃)*e* is the verb. However, Beghelli and Stowell's analysis relies on the idea that the event argument is *on par* with thematic arguments, which clearly does not hold if the event argument is reinterpreted as the verb.

We submit that the right way to go is to take the Relativized Minimality account outlined above, and supplement it with the core of Beghelli and Stowell's analysis

(DistP and ShareP). The basic case of a negated intransitive verb is covered by Relativized Minimality alone, as shown in (49) above; 'every' cannot move over a negation because [$\text{Neg}_\forall$], the characteristic feature of 'every', belongs to the same class as the sentential negation feature [Neg]. To nevertheless derive sentences such as (50), we first follow Beghelli and Stowell in assuming that the structure of this sentence is (51), but then also adopt Starke's (2001) version of Relativized Minimality with the tree-like structure of RM classes (see subsection 12.2.2). Specifically, we assume that [Dist] belongs to a subclass of [$\text{Neg}_\forall$]. The movement of 'every dog' to [Spec, DistP] in (51) is then allowed because it is an instance of the movement of a subclass feature ([Dist]) over a superclass feature ([Neg]; remember that [Neg] and [$\text{Neg}_\forall$] belong to the same class).

Our goal here is obviously not to provide a bullet-proof analysis of the interaction between universals and sentential negation. Rather our goal is to show how the L^{**} analysis interacts with the current state of the syntactic theory. One might have expected that once our analysis was introduced it would give rise to puzzles and the need for clever hacks. However, we believe the opposite is the case. We are convinced that the L^{**} paradigm resolves more puzzles than it introduces, and tells us that we are on the right track—that L^{**} will embed into linguistic theory and the Minimalist Program in a natural way. This isn't to say that it will solve all problems nor that it will generate none. It is rather to say that the sign of a successful integration is that the proposal is a plausible and helpful addition to the state of ongoing linguistics research.

13
Remaining Conceptual Issues

We began this journey with a discussion of the Medieval logicians' goal of reducing all logical rules down to two basic rules—each triggered by the polarity of the environment in which it is applied. This was the *dicta de omni et nullo*. We also noted that the Medievals saw an important connection between polarity and quantification. In subsequent chapters we have shown how these core ideas can be refined and supplemented with a natural deduction system, how the resulting system for Natural Logic can be sound and complete, and how this system can cash out the Medieval ideas about the deep connection between polarity and quantification. In addition, we have seen how the resulting system can provide us a syntactic reflex of conservativity—i.e. restrictedness. We have also seen how polarity is the common thread running through our accounts of restrictedness and quantification and inference, and how it plays a critical role in our deductive system.

After vindicating the core elements of the Medieval project—its Holy Grail—one option might be to declare victory and end the book right here. However, the success of the general project seems to raise at least as many questions as it answers. First, it asks us to reflect on the sense in which a theory of natural deduction can be said to be our *theory* of deduction, humans being notoriously bad at making logical inferences, after all. Thus we need to say some things about the competence/performance distinction. Second, we need to pause and reflect on the deep connection between polarity and deduction. Why is polarity so important? Third, one wonders if our project can be used to anchor an alternative theory of meaning—a proof-theoretic conception of meaning that might replace traditional denotational semantics. To put it another way, can proof theory be the new theory of meaning? Of course we should also pause to reflect on the topic that got this whole project rolling—the Holy Grail of Natural Logic. What have we learned about that quest? In this final chapter, we thus begin exploring some of those fascinating foundational questions.

13.1 L** and the Theory of Human Inferential Capacity

In earlier chapters we suggested that our inline dynamic deductive system had a certain kind of psychological reality to it—that it might serve as a part of an explanation for the kinds of deductive abilities that human agents actually have.

Language, Form, and Logic: In Pursuit of Natural Logic's Holy Grail. Peter Ludlow and Sašo Živanović,
Oxford University Press. © Peter Ludlow and Sašo Živanović 2022. DOI: 10.1093/oso/9780199591534.003.0014

This is a tricky claim, and it needs to be handled with lots of care. Obviously, as anyone who has ever taught logic knows, people can at times be atrocious at distinguishing between valid and invalid arguments. And for anyone who finds themselves caught up in disagreements at family gatherings or in bars (and faculty meetings!), one can expect to see lots of basic fallacies committed.

The problem here is that one needs to distinguish a certain logical capacity that humans have from the way in which they execute that capacity. Here we get into a phenomenon that will be familiar to linguists, and which Chomsky has characterized as the *competence/performance* distinction. Sometimes, for any number of reasons, ranging from memory limitations to distractions, we behave in ways that are antithetical to our linguistic competence. Perhaps not surprisingly, the same phenomenon applies in the realm of logical agents.

We will soon get to the issue of competence and performance as it applies to the logical realm, but before we do, we need to clean up some misunderstandings about human inferential abilities and what breakdowns in performance are indicative of.

For several decades now, the received view has been that humans are, in some sense "not rational" because they are prone to commit a number of errors in logical reasoning. There are now several studies that are thought to support this, including Wason and Johnson-Laird (1972). However, while the methodology of these studies is sound, there are concerns about the kinds of conclusions that one can extract from the data.

For example, while it is not our concern in this book, conclusions about the "rationality" of agents really cannot be extruded from adherence to logical principles. Logic is a tool for sorting out problems of arbitrary complexity, it is not necessarily a governing principle for the day-to-day activities of people. Typically, we do quite well using fast reasoning techniques that don't track the laws of logic but which get you through the day with a minimum of cognitive labor and infrequent errors of consequence. There is nothing non-rational about that.

Quite apart from the question of rationality, one can ask if people are behaving in a way that is non-logical. Here again, caution is necessary. As Rips (2008) notes, there is more than one kind of logic (or at least multiple candidates for whatever ultimately counts as logic), and the famous psychological experiments seem to assume that the logics typically taught in intro courses are the ones that get to decide what is logical and hence what is rational. But, as noted earlier, there are numerous candidate logics, and it is also worth noting that sometimes people are reasoning inductively rather than deductively. Thus, while it is true that reasoning 'all As are Bs; this is a B; therefore, this is an A' is an atrocious fallacy in deductive logic, if one is reasoning inductively it might be just fine when supplemented by appropriate priors. Thus, if I am newly informed that all dogs bark, when I am then told that something is barking I might infer that it is a dog, for I consider the probability that I am hearing a seal or other barking thing to be quite low given my prior knowledge that there are lots of dogs in the neighborhood and precious few seals.

Nor can one discount the pragmatic force of proffering 'all dogs bark' in the first place. When the continuation is 'something is barking' one might naturally suppose that the goal of the first utterance was to raise dogs to salience. Otherwise the utterance hardly seems relevant.

The moral here is that there is more to human reasoning capacity than adherence to a particular deductive (or inductive) reasoning strategy, and humans seem to be optimized to solve everyday problems with astounding efficiency. People do show up for important meetings after all, and when they don't it is typically due to a breakdown in health, or transportation, or lack of desire to be there, rather than some reasoning error—even when the meeting is on the other side of the world.

The point is that before we begin drawing conclusions about logical competence from errors on logic exams, we need to control for pragmatic effects and task demands and somehow assure ourselves that the subject is reasoning deductively with their real world knowledge completely siloed and that they are not reasoning inductively or deploying knowledge about things like the relative frequency of dogs vs. seals in the local suburban *cul de sac*, and finally we need to know that they in fact are reasoning with care and not using some fast but typically adequate system that is more efficient and burns fewer mental calories and is hence, in the normal course of events, completely rational to deploy.

This takes us to another issue. The mere fact that an agent is competent or "has" a logic in some yet-to-be-determined sense leaves wide open the system that the agent uses to carry out deductions. If studies were conducted with angels as experimental subjects, and if those angels had (let us say) infinite minds that could carry out an infinite number of deductions in an instant, inferential failures would be pretty damning of the logical systems of angels. But as humans typically do not have infinite minds, they must utilize imperfect strategies for carrying out inferences before they perish of old age.

For example, given that Disjunction Introduction (Addition) is a perfectly valid logical rule, there is no reason why an agent, attempting to reason from A to $A \vee C$, could not spend an eternity utilizing Addition of the following sort: $A \vee B$, $A \vee (A \vee B)$, $B \vee (A \vee (A \vee B))$, etc. This sort of blind inference is entirely logical and hence, one imagines, rational according to bad theories of rationality, but it is just a lot of logical wheel spinning. The inferences need to be guided by the conclusion, or, if you prefer, one needs to reason backwards from the conclusion in order to guide the process. If you know that the conclusion is $A \vee C$, then you have a good idea that introducing Bs is not going to be fruitful, no matter how logical, given your goals.

Accordingly, agents need more than logic, they need heuristics to guide them in the application of the rules of their logic. They may even have dispositions to use some rules more than others. Thus as noted by Rips and Conrad (1983), human agents really do not like to deploy Disjunction Introduction. It is a perfectly valid logical rule, but its deployment can be unproductive when we engage in logical reasoning.

With these considerations in mind, Rips developed a theory that incorporated not just logic, but a theory of the heuristics that people might use to carry on their logical reasoning. Given that total picture, one might have a theory that can predict when human agents are more apt to be successful and when they are more apt to fail in their logical reasoning.

The picture we thus have entering into this topic is not entirely clear-cut. We can't extract interesting conclusions about the naturalness of a logical system in isolation, but we need to consider it (the system, including its proof theory) in concert with a theory of the heuristics that guide people through the deductive process. Even with that much in play, one needs to be sure that, somehow, one has successfully controlled for pragmatic effects and that the task demands are clear enough so that inductive reasoning is not overriding the deductive task. Controlling all of these things is not trivial. However, as we will see, one can get glimpses into deductive reasoning abilities.

In the remainder of this section we proceed as follows. In subsection 13.1.1, we review Chomsky's idea of the competence/performance distinction in an effort to see if it is applicable in the deductive setting. In subsection 13.1.2 we look at the Psychology of Proof (PSYCOP) program of Rips (1983, 1994, 2008) and see how it can be used as a model for extracting predictions of human logical behavior from understanding that behavior as a product of the logical system in concert with a proof procedure plus heuristics. We then reflect on the lessons of that model of investigation to our DDS system, with an eye to how it squares up with the inferential failures and successes of human agents. In effect, we want a system that, in the hands of an agent with infinite resources, reasons perfectly, but in the hands of finite agents with heuristics optimized for efficiency, will fail where they fail and succeed where they succeed.

13.1.1 Competence, performance, and the ontology of Natural Logic

So far in this book we have been avoiding three very large elephants in the room. We have talked about Natural Logic as being concerned with natural language forms, but we haven't said very much about what the ontology of natural language is. And we have been talking as though Natural Logic is a thing that we have (or have access to), but without much concern for how we come to have it or even what it means to have it. Finally, we have said just a little about how Natural Logic is deployed by agents who have it (and how it is misdeployed). As it turns out, these three elephants are closely related. In this section we want to get a bit clearer on the nature of those elephants, how they are related, and how they can be removed from the room without making too much of a mess.

First, let's take up the issue of what the ontology of natural language is. This is a natural place to start, since Natural Logic is supposed to be concerned with natural language syntax, and it is reasonable to ask what *that* comes to. Anyone who is familiar with the linguistics literature over the last few decades knows that this is a topic on which there is plenty of disagreement. We are putting our cards on the table here, not so much to convince others of our views on the ontology of natural language, but to lay a foundation for what we have to say about the psychological reality of Natural Logic. We know of no reasons why the results achieved in previous chapters cannot be incorporated into other approaches to the ontology of language, and perhaps what we have to say in this chapter will give hints as to how they can be incorporated into other frameworks. Thus, what we say here is similar to what we had to say about meshing our understanding of L^{**} with the Minimalist Program. We aren't saying this is the only way to ground things, but it is *a* way, and perhaps it will be illuminating for other ways.

To keep this discussion as sharp as possible, let's first focus on the ontology of grammars, which are systems of rules that determine languages. We take our point of departure on the ontology of grammars from the proposal in Ludlow (2011) which in turn owes much to a proposal by George (1989). Linguists often disagree over whether a grammar (understood as an object of empirical study in linguistics) is an abstract object or a psychological object; our approach here is more of a hybrid approach.

Our idea is that while grammars are indeed abstract (mathematical) objects, one has the grammar one does by virtue of being in a certain psychological state. Sometimes Chomsky speaks of us "knowing" a grammar and sometimes of us "cognizing" a grammar, but we will remain neutral on this and simply speak of "having" a grammar.

As George puts the idea, we have the grammar we do because we have a *psy-grammar*[1] that in some sense grounds that grammar. Ludlow (2011) updates the idea for the Minimalist Program by suggesting that the psy-grammar is in turn grounded by a *physio-grammar*—a level of grounding where low level biophysical principles come into play. It is by virtue of some state of our physio-grammar and our psy-grammar that we have the grammar (the abstract object) that we do.

This still leaves open interesting questions about how the grammar is or can be grounded by the psy-grammar or the physio-grammar. There is no easy answer to this question because there is no easy answer to the question of how physical (or psychological) states correspond to abstract structures. Obviously, this is

[1] George uses the term 'psychogrammar', but we prefer 'psy-grammar' for obvious reasons.

related to the ancient problem of how objects can "participate" in the forms, first faced by Plato some two and a half millennia ago. If we had a solution to this problem (which we don't) this would not be the place to present it, but we can share some basic observations about what one seeks in such relations, with an eye to avoiding theories that are non-starters.

One idea is the thought that we only have a grammar by virtue of directly hard-coding the grammar in some computational language. In the case of computers this initially appears to be a coherent idea, and the thought is that one could appeal to a direct isomorphism between computational structures and the rules of the grammar. This is a lot to ask for human cognition, where we possess no established computer language in which the mind/brain executes its operations. That is just the way it is with systems of natural origin in which we don't have access to documented computer languages. Even for computers, however, matters are not as tidy as these first appear.

Our preferred view is that there is a subjective element to our judgment of the relation between physical and psychological processes that we study and the abstracta that those processes ground. One way to look at this is that we look for patterns in lower level phenomena that map in a natural way to our abstract models. The reason these judgments must be subjective is first that, following arguments from Kripke (1982) and Wittgenstein ([1953] 2009), there can be no fact of the matter about the syntactic states of a physical system in isolation (that is, without knowledge of its function or embedding environment, or the access to the programmer, if there is one). Its syntactic states are grounded by states that are external to the machine.[2]

Second, as Gallistel and King (2010) note, any system that is computational is an information-processing system and information itself must be understood in subjective terms. That is, since the information carried by a signal (or being processed by a machine) is a function of the possible messages, what counts as a possible message depends very much on our subjective positions—indeed, Ludlow (2019) argues that it depends upon our perspectival positions as well.

What this state of affairs means is that there are no hard physical facts by virtue of which psychological and physical states ground grammars (or, in this case, logics), but the enterprise is more a matter of the legibility of the actions of machines and computational (and logical systems), where legibility involves our being able to map between the world and the abstract structures (whether they be grammars or logics or abstract physical theories).

Now, of critical interest here (and one of the elephants in the room) is the fact that whatever grammar a person might have, they do not always act in

[2] See Ludlow (2019, chapter 5) for discussion.

compliance with the rules of that grammar. Indeed, at least as Chomsky envisions the linguistic project, the grammar is distinct from the speech production and speech comprehension system. That is to say, it is not merely that there are memory limitations and distractions and motor-malfunctions at play, but the grammar may be something quite distinct from the parser. One possibility explored in Ludlow (2011) is that the grammar is a kind of regulator that issues a warning (a feeling that something isn't quite right) when performance diverges from the target. In this sense an individual's grammar (what linguists would call a parametric state of the grammar) encodes individual norms for the language user.

This picture admittedly leads to a number of messy questions, not least of which is determining when some feature of human language use is a function of the grammar and when it is a function of something else altogether.

For example, one classic case involves center-embedding within a relative clause. One embedding is fine.

(1) The cat the dog chased is black.

But what happens when we try another level of embedding?

(2) #The mouse the cat the dog chased ate is brown.

It doesn't exactly roll off the tongue, and so the question is whether this is because of some processing difficulties that center-embedding causes or whether cases like this are ungrammatical for some grammar-internal reason. There is clearly no pre-theoretical way to determine this. We need to know more about processing or the grammar to know the answer.

Another case involves what appears to involve interference from prescriptive language rules. Some people really cannot abide dangling prepositions and split infinitives, but these attitudes appear to be the result of prescriptive language training rather than a function of the grammar itself. If Pinker (1994) is right we even know the etiology of these prescriptive rules, which was a desire to make "proper" use of English language conform to structures found in Latin.

There is a moral in this discussion of grammars that will be useful to us going forward. But first, perhaps we can help ourselves to some key ideas from linguistics (one would hope so given that Natural Logic operates over natural language structures!). Following the above ontology of linguistics, we want to distinguish the Natural Logic itself, which is a kind of abstract mathematical object, from the psy-logic and perhaps physio-logic by virtue of which a person has that Natural Logic. This relation is perhaps most easily seen by reflecting on the physio-logic and thinking of it on analogy with the relation between a schematic drawing of Boolean logic gates and the actual physical instantiation of the drawing in physical circuits and the representation of logical polarity by voltage changes.[3]

Now the moral: one cannot expect performance to track the Natural Logic that one has. Or to put it another way, all those terrible logic exams that you graded are not evidence against Natural Logic, but are rather evidence that performance has degraded for some reason, and it invites an investigation as to why.

What we need are models that are predictive of performance errors for Natural Logic. But, before we look for those models, we need to say one last thing about the ontology of Natural Logic, because so far we have said nothing about the nature of logical judgments, and their relationship to logical theory (in our case, to Natural Logic).

Clearly, one needs to say something about the nature of the data here, because it is going to drive any interesting discussion involving theory choice, and indeed this book is full of judgments about what sentence follows from what and what doesn't, where NPIs might be located, and on and on. Indeed, such judgments are presumably found in most interesting logical projects. Following advice from Williamson (2004) we should be careful to avoid calling these judgments "intuitions," because that descriptor suggests that we are in some sense reporting on our inner life. That kind of Cartesianism needs to be rejected both in the case of linguistics and in the case of logic. As Williamson diagnoses the talk of intuitions, it is an attempt to insulate the theory from error, but in doing so, it pushes the project into being a chapter in narrow (Cartesian) psychology, and this is precisely the sort of thing that Frege warned against as well.

The talk of judgments is helpful, because it opens the door to a distinction between our judgments and the subject matter or phenomenon about which we judge—in this case, the laws of Natural Logic. This distinction is in the spirit of Bogen and Woodward (1988) on their distinction between data and the phenomena for which the data is illuminating. Here, we need to understand phenomena not as mental phenomenal experience, but as features of the subject of inquiry. Following Bogen and Woodward, we thus take *phenomena* to be stable and replicable effects or processes that are potential objects of explanation and prediction for scientific theories. In this case the phenomena will include the domain of entailment-related facts.

To understand the role of logical judgments in constructing the theory of Natural Logic it will be useful if we can get clear on the relation between logical *theory*,

[3] There are interesting questions about whether the (linguistic) theory of grammar and Natural Logic are even different enterprises. We think that they are, not least because the rules of grammar are concerned with well-formed structures and the rules of Natural Logic are concerned with the entailment relations holding between those structures. It is of course possible that the grammar is optimized for generating languages that are optimal for expressing logical relations, but this is certainly not obvious. For now we remain agnostic on the question.

logical *phenomena* or *facts*, and logical *data*. Following Bogen and Woodward we can illustrate the relation between theory, phenomena, and data as follows:

(3) a. *Theory* explains/predicts *Phenomena*.
b. *Phenomena* are evidence for *Theory*.
c. *Data* are evidence for *Phenomena*.

In our case, *Theory* is the theory of Natural Logic in its application, which means we are concerned not just with the formulas and inference rules, but with the deductive strategies and heuristics that are executed by human agents—agents that are susceptible to memory limitations, etc. We could call it the theory of Natural Logic+, because we are interested in the theory of how Natural Logic gets (mis)deployed given human processing limits and attention spans.

Some facts will provide evidence for the theory of Natural Logic+. The theory *contributes to* the explanation and prediction of these facts. We can also say that we have knowledge of some of the phenomena (that is, we have knowledge of entailment facts). All of this will be useful in the next section in which we begin to explore the theory of Natural Logic in the context of deductive strategies and heuristics, and in which data (judgments of entailment relations) will be called upon to illuminate aspects of this overarching picture of Natural Logic+.

13.1.2 Modeling logical performance

We've made a distinction between logical competence and logical performance, which helps clarify why apparent breakdowns in logical reasoning should not be too troubling, but what makes the distinction really interesting is that we can use it to build models that help predict and explain those apparent breakdowns.

There are a lot of moving parts here. The story in its entirety would consist of our account of Natural Logic, plus a theory of deductive heuristics, plus whatever processing limitations might be in play, plus pragmatic considerations that override rules that a human might otherwise follow. The total package will be complex, and is really more neatly situated in a disciplinary research program than a book section. There is suggestive work in this area, however, and we want to highlight one well-known example—from Rips (1994)—to illustrate how one might proceed. Rips' principal idea is that you can purchase a lot of insight into human logical performance by constructing models that reflect the heuristics that people naturally deploy in logical reasoning. The results aren't perfect, but they do seem to illuminate the source of apparent breakdowns in reasoning.

The driving point here, and one that we alluded to earlier, is that one cannot proceed with blind application of logical rules but that at some point they must be goal driven. We touched on the example of Disjunction Introduction, which is pretty clearly useless without some attention to the conclusion one is trying to derive. One strategy—indeed, one of the very earliest strategies in AI—was the suggestion by Newell et al. (1957) that we reason backwards from the conclusion.

Rips' model, called PSYCOP, is more of a hybrid mix of forward reasoning and backwards reasoning heuristics. The idea is that some rules, like traditional Modus Ponens, are not particularly cost expensive, in that they produce a limited number of outputs (they don't apply to their own output like Disjunction Introduction does). Some rules, like Conjunction Introduction, work best in a backwards direction (as in the disjunction case, you can combine premises with 'and' all day long, but there is no point in doing it unless you have a good reason). So, the idea is to have some forward chaining rules and some backwards chaining rules as in Tables 13.1 and 13.2.

The model also assumes short term memory limitations. Now, in some respects that is a pretty simple model, but it does an excellent job of failing exactly where it is supposed to, as the data in Table 13.3 indicate.

There are a lot of interesting takeaways from these results, but one of the most interesting, as highlighted by Table 13.4 from Rips, is the distribution of rule use in the model, where the forward and backward versions of the rules are combined.

Table 13.1 Forward and backward rules in PSYCOP

Forward ⇒ elimination	
$P \Rightarrow Q$	(a) If a sentence of the form $P \Rightarrow Q$ holds in some
P	domain D,
$\therefore Q$	(b) and P holds in D,
	(c) and Q does not yet hold in D,
	(d) then add Q to D.
Forward De Morgan (¬ over ∧)	
$\neg(P \wedge Q)$	(a) If a sentence of the form $\neg(P \wedge Q)$ holds in some domain D,
$\therefore \neg P \vee \neg Q$	(b) and $\neg P \vee \neg Q$ does not yet hold in D,
	(c) then add $\neg P \vee \neg Q$ to D.
Forward Disjunctive Syllogism	
$P \vee Q$	(a) If a sentence of the form $P \vee Q$ holds in some
$\neg Q$	domain D,
$\therefore P$	(b) then if $\neg P$ holds in D and Q does not yet hold in D,
	(c) then add Q to D.
	(d) Else, if $\neg Q$ holds in D and P does not yet hold in D,
	(e) then add P do D.
Forward Disjunctive Modus Ponens	
$P \vee Q \Rightarrow R$	(a) If a sentence of the form $P \vee Q \Rightarrow R$ holds in some domain D,
P	(b) and P or Q also holds in D,
$\therefore R$	(c) and R does not yet hold in D,
	(d) then add R do D.

Table 13.1 *Continued*

Forward ∧ Elimination	
$P \wedge Q$ $\therefore$ P	(a) If a sentence of the form $P \wedge Q$ holds in some domain D, (b) then if P does not yet hold in D, (c) then add P to D, (d) and if Q does not yet hold in D, (e) then add Q to D.
Backward ⇒ Elimination	
$P \Rightarrow Q$ P $\therefore Q$	(a) Set D to domain of current goal. (b) Set Q to current goal. (c) If the sentence $P \Rightarrow Q$ holds in D, (d) and D does not yet contain subgoal P, (e) then add P to the list of subgoals.
Backward De Morgan (¬ over ∧)	
$\neg(P \wedge Q)$ $\therefore \neg P \vee \neg Q$	(a) Set D to domain of current goal. (b) If current goal is of the form $\neg P \vee \neg Q$, (c) and $\neg(P \wedge Q)$ is a subformula of a sentence that holds in D, (d) then add $\neg(P \wedge Q)$ to the list of subgoals.
Backward Disjunctive Syllogism	
$P \vee Q$ $\neg Q$ $\therefore P$	(a) Set D to domain of current goal. (b) Set Q to current goal. (c) If a sentence of the form $P \vee Q$ or $Q \vee P$ holds in D, (d) and $\neg P$ is a subformula of a sentence that holds in D, (e) and D does not yet contain the subgoal $\neg P$, (f) then add $\neg P$ to the list of subgoals.

Table 13.2 Additional backward rules in PSYCOP

Backward Disjunctive Modus Ponens	
$P \vee Q \Rightarrow R$ P $\therefore R$	(a) Set D to domain of current goal. (b) Set R to current goal. (c) If the sentence $P \vee Q \Rightarrow R$ holds in D, (d) and D does not yet contain the subgoal P, (e) then add P to the list of subgoals. (f) If the subgoal in (e) fails, (g) and D does not yet contain the subgoal Q, (h) then add Q to the list of subgoals.
Backward ∧ Elimination	
$P \wedge Q$ $\therefore P$	(a) Set D to domain of current goal. (b) Set P to current goal. (c) If the sentence $P \wedge Q$ subformula of a sentence that holds in D, (d) and D does not yet contain the subgoal $P \wedge Q$, (e) add $P \wedge Q$ to the list of subgoals. (f) If the subgoal in (e) fails, (g) and the sentence $Q \wedge P$ is a subformula of a sentence that holds in D, (h) and D does not yet contain the subgoal $Q \wedge P$, (i) then add $Q \wedge P$ to the list of subgoals.

Table 13.2 *Continued*

Backward ∧ Introduction	
P	(a) Set D to domain of current goal.
Q	(b) If current goal is of the form $P \wedge Q$,
$\therefore P \wedge Q$	(c) and D does not yet contain the subgoal P,
	(d) then add the subgoal of proving P in D to the list of subgoals.
	(e) If the subgoal in (d) succeeds,
	(f) and D does not yet contain the subgoal Q,
	(g) then add the subgoal of proving Q in D to the list of subgoals.
Backward ∨ Introduction	
P	(a) Set D to domain of current goal.
$\therefore P \vee Q$	(b) If current goal is of the form $P \vee Q$,
	(c) and D does not yet contain the subgoal P,
	(d) then add the subgoal of proving P in D to the list of subgoals.
	(e) If the subgoal in (d) fails,
	(f) and D does not yet contain the subgoal Q,
	(g) then add the subgoal of proving Q in D to the list of subgoals.
Backward ¬ Introduction	
$+P$	(a) Set D to domain of current goal.
...	(b) If current goal is of the form $\neg P$,
...	(c) and P is a subformula of the premises or conclusion,
$Q \wedge \neg Q$	(d) and Q is an atomic subformula of the premises or conclusion,
$\therefore \neg P$	(e) and neither D nor its superdomains nor its immediate subdomain contain suppositions P and subgoal $Q \wedge \neg Q$,
	(f) then set up a subdomain of D, D', with supposition P,
	(g) and add the subgoal of proving $Q \wedge \neg Q$ in D' to the list of subgoals.

The aversion to Disjunction Introduction has already been noted, and it has been fairly widely discussed in the literature, although it remains a bit of an open question as to why the rule should be problematic in a way that Conjunction Introduction is not (given that both rules can forward chain to infinity). Gazdar (1979), McCawley (1981), and Grice (1989) for example have suggested that there is a pragmatic constraint on the rule.[4]

It is certainly premature to make bold claims about why Disjunction Introduction and Negation Introduction (found at the bottom of Table 13.4) are so seldom deployed, but the structure of our inline Dynamic Deductive System suggests some possible answers.

With respect to Disjunction Introduction (Add), it is completely alien from the perspective of our DDS. All the other rules operate on existing inline material and those operations are basically limited to our rule Copy and Prune, and operate under the control of p-scope. They operate on material that is already on the table. Add is different in that it adds new material to the inline deductive

[4] See Rips and Conrad (1983) for further disussion of Disjunction Introduction.

Table 13.3 Percentage of correct responses from experiment, and predictions from the PSYCOP model

	Argument	Obs.	Pred.
A	$(p \vee q) \wedge \neg p$ $\therefore q \vee r$	33.3	33.3
B	s $p \vee q$ $\therefore \neg p \Rightarrow (q \wedge s)$	66.7	70.2
C	$p \Rightarrow \neg(q \wedge r)$ $(\neg q \vee \neg r) \Rightarrow \neg p$ $\therefore \neg p$	16.7	32.4
D	$\neg\neg p$ $\neg(p \wedge q)$ $\therefore \neg q \vee r$	22.2	30.6
E	$(p \vee r) \Rightarrow q$ $\therefore (p \vee q) \Rightarrow q$	83.3	79.9
F	$\neg p \wedge q$ $\therefore q \wedge \neg(p \wedge r)$	41.7	40.5
G	$(p \vee q) \Rightarrow \neg r$ $r \vee s$ $\therefore p \Rightarrow s$	61.1	70.2
H	$(p \Rightarrow q) \wedge (p \wedge r)$ $\therefore q \wedge r$	80.6	76.6
I	$(p \vee q) \Rightarrow \neg s$ s $\therefore \neg p \wedge s$	55.6	41.2
J	q $\therefore p \Rightarrow ((p \wedge q) \vee r)$	33.3	36.0
K	$(p \vee \neg q) \Rightarrow \neg p$ $p \vee \neg q$ $\therefore \neg(q \wedge r)$	22.2	35.6
L	$(p \vee q) \Rightarrow \neg(r \wedge s)$ $\therefore p \Rightarrow (\neg r \vee \neg s)$	75.0	70.4
M	$\neg p$ q $\therefore \neg(p \wedge r) \wedge (q \vee s)$	22.2	26.4
N	$(p \vee r) \Rightarrow \neg s$ $\therefore p \Rightarrow \neg(s \wedge t)$	50.0	38.1
O	$\neg(p \wedge q)$ $(\neg p \vee \neg q) \Rightarrow r$ $\therefore \neg(p \wedge q) \wedge r$	77.8	75.8
P	$(q \vee r) \wedge s$ $\therefore \neg q \Rightarrow r$	69.4	68.5
Q	p $(p \vee q) \Rightarrow \neg r$ $\therefore p \wedge \neg(r \wedge s)$	33.3	40.5
R	$p \Rightarrow r$ $\therefore (p \wedge q) \Rightarrow r$	58.3	69.1
S	s $p \Rightarrow r$ $\therefore p \Rightarrow (r \wedge s)$	75.0	70.9
T	$p \vee q$ $\therefore \neg p \Rightarrow (q \vee r)$	33.3	32.2
U	p $(p \vee q) \Rightarrow r$ $r \Rightarrow s$ $\therefore s \vee t$	38.9	33.9
V	$p \wedge q$ $\therefore q \wedge (p \vee r)$	47.2	37.6
W	$\neg(p \wedge q)$ $(\neg p \vee \neg q) \Rightarrow \neg r$ $\therefore \neg(r \wedge s)$	23.0	35.5
X	$(p \vee s) \Rightarrow r$ s $\therefore \neg(r \Rightarrow \neg s)$	50.0	36.1
Y	$p \Rightarrow \neg q$ $\therefore p \Rightarrow \neg(q \wedge r)$	36.1	33.9
Z	$\neg(p \wedge q) \wedge r$ $(\neg p \vee \neg q) \Rightarrow s$ $\therefore s$	66.7	73.9
A'	$(p \vee q) \Rightarrow (r \wedge s)$ $\therefore p \Rightarrow r$	91.7	86.9
B'	$\neg r$ $q \vee r$ $\therefore r \Rightarrow \neg\neg q$	36.1	38.7
C'	$\neg(p \wedge q)$ $\neg\neg q$ $\therefore \neg p \wedge \neg(p \wedge q)$	72.2	62.2
D'	$(p \vee q) \wedge ((r \vee s) \Rightarrow \neg p)$ r $\therefore q$	83.3	75.8
E'	$p \vee s$ $(p \vee r) \Rightarrow s$ $\therefore s \vee t$	26.1	36.0
F'	t $\neg(r \wedge s)$ $\therefore ((\neg r \vee \neg s) \wedge t) \vee u$	36.1	33.8

Table 13.4 Parameters for rule use in PSYCOP

Rule	**Estimate**
Disjunctive Modus Ponens	1.000
$\wedge$ Introduction	1.000
$\wedge$ Elimination	0.963
$\Rightarrow$ Introduction	0.861
$\vee$ Elimination	0.858
$\Rightarrow$ Elimination	0.723
De Morgan ($\neg$ over $\wedge$)	0.715
Disjunctive Syllogism	0.713
$\neg$ Introduction	0.238
$\vee$ Introduction	0.197

process. It is pretty clear why it should be a process of last resort, as unguided and unmotivated addition can continue without end. Furthermore, such a rule seems to be in conflict with a property like conservativity, which, recall, constrains the aboutness of a sentence to material that is there in the sentence. There is a sense, perhaps metaphorical, in which a logical rule that introduces new material is itself non-conservative, and thus understandably a rule of last resort given the general principles we have been operating with.

The situation with Negation Introduction (*reductio*) is different. Here, we are looking at an inline rule that is supposed to take us from $A \Rightarrow B$ and $A \Rightarrow \neg B$ to $\neg A$. This might be a straightforward rule in the abstract, but in the context of our DDS it is complex, and in a way, somewhat "expensive" within our system. To see this, consider (4a) and (4b) below:

(4) a. If it rains, it pours. $R \Rightarrow P$
b. If it rains, it does not pour. $R \Rightarrow \neg P$

If these are combined into a single formula, the deduction of $\neg R$ proceeds as follows:

(5) 1. $\underbrace{(\neg R \vee P)}_{+} \wedge (\neg R \vee \underbrace{\neg P}_{\substack{+\\ \text{COPY}}})$

2. $(\neg R \vee P) \wedge (\neg \underset{+}{R} \vee (\neg P \wedge \underbrace{(\neg \underset{-}{R} \vee P)}_{\substack{+\\ \text{PRUNE}}}))$

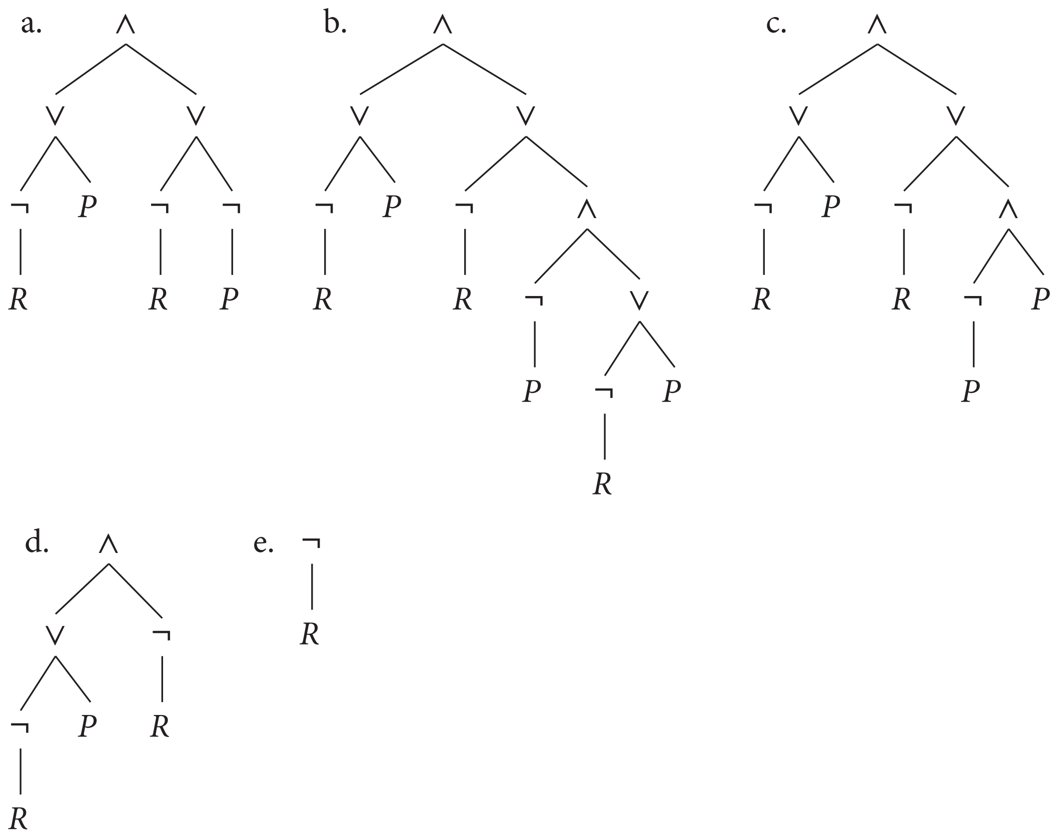
3. $(\neg R \vee P) \wedge \underbrace{(\neg R \vee (\neg \underset{\mp}{P} \wedge \underset{+}{P}))}_{\text{PRUNE}}$

4. $\underbrace{(\neg R \vee P)}_{\substack{+\\ \text{DELETE}}} \wedge \neg R$

5. $\neg R$

We can also view this deduction in tree form as below, where the trees in (6) replace the steps of (5).

(6)



One take on this rule is that it is inferentially expensive within DDS. Clearly, we need to be cautious here, because we ultimately need to formalize a theory-internal complexity measure, but it is clear that this can be done in terms of either steps or some computation of the structures that need to be built, moved, pruned, etc.

We should make it clear that our language L** is not tied to a single deductive system, and our DDS is certainly not the only possiblity. Our point in this section is that DDS (and presumably its competitors) can play a role in a model of inferential success and failures when it is applied to creatures with finite memories and other cognitive limitations. Our message is thus that performance failures in logic are not any sort of evidence against Natural Logic, but rather clues into (among other things) the deductive methods that people employ. L** and DDS can play an important part in any new systematic investigation into logical competence and performance.

13.2 Foundational Questions about Polarity and Directional Entailingness

Throughout this book, we have been exploring the traditional insight into the role of polarity in natural language. We hope the resiliance of the insight comes as surprising and possibly even shocking to you. It truly is a remarkable insight, and it raises the question of why this should be. Why should all of Natural Logic (and perhaps logic itself) have polarity-sensitivity running through the entire system as it does?

You might think that this question rests on a false assumption, since there are lots of ways to ground logic, and none of them are in any interesting sense special. For example, in the preceding chapters it was critical that we understand the implication $A \Rightarrow B$ as a disjunction $\neg A \vee B$, revealing the polarity of A as negative (downward entailing). But there is no straight logical reason to think that negation and disjunction must be primitives. To illustrate this concern, let's restrict our attention to basic Boolean operations.

The set $\{\neg, \vee\}$ is certainly a *functionally complete (FC)* set of connectives, meaning that with just those operators you can express all of Boolean logic. But it is not the only set of operators that is FC, e.g. $\{\neg, \wedge\}$ and $\{\neg, \Rightarrow\}$ are FC as well. Notably, FC sets without any negation at all exist. There are two of them that computer scientists particularly like: the singleton sets $\{\uparrow\}$ and $\{\downarrow\}$.

$\uparrow$ "nand" and $\downarrow$ "nor" can be understood using familiar connectives, as in (7), which also show that $\uparrow$ and $\downarrow$ are *polarity flippers*.[5] Note, however, that this definition is just for your convenience. One can define them simply by a truth table (8). And yes, you can implement $\uparrow$ and $\downarrow$ gates, and this is actually the usual case, and due to the FC property of $\{\uparrow\}/\{\downarrow\}$, entire circuits can (and are) implemented using only $\uparrow/\downarrow$. By the way, $\neg A \sim A \uparrow A \sim A \downarrow A$. Given this, why imagine that there is anything deep or special about polarity?

(7) a. $A \uparrow B :\equiv \neg(A \wedge B)$
    b. $A \downarrow B :\equiv \neg(A \vee B)$

(8)

A	B	$A \uparrow B$	$A \downarrow B$
F	F	T	T
F	T	T	F
T	F	T	F
T	T	F	F

[5] In this context a polarity flipper is something that can change the polarity of a formula from true to false (and vice versa). More generally we can say that in a binary system where one state represents true and the other represents false, a polarity flipper transitions from one of the binary states into the other. So, for example, we will look at the example of an electronic circuit. In this case, the binary states are high voltage (representing true) and low voltage (representing false), and a polarity flipper is that which transitions from high (low) voltage into low (high) voltage.

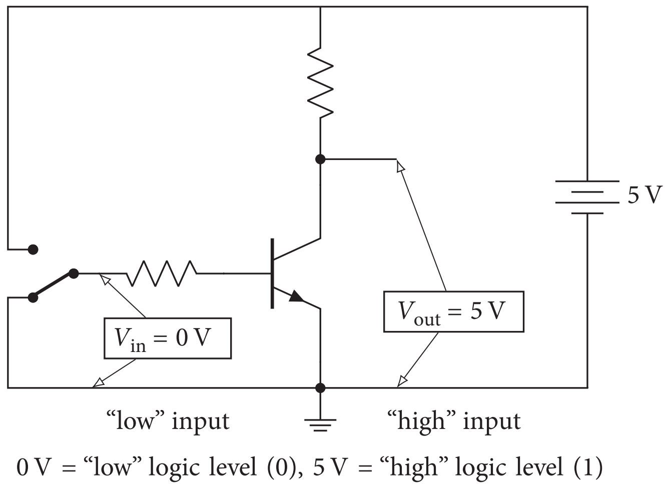


Figure 13.1 A state flipper (transistor in cutoff)

But let's take a closer look. Suppose we built a circuit for $A \Rightarrow B$ using $\downarrow$. One way to do this is shown in (9). However—and here is the big point—since $\downarrow$ is a polarity flipper, all occurrences of A are in the scope of an odd number of polarity flippers and all occurrences of B are in the scope of an even number of polarity flippers, so the polarity is, as usual, available upon inspection.

(9) $((A \downarrow A) \downarrow B) \downarrow ((A \downarrow A) \downarrow B)$

You might respond to this by claiming that we could simply design a primitive conditional logic gate. In doing so, we would not have any reason to posit or to suppose that there is negation at play in the antecedent of the conditional. But this idea is mistaken.

It is not accidental that we have a strong preference for starting with basic Boolean operators like negation and disjunction rather than a conditional, and it is not accidental that were we to build an electronic circuit for a conditional we would likewise "see" a polarity flipper in it.

What does the polarity flipper come to at the level of implementation? Well it amounts to something that can, for example, change a voltage state from low to high (and vice versa), for example from closer to 0 V to closer to 5 V. Take a look at Figure 13.1 for a schematic which shows how such a state flipper works.

A logic gate for a conditional would have to include such a voltage flipper, for if the consequent has negative polarity, it must flip the antecedent to negative polarity as well. This is not just a point about electronic circuits. The same point would hold if we made the logic gates out of DNA as in Stojanovic et al. (2002). The system would need a state flipper.

Our point is that any implementation of a conditional logic gate will naturally have these subcomponents and will likewise have an element (representing the antecedent) which has a negative polarity. But isn't there a possible objection here? Let's say we build a circuit for $\Rightarrow$ using $\downarrow$ in the usual way. We say, "see, the polarity flips three times for the antecedent because the voltage gets flipped three times." But suppose an alien shows up from planet $\Rightarrow$, and this alien looks at our circuit and says "I don't know what you are talking about—I just see a transistor that instantiates $\Rightarrow$." So then we say, "can't you see the voltage flips? Those are implementing syntactic operations." But the alien says, "no, I don't see anything happening—just a lot of buzzing confusion including some voltage changes going on inside your transistor, which obviously you designed to be a primitive implementation of $\Rightarrow$." So then we say "this is some confusion, can you bring me some other examples of what you consider primitive $\Rightarrow$ transistors?" and sure enough the alien shows up with a hundred of them, but every one of them has a Boolean structure that we can see and we clearly see the polarity (voltage) flipping happening in them. But the alien insists that nothing is going on in those circuits that represents polarity flips—they are just primitive implementations of $\Rightarrow$.

What can we say about this objection? One thought would be that such a creature would not be possible, but we don't know what an argument for that would look like. Another idea would be that this particular natural logic only works for human psychology and not for alien psychologies. Alien psychologies might yield logics that are functionally equivalent to ours but which bottom out in other primitives. But there is another possible response, one that stems from a certain understanding about the theory of information as Shannon and Weaver (1949) developed it.

The alternative view is that information states are a function of their legibility to us. So that there is no such thing as information flow happening in the absence of our ability to interpret that information. The same holds for computation. (See Ludlow 2019 for a discussion of this idea.) The computational state of a system is a function of the legibility of the inputs, outputs, and internal syntactic states *to us*. Thus, the Putnam (1988) argument about a wall being able to implement word processing software is false because such a system would not be legible to us. But what about the alien? Well you might say that the alien can see different information states than we can, but we're inclined to be chauvinistic about this. It only counts as information to creatures like us. Maybe the aliens have their own scientists and their own Shannon (named Schmannon) and their own Bolzman (named Schmolzman) but they weren't studying "information," they were studying "schminformation." You might think that sounds like a psychological theory, since after all we are talking about legibility and isn't that part of psychology? And maybe, but we don't have an existing psychological theory that explains what is going on. We are instead leaning on a point about the subjectivity of information. Here is how Gallistel and King (2010) put the idea (using the midnight ride of Paul Revere as their example).

> Shannon defined the amount of information *communicated* to be the difference between the receiver's uncertainty before the communication and the receiver's uncertainty after it. Thus, the amount of information that Paul got when he saw the lights depends not only on his knowing beforehand the two possibilities (knowing the set of possible messages) but also on his prior assessment of the probability of each possibility. This is an absolutely critical point about communicated information—and the subjectivity that it implies is deeply unsettling. By subjectivity, we mean that the information communicated by a signal depends on the receiver's (the subject's) prior knowledge of the possibilities and their probabilities. Thus, the amount of information actually communicated is not an objective property of the signal from which the subject obtained it!
>
> (Gallistel and King 2010, p. 9)

How does this apply to questions about the logical structure of a circuit instantiated by a machine? Well, states of a machine are information states (in Shannon and Weaver's sense of information). And those states are a function of the information they carry. And the information that they carry is a subjective property—it is a relational property that depends upon us, and the legibility of the information state to us.

Where does this leave us? It might be possible to push this discussion still deeper, but as it stands, it seems that there is a very deep level of analysis at which Natural Logic must fundamentally be a polarity logic—one in which all logical operations are fundamentally polarity-sensitive. That is the level of analysis in which we ground our basic understanding of the nature of information and computation. If the discussion can be pushed deeper than that, we don't know how.

13.3 Is Proof Theory the New Semantics (and Theory of Meaning)?

Earlier, we quoted a passage from Moss (2005) in which he suggested that proof theory might well supercede semantics. The passage is worth repeating here.

> If one is seriously interested in entailment, why not study it axiomatically instead of building models? In particular, if one has a complete proof system, why not declare it to be the semantics? Indeed, why should semantics be founded on model theory rather than proof theory?

It is certainly a heady idea, and it can even be taken further, for as we noted, it has been suggested that proof theory might, at the end of the day, yield a theory of word meaning—certainly for connectives, but perhaps for word meaning *tout court*. This idea perhaps has its origins in a suggestion made in Gentzen (1934) in the context of his discussion of his natural deduction system. Here is the relevant passage.

> The introductions constitute, as it were, the 'definitions' of the symbols concerned, and the eliminations are, in the final analysis, only consequences of this, which may be expressed something like this: At the elimination of a symbol, the formula with whose outermost symbol we are dealing may be used only 'in respect of what it means according to the introduction of that symbol.'

So, what is Gentzen's idea here? Recall that an introduction rule is a rule that introduces a connective. Conjunction introduction would be a case in point.

(10) $\dfrac{A \quad B}{A \wedge B} \wedge_I$

If the truth of a proposition can be established in more than one way, then the corresponding connective has multiple introduction rules.

(11) a. $\dfrac{A}{A \vee B} \vee_{I_1}$ b. $\dfrac{B}{A \vee B} \vee_{I_2}$

So the headline idea is that for connectives, those introduction rules, introduced in order to get inferential patterns right, are effectively the word meanings for these expressions.

One can ask whether there isn't something idiosyncratic about Gentzen's choice of introduction rules to characterize meaning. Why not elimination rules? Or some combination? There is certainly flexibility here, and one idea is that introduction rules are not the route to the question of meaning, but rather that the inferential roles of the terms are key. As we noted in Chapter 3, this idea has been advanced by Dummett (1991) and Brandom (1998),

The interesting question is whether this Brandom-Dummett idea (or something in its neighborhood) can be executed. Let's give an illustration of the idea, using expressions that should be familiar to us by now.

In Chapter 3, we expressed reservations about certain polarity marking approaches because they were not driven by the desire to get the truth conditions correct. They were just there to get the inferences to fall out. But now suppose we responded like this: Exactly! We don't need truth conditions; we are only in the business of getting the inferential relations to work out, and to do that we can start with inferences that appear valid, and which are directional entailing, and then we assign polarity markings accordingly. And voilà! We deduce where the polarity markings should go, and if you want to work in L** we can deduce whether there should be a disjunction or a conjunction as part of the determiner definitions. Truth conditions are not needed.

But how does all this work in the case of word meanings beyond the connectives? Well, rather than start with word meanings and build out the inferential relations, you assume some knowledge of what the inferential relations are, and then deduce some things about word meanings. There are a number of ways this idea could play out, but let's stick with polarity. We have frequently spoken in this book of substitutions that can apply in upward entailing environments and substitutions that can apply in downward entailing environments. Suppose you used these substitutions to figure out things like 'dog' $<$ 'bark' and 'dog' $<$ 'animal', and think of those as being meaning rules—two sets of meaning rules actually, one for positive polarity environments and one set for negative polarity environments. Once you assemble all those meaning rules for 'dog' you in effect have the meaning of 'dog'.

We can't dispute that this should work in theory. If you claim you have a successful referential semantics for a language and that the inferential relations then fall out naturally, then it must surely be possible to run things in reverse: if you have the inferential relations figured out, then you can deduce something very much like those truth conditions.

Philosophers can go back and forth on which is the more sensible place to start—top-down with the inference patterns, or bottom-up with a referential semantics for word meanings. One might say that it is crazy to begin with all those inferential relations—there are just too many! And alternatively it can be argued that stating truth conditions is not the simple enterprise that referential semanticists imagine it to be.

It is also true that proof theory can, theoretically, carry out the work of model-theoretic semantics, in that we take both accounts to be theories about inferential relations in natural language, and if we are right, the systems should run in parallel. However from the perspective of theoretical discovery, doing proof theory without having the model theory for guidance is simply flying blind.

We've seen several examples in this book of where the syntax and proof theory have had to take the lead from model-theoretic semantics. For example we moved from accounts of conservativity in the model theory of determiners, to a syntactic account of restrictedness. Similarly, we looked at semantic properties like anti-additivity and then looked for their syntactic counterparts. This isn't accidental. Semantics is easier. That is no secret. So, perhaps the best one can hope for is the opportunity to kick away the ladder of model theory when our proof theory is finished. Good luck with that.

It is sometimes supposed that model-theoretic semantics also has the task of establishing truth conditions in its portfolio, but one needs to be careful here. One thing that model-theoretic semantics cannot do is provide truth conditions in the sense of absolute truth conditions for the expressions in a language. That is to say, model-theoretic semantics does not get one from the expression 'dog' to actual dogs. One needs to specify an "intended model" that specifically identifies the denotations of terms, or alternatively one needs an absolute truth-theoretic semantics in the spirit of Higginbotham (1985) to anchor the meanings of the expressions in the world.

Now, there are not a few linguists (most notably Chomsky) who are not fans of truth-conditional semantics—see the exchange between Ludlow (2008) and Chomsky (2008) and Ludlow's (2011) interview with Chomsky for background. And for sure, if Chomsky is right about this, then there is really no reason to think that a proof-theoretic account could not do everything one expects a semantic theory to do. Whether that in itself is a case against model-theoretic semantics is a different question altogether.

However, it should be noted that since the 1970s philosophers (notably Putnam 1975; Kripke 1980) have built a strong case that the business of semantics is not merely to get the truth conditions right, but also to provide anchoring conditions in the world. Sometimes the view is characterized as externalism, and the thought is that proof theory is an internalist enterprise and can't possiby provide the necessary anchoring conditions.

To see this, consider the famous example from Putnam (1975) in which two perfect duplicates use the word 'water', but one is in the land of H_2O and the other is in the land of XYZ (which is phenomenologically like water). The idea is that if the two twins are molecule-for-molecule duplicates, then they must be in the same syntactic states. We might add then, that whatever natural deduction they are engaged in, it must be the same proof, since the proof supervenes on whatever syntactic states they are in. So, proof theory can't deliver meanings. It can at best deliver the inferential roles of terms, and, if you believe Putnam et al., that is not enough to deliver meaning.

It may seem that we are being uncharacteristially non-commital on this question, but it is hard to take a bold stand given that the syntax and semantics run in parallel and questions of what anchors what are notoriously difficult. Anti-referentialists like Chomsky will find the top-down approach to meaning preferable. Referentialists will find the bottom-up approach preferable. The only thing that we are sure of on this question is that the framework developed here is theory neutral on the question.

13.4 The Holy Grail

The title of this book alludes to there being a Holy Grail for natural logic. That Holy Grail was originally the Medieval project to reduce all of logic down to two polarity-sensitive rules. We have taken this project a bit further by suggesting that we can actually get by with four very basic sub-logical syntactic operations, guided by the polarity-sensitive relation p-scope.

Sometimes you know an idea is on the right track when the results carry over into related topics and questions. And we have seen that our use of p-scope has application in the syntactic notion of restrictedness, which cashes out the semantic notion of conservativity, which in turn cashes out our philosophical understanding of aboutness. This result in turn illuminated the deep relation between polarity and quantification, and showed how polarity was the thread running through our notions of deduction, directional entailingness, and quantification. More, we showed how the result had consequences for our understanding of discourse anaphora and even our understanding of the Minimalist Program in linguistics.

In the traditional story of the Grail, Perceval was not so much interested in acquiring the Grail as he was in a far more important question: What is the meaning of the Grail? What Perceval learned was that the meaning was the quest itself. Is there a similar lesson for us? We think so. The deep connections between polarity and inference, quantification and linguistic form are yet to be completely illuminated. This book, at the end of the day, is a map for that quest. And if the message of the Grail is the journey, that is good news for everyone. Because the journey has only begun, and everyone is invited to join.

APPENDIX

The Mathematical Foundations

This book introduced quite a few fairly technical concepts, from directional entailingness and polarity through p-scope and inline rules to conservativity and restrictedness. In the main text, we tried to evoke an intuitive understanding of what these concepts are and how they are related. That is more than enough to understand—in detail—what our proposal is all about. But a more mathematically inclined reader might want to see our major claims formulated with mathematical rigor, and of course to see those claims proven as well. If you are such a reader, then this Appendix is for you. And don't worry; while the Appendix might not be light reading, it does not approach anything resembling hard-core math—you certainly don't need a PhD in math to read it (we would know as neither of us holds one).

The first half of the Appendix is dedicated to our Dynamic Deductive System: in section A.1 we provide the formal definition of DDS, and in section A.2 we prove that it is sound (remember that we have already proved completeness in section 7.3). The second half of the Appendix is devoted to the syntactic characterization of the two major semantic properties discussed in the book: section A.3 relates directional entailingness and polarity, while section A.4 is about conservativity and restrictedness, and it also formulates our most general notion of restrictedness.

A.1 Formal Definition of Dynamic Deductive System

In the main text, Dynamic Deductive System was introduced in Chapters 6 and 7, and further investigated in Chapter 8. In those chapters, we wanted to give you an intuitive understanding of the system and some hands-on experience in deploying it in natural language analysis. This section aims to provide a rigorous mathematical definition of DDS. Besides explicating the DDS-related notions in full detail, the concepts introduced here will serve as a foundation for the proof of soundness in the following section.

We begin the section by defining some general concepts related to parse trees and polarity deployed in the definition of DDS (subsection A.1.1). We then turn to the definition of p-scope (subsection A.1.2). The rigorous definition of this relation will proceed through the introduction of a new central notion, the *p-scope path*, which will be given both a mathematical and an algorithmic definition. We conclude by presenting the general architecture of DDS, paying attention to all the details concerning the application of inline rules (subsection A.1.3).

A.1.1 Generic notions

We equate formulas with their parse trees. By NODE, we mean a particular location in the tree; nodes can be terminal or non-terminal, and non-terminal nodes can be unary or branching. By CONSTITUENT, we refer to the entire subtree rooted in some node. As nodes and constituents are in one-to-one correspondence, we are often lax in the distinguishing between them.

We use the relational terms PARENT, ANCESTOR, DESCENDANT, SISTER, etc., in their intuitive meanings. Being an ancestor, dominating and containing are used as synonyms. When we state that α is an ancestor (descendant, constituent) of β, we always permit the possibility that α equals β. Whenever we want to exclude this option, we say that α is a PROPER ancestor (descendant, constituent) of β. The IMMEDIATE ANCESTOR of a node is obviously its parent; the term IMMEDIATE ANCESTORS (note the plural) refers to any set of ancestors of a node such that no ancestor not belonging to this set is dominated by an ancestor in this set.

Definition 1. Let α and β be constituents of formula φ. A constituent of φ is a COMMON ANCESTOR of α and β iff it contains both α and β.

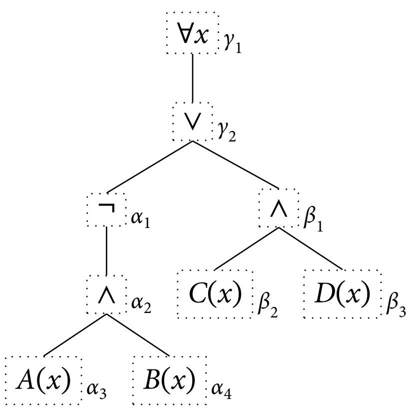
Definition 2. Let α and β be constituents of the same formula. The LOWEST COMMON ANCESTOR (LCA) of α and β, denoted by LCA(α, β), is the common ancestor of α and β which does not properly contain any common ancestor of α and β. (Obviously, any two nodes have a unique LCA.)

Definition 3. Constituents α and β are γ-RELATED (or, β is a γ-RELATIVE of α) iff γ is the LCA of α and β distinct from both α and β. (Obviously, if α and β are γ-related, neither of them dominates the other, and γ is either a conjunction or a disjunction.)

Definition 4. Let α and β be constituents of a formula. The UP-PATH from α to β is the set of proper ancestors of α dominated by the LCA of α and β.

Example. We illustrate the above concepts on (1). Node α_2, $\wedge$, corresponds to constituent rooted in α_2, of form $A(x) \wedge B(x)$. Nodes α_1 and γ_2 count as immediate ancestors of α_2, but α_1 and γ_1 do not. Common ancestors of α_i and β_j (for any i and j) are γ_1 and γ_2; their LCA is γ_2, so any α_i and β_j are γ_2-related. The up-path from α_3 to β_3 consists of nodes α_2, α_1, and γ_2.

(1)



Definition 5. Let $\wedge$, $\vee$, and $\neg$ be the only (non-zero-place) propositional connectives of the formal language; any other propositional connectives are defined as abbreviations, e.g. $A \Rightarrow B :\equiv \neg A \vee B$. A constituent α of formula φ has POSITIVE/NEGATIVE CONSTITUENT POLARITY in φ iff it lies within the scope of an even/odd number of negations in φ. (We often refer to constituent polarity simply as polarity.)

In symbols, we use α^π to express the statement that constituent α has polarity π. As polarity is a relative notion—a constituent has a certain polarity *within a given formula*—the notation only makes sense given the context, i.e. the formula φ that the polarity of α is relative to.

The set of polarities can be naturally thought of as a multiplication group $\{1, -1\}$, where 1 and -1 correspond to positive and negative polarity, respectively: if α has polarity π in β, and β has polarity ρ in γ, then α has polarity $\pi\rho$ in γ. Using this notation, we can easily refer to the opposite polarity of π by $-\pi$. Instead of referring to polarities by 1 and -1, we write $+$ for 1 and $-$ for -1.

Connectives often exhibit *dual* behavior with respect to polarity, e.g. $\wedge$ in a positive polarity environment has the same properties as $\vee$ in a negative polarity environment. We use polarity superscripts on connectives to easily to refer to members of the dual pairs, as shown below.

Definition 6 (Dual connectives).

Conjunction and disjunction:

(a) $\wedge^+ := \vee^- := \wedge$
(b) $\vee^+ := \wedge^- := \vee$

True and false:

(a) $\top^+ := \bot^- := \top$
(b) $\bot^+ := \top^- := \bot$

Universal and existential quantifier:

(a) $\forall^+ := \exists^- := \forall$
(b) $\exists^+ := \forall^- := \exists$

Absence and presence of negation (for any formula φ):

(a) $\neg^+\varphi :\equiv \varphi$
(b) $\neg^-\varphi :\equiv \neg\varphi$

We treat $\top$ and $\bot$ as 0-place connectives returning true and false, respectively. Negation stands out. Analogously to other dual pairs, negation is the dual of identity: $\mathrm{id}^+ := \neg^- := \mathrm{id}$ and $\neg^+ := \mathrm{id}^- := \neg$. However, as the identity connective is not a part of our formal language, we adopt a different perspective and define the absence vs. presence of negation as a "dual pair." The rationale behind the above definition is that the polarity of φ within $\neg^\tau\varphi$ is τ.

Given a set C of connectives not containing negation, let $C^\pi := \{c^\pi;\ c \in C\}$; clearly, $C^+ = C$.

A.1.2 Premise scope

The steps toward the definition of *premise scope (p-scope)* in this section are as follows. We start by defining the notion of *alternating path* (and its root), which we then use to define *p-scope pivot*. Together, these two auxiliary notions provide us the tools we need to define *p-scope path* and p-scope path polarity, which are central notions in the definition of p-scope itself. The latter definition comprises the definition of the three familar p-scope domains (descendant, relative, and ancestor p-scope) and the definition of p-scope polarity. We conclude the section by visually presenting the relations between the p-scope domains and their polarities and by providing an alternative, algorithmic definition of the central notion of this section, the p-scope path.

Definition 7. Let α be a constituent of formula φ. Let C be a set of logical connectives not containing $\neg$. A constituent γ is a member of the ALTERNATING C-PATH of α within φ, in symbols $\mathbb{A}^{\varphi}_{C}(\alpha)$, iff it is a proper ancestor of α and the following holds for every proper ancestor β of α contained in γ: β is either a negation, or

(a) if α has positive polarity in β, the connective of β is a member of C^+; and
(b) if α has negative polarity in β, the connective of β is a member of C^-.

Definition 8. The ALTERNATING C-PATH ROOT of α within φ is the highest node of the alternating C-path of α within φ, if the latter is non-empty, and α otherwise.

For brevity, we omit the phrase "within φ" and the superscript φ of $\mathbb{A}^{\varphi}_{C}(\alpha)$ whenever the context makes it clear which formula we are considering, and the braces in the subscript of $\mathbb{A}_C(\alpha)$ when listing the connectives explicitly, e.g. $\mathbb{A}_{\wedge,\forall}(\alpha) = \mathbb{A}_{\{\wedge,\forall\}}(\alpha)$.

Intuitively, an alternating C-path of α is the largest set of immediate ancestors of α whose type is among the allowed types (C) and such that the type of the ancestor's connective alternates with its polarity. The alternating path of α contains all the nodes between α and the alternating path root—this is why we call it a path—including the root (if distinct from α) but excluding α. An alternating path may be empty, which is always the case when α is the root of the formula.

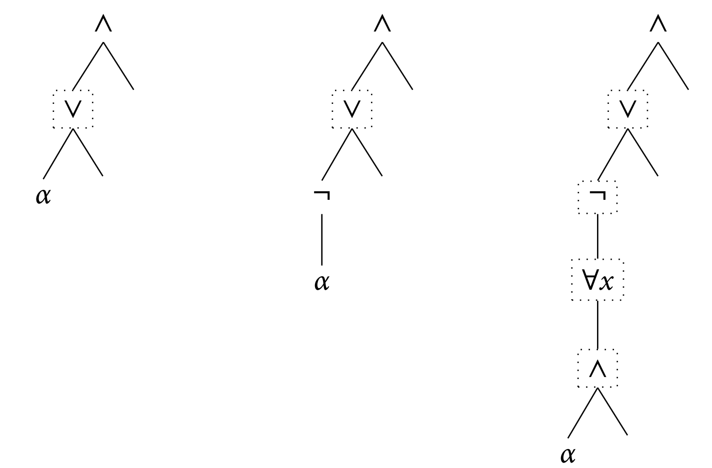
To compute the alternating C-path of node α, one can follow the following algorithm. Set the current node to α, the set of expected connectives to C, and the alternating path to an empty set. Then repeat the following instructions:

(a) if the parent of the current node is a negation or one of the expected connectives, add it to the alternating path;
(b) if the parent of the current node is a negation, replace the set of expected connectives by its dual, i.e. replace $\wedge$ by $\vee$ and $\forall$ by $\exists$, and vice versa;
(c) if the parent of the current node is a negation or one of the expected connectives, set the current node to the parent, and repeat; otherwise, finish.

The current node at the end of the procedure is the root of the alternating path.

Consider the trees in (2) for some examples. For each tree, the alternating path (for the set of connectives noted above the tree) consists precisely of the framed nodes. Also note that $\mathbb{A}_{\wedge,\forall}(\alpha) = \varnothing$ in (2a), $\mathbb{A}_{\vee,\exists}(\alpha) = \varnothing$ in (2b), and $\mathbb{A}_{\vee,\exists}(\alpha) = \varnothing$ in (2c).

(2) a. $\mathbb{A}_{\vee(\alpha)} = \mathbb{A}_{\vee,\exists}(\alpha)$ b. $\mathbb{A}_{\wedge(\alpha)} = \mathbb{A}_{\wedge,\forall}(\alpha)$ c. $\mathbb{A}_{\wedge,\forall}(\alpha)$



The alternating C-path of a node contains all the negations immediately dominating the node, regardless of the choice of C. In particular, when C is empty, the alternating path consists precisely of the immediately dominating negations. The p-scope pivot of α, defined below, is thus the highest negation immediately dominating α, or α itself, if the parent of α is not a negation.

Definition 9. Let α be a constituent of formula φ. The P-SCOPE PIVOT of α is the alternating $\varnothing$-path root of α.

In (2a) and (2c) above, the p-scope pivot of α is α itself; in (2b), the p-scope pivot is the negation.

Assume that the set of connectives C contains at most one member of each dual pair of connectives, i.e. if $c \in C$, then $c^- \notin C$. Then, C^+ and C^- have no common members, so either the alternating C-path or the alternating C^--path of any constituent consists solely of negations. If α' is the p-scope pivot of some constituent α, it is clearly never dominated by a negation, so at least one of $\mathbb{A}_C(\alpha')$ and $\mathbb{A}_{C^-}(\alpha')$ is empty. We use this fact in the definition of p-scope path below, with $C = \{\wedge, \forall\}$ (and $C^- = \{\vee, \exists\}$).

Definition 10. Let α be a constituent of formula φ, α' the p-scope pivot of α, and π the polarity of α within α'. The PREMISE SCOPE PATH (P-SCOPE PATH) of α:

(a) has polarity π and equals $\mathbb{A}_{\wedge,\forall}(\alpha')$, if the latter is non-empty;
(b) has polarity $-\pi$ and equals $\mathbb{A}_{\vee,\exists}(\alpha')$, if the latter is non-empty;
(c) is empty otherwise (with undefined polarity).

Definition 11. The P-SCOPE ROOT of α is the highest node of the p-scope path of α, if the latter is non-empty, and equals the p-scope pivot of α otherwise.

Intuitively, the p-scope path of a constituent α is an unbroken path from (and excluding) α up the tree which extends as far as the type of binary propositional connectives and quantifiers alternates with their polarity. Negations require special attention: while all the negations immediately dominating any node in the p-scope path are contained in the p-scope path as well, the negations immediately dominating α do not belong to its p-scope path.

Consider (2) again for some examples. The p-scope path of α in each tree consists exactly of the framed nodes; its polarity is negative in (2a) and positive in (2b) and (2c).

Definition 12. Let α be a constituent of formula φ, and π the polarity of α in φ. If the p-scope path of α is non-empty, let σ be its polarity. The PREMISE SCOPE (P-SCOPE) of α is the union of the descendant, relative, and ancestor p-scope defined below, and can be partitioned into POSITIVE AND NEGATIVE P-SCOPE (p-scope$^+$ and p-scope$^-$, respectively), so that the following holds:

(a) The DESCENDANT P-SCOPE of α equals the set of descendants of the p-scope pivot of α. It is a subset of p-scope$^\pi$.
(b) The RELATIVE P-SCOPE of α is the union of the sets of γ-relatives of α for every node γ in the p-scope path of α. It is a subset of p-scope$^\sigma$.
(c) The ANCESTOR P-SCOPE of α equals the p-scope path of α, if the latter is non-empty and $\pi = \sigma$, and is empty otherwise. It is a subset of both p-scope$^\pi$ and p-scope$^\sigma$.

If the (descendant/relative/ancestor) p-scope is a subset of the positive or negative p-scope, we say that it has positive or negative polarity, respectively. If a constituent t is in a (positive/negative) p-scope of constituent p, we say that p p-scopes (positively/negatively) over t.

If constituent p p-scopes over all nodes of some set with the same polarity, we say that the p-scope polarity of p over these nodes is uniform. Clearly, each of the p-scope domains (descendant, relative, and ancestor p-scope) has uniform polarity. If the ancestor p-scope is non-empty, the p-scope as a whole has uniform polarity as well, and equals the set of descendants of the p-scope root.

Consider the deduction of inference (3) in (4) for an example of a deduction involving all three p-scope domains. The application of iMP in (4.1) relies on two instances of p-scope. First, the target is in the positive relative p-scope of the (conditional) premise; second, the target (i.e. the antecedent premise) is in the positive descendant p-scope of itself. In the application of Copy in (4.2), the target is in the positive ancestor p-scope of the premise. The three relevant p-scope relations are visualized in (5).

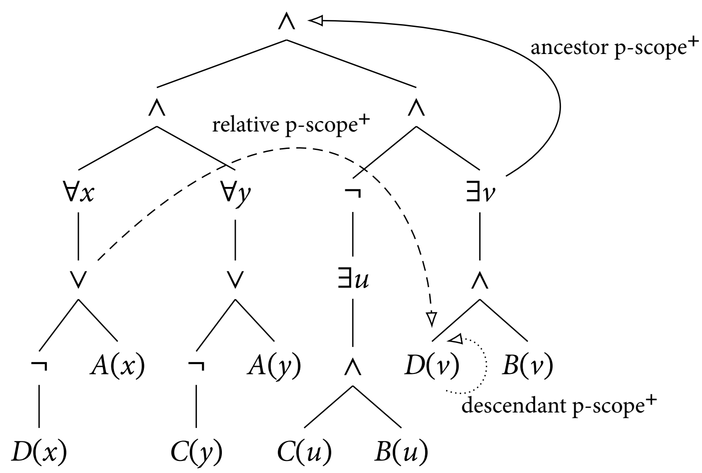
(3) a. Every dog is an animal and every cat is an animal.
b. No cat barks and some dog barks.
c. ∴ Some animal barks.

(4) 1. $((\forall x\colon \underbrace{D(x) \overset{}{\underset{+}{\Rightarrow}} A(x)}_{x\mapsto v}) \wedge (\forall y\colon C(y) \Rightarrow A(y)))\wedge$
$\wedge ((\neg\exists u\colon C(u) \wedge B(u)) \wedge (\exists v\colon \underbrace{D(v)}_{\substack{+\\ \text{IMP}}} \wedge B(v)))$

2. $((\forall x\colon D(x) \Rightarrow A(x)) \wedge (\forall y\colon C(y) \Rightarrow A(y)))\wedge$
$\underbrace{\wedge ((\neg\exists u\colon C(u) \wedge B(u)) \wedge \underline{(\exists v\colon \overline{A(v)} \wedge B(v))}_{+})}_{\substack{+\\ \text{COPY}}}$

3. $\overline{\exists v\colon A(v) \wedge B(v)}$

(5)



The main p-scope-related notions defined in this section are visualized in Figure A.1. In particular, this figure shows all three p-scope domains (descendant, relative, and ancestor p-scope) and highlights the relations between their polarities. The descendant p-scope polarity (π) equals the (global) constituent polarity of the premise. The relative p-scope polarity (σ) equals the p-scope path polarity—we'll say more about the latter below. If π and σ match, the ancestor p-scope equals the p-scope path, and the ancestor p-scope

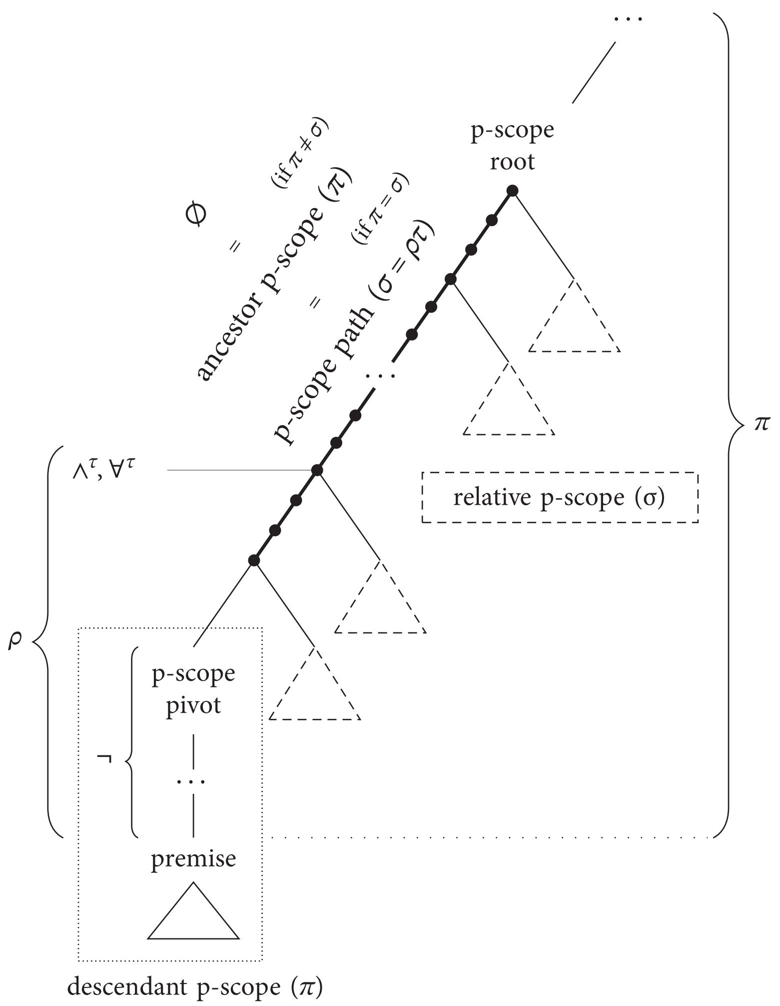


Figure A.1 Premise scope domains and their polarities

polarity equals the p-scope path polarity; if π and σ don't match, the ancestor p-scope is empty, and its polarity is undefined.

Figure A.1 also shows how to determine the polarity of the p-scope path (σ). Let's say constituent α is our premise. Take any constituent β belonging to the p-scope path of α which is not a negation. In the figure, the choice of β is marked by "$\wedge^\tau$, $\forall^\tau$": so if β is a conjunction or a universal quantifier, $\tau = +$, but if β is a disjunction or an existential quantifier, $\tau = -$. Now count the number of the negations between α and β to determine ρ, the polarity of α within β: if there is an even number of them, $\rho = +$, and if there is an odd number of them, $\rho = -$. As shown in the figure, the p-scope path polarity σ then equals the product of the polarities determined above: $\sigma = \rho\tau$. So in summary, to determine the p-scope path polarity of α, compute the polarity of α within any (non-negation) β on the p-scope path, and flip it if β is a disjunction or an existential quantifier.

There are two common choices of β from the previous paragraph. The first one is the LCA of the premise and the given target, and the second one is the parent of the p-scope pivot, i.e. the first non-negation node above α, as this is a node which always belongs to the p-scope path (if it exists; the only situation where it doesn't exist is when the p-scope pivot is the root of the formula).

Returning to the p-scope path, as this is a central concept in DDS, we provide an alternative definition for a reader familiar with computer algorithms. Algorithm 1 has three parts. In the first part (the first loop), we walk up the tree from the premise to the p-scope pivot, computing the polarity of the premise within the p-scope pivot along the way. Afterwards (between the loops), we look at the parent of the p-scope pivot. As explained above, this is the bottom-most node on the p-scope path, and the type of its connective ($\wedge$ or $\forall$ vs. $\vee$ or $\exists$), together with the polarity of the premise within the p-scope pivot computed earlier, determines the p-scope path polarity. In the final part (the second loop), we walk up to the p-scope root, adding nodes to the p-scope path along the way.

Algorithm 1 P-scope path and its polarity

```
function P-Scope Path(φ, α)                          ▷ where α is a constituent of formula φ
    σ ← +                                  ▷ The initial p-scope path polarity is positive.
    while α has a parent in φ do                           ▷ Walk up to the p-scope pivot.
        β ← Parent(α, φ)
        if Type(β) ≠ ¬ then                   ▷ Only negations up to the p-scope pivot.
            break
        end if
        σ ← −σ                                           ▷ Flip the p-scope path polarity.
        α ← β
    end while
    if α does not have a parent in φ then                ▷ If the p-scope pivot is the root:
        return ∅                      ▷ The p-scope is empty, the p-scope polarity undefined.
    end if
    α ← Parent(α, φ)                                ▷ The lowest node on the p-scope path.
    P ← {α}                                            ▷ Start building the p-scope path.
    π ← TypePolarity(α)                        ▷ The expected "polarity of the connective."
    σ ← σ · π                              ▷ The p-scope path polarity is determined now.
    while α has a parent in φ do                              ▷ Move up the p-scope path.
        β ← Parent(α, φ)
        if Type(β) = ¬ then                                ▷ A negation can occur anytime.
            π ← −π                                      ▷ Flip the expected connectives.
        else if π ≠ TypePolarity(β) then                              ▷ Wrong connective!
            break                                           ▷ We're out of p-scope path.
        end if
        α ← β
        P ← P ∪ {α}                        ▷ Add the node to the path (negations, too).
    end while
    return (P, σ)
end function
```

A.1.3 Inline derivation

This section aims to define two notions. The simpler of the two is the notion of *inline deduction*, which reflects the dynamic/linear nature of DDS, where a deduction is envisioned as a dynamic process which keeps no record of the history of the preceding steps. The other notion will serve as a general characterization of derivability in DDS. All the inline rules proposed in this book (and possibly all inline rules in general) can be thought of as (sequences of) *canonical inline rules*.[1] A canonical inline rule can be seen as a DDS wrapper around a classical derivation—we will call it a *core derivation*—which we wish to apply in an arbitrary syntactic environment. The wrapper precisely specifies the conditions which must obtain for "inlining" the core derivation, and will serve as a general setup in the proof of validity of inline rules in the following section.

In a Hilbert-style system, every premise of a rule application forms the entire line of proof, and so does the conclusion of the rule. Seeing this through the lens of DDS, we can say the premises and the conclusion are the *root* constituents of their lines of proof. We therefore call the rules of a classical system ROOT RULES, and the deductions, ROOT DEDUCTIONS. In DDS, root deductions play the role of the core derivations of canonical inline rules. To facilitate statement of these, we do not limit the root rules to the smallest possible set (Modus Ponens and Universal Generalization), but allow a core derivation to deploy any valid derived rule, in particular including Deduction Theorem and Existential Instantiation.

In contrast, rules and derivations of Dynamic Deductive System are called *inline rules* and *inline derivation*, as they apply arbitrarily deep within the dynamic formula. Starting with the formal definitions of these concepts, observe that the dynamic/linear character of DDS relies on inline rules being binary relations between *formulas*—in DDS, the conclusion is a consequence of a single (immediately preceding) formula, not of all the preceding formulas in the deduction.

Definition 13. A INLINE RULE R is a binary relation on formulas; we write $\varphi \vdash_R \varphi'$ for $(\varphi, \varphi') \in R$.

Definition 14. An INLINE DERIVATION is a sequence of formulas $\varphi_0, \ldots, \varphi_N$ such that:

- φ_0 is an axiom or a hypothesis,
- for every $0 \leq n < N$, there is an inline rule R such that $\varphi_n \vdash_R \varphi_{n+1}$.

The final formula of the sequence, φ_N, is the CONCLUSION of the derivation.[2]

With the exception of Prune, the inline rules defined in the book belong to the special class of *canonical inline rules*.[3] The purpose of this notion is to streamline the proof of soundness. The main ingredient of a canonical inline rule is its *core derivation*, which is a classical (root) derivation we wish to apply inline.

[1] Copy, Delete, and Add are canonical inline rules. Prune, however, is not. The effect of Prune can be achieved by a sequence of certain inline rules, but we don't define those inline rules in the book. We prove the validity of Prune in an easier way, by slightly adjusting the proof of validity for canonical inline rules; see subsection A.2.6 for details.

[2] Alternatively, we could have defined *a* conclusion of a derivation as the form of any constituent of φ_N positively p-scoping over the root of φ_N, and the negation of the form of any constituent of φ_N negatively p-scoping over the root of φ_N. The deductive power of the alternative system is clearly the same: *the* conclusion (in the sense of the official definition) is *a* conclusion (in the sense of the alternative definition) as the root positively p-scopes over itself, and Copying a conclusion to the root derives it as the conclusion; see also subsection 6.5.3.

[3] Strictly speaking, this statement is not true: it turns out that Add, Delete, Copy, and any other polarity-sensitive inline rules we might devise are not canonical. Each of them is rather a *union* of several canonical inline rules; see subsection A.2.6 for discussion.

Definition 15. A CANONICAL INLINE RULE specifies:

(a) the required number of (non-target) premises (M);
(b) the required p-scope polarities of the target (τ) and the premises (σ_i) at the target;
(c) the formal requirements on the target and the premises, which may be interdependent and furthermore depend on the p-scope polarities;
(d) the CORE DERIVATION SCHEMA: a schema for a Hilbert-style deduction with hypotheses $\neg^{\sigma_1}\mathcal{P}_1, \ldots, \neg^{\sigma_M}\mathcal{P}_M$ and $\neg^{\tau}\mathcal{T}$. (The core derivation schema may be of any length, and may deploy any valid inference rules, including Deduction Theorem and Existential Instantiation.)

Before turning to the generic characterization of a canonical inline rule, we need to introduce some further terminology.

Definition 16. A constituent t of formula φ is PROPERLY QUANTIFIED in φ iff

(a) no variable of the formal language is the bound variable of more than one quantifier properly dominating t, and
(b) no free variable of φ is the bound variable of a quantifier properly dominating t.

Definition 17.

(a) A VARIABLE SUBSTITUTION is a mapping from a set of variables to a set of terms.
(b) A VARIABLE SUBSTITUTION ON a set of nodes N is a variable substitution whose domain equals the set of variables bound by the quantifier nodes in N.

Definition 18. For a canonical inline rule R, let $\varphi \vdash_R \varphi'$ iff there are constituents t and $p_1, \ldots, p_M$ of φ, variable substitutions $f_1, \ldots, f_M$, and formula γ such that the five items below hold, where α is the form of t, τ is the constituent polarity of t in φ, and β_i is the form of p_i for every $i \in \mathbb{N}_1^M$.

1. M is the number of premises required by R, and for each $i \in \mathbb{N}_1^M$:
   (a) p_i p-scopes over t (let σ_i be the p-scope polarity of p_i at t);
   (b) f_i is a variable substitution on the up-path from p_i to t; and
   (c) for every variable x in the domain of f_i, $f_i(x)$ is free for x in β_i.
2. τ is the constituent polarity of the target required by R, and α and $\beta_1(f_1), \ldots, \beta_M(f_M)$ satisfy the formal requirements on the target and the premises specified by R.
3. t is properly quantified in φ.
4. The instantiation of the core derivation schema of R with α instantiating $\mathcal{T}$ and $\beta_i(f_i)$ instantiating $\mathcal{P}_i$ for every $i \in \mathbb{N}_1^M$—we call it the CORE DEDUCTION—satisfies the following conditions:
   (a) In every application of Universal Generalization within the core deduction, the quantified variable is neither free in φ nor bound in φ by any proper ancestor of the target t.[4] (This requirement also applies to any UGs implicitly used by a derived rule or a theorem.)
   (b) The conclusion of the core deduction is $\neg^{\tau}\gamma$.

[4] This requirement could be made more precise, but at the cost of making it more complicated. The exact formulation would replicate the requirements made explicit in the proof of Theorem 1, see (T3), (T5), and (R4) in the proof. We refrain from explicating it as the simpler formulation above suffices for all the applications in the book.

5. $\varphi' \equiv \varphi(\gamma'/t)$, where $\gamma' :\equiv \forall^\tau x_1 \dots \forall^\tau x_l\, \gamma$ for some non-negative integer l and variables $x_1, \dots, x_l$ not free in φ or bound by a proper ancestor of t in φ. We call γ' the REPLACEMENT; $\forall^\tau x_1 \dots \forall^\tau x_l$ is the (e-universal) quantifier prefix; we also say that the conclusion was implicitly (e-universally) generalized (by $\forall^\tau x_1 \dots \forall^\tau x_l$).

A.2 Soundness of Dynamic Deductive System

In this section, we prove the soundness of Dynamic Deductive System (DDS) with respect to the semantics of classical first-order logic. The proof is indirect; we show that an application of a canonical inline rule can be "simulated" in a Hilbert-style deductive system.

To prove the soundness of a deductive system through another deductive system, we must perform two tasks: deduce each axiom of the former system in the latter, and prove the validity of each rule of the former system by constructing a deduction in the latter system having the same premises and the same conclusion. In our case, the first task is trivial, as DDS has a single axiom, $\top$. However, a thorough execution of the second task will take some care.

In the previous section, Definition 15 introduced the notion of a canonical inline rule. At the heart of such a rule is a (classical) root deduction which we have called a core deduction, and one can see Definition 18 as stating the conditions which allow the core deduction to be inlined. The bulk of the present section (subsections A.2.1–A.2.5) is dedicated to proving that the stated conditions in fact suffice for the validity of inlining. In subsection A.2.6 we then apply this generic result to individual rules; to prove their validity, we need only exhibit their core deductions. (As we shall see, this does not fully apply to Prune. This rule will require a bit more work.)

The central result of this section, which proves the validity of canonical inline rules (see Definitions 15 and 18), is the following.

Theorem 1. *Let R be a canonical inline rule. If $\varphi \vdash_R \varphi'$, then φ' can be deduced from φ by a Hilbert-style deduction.*

The proof of this theorem is laid out in subsections A.2.1–A.2.5. Though fairly long, as there are many details to consider, it is also quite straightforward. For a given canonical inline rule R, applied to target t to yield a replacement ρ, we explicitly construct a root deduction from φ to $\varphi(\rho/t)$, which we call the *root validation*. The root validation has four parts:

- the *descent to the target* yields the target (or its negation, depending on the target's polarity); more precisely, it yields the (negated) target with the effectively existentially bound variables replaced by novel constants;
- for each premise, the *descent to the premise* yields the premise (or its negation, depending on the p-scope polarity); more precisely, it yields the (negated) premise with the (effectively universally bound) variables on the up-path from the premise to the target substituted by the terms specified by the given variable substitution;
- the *derivation of the replacement* is an instance of the core derivation schema defined by the canonical inline rule; and
- the *ascent from the target*, where we rebuild the matrix with the replacement substituted for the target.

Each of these parts of the root validation is constructed in a separate subsection. But first we need to introduce some tools we will find useful in the process of construction.

A.2.1 Auxiliary notions and results

For simplicity, we employ the following derived classical rules in the root validation: Conjunction Elimination (CE), Conjunction Introduction (CI), Disjunctive Syllogism (DS), Replacement (using Double Negation, De Morgan's laws, Quantifier Negation, and Material Implication), Reiteration, Universal Generalization (UG), Universal Instantiation (UI), Existential Generalization (EG), Existential Instantiation (EI), and Deduction Theorem (DT).

Deduction Theorem and the theorem supporting Existential Instantiation (we rely on Mendelson's (1997) implementation of EI, which he calls Rule C) state that, given the existence of a derivation which satisfies certain conditions, we can infer the existence of a corresponding regular derivation (with the conclusion stated by the theorem). Semi-formally, we shall refer to the former derivation as a *subderivation*, which is opened at some point and closed at another. Opening a subderivation makes available a certain additional hypothesis, but forces us to adhere to the additional conditions until the subderivation is closed. By closing the subderivation, we implicitly replace it by its corresponding regular derivation. As we will have no need to refer to any line of the corresponding derivation except for its conclusion, we will not formalize the replacement. (Within a subderivation, we may of course refer to preceding lines of any outer derivation.)

To use EI, we need to enrich the original formal language L by a countably infinite set of new individual constants $D = \{d_0, d_1, \dots\}$. To apply substitutions f_i to premises p_i, we need to enrich it by a countably infinite set of new variables $Y = \{y_0, y_1, \dots\}$. We refer to the enriched language as L'. Theorem 2, Definition 19, and Theorem 3 are adapted from (Mendelson 1997, pp. 67, 74–75).

Theorem 2 (Deduction Theorem). *If a deduction showing that* $\Gamma, \mathcal{B} \vdash \mathcal{C}$ *involves no application of Universal Generalization over a variable free in* $\mathcal{B}$*, then* $\Gamma \vdash \mathcal{B} \Rightarrow \mathcal{C}$.

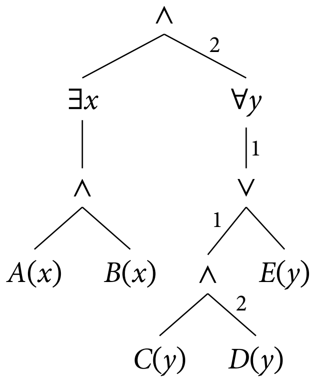
Definition 19. A RULE C DEDUCTION in a first-order theory K is defined as follows: $\Gamma \vdash^{C} \mathcal{B}$ iff there is a sequence of formulas $\langle \mathcal{D}_1, \dots, \mathcal{D}_n \rangle$ such that $\mathcal{D}_n \equiv \mathcal{B}$ and the following conditions hold:

1. For $i \in \mathbb{N}_1^n$, either
   (a) $\mathcal{D}_i$ is an axiom of K, or
   (b) $\mathcal{D}_i \in \Gamma$, or
   (c) $\mathcal{D}_i$ follows from preceding formulas in the sequence by some rule, or
   (d) there is a preceding formula $\exists x\, \mathcal{C}$ such that $\mathcal{D}_i \equiv \mathcal{C}(d/x)$, where d is a new individual constant (rule C).
2. As axioms in condition 1a, we can also use all logical axioms that involve the new individual constants already introduced in the sequence by applications of rule C.
3. No application of Universal Generalization is made using a variable that is free in some $\exists x\, \mathcal{C}$ to which rule C has been previously applied.
4. $\mathcal{B}$ contains none of the new individual constants introduced in the sequence in any application of rule C.

Theorem 3 (Rule C Theorem). *If* $\Gamma \vdash^{C} \mathcal{B}$*, then* $\Gamma \vdash \mathcal{B}$.

In both the main text and the mathematical definition of DDS in the previous section, we have relied on an intuitive understanding of the constituency of a formula. In the proof of soundness, however, we will require an explicit encoding of the constituent structure of formulas, i.e. a way to refer to the constituents of their parse trees. The notion of *constituent path* defined below presents such an explicit encoding. The idea is simply to enumerate the children of a node (from left to right, starting at 1), and say that a constituent path is a sequence of integers telling us which road to take at a junction of some depth. For example, the constituent path of $D(y)$ below is $\langle 2, 1, 1, 2\rangle$. Of course, the constituent path is relative to the chosen root. For example, $\langle 2, 1, 1, 2\rangle$ is the constituent path of $D(y)$ with respect to the root of (6), while the constituent path of $D(y)$ with respect to the node $\vee$ is $\langle 1, 2\rangle$. And when we want to refer to the form of some constituent, we will write things like $\varphi(\langle 2, 1, 1\rangle)$—if we take φ to stand for the tree in (6), this expression stands for $C(y) \wedge D(y)$.

(6)



Definition 20. A CONSTITUENT PATH is a (possibly empty) sequence of positive integers, i.e. an element of $\bigcup_{n=0}^{\infty} \mathbb{N}_+^n$.

Definition 21. For a constituent path c, the let $|c|$ be the unique non-negative integer such that $c \in \mathbb{N}_+^{|c|}$.

Definition 22. For constituent paths $a = \langle a_1, \ldots, a_m\rangle$ and $b = \langle b_1, \ldots, b_n\rangle$, let $a + b := \langle a_1, \ldots, a_m, b_1, \ldots, b_n\rangle$.

Definition 23. For a constituent path $a = \langle a_1, \ldots, a_m\rangle$ and non-negative integer $n \leq m$, let $a^n := \langle a_1, \ldots, a_n\rangle$.

Let L be a language of first-order predicate logic. Assume that the only propositional connectives of L are $\neg$, $\wedge$, and $\vee$. Let $\mathbb{F}$ be the set of formulas, $\mathbb{P}$ the set of predicates symbols and $\mathbb{V}$ the set of variables of L. Following the formation rules of L, we define the following mappings.

Definition 24. A HEAD MAPPING $\mathbb{H}$ is a mapping from formulas to logical symbols and predicate symbols:

$$\mathbb{H}\colon\ \mathbb{F} \to \{\neg, \wedge, \vee, \forall, \exists\} \cup \mathbb{P}$$

Definition 25. A CONSTITUENT MAPPING $\mathbb{C}$ is a partial mapping from formulas and constituent paths to formulas

$$\mathbb{C}\colon\ \mathbb{F} \times \bigcup_{n=0}^{\infty} \mathbb{N}_+^n \supset \mathrm{dom}(\mathbb{C}) \to \mathbb{F},$$

where the domain of $\mathbb{C}$ is the smallest subset of $\mathbb{F} \times \bigcup_{n=0}^{\infty} \mathbb{N}_+^n$ such that:

(a) $(\varphi, \langle\rangle) \in \mathrm{dom}(\mathbb{C})$ and $\mathbb{C}(\varphi, \langle\rangle) :\equiv \varphi$, for any formula φ;
(b) $(\varphi, \langle 1\rangle) \in \mathrm{dom}(\mathbb{C})$, for any formula φ such that $\mathbb{H}(\varphi) \in \{\neg, \wedge, \vee, \forall, \exists\}$;
(c) $(\varphi, \langle 2\rangle) \in \mathrm{dom}(\mathbb{C})$, for any formula φ such that $\mathbb{H}(\varphi) \in \{\wedge, \vee\}$;
(d) if $(\varphi, a) \in \mathrm{dom}(\mathbb{C})$ and $(\mathbb{C}(\varphi, a), b) \in \mathrm{dom}(\mathbb{C})$, then $(\varphi, a + b) \in \mathrm{dom}(\mathbb{C})$ and $\mathbb{C}(\varphi, a + b) \equiv \mathbb{C}(\mathbb{C}(\varphi, a), b)$, for any formula φ and constituent paths a and b.

Definition 26. The BOUND VARIABLE MAPPING $\mathbb{B}$ is a mapping from formulas headed by a quantifier to variables:

$$\mathbb{B}\colon \{\varphi \in \mathbb{F};\ \mathbb{H}(\varphi) \in \{\forall, \exists\}\} \to \mathbb{V}.$$

Definition 27. For any formula φ of L (below, α and β are arbitrary formulas, and x is an arbitrary variable):

(a) if $\varphi \equiv \neg\alpha$: $\mathbb{H}(\varphi) := \neg$, $\mathbb{C}(\varphi, \langle 1\rangle) :\equiv \alpha$;
(b) if $\varphi \equiv \alpha \wedge \beta$: $\mathbb{H}(\varphi) := \wedge$, $\mathbb{C}(\varphi, \langle 1\rangle) :\equiv \alpha$, $\mathbb{C}(\varphi, \langle 2\rangle) :\equiv \beta$;
(c) if $\varphi \equiv \alpha \vee \beta$: $\mathbb{H}(\varphi) := \vee$, $\mathbb{C}(\varphi, \langle 1\rangle) :\equiv \alpha$, $\mathbb{C}(\varphi, \langle 2\rangle) :\equiv \beta$;
(d) if $\varphi \equiv \forall x\, \alpha$: $\mathbb{H}(\varphi) := \forall$, $\mathbb{B}(\varphi) := x$, $\mathbb{C}(\varphi, \langle 1\rangle) :\equiv \alpha$;
(e) if $\varphi \equiv \exists x\, \alpha$: $\mathbb{H}(\varphi) := \exists$, $\mathbb{B}(\varphi) := x$, $\mathbb{C}(\varphi, \langle 1\rangle) :\equiv \alpha$.

Definition 28. A CONSTITUENT is a member of $\mathrm{dom}(\mathbb{C})$. Given a constituent (φ, c), we call φ its MATRIX, c its PATH, and $\mathbb{C}(\varphi, c)$ its FORM. We normally state that (φ, c) is a constituent by saying that c is a CONSTITUENT OF φ, and refer to its form by $\varphi(c)$, i.e. we abbreviate $\varphi(c) :\equiv \mathbb{C}(\varphi, c)$.

Definition 29. Constituent (φ, a) is an ANCESTOR of constituent (φ, b), in symbols $(\varphi, a) \geq (\varphi, b)$, iff $a = b^{|a|}$. When it is clear that we are talking about constituents of formula φ, we usually abbreviate $(\varphi, a) \geq (\varphi, b)$ to $a \geq b$.

Definition 30. Constituent a is a PROPER ANCESTOR of constituent b, in symbols $a > b$, iff $a \geq b$ and $a \neq b$.

Definition 31. a is a (PROPER) DESCENDANT of b iff b is a (proper) ancestor of a. a (PROPERLY) DOMINATES b iff a is a (proper) ancestor of b.

Definition 32. Constituent a is a COMMON ANCESTOR of constituents b and c iff $a \geq b$ and $a \geq c$.

Definition 33. Constituent c is the LOWEST COMMON ANCESTOR (LCA) of constituents a and b, in symbols $\mathrm{LCA}(a, b)$, iff c is a common ancestor of a and b and there is no common ancestor c' of a and b such that $c' < c$.

Lemma 1.

(a) Let $a = \langle a_1, \ldots, a_m\rangle$ and $b = \langle b_1, \ldots, b_n\rangle$. Then $\mathrm{LCA}(a, b) = a^i = b^i$, where i is the unique integer $0 \leq i \leq \min(m, n)$ such that: $a_j = b_j$ for every integer $1 \leq j \leq i$, and $a_{i+1} \neq b_{i+1}$ if $i < \min(m, n)$.

(b) *LCA*$(a, b) =$ *LCA*(b, a).
(c) *a is an ancestor of b iff a equals* LCA(a, b).

Definition 34. The UP-PATH from constituent a to constituent b is the set of constituents $\{c;\ a < c \leq \mathrm{LCA}(a, b)\}$.

Definition 35. Let constituent a be an ancestor of constituent c. Let p be the number of negations on the up-path from a to c. Constituent c has POSITIVE POLARITY in a, in symbols $c \leq^{+} a$, iff p is even; c has NEGATIVE POLARITY in a, in symbols $c \leq^{-} a$, iff p is odd. The polarity of constituent c of φ in φ is the polarity of (φ, c) in $(\varphi, \langle\rangle)$.

Definition 36. A VARIABLE SUBSTITUTION ON a set of nodes N is a partial mapping

$$f\colon\ N \supset \{n \in N;\ \mathbb{H}(n) \in \{\forall, \exists\}\} \to \mathbb{T}$$

Definition 37. A PREDICATE SUBSTITUTION is a partial mapping $f\colon\ \mathbb{P} \hookrightarrow \mathbb{P}$ such that the arity of $f(P)$ equals the arity of P for each P in $\mathrm{dom}(f)$.

This concludes the part of the section containing generic definitions. Below, we define a special kind of substitution which will be useful in the proof of soundness, and prove several simple statements about it. (We don't provide the proof of Lemma 2, as it is obvious. For the definition of p-scope path, see subsection A.1.1.)

Definition 38 (Step-wise substitution). Let φ be a formula, $a = \langle a_1, \ldots, a_n\rangle$ a constituent of φ, and $u\colon\ \{i \in \mathbb{N}_0^{n-1};\ \mathbb{H}(\varphi(a^i)) \in \{\forall, \exists\}\} \to \mathbb{T}$ a partial mapping from natural numbers to terms. Then $\varphi(a, u)$ denotes the formula α_n generated by the following algorithm. First, set $\alpha_0 :\equiv \varphi$. Then, for every $i \in \mathbb{N}_1^n$, let

- $\alpha_i :\equiv \alpha_{i-1}(\langle a_i\rangle)(u(i-1)/\mathbb{B}(\varphi(a^{i-1})))$, if $i \in \mathrm{dom}(u)$; and
- $\alpha_i :\equiv \alpha_{i-1}(\langle a_i\rangle)$ otherwise.

Lemma 2. *With assumptions as in Definition 38:*

(a) *For every $i \in \mathbb{N}_0^n$, $\varphi(a^i, u) \equiv \alpha_i$, with α_i as in the definition.*
(b) *For every $i \in \mathbb{N}_0^{n-1}$, $\mathbb{H}(\varphi(a^i, u)) = \mathbb{H}(\varphi(a^i))$. (Step-wise substitution does not affect the heads.)*
(c) *For every $i \in \mathrm{dom}(u)$, $\mathbb{B}(\varphi(a^i, u)) = \mathbb{B}(\varphi(a^i))$. (Step-wise substitution does not affect which variable a quantifier binds.)*
(d) *Let $\mathcal{A}$ be any formula and $\varphi' :\equiv \varphi(\mathcal{A}/a)$. Then, for every $i \in \mathbb{N}_0^{n-1}$ where $\mathbb{H}(\varphi(a^i)) \in \{\wedge, \vee\}$, $\varphi(b_i, u) \equiv \varphi'(b_i, u)$ for $b_i := a^i + \langle 3 - a_{i+1}\rangle$. (Step-wise substitution produces the same uncles for different forms of the target.)*
(e) *Define $\chi\colon\ \mathrm{dom}(u) \to \mathbb{F}$ by $\chi(i) := \mathbb{B}(\varphi(a^i))$ and $f\colon\ \mathrm{im}(\chi) \to \mathbb{T}$ by $f(x) := u(max(\chi^{-1}(x)))$. Then, $\varphi(a, u) \equiv \varphi(a)(f)$. (Step-wise substitution yields the same result as the variable substitution.)*
(f) *Define χ and f as in point (e). If a is properly quantified in φ, χ is an injection, therefore, for any $x \in \mathrm{dom}(f)$, $\chi^{-1}(x)$ is a singleton and $f(x) = u(i)$ for the unique $i \in \mathrm{dom}(u)$ such that $\mathbb{B}(\varphi(a^i)) = x$.*

Lemma 3. *Let β be a member of the p-scope path of α and assume that $\mathbb{H}(\beta) \in \{\wedge, \forall\}^{\tau}$ for some polarity τ. Let σ be the polarity of the p-scope path of α, and ρ the polarity of α within β. Then $\sigma = \rho\tau$.*

Proof. Let α' be the p-scope pivot of α and π the polarity of α within α'. Let ρ' be the polarity of α' within β. As β is a member of the p-scope path of α, the latter is obviously non-empty, so exactly one of $\mathbb{A}_{\wedge,\forall}(\alpha')$ and $\mathbb{A}_{\vee,\exists}(\alpha')$ is non-empty and β is its member.

- If $\mathbb{A}_{\wedge,\forall}(\alpha')$ is non-empty: by Definition 7, $\mathbb{H}(\beta) \in \{\wedge, \forall\}^{\rho'}$, so $\tau = \rho'$; by Definition 10, $\sigma = \pi$. Thus $\rho\tau = (\pi\rho')\tau = \pi(\rho'\tau) = \pi = \sigma$.
- If $\mathbb{A}_{\vee,\exists}(\alpha')$ is non-empty: by Definition 7, $\mathbb{H}(\beta) \in \{\vee, \exists\}^{\rho'} = \{\wedge, \forall\}^{-\rho'}$, so $\tau = -\rho'$; by Definition 10, $\sigma = -\pi$. Thus $\rho\tau = (\pi\rho')\tau = \pi(\rho'\tau) = -\pi = \sigma$.

■

A.2.2 The descent to the target

In this subsection, we derive a sequence of formulas $\gamma_{i,j}$ while "descending" from the root of the (dynamic) formula to the target. The sequence is doubly indexed to facilitate keeping track of the correspondence between the constructed derivation and the ancestors of t, with the first index corresponding to the ancestor and the second index being 0 for the final formula corresponding to an ancestor. Formally, γ_{i_1,j_1} precedes γ_{i_2,j_2} iff $i_1 < i_2$, or $i_1 = i_2$ and $j_1 < j_2$.

Let τ be the polarity of t in φ. Let $\langle t_1, \ldots, t_n\rangle := t$. For each $i \in \mathbb{N}_0^n$, let τ_i be the polarity of t^i in φ (we obviously have $\tau = \tau_n$). Define mapping $u\colon \{i \in \mathbb{N}_0^{n-1};\ \mathbb{H}(\varphi(t^i)) \in \{\forall, \exists\}\} \to \mathbb{T}$ by

$$u(i) := \begin{cases} \mathbb{B}(\varphi(t^i)) & \text{if } \mathbb{H}(\varphi(t^i)) = \forall^{\tau_i} \\ d_i & \text{if } \mathbb{H}(\varphi(t^i)) = \exists^{\tau_i}. \end{cases}$$

For $i \in \mathbb{N}_0^n$, let $t_i' := \varphi(t^i, u)$ (see Definition 38). By points (b) and (c) of Lemma 2, we have $\mathbb{H}(t_i') = \mathbb{H}(\varphi(t^i))$ for all $i \in \mathbb{N}_0^n$, and $\mathbb{B}(t_i') = \mathbb{B}(\varphi(t^i))$ for all $i \in \mathbb{N}_0^n$ where $\mathbb{H}(\varphi(t^i)) \in \{\forall, \exists\}$. As t is properly quantified (see Definition 16) in φ, all $\mathbb{B}(\varphi(t^i))$ are distinct, so u is an injection.

At each step below, we assume $\gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t_{i-1}'$ and derive that $\gamma_{i,0} \equiv \neg^{\tau_i} t_i'$ holds as well. We start by introducing the matrix, $\gamma_{0,0} :\equiv \varphi$. As $\varphi \equiv \varphi(t^0) \equiv t_0' \equiv \neg^{\tau_0} t_0'$, the assumption holds for $i = 1$. For $i \in \langle 1, \ldots, n\rangle$:

(T1) If $\mathbb{H}(t_{i-1}') = \neg$:
By construction of t_i', we have $t_i' \equiv t_{i-1}'(\langle t_i\rangle)$ (and $t_i = 1$), so $t_{i-1}' = \neg t_i'$. Clearly, $\tau_{i-1} = -\tau_i$.
   (a) If $\tau_i = +$ (and $\tau_{i-1} = -$), let $\gamma_{i,0} :\equiv t_i'$; justification: Replacement (by Double Negation) from $\gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t_{i-1}' \equiv \neg t_{i-1}' \equiv \neg\neg t_i'$.
   (b) If $\tau_i = -$ (and $\tau_{i-1} = +$), let $\gamma_{i,0} :\equiv \neg t_i'$; justification: Reiteration from $\gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t_{i-1}' \equiv t_{i-1}' \equiv \neg t_i'$.
   In both cases, $\gamma_{i,0} \equiv \neg^{\tau_i} t_i'$.

(T2) If $\mathbb{H}(t_{i-1}') = \wedge^{\tau_{i-1}}$:
By construction of t_i', we have $t_i' \equiv t_{i-1}'(\langle t_i\rangle)$. Let $\beta_i :\equiv t_{i-1}'(\langle 3 - t_i\rangle)$ be the other immediate constituent of t_{i-1}'. Clearly, $\tau_{i-1} = \tau_i$.

(a) We only show the case of $t_i = 1$, where $t'_{i-1} \equiv t'_i \wedge^{\tau_{i-1}} \beta_i$; reverse the order of conjuncts/disjuncts for $t_i = 2$.

i. If $\tau_{i-1} = \tau_i = +$, let $\gamma_{i,-2} :\equiv \gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t'_{i-1} \equiv t'_{i-1} \equiv t'_i \wedge^{\tau_{i-1}} \beta_i \equiv t'_i \wedge \beta_i$; justification: Reiteration.

ii. If $\tau_{i-1} = \tau_i = -$, let $\gamma_{i,-2} :\equiv \neg t'_i \wedge \neg\beta_i$; justification: Replacement (by De Morgan's law) from $\gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t'_{i-1} \equiv \neg t'_{i-1} \equiv \neg(t'_i \wedge^{\tau_{i-1}} \beta_i) \equiv \neg(t'_i \vee \beta_i)$.

For both values of τ_i, we have $\gamma_{i,-2} \equiv \neg^{\tau_i} t'_i \wedge \neg^{\tau_i} \beta_i$.

(b) Let $\gamma_{i,-1} :\equiv \neg^{\tau_i}\beta_i$; justification: Conjunction Elimination from $\gamma_{i,-2}$. (This step will be used in the descent to the premises and the ascent from the target.)

(c) Let $\gamma_{i,0} :\equiv \neg^{\tau_i} t'_i$; justification: Conjunction Elimination from $\gamma_{i,-2}$.

(T3) If $\mathbb{H}(t'_{i-1}) = \vee^{\tau_{i-1}}$:
By construction of t'_i, we have $t'_i \equiv t'_{i-1}(\langle t_i \rangle)$. Let $\beta_i :\equiv t'_{i-1}(\langle 3 - t_i \rangle)$ be the other immediate constituent of t'_{i-1}. Clearly, $\tau_{i-1} = \tau_i$.

(a) We only show the case of $t_i = 1$, where $t'_{i-1} \equiv t'_i \vee^{\tau_{i-1}} \beta_i$; reverse the order of conjuncts/disjuncts for $t_i = 2$.

i. If $\tau_{i-1} = \tau_i = +$: $\gamma_{i,-2} :\equiv \gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t'_{i-1} \equiv t'_{i-1} \equiv t'_i \vee^{\tau_{i-1}} \beta_i \equiv t'_i \vee \beta_i$; justification: Reiteration.

ii. If $\tau_{i-1} = \tau_i = -$: Let $\gamma_{i,-2} :\equiv \neg t'_i \vee \neg\beta_i$; justification: Replacement (by De Morgan's law) from $\gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t'_{i-1} \equiv \neg t'_{i-1} \equiv \neg(t'_i \vee^{\tau_{i-1}} \beta_i) \equiv \neg(t'_i \wedge \beta_i)$.

For both values of τ_i, we have $\gamma_{i,-2} \equiv \neg^{\tau_i} t'_i \vee \neg^{\tau_i} \beta_i$.

(b) Let $\gamma_{i,-1} :\equiv \neg\neg^{\tau_i}\beta_i$; justification: assumption for Deduction Theorem.
This line opens a Deduction Theorem subderivation. The Deduction Theorem requires (Theorem 2) that the subderivation involves no application of Universal Generalization of which the quantified variable is free in the assumption.[5] In our case, this means that, until we close this subderivation, we may not universally generalize over any variable having a free occurrence in β_i. In principle, any free variable of φ and any variable effectively universally bound by any ancestor of t^{i-1} could be free in β_i (effectively existentially bound variables were substituted for individual constants; see (T5)). We therefore disallow UG over variables belonging to sets $\mathbb{F}(\varphi)$ and $\{\mathbb{B}(\varphi(t^j)) ; j \in \mathbb{N}_0^{i-1} \wedge \mathbb{H}(\varphi(t^j)) = \forall^{\tau_j}\}$ until the derivation of $\zeta_{i-1,0}$ in (B3) in the ascent from the target.

(c) Let $\gamma_{i,0} :\equiv \neg^{\tau_i} t'_i$; justification: Disjunctive Syllogism from $\gamma_{i,-1}$ and $\gamma_{i,-2}$.

(T4) If $\mathbb{H}(t'_{i-1}) = \forall^{\tau_{i-1}}$:
Let $x := \mathbb{B}(t'_{i-1})$ and $\beta :\equiv t'_{i-1}(\langle t_i \rangle)$ (we have $t_i = 1$); so $t'_{i-1} \equiv \forall^{\tau_{i-1}} x\, \beta$. By construction of t'_i and the definition of u, we have $t'_i \equiv \beta(u(i-1)/x) \equiv \beta(x/x) \equiv \beta$, so $t'_{i-1} \equiv \forall^{\tau_{i-1}} x\, t'_i$. Clearly, $\tau_{i-1} = \tau_i$.

(a) i. If $\tau_{i-1} = \tau_i = +$, let $\gamma_{i,-1} :\equiv \gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t'_{i-1} \equiv t'_{i-1} \equiv \forall^{\tau_{i-1}} x\, t'_i \equiv \forall x\, t'_i$; justification: Reiteration.

ii. If $\tau_{i-1} = \tau_i = -$, let $\gamma_{i,-1} :\equiv \forall x \neg t'_i$; justification: Replacement (by Quantifier Negation) from $\gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t'_{i-1} \equiv \neg t'_{i-1} \equiv \neg\forall^{\tau_{i-1}} x\, t'_i \equiv \neg\exists x\; t'_i$.

For both values of τ_i, we have $\gamma_{i,-1} \equiv \forall x \neg^{\tau_i} t'_i$.

(b) $\gamma_{i,0} :\equiv (\neg^{\tau_i} t'_i)(x/x) \equiv \neg^{\tau_i} t'_i$; justification: Universal Instantiation from $\gamma_{i,-1}$ with x/x.

[5] More precisely, Mendelson's (1997, p. 67) Deduction Theorem states that UG generalizing a free variable of the assumption of DT should not be applied to a line of proof which *depends* on the assumption. Mendelson states Theorem 2 as a useful consequence of the general theorem. As the next line in the root validation, $\gamma_{i,0}$, depends on the assumption $\gamma_{i,-1}$, the general theorem and its consequence given as Theorem 2 are equivalent for the purpose of this proof.

(T5) If $\mathbb{H}(t'_{i-1}) = \exists^{\tau_{i-1}}$:

Let $x := \mathbb{B}(t'_{i-1})$ and $\beta :\equiv t'_{i-1}(\langle t_i \rangle)$ (we have $t_i = 1$); so $t'_{i-1} \equiv \exists^{\tau_{i-1}} x\, \beta$. By construction of t'_i and definition of u, we have $t'_i \equiv \beta(u(i-1)/x) \equiv \beta(d_{i-1}/x)$. Clearly, $\tau_{i-1} = \tau_i$.

(a) i. If $\tau_{i-1} = \tau_i = +$, let $\gamma_{i,-1} :\equiv \gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t'_{i-1} \equiv t'_{i-1} \equiv \exists^{\tau_{i-1}} x\, \beta \equiv \exists x\, \beta$; justification: Reiteration.

ii. If $\tau_{i-1} = \tau_i = -$, let $\gamma_{i,-1} :\equiv \exists x \neg \beta$; justification: Replacement (by Quantifier Negation) from $\gamma_{i-1,0} \equiv \neg^{\tau_{i-1}} t'_{i-1} \equiv \neg t'_{i-1} \equiv \neg \exists^{\tau_{i-1}} x\, \beta \equiv \neg \forall x\, \beta$.

For both values of τ_i, we have $\gamma_{i,-1} \equiv \exists x \neg^{\tau_i} \beta$.

(b) $\gamma_{i,0} :\equiv (\neg^{\tau_i} \beta)(d_{i-1}/x) \equiv \neg^{\tau_i} \beta(d_{i-1}/x) \equiv \neg^{\tau_i} t'_i$; justification: Existential Instantiation from $\gamma_{i,-1}$ with d_{i-1}/x (the new individual constant d_{i-1} has clearly not yet been used in the derivation).

This line opens a Rule C subderivation. In a Rule C subderivation, we may not apply Universal Generalization using a variable that has a free occurrence in the existential statement that we have applied Existential Instantiation to (see Theorem 3 and Definition 19). In our case, this means that, until we close this subderivation, we may not universally generalize over any variable having a free occurrence in $\gamma_{i,-1}$. In principle, any free variable of φ and any variable effectively universally bound by any ancestor of t^{i-1} could be free in $\gamma_{i,-1}$ (effectively existentially bound variables were substituted for individual constants; see (T5)). We therefore disallow UG over variables belonging to sets $\mathbb{F}(\varphi)$ and $\{\mathbb{B}(\varphi(t^j)); j \in \mathbb{N}_0^{i-1} \wedge \mathbb{H}(\varphi(t^j)) = \forall^{\tau_j}\}$ until the derivation of $\zeta_{i-1,-4}$ in point (B5).

The final line of the descent to the target reads $\gamma_{n,0} \equiv \neg^{\tau_n} t'_n \equiv \neg^{\tau_n} \varphi(t^n, u) \equiv \neg^{\tau} \varphi(t, u)$. Let e be the variable substitution defined in point (e) of Lemma 2 applied to $\varphi(t, u)$. We then have $\varphi(t, u) \equiv \varphi(t)(e)$, so $\gamma_{n,0} \equiv \neg^{\tau} \varphi(t)(e)$.

(The domain of e is the set of all variables bound by the proper ancestors of t. As u is injective, so is e. For any $x \in \mathbb{F}_L$, either $x(e) = x$ or $x(e) \in D$, so the application of e to a formula substitutes all the free variables of the formula which are effectively existentially bound in φ by the proper ancestors of the target t by the new individual constants introduced in (T5).)

A.2.3 The descent to the premises

The derivations for the individual premises are mutually independent, so we fix the current premise index $K \in \mathbb{N}_1^M$ for this subsection and define the premise constituent $p := p_K$, its p-scope polarity at the target $\sigma := \sigma_K$, and the variable substitution $f := f_K$. Remember that $f\colon \mathbb{B}(A) \to \mathbb{T}$, where A is the p-scope path from p to t, and that $f(x)$ is free for x in $\varphi(p)$ for every $x \in \mathrm{dom}(f)$. Below, we construct sequences $\delta_{i,j}$ (the descent to the premise) and $\delta'_{i,j}$ (the derivation of the variable substitution), the final line of the latter being the result we use in the next subsection.

Let $\langle p_1, \ldots, p_m \rangle := p$ be the components of the constituent p (note that this redefines the meaning of the indices on p for the subsection). Let π be the polarity of p within φ and for every $i \in \mathbb{N}_0^m$, let π_i be the polarity of p^i within φ and π'_i be the polarity of p within p^i. Obviously, $\pi_i \pi'_i = \pi$.

In the descent to the target, we had set the components of the target to be $t = \langle t_1, \ldots, t_n \rangle$. Let q be the greatest member of $\mathbb{N}_0^{\min(m,n)}$ such that $p^i = t^i$ for each $i \in \mathbb{N}_0^q$; p^q is the LCA of p and t. We obviously also have $\pi_i = \tau_i$ for every $i \in \mathbb{N}_0^q$. Define $\sigma_i :\!= \sigma\pi'_i$ for $i \in \mathbb{N}_q^m$ (note that this redefines the meaning of the indices on σ for the subsection). Define mapping $v\colon \{i \in \mathbb{N}_0^{m-1};\ \mathbb{H}(\varphi(p^i)) \in \{\forall, \exists\}\} \to \mathbb{T}$ by

$$v(i) :\!= \begin{cases} \mathbb{B}(\varphi(p^i)) & \text{if } i \in \mathbb{N}_0^q \text{ and } \mathbb{H}(\varphi(p^i)) = \forall^{\tau_i}, \\ d_i & \text{if } i \in \mathbb{N}_0^q \text{ and } \mathbb{H}(\varphi(p^i)) = \exists^{\tau_i}, \text{ and} \\ y_i & \text{if } i \in \mathbb{N}_{q+1}^{m-1} \text{ and } \mathbb{H}(\varphi(p^i)) \in \{\forall, \exists\}. \end{cases}$$

For every $i \in \mathbb{N}_0^m$, let $p'_i :\!= \varphi(p^i, v)$ (see Definition 38). By points (b) and (c) of Lemma 2, we have $\mathbb{H}(p'_i) = \mathbb{H}(\varphi(p^i))$ for all $i \in \mathbb{N}_0^m$, and $\mathbb{B}(p'_i) = \mathbb{B}(\varphi(p^i))$ for all $i \in \mathbb{N}_0^m$ where $\mathbb{H}(\varphi(p^i)) \in \{\forall, \exists\}$.

Remember that $t'_i = \varphi(t^i, u)$. For $i \in \mathbb{N}_0^n$: as $p^i = t^i$ and $u(i) = v(i)$, we have $p'_i = t'_i$. As the target t is properly quantified, $\mathbb{B}(\varphi(p^i)) = \mathbb{B}(\varphi(t^i))$ are distinct for $i \in \mathbb{N}_0^n$, so v is injective.

As the first step in the descent to the premise, we Reiterate the result of the part of the derivation which is shared with the descent to the target (we need to Reiterate it, as descending from p^0 to p^q would open new Deduction Theorem and Rule C subderivations; see (T3) and (T5)), and define the starting point (s) for the subsequent steps.

(P1) If t belongs to the relative p-scope of p (see Definition 12), then p'_q is a branching node, i.e. $\mathbb{H}(p'_q) \in \{\wedge, \vee\}$. In the descent to the target, $\gamma_{q+1,0}$ was constructed by (T2) or (T3). Both define β_{q+1} to be the constituent on the non-target branch. (T2), where $\mathbb{H}(p'_q) = \wedge^{\tau_q} = \wedge^{\tau_{q+1}}$, sets $\gamma_{q+1,-1} \equiv \neg^{\tau_{q+1}}\beta_{q+1}$, while (T3), where $\mathbb{H}(p'_q) = \vee^{\tau_q} = \wedge^{-\tau_q} = \wedge^{-\tau_{q+1}}$ sets $\gamma_{q+1,-1} \equiv \neg\neg^{\tau_{q+1}}\beta_{q+1} \equiv \neg^{-\tau_{q+1}}\beta_{q+1}$. Summing up, $\gamma_{q+1,-1} \equiv \neg^{\rho}\beta_{q+1}$, where ρ is the polarity such that $\mathbb{H}(p'_q) = \wedge^{\rho}$.

Let $\delta_{q+1,0} :\equiv \gamma_{q+1,-1} \equiv \neg^{\rho}\beta_{q+1}$; justification: Reiteration. As $p'_q = t'_q$ but $p_{q+1} \neq t_{q+1}$, p'_{q+1} is the non-target branch of t'_q, i.e. $\beta_{q+1} \equiv p'_{q+1}$, therefore $\delta_{q+1,0} \equiv \neg^{\rho}p'_{q+1}$.

As t belongs to the relative p-scope of p, the polarity of p-scope of p at t (σ) is the polarity of p-scope path of p, by Definition 12. By Lemma 3, $\sigma = \rho\pi'_q$. As $\mathbb{H}(p'_q)$ is not a negation, $\pi'_q = \pi'_{q+1}$, so $\sigma = \rho\pi'_{q+1}$, therefore $\rho = \sigma\pi'_{q+1}$. Summing up, we have $\delta_{q+1,0} \equiv \neg^{\sigma\pi'_{q+1}}p'_{q+1}$. Finally, we define $s :\!= q+2$ and use $\sigma_{s-1} :\!= \sigma\pi'_{s-1}$ to get $\delta_{s-1,0} \equiv \neg^{\sigma_{s-1}}p'_{s-1}$.

(P2) If t belongs to the descendant or the ancestor p-scope of p (see Definition 12), we set $\delta_q :\equiv \gamma_{q,0}$; justification: Reiteration. From the construction of the descent, we have $\gamma_{q,0} \equiv \neg^{\tau_q}t'_q$, so $\delta_{q,0} \equiv \neg^{\tau_q}t'_q \equiv \neg^{\pi_q}p'_q$. As t belongs to the descendant or the ancestor p-scope of p, the polarity of p-scope of p at t (σ) is the polarity of p within φ, i.e. $\sigma = \pi$, by Definition 12. As $\pi = \pi_q\pi'_q$, we get $\pi_q = \pi\pi'_q = \sigma\pi'_q$. Defining $s :\!= q+1$ and using $\sigma_{s-1} :\!= \sigma\pi'_{s-1}$, we arrive at $\delta_{s-1,0} \equiv \neg^{\sigma_{s-1}}p'_{s-1}$.

Let $I = \{i \in \mathbb{N}_0;\ s \leq i \leq m\}$. Let us prove that $\mathbb{H}(p'_{i-1}) \notin \{\vee, \exists\}^{\sigma_{i-1}}$ for every $i \in I$. I can be partitioned into two subsets. The p-scope pivot (see Definition 9) is an ancestor of p, so there is a unique m' such that $p^{m'}$ is the p-scope pivot of p. Define $I_1 = \{i \in I;\ i < m'\}$, $I_2 = \{i \in I;\ m' \leq i\}$ and $P'_j = \{p'_{i-1};\ i \in I_j\}$ for $j \in \{1,2\}$.

P'_1 is a (possibly empty) subset of the p-scope path of p. Thus, if $\mathbb{H}(p'_{i-1}) \in \{\vee, \exists\}^{\sigma_{i-1}} = \{\wedge, \forall\}^{-\sigma_{i-1}}$ for some $i \in I_1$, then, by Lemma 3, $\sigma = \pi'_{i-1}(-\sigma_{i-1})$, and thus $\sigma = -\pi'_{i-1}\sigma\pi'_{i-1} = -\sigma$, a contradiction. Therefore, $\mathbb{H}(p'_{i-1}) \notin \{\vee, \exists\}^{\sigma_{i-1}}$ for $i \in I_1$.

P_2' is a (possibly empty) subset of the alternating $\varnothing$-path of p (see the application of Definition 7 in Definition 9). As such, it consists only of negations. It follows that $\mathbb{H}(p_{i-1}') = \neg \notin \{\vee, \exists\}^{\sigma_{i-1}}$ for any $i \in I_2$.

Having proven that $\mathbb{H}(p_{i-1}') \notin \{\vee, \exists\}^{\sigma_{i-1}}$, we can run the below procedure for every $i \in I$. At each step, we assume $\delta_{i-1,0} \equiv \neg^{\sigma_{i-1}} p_{i-1}'$ and show that $\delta_{i,0} \equiv \neg^{\sigma_i} p_i'$ holds as well. The starting condition was shown to hold in (P1) and (P2) above.

(P3) If $\mathbb{H}(p_{i-1}') = \neg$:
By construction of p_i', we have $p_i' \equiv p_{i-1}'(\langle p_i \rangle)$ (and $p_i = 1$), so $p_{i-1}' = \neg p_i'$. Clearly, $\sigma_{i-1} = -\sigma_i$.
- (a) If $\sigma_i = +$ (and $\sigma_{i-1} = -$), let $\delta_{i,0} :\equiv p_i' \equiv$; justification: Replacement (by Double Negation) from $\delta_{i-1,0} \equiv \neg^{\sigma_{i-1}} p_{i-1}' \equiv \neg p_{i-1}' \equiv \neg\neg p_i'$.
- (b) If $\sigma_i = -$ (and $\sigma_{i-1} = +$), let $\delta_{i,0} :\equiv \neg p_i'$; justification: Reiteration from $\delta_{i-1,0} \equiv \neg^{\sigma_{i-1}} p_{i-1}' \equiv p_{i-1}' \equiv \neg p_i'$.

In both cases, $\delta_{i,0} \equiv \neg^{\sigma_i} p_i'$.

(P4) If $\mathbb{H}(p_{i-1}') = \wedge^{\sigma_{i-1}}$:
By construction of p_i', we have $p_i' \equiv p_{i-1}'(\langle p_i \rangle)$. Let $\beta :\equiv p_{i-1}'(\langle 3 - p_i \rangle)$ be the other immediate constituent of p_{i-1}'. Clearly, $\sigma_{i-1} = \sigma_i$.
- (a) We only show the case of $p_i = 1$, where $p_{i-1}' \equiv p_i' \wedge^{\sigma_{i-1}} \beta$; reverse the order of conjuncts/disjuncts for $p_i = 2$.
  - i. If $\sigma_{i-1} = \sigma_i = +$, let $\delta_{i,-1} :\equiv \delta_{i-1,0} \equiv \neg^{\sigma_{i-1}} p_{i-1}' \equiv p_{i-1}' \equiv p_i' \wedge^{\sigma_{i-1}} \beta \equiv p_i' \wedge \beta$; justification: Reiteration.
  - ii. If $\sigma_{i-1} = \sigma_i = -$, let $\delta_{i,-1} :\equiv \neg p_i' \wedge \neg\beta$; justification: Replacement (by De Morgan's law) from $\delta_{i-1,0} \equiv \neg^{\sigma_{i-1}} p_{i-1}' \equiv \neg p_{i-1}' \equiv \neg(p_i' \wedge^{\sigma_{i-1}} \beta) \equiv \neg(p_i' \vee \beta)$.

  For both values of σ_i, we have $\delta_{i,-1} \equiv \neg^{\sigma_i} p_i' \wedge \neg^{\sigma_i} \beta$.
- (b) Let $\delta_{i,0} :\equiv \neg^{\sigma_i} p_i'$; justification: Conjunction Elimination from $\delta_{i,-1}$.

(P5) If $\mathbb{H}(p_{i-1}') = \forall^{\sigma_{i-1}}$:
Let $y := \mathbb{B}(p_{i-1}')$ and $\beta :\equiv p_{i-1}'(\langle p_i \rangle)$ (we have $p_i = 1$); so $p_{i-1}' \equiv \forall^{\sigma_{i-1}} y\, \beta$. By construction of p_i' and definition of v (note that $i \geq s > q$), we have $p_i' \equiv \beta(v(i-1)/y) \equiv \beta(y_{i-1}/y)$. Clearly, $\sigma_{i-1} = \sigma_i$.
- (a)
  - i. If $\sigma_{i-1} = \sigma_i = +$, let $\delta_{i,-1} :\equiv \delta_{i-1,0} \equiv \neg^{\sigma_{i-1}} p_{i-1}' \equiv p_{i-1}' \equiv \forall^{\sigma_{i-1}} y\, \beta \equiv \forall y\, \beta$; justification: Reiteration.
  - ii. If $\sigma_{i-1} = \sigma_i = -$, let $\delta_{i,-1} :\equiv \forall y \neg\beta$; justification: Replacement (by Quantifier Negation) from $\delta_{i-1,0} \equiv \neg^{\sigma_{i-1}} p_{i-1}' \equiv \neg p_{i-1}' \equiv \neg\forall^{\sigma_{i-1}} y\, \beta \equiv \neg\exists y\, \beta$.

  For both values of σ_i, we have $\delta_{i,-1} \equiv \forall y \neg^{\sigma_i} \beta$.
- (b) $\delta_{i,0} :\equiv (\neg^{\sigma_i} \beta)(y_{i-1}/y) \equiv \neg^{\sigma_i} \beta(y_{i-1}/y) \equiv \neg^{\sigma_i} p_i'$; justification: Universal Instantiation from $\delta_{i,-1}$ with y_{i-1}/y (the new variable y_{i-1} is clearly free for y in $\neg^{\sigma_i} \beta$).

The final line of the above descent reads $\delta_{m,0} \equiv \neg^{\sigma_m} p_m' \equiv \neg^{\sigma_m} \varphi(p^m, v) \equiv \neg^{\sigma} \varphi(p, v)$. Let g be the variable substitution defined in point (e) of Lemma 2 applied to $\varphi(p, v)$. We then have $\varphi(p, v) \equiv \varphi(p)(g)$ and thus $\delta_{m,0} \equiv \neg^{\sigma} \varphi(p)(g)$. As v is an injection, so is g.

We now construct the derivation justifying the application of substitution f to the premise p. Let $\{i_1, \ldots, i_k\} := \{i \in \mathbb{N}_q^{m-1};\ \mathbb{H}(\varphi(p^i)) \in \{\forall, \exists\}\}$ with $i_j \neq i_{j'}$ for $j \neq j'$. The set $\{\mathbb{B}(\varphi(p^i));\ i \in \{i_1, \ldots, i_k\}\}$ is the set of variables bound in the up-path from p to t and thus equals $\mathrm{dom}(f)$. Let us define $f' := e \circ f \circ g_2^{-1}$, where g_2 is the narrowing of injection g to $\mathrm{dom}(f)$, i.e. $g_2 := g|_{\mathrm{dom}(f)}$.

Let $Y_j := \{y_{i_1}, \ldots, y_{i_j}\}$ for each $j \in \mathbb{N}_0^k$. Note that $\mathrm{dom}(f') = \mathrm{im}(g_2) = Y_k$. Let $\delta_{0,0}' := \delta_{m,0}$; justification: Reiteration. For each $j \in \mathbb{N}_1^k$, we will assume that $\delta_{j-1,0} \equiv \neg^{\sigma} \varphi(p)(g)(f'|_{Y_{j-1}})$

and prove that $\delta_{j,0} \equiv \neg^\sigma \varphi(p)(g)(f'|_{Y_j})$. $\delta_{0,0}$ clearly satisfies the starting condition. For each $j \in \langle 1, \ldots, k \rangle$, let $i := i_j$ and perform the following steps.

(P6) Let $\delta'_{j,-1} :\equiv \forall y_i\, \delta'_{j-1,0}$; justification: Universal Generalization from $\delta'_{j-1,0}$ on y_i. As $y_i \notin \mathbb{F}_L$, y_i has no occurrences in φ, so the application of UG is allowed by the conditions imposed by Deduction Theorem and Rule C subderivations, see (T3) and (T5), respectively.

(P7) Let $\delta'_{j,0} :\equiv \delta'_{j-1,0}(f'(y_i)/y_i)$; justification: Universal Instantiation on $\delta'_{j,-1}$ with $f'(y_i)/y_i$. For the UI to be applicable, $f'(y_i)$ must be free for y_i in $\delta_{j-1,0} \equiv \neg^\sigma \varphi(p)(g)(f'|_{Y_{j-1}})$. Let us prove that this is indeed the case.[6]

We have required that $f(y)$ be free for y in $\varphi(p)$ for every $y \in \mathrm{dom}(f)$. This means that for every variable z in $f(y)$, where $y \in \mathrm{dom}(f)$, no free occurrence of y in $\varphi(p)$ lies within the scope of a quantifier binding z.

$y_i \in \mathrm{dom}(g_2^{-1})$, so we can define $y := g_2^{-1}(y_i)$. Obviously, $y \in \mathrm{im}(g_2^{-1}) = \mathrm{dom}(g_2) = \mathrm{dom}(f)$. Therefore, for every variable z in $f(y) = f(g_2^{-1}(y_i))$, no free occurrence of y in $\varphi(p)$ lies within the scope of a quantifier binding z.

For any variable z', either $z'(e) = z'$ or $z'(e) \in D$. Therefore the set of variables in term $f'(y_i) \equiv e \circ (f \circ g_2^{-1}))(y_i)$ is a subset of the set of variables in term $f(y) = (f \circ g_2^{-1})(y_i)$. It follows that for every variable z in $f'(y_i)$, no free occurrence of y in $\varphi(p)$ lies within the scope of a quantifier binding z.

As g is a bijection, the free occurrences of y in $\varphi(p)$ lie within the scope of the same quantifiers as the free occurrences of $g(y) = y_i$ within $\varphi(p)(g)$. Furthermore, as substitution by $f'|_{Y_{j-1}}$ neither affects the existing occurrences of y_i in $\varphi(p)(g)$, nor adds any new occurrences of y_i, we conclude that the free occurrences of y_i within $\varphi(p)(g)(f'|_{Y_{j-1}})$ lie within the scope of the same quantifiers as the free occurrences of y in $\varphi(p)$. Specifically, none of them lies within the scope of any variable z in term $f'(y_i)$, so $f'(y_i)$ is free for y_i in $\varphi(p)(g)(f'|_{Y_{j-1}})$.

Finally note that the fact that $f'|_{Y_{j-1}}$ does not affect the existing occurrences of y_i in $\varphi(p)(g)$ also implies that substitutions $f'|_{Y_{j-1}}$ and $f'(y_i)/y_i$ may be performed simultaneously, so that $\delta'_{j,0} \equiv \delta'_{j-1,0}(f'(y_i)/y_i) \equiv \neg^\sigma \varphi(p)(g)(f'|_{Y_{j-1}})(f'(y_i)/y_i) \equiv \neg^\sigma \varphi(p)(g)(f'|_{Y_j})$.

The final line reads $\delta'_k \equiv \neg^\sigma \varphi(p)(g)(f'|_{Y_k}) \equiv \neg^\sigma \varphi(p)(g)(f')$. We want to show that $\delta'_k \equiv \neg^\sigma \varphi(p)(f)(e)$. Remember that $f' = e \circ f \circ g_2^{-1}$. We need to show that for every variable y in $\mathbb{F}(\varphi(p))$, $y(f' \circ g) = y(e \circ f)$.

- If $y \in \mathrm{dom}(g)$ (y is bound by an ancestor of p), then $g(y) = v(i)$ for the maximal $i \in \mathbb{N}_0^{m-1}$ such that $\mathbb{H}(\varphi(p^i)) \in \{\forall, \exists\}$ and $\mathbb{B}(\varphi(p^i)) = y$.
  - If $i \in \mathbb{N}_0^q$ (y is bound at or above the LCA p^q), we have $g(y) \notin Y_k$, which entails $y(f' \circ g) = y(g) = g(y)$, and $y \notin \mathrm{dom}(f)$ but $y \in \mathrm{dom}(e)$, which entails

[6] In (P5)b above, the application of Universal Instantiation with $(e \circ f)(y)/y$ (and $u(i-1)$ set to $f(y)$) would in general not be justified. While $f(y)$ is required to be free for y in $\varphi(p)$ for any $y \in \mathrm{dom}(f)$, it is not necessarily free for y in β, which in general contains not only the (suitably transformed) premise $\varphi(p)$ but also the other branches (β_i) of binary nodes on the up-path from the premise to the target. Example: let $\varphi \equiv \forall y\, (\forall z\, A(y,z) \wedge B(y)) \wedge \psi$, where ψ is the target and $B(y)$ the premise, and let $f\colon y \mapsto z$. In the descent to the premise, we derive $B(z)$ from $\forall y\, ((\forall z\, A(y,z)) \wedge B(y))$. A "direct" but invalid derivation would be: $\forall z\, A(z,z) \wedge B(z)$ by UI with z/y (this step is invalid as z is not free for y in $\forall z\, A(y,z) \wedge B(y)$); $B(z)$ by CE. The sequence of the descent followed by the UG+UI pairs is designed to first "isolate" the premise (which of course works only because the premise p-scopes over the target) and only then apply the instantiations by f. The valid derivation is thus: $\forall z\, A(y_1,z) \wedge B(y_1)$ by UI with y_1/y; $B(y_1)$ by CE; $\forall y_1\, B(y_1)$ by UG, $B(z)$ by UI with z/y_1.

$y(e \circ f) = y(e) = e(y)$. As t is properly quantified in φ, $e(y) = u(j)$ for the unique $j \in \mathrm{dom}(u)$ such that $\mathbb{B}(\varphi(t^j)) = y$, by point (f) of Lemma 2. However, as $i \leq q$, $\mathbb{B}(\varphi(t^i)) = \mathbb{B}(\varphi(p^i)) = y$, so we have $j = i$ and therefore $g(y) = v(i) = u(j) = e(y)$.

– If $i \in \mathbb{N}_{q+1}^{m-1}$ (y is bound between the LCA p^q and the premise p), then $g(y) = y_i$ and $y = g_2^{-1}(y_i)$. Therefore, $(y)(f' \circ g) = y_i(f') = y_i(g_2^{-1})(f)(e) = y(f)(e)$.

- The case of $y \in \mathrm{dom}(e) - \mathrm{dom}(g)$ (a free occurrence of y in $\varphi(p)$ is free in φ, but y is the bound variable of some quantifier properly dominating the target t) is in contradiction with the requirement that t be properly quantified in φ (see clause (b) of Definition 16).
- If $y \notin \mathrm{dom}(e) \cup \mathrm{dom}(g)$ (y is free in φ and not bound by an ancestor of the target), then, as $y \notin \mathrm{dom}(f') \cup \mathrm{dom}(f)$ either, both $y(g)(f') = y$ and $y(f)(e) = y$.

Summing up, we have $\delta_k' \equiv \neg^\sigma \varphi(p)(f)(e)$. Reintroducing the current premise index $K \in \mathbb{N}_1^M$ on p, σ and f fixed at the beginning of the subsection, define $\delta_K'' := \delta_k' \equiv \neg^{\sigma_K} \varphi(p_K)(f_K)(e)$; justification: Reiteration.

A.2.4 The derivation of the replacement

Each canonical inline rule specifies a core derivation which derives the replacement from the target and the premises; the conditions on core derivations are given in Definition 18. Formulas in the core derivation belong to the original language L. We will prove that applying variable substitution e to each line yields a valid continuation of the root validation (a derivation in L') constructed so far. Consider each line α of the core derivation:

(R1) If α is an axiom in L, we will show $\alpha(e)$ is an axiom of L' (clearly using only the new individual constants introduced by EI) and may thus be introduced into the derivation by point 2 of Definition 19. The proof is trivial for the propositional axioms, so let us focus on the additional axioms of predicate logic.

(a) Let $\alpha \equiv (\forall x\, \mathcal{A}) \Rightarrow \mathcal{A}(t/x)$, where t is free for x in $\mathcal{A}$. Then $\alpha(e) \equiv (\forall x\, \mathcal{A}(e')) \Rightarrow \mathcal{A}(t/x)(e)$, where $e' := e|_{\mathrm{dom}(e)-\{x\}}$. Take any variable $y \in \mathbb{F}_L$: if $y \neq x$, then $x \notin \mathrm{im}(e')$ entails $y(e')(t(e)/x) = y(e') = y(e) = y(t/x)(e)$; for $y = x$, $x \notin \mathrm{dom}(e')$ entails $x(e')(t(e)/x) = x(t(e)/x) = t(e) = x(t/x)(e)$. Therefore, $\mathcal{A}(t/x)(e) \equiv \mathcal{A}(e')(t(e)/x)$. It follows that $\alpha(e) \equiv (\forall x\, \mathcal{A}(e')) \Rightarrow \mathcal{A}(e')(t(e)/x)$ is an instance of this axiom scheme if we can prove that $t(e)$ is free for x in $\mathcal{A}(e')$.

For any variable y, $y(e) = y$ or $y(e) \in D$, so $\mathbb{F}(t(e)) \subset \mathbb{F}(t)$. Applying substitution e' to $\mathcal{A}$ affects neither the quantifiers of $\mathcal{A}$ nor (as $x \notin \mathrm{dom}(e') \cup \mathrm{im}(e')$) the free occurences of x. Therefore, $t(e)$ is free for x in $\mathcal{A}(e')$ because t is free for x in $\mathcal{A}$.

(b) Let $\alpha \equiv \forall x\, (\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \forall x\, \mathcal{B})$, where $x \notin \mathbb{F}(\mathcal{A})$. Then $\alpha(e) \equiv \forall x\, (\mathcal{A}(e') \Rightarrow \mathcal{B}(e')) \Rightarrow (\mathcal{A}(e) \Rightarrow \forall x\, \mathcal{B}(e'))$, where $e' := e|_{\mathrm{dom}(e)-\{x\}}$. As $x \notin \mathbb{F}(\mathcal{A})$, $\mathcal{A}(e') = \mathcal{A}(e)$. For any variable y, $y(e') = y$ or $y(e') \in D$, so $\mathbb{F}(\mathcal{A}(e')) \subset \mathbb{F}(\mathcal{A}) \not\ni x$. Therefore, $\alpha(e) \equiv \forall x\, (\mathcal{A}(e') \Rightarrow \mathcal{B}(e')) \Rightarrow (\mathcal{A}(e') \Rightarrow \forall x\, \mathcal{B}(e'))$ is an instance of this axiom scheme in L'.

(R2) If α equals $\neg^\tau \varphi(t)$ or one of $\neg^{\sigma_K} \varphi(p_K)(f_K)$ for $K \in \mathbb{N}_1^M$, then $\alpha(e)$ is justified by Reiteration of $\gamma_{n,0} \equiv \neg^\tau \varphi(t)(e)$ or one of $\delta_K'' \equiv \neg^{\sigma_K} \varphi(p_K)(f_K)(e)$, respectively.

(R3) If α follows from $\mathcal{A}$ and $\mathcal{B} \equiv \mathcal{A} \Rightarrow \alpha$ by Modus Ponens, then clearly $\alpha(e)$ follows by MP from $\mathcal{A}(e)$ and $\mathcal{B}(e) \equiv (\mathcal{A} \Rightarrow \alpha)(e) \equiv \mathcal{A}(e) \Rightarrow \alpha(e)$.

(R4) Let $\forall x\, \mathcal{A}$ follow from $\mathcal{A}$ by Universal Generalization, where x satisfies the constraint on UG in core derivations given in point 4a of Definition 18 and is thus neither free in φ nor bound in φ by any proper ancestor of the target t.

In the root validation, we want to apply UG on $\mathcal{A}(e)$ with x as the bound variable to derive $\forall x\, \mathcal{A}(e)$. For UG to be applicable, it needs to satisfy the conditions imposed by any Deduction Theorem and Rule C subderivations that may have been opened in the descent to the target in (T3) and (T5), respectively. Both conditions only disallow UG over the free variables of φ and a subset of variables bound by some quantifer properly dominating the target t. The constraint on UG in core derivations stated above is stronger, so the rule is applicable.

The application of UG on $\mathcal{A}(e)$ with x derives $\forall x\, \mathcal{A}(e)$. We need to show that the latter formula is identical to $(\forall x\, \mathcal{A})(e)$. Remember that the domain of e equals the set of variables bound by the proper ancestors of the target. The constraint on UG in core derivations therefore ensures that $x \notin \mathrm{dom}(e)$. It follows that $x(e) = x$ and therefore $\forall x\, \mathcal{A}(e) \equiv (\forall x\, \mathcal{A})(e)$.

(The condition imposed by DT and Rule C subderivations only disallows UG over variables which are effectively *universally* bound by a proper ancestor of the target. They allow UG over variables which are effectively existentially bound by a proper ancestor of the target, simply because these variables cannot occur in the assumption $\gamma_{i,-1}$, as they were mapped to the new individual constants of D in (T5). However, if x is a variable effectively existentially bound by a proper ancestor of the target, then $e(x) \in D$ and therefore (contrary to the conclusion of the previous paragraph) $\forall x\, \mathcal{A}(e) \not\equiv (\forall x\, \mathcal{A})(e)$ whenever $\mathcal{A}$ contains a free occurrence of x. It is therefore crucial that point 4a of Definition 18 disallows UG over variables which are effectively existentially bound by a proper ancestor of the target.)

(R5) The core derivation may use derived rules and theorems. We need not prove the their validity here. Their definitions or proofs construct or at least prove the existence of a derivation which does not employ them. Ultimately, we are provided with a core derivation which employs only Modus Ponens and Universal Generalization, whose validity as core derivation rules was proved above.

Take Universal Instantiation as an example. UI is a combination of Axiom Introduction and Modus Ponens. Assume a step $\alpha :\equiv \forall x\, \mathcal{A} \vdash \mathcal{A}(t/x)$ in the core derivation, where t is free for x in $\mathcal{A}$. In the decomposition of UI, we first introduce $\beta :\equiv \forall x\, \mathcal{A} \Rightarrow \mathcal{A}(t/x)$, an instance of Axiom Scheme $(\forall x\colon\ \varphi(x)) \Rightarrow \varphi(t)$, and then apply MP to α and β, yielding $\mathcal{A}(t/x)$. In the L' derivation, we first introduce $\beta(e) \equiv \forall x\, \mathcal{A}(e') \Rightarrow \mathcal{A}(e')(t(e)/x)$, where $e' := e|_{\mathrm{dom}(e)-\{x\}}$ and then apply MP to $\alpha(e)$ and $\beta(e)$, arriving at $\mathcal{A}(t/x)(e)$; see (R1) for details. Summing up, the L'-correspondant to the original UI with t/x is again an instance of UI, but with $t(e)/x$.

Let α be the last line of the core derivation. Given requirement 4b of Definition 18, we have $\alpha \equiv \neg^{\tau} \epsilon$ for some formula ϵ (the replacement). Obviously, we have just derived $\neg^{\tau} \epsilon(e)$.

A.2.5 The ascent from the target

For $i \in \mathbb{N}_0^n$, let $t''_i := \varphi(\epsilon/t)(t^i, u)$. Obviously, $t''_0 = \varphi(\epsilon/t)$. Note that for every $i \in \mathbb{N}_0^{n-1}$, $\mathbb{H}(t''_i) = \mathbb{H}(\varphi(\epsilon/t)(t^i)) = \mathbb{H}(\varphi(t^i)) = \mathbb{H}(t'_i)$, and when $\mathbb{H}(t''_i) \in \{\forall, \exists\}$, also $\mathbb{B}(t''_i) = \mathbb{B}(\varphi(\epsilon/t)(t^i)) = \mathbb{B}(\varphi(t^i)) = \mathbb{B}(t'_i)$. Applying point (e) of Lemma 2 to $\varphi(\epsilon/t)(t, u)$ and e, we also see that $t''_n = \varphi(\epsilon/t)(t, u) \equiv \varphi(\epsilon/t)(t)(e) \equiv \epsilon(e)$.

At each step below, we assume $\zeta_{i,0} \equiv \neg^{\tau_i} t''_i$ and derive $\zeta_{i-1,0} \equiv \neg^{\tau_{i-1}} t''_{i-1}$. The lines of the ascent are ordered in the reverse order in the first coordinate: ζ_{i_1,j_1} precedes ζ_{i_2,j_2} iff $i_1 > i_2$, or $i_1 = i_2$ and $j_1 < j_2$. The first step: $\zeta_{n,0} :\equiv \neg^{\tau} \epsilon(e) \equiv \neg^{\tau_n} t''_n$; justification: Reiteration (of the final line produced in the derivation of the replacement). For each $i \in \langle n, \ldots, 1\rangle$:

(B1) If $\mathbb{H}(t''_{i-1}) = \neg$:
By construction of t''_i, we have $t''_i \equiv t''_{i-1}(\langle t_i \rangle)$ (and $t_i = 1$), so $t''_{i-1} = \neg t''_i$. Clearly, $\tau_{i-1} = -\tau_i$.

(a) If $\tau_i = +$ (and $\tau_{i-1} = -$), let $\zeta_{i-1,0} :\equiv \neg\neg\zeta_{i,0} \equiv \neg\neg\neg^{\tau_i} t''_i \equiv \neg\neg t''_i \equiv \neg t''_{i-1} \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Replacement (by Double Negation) of $\zeta_{i,0}$.

(b) If $\tau_i = -$ (and $\tau_{i-1} = +$), let $\zeta_{i-1,0} :\equiv \zeta_{i,0} \equiv \neg^{\tau_i} t''_i \equiv \neg t''_i \equiv t''_{i-1} \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Reiteration.

(B2) If $\mathbb{H}(t''_{i-1}) = \wedge^{\tau_{i-1}}$:
By construction of t''_i, we have $t''_i \equiv t''_{i-1}(\langle t_i \rangle)$. Consider the other immediate constituent of t''_{i-1}: by point (d) of Lemma 2, $t''_{i-1}(\langle 3 - t_i \rangle) \equiv t'_{i-1}(\langle 3 - t_i \rangle) \equiv \beta_i$; also remember that $\neg^{\tau_i}\beta \equiv \gamma_{i,-1}$, see (T2). Clearly, $\tau_{i-1} = \tau_i$. We only show the case of $t_i = 1$, where $t''_{i-1} \equiv t''_i \wedge^{\tau_{i-1}} \beta_i$; reverse the order of conjuncts/disjuncts for $t_i = 2$.

(a) $\zeta_{i-1,-1} :\equiv \zeta_{i,0} \wedge \gamma_{i,-1} \equiv \neg^{\tau_i} t''_i \wedge \neg^{\tau_i}\beta_i$; justification: Conjunction Introduction.

(b) i. If $\tau_{i-1} = \tau_i = +$, let $\zeta_{i-1,0} :\equiv \zeta_{i-1,-1} \equiv \neg^{\tau_i} t''_i \wedge \neg^{\tau_i}\beta_i \equiv t''_i \wedge^{\tau_{i-1}} \beta_i \equiv t''_{i-1} \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Reiteration.

ii. If $\tau_{i-1} = \tau_i = -$, let $\zeta_{i-1,0} :\equiv \neg(t''_i \vee \beta_i) \equiv \neg^{\tau_{i-1}}(t''_i \wedge^{\tau_{i-1}} \beta_i) \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Replacement (by De Morgan's law) from $\zeta_{i-1,-1} \equiv \neg^{\tau_i} t''_i \wedge \neg^{\tau_i}\beta_i \equiv \neg t''_i \wedge \neg\beta_i$.

(B3) If $\mathbb{H}(t''_{i-1}) = \vee^{\tau_{i-1}}$:
By construction of t''_i, we have $t''_i \equiv t''_{i-1}(\langle t_i \rangle)$. Consider the other immediate constituent of t''_{i-1}: by point (d) of Lemma 2, $t''_{i-1}(\langle 3 - t_i \rangle) \equiv t'_{i-1}(\langle 3 - t_i \rangle) \equiv \beta_i$; also remember that $\neg\neg^{\tau_i}\beta_i \equiv \gamma_{i,-1}$, see (T3). Clearly, $\tau_{i-1} = \tau_i$.

(a) $\zeta_{i-1,-4} :\equiv \gamma_{i,-1} \Rightarrow \zeta_{i,0}$; justification: Deduction Theorem from $\gamma_{i,-1}$ to $\zeta_{i,0}$. $\zeta_{i,0}$ closes the Deduction Theorem subderivation opened in (T3) by derivation of $\gamma_{i,-1}$ as a DT assumption. To justify $\zeta_{i-1,-4}$, the conclusion of DT, we need to show that none of the lines of the subderivation involved an application of UG over a variable free in the assumption. To see that this is the case, consider (P6), (R4) and (B4), the only points in the subderivation where UG was applied.

(b) $\zeta_{i-1,-3} :\equiv \neg\gamma_{i,-1} \vee \zeta_{i,0} \equiv \neg\neg\neg^{\tau_i}\beta_i \vee \zeta_{i,0}$; justification: Replacement (by Material Implication) from $\zeta_{i-1,-4}$.

(c) $\zeta_{i-1,-2} :\equiv \neg^{\tau_i}\beta_i \vee \zeta_{i,0} \equiv \neg^{\tau_i}\beta_i \vee \neg^{\tau_i} t''_i$; Replacement (by Double Negation) from $\zeta_{i-1,-3}$.

(d) i. If $t_i = 1$: $\zeta_{i-1,-1} :\equiv \neg^{\tau_i} t''_i \vee \neg^{\tau_i}\beta_i$; justification: Replacement (by Commutation) from $\zeta_{i-1,-2}$.

ii. If $t_i = 2$: $\zeta_{i-1,-1} :\equiv \zeta_{i-1,-2} \equiv \neg^{\tau_i}\beta_i \vee \neg^{\tau_i} t''_i$ Reiteration.

(e) We only show the case of $t_i = 1$, where $t''_{i-1} \equiv t''_i \vee^{\tau_{i-1}} \beta_i$; reverse the order of conjuncts/disjuncts for $t_i = 2$.

i. If $\tau_{i-1} = \tau_i = +$, let $\zeta_{i-1,0} :\equiv \zeta_{i-1,-1} \equiv \neg^{\tau_i} t''_i \vee \neg^{\tau_i}\beta_i \equiv t''_i \vee^{\tau_{i-1}} \beta_i \equiv t''_{i-1} \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Reiteration.

ii. If $\tau_{i-1} = \tau_i = -$, let $\zeta_{i-1,0} :\equiv \neg(t''_i \wedge \beta_i) \equiv \neg^{\tau_{i-1}}(t''_i \vee^{\tau_{i-1}} \beta_i) \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Replacement (by De Morgan's law) from $\zeta_{i-1,-1} \equiv \neg t''_i \vee \neg\beta_i$.

(B4) If $\mathbb{H}(t''_{i-1}) = \forall^{\tau_{i-1}}$:
Let $x := \mathbb{B}(t''_{i-1})$ and $\beta :\equiv t''_{i-1}(\langle t_i \rangle)$ (we have $t_i = 1$); so $t''_{i-1} \equiv \forall^{\tau_{i-1}} x\,\beta$. By construction of t''_i and definition of u, we have $t''_i \equiv \beta(u(i-1)/x) \equiv \beta(x/x) \equiv \beta$, therefore $t''_{i-1} \equiv \forall^{\tau_{i-1}} x\, t''_i$. Clearly, $\tau_{i-1} = \tau_i$.

(a) $\zeta_{i-1,-1} :\equiv \forall x \neg^{\tau_i} t''_i$; justification: Universal Generalization from $\zeta_{i,0} \equiv \neg^{\tau_i} t''_i$ on x.

Let us prove that the application of UG is allowed by conditions imposed by Deduction Theorem (T3) and Rule C (T5) subderivations. As we are proceeding bottom-up, any currently open subderivation was opened during the derivation of $\gamma_{j,0}$ for some $j < i$, and disallows only UG over:

- variables free in φ: as t is properly quantified in φ (see clause (b) of Definition 16) and $x = \mathbb{B}(\varphi(t^{i-1}))$ is bound by a proper ancestor of the target, x cannot be free in φ; and
- variables from $\{\mathbb{B}(\varphi(j')); j' \in \mathbb{N}_0^{j-1} \wedge \mathbb{H}(\varphi(t^{j'})) = \forall^{\tau_{j'}}\}$: as t is properly quantified in φ, there is (by point (f) of Lemma 2) no $i' \in \mathbb{N}_0^{n-1} - \{i\}$ such that $x = \mathbb{B}(\varphi(t^{i'-1}))$; specifically, there is no such $i' < i$; therefore, as $j < i$, there is no prohibition against UG over $x = \mathbb{B}(\varphi(t^{i-1}))$.

(b) i. If $\tau_{i-1} = \tau_i = +$, let $\zeta_{i-1,0} :\equiv \zeta_{i-1,-1} \equiv \forall x \neg^{\tau_i} t''_i \equiv \forall x\, t''_i \equiv \neg^{\tau_{i-1}} \forall^{\tau_{i-1}} x\, t''_i \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Reiteration.

ii. If $\tau_{i-1} = \tau_i = -$, let $\zeta_{i-1,0} :\equiv \neg\exists x\, t''_i \equiv \neg^{\tau_{i-1}} \forall^{\tau_{i-1}} x\, t''_i \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Replacement (by Quantifier Negation) from $\zeta_{i-1,-1} \equiv \forall x \neg^{\tau_i} t''_i \equiv \forall x \neg t''_i$.

(B5) If $\mathbb{H}(t''_{i-1}) = \exists^{\tau_{i-1}}$:

Let $x := \mathbb{B}(t''_{i-1})$ and $\beta :\equiv t''_{i-1}(\langle t_i \rangle)$ (we have $t_i = 1$); so $t''_{i-1} \equiv \exists^{\tau_{i-1}} x \beta$. By construction of t''_i and definition of u, we have $t''_i \equiv \beta(u(i-1)/x) \equiv \beta(d_{i-1}/x)$. Clearly, $\tau_{i-1} = \tau_i$.

(a) $\zeta_{i-1,-1} :\equiv \exists x \neg^{\tau_i} \beta$; justification: Existential Generalization from $\zeta_{i,0} \equiv \neg^{\tau_i} t''_i \equiv (\neg^{\tau_i} \beta)(d_{i-1}/x)$, where d_{i-1}, being an individual constant, is obviously free for x in β.

By this line, we close the rule C subderivation opened in (T5) by the derivation of $\gamma_{i,0}$, so we need to show that the lines from $\gamma_{i,0}$ to $\zeta_{i-1,-1}$ conform to the requirements for Rule C derivation given in Definition 19. The only non-obvious requirements are given in points 3 and 4 of the definition.

- No Universal Generalization had as its quantified variable a free variable of $\gamma_{i,0}$. To see that this is the case, consider (P6), (R4) and (B4), the only points in the subderivation where UG was applied.
- $\varphi(\epsilon/t)$ is a formula of L, i.e. it does not contain any individual constant from D. The conclusion of the subderivation is $\zeta_{i-1,-1} \equiv \exists x \neg^{\tau_i} t''_{i-1}(\langle t_i \rangle)$, which, given the definition of u and the construction of α_i in Definition 38, contains no individual constants d_j for $j \geq i-1$.

(b) i. If $\tau_{i-1} = \tau_i = +$, let $\zeta_{i-1,0} :\equiv \zeta_{i-1,-1} \equiv \exists x \neg^{\tau_i} \beta \equiv \exists x\, \beta \equiv \neg^{\tau_{i-1}} \exists^{\tau_{i-1}} x\, \beta \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Reiteration.

ii. If $\tau_{i-1} = \tau_i = -$, let $\zeta_{i-1,0} :\equiv \neg\forall x\, \beta \equiv \neg^{\tau_{i-1}} \exists^{\tau_{i-1}} x\, \beta \equiv \neg^{\tau_{i-1}} t''_{i-1}$; justification: Replacement (by de Quantifier Negation) from $\zeta_{i-1,-1} \equiv \exists x \neg^{\tau_i} \beta \equiv \exists x \neg \beta$.

The final line reads $\zeta_{0,0} \equiv \neg^{\tau_0} t''_0 \equiv \varphi(\epsilon/t)$. Clearly, all subderivations were closed. Summing up, assuming φ, we have derived $\varphi(\epsilon/t)$.

A.2.6 Validity of individual inline rules

Having proven Theorem 1, which proves that the design of DDS presented in section A.1 leads to valid inferences, we can now apply this theorem to the rules we are using in the book. We do this by exhibiting the core deductions powering these inline rules. We first deal with the simpler cases of Add, Delete, and Copy, and then proceed to Prune, which requires some extra work.

However, before proceeding, we need to address a detail concerning the status of Add, Delete, and Copy as canonical inline rules. Strictly speaking, these rules are not canonical after all. While a canonical inline rule (see Definition 15 in section A.1) requires fixed p-scope polarities of the target and the premises, our familiar inline rules are all sensitive to polarity, in the sense that the forms of the target and the premises "automatically" adapt to various polarity configurations. However, the discrepancy is not really problematic. While our official inline rules are not canonical, they are *unions* of canonical inline rules, one for each polarity configuration. Addressing the complication is therefore very simple; we merely need to exhibit several (very similar) core deductions for each of our inline rules. The formal setup presented in section A.1 was chosen because it facilitates the proof of soundness in section A.2, and while it demands a bit of extra work in this section, this extra work (in the form of multiple core deductions for *individual* inline rules) is negligible to the amount of extra machinery we would need to bring in to resolve the issue generally.

We emphasize that seeing Add, Delete, and Copy as composed of several canonical inline rules does not negate their status of *basic*, i.e. non-derived rules in the rule system of DDS. The derived rules of DDS are those which we use to abbreviate sequences of basic rules (for example, like the center Copy abbreviates the sequence of right Copy and left Delete). The decomposition of Add, Copy, and Delete into canonical inline rules has nothing to do with the application of the system; it is merely an aid toward a mathematical goal, the proof of validity of these rules. The necessity, or at least the facility of the decomposition is an artefact of our decision to execute the proof of soundness by simulation of DDS—a polarity-sensitive deductive system—in a Hilbert system—a system not sensitive to polarity.

We are finally ready to write down the core deductions of Add, Delete, and Copy. Add and Delete, depending only on target polarity, depend on two core deductions each, one for positive target polarity and one for negative target polarity. Copy comprises four core deductions, as not only the target polarity, but also the p-scope polarity of the premise can vary. However, the multiple core deductions are compressed below: the lines with the note starting by "if" are to be included only if the polarity condition is satisfied.

When reading these core deductions, remember that a target can be introduced into the deduction "as is" when its polarity is positive, but must be negated when its polarity is negative; in other words, if we have a target t of polarity τ, the form of the target hypothesis is $\neg^\tau t$. In a similar fashion (but with p-scope polarity instead of constituent polarity), given a premise p with p-scope polarity σ at the target, the form of the premise hypothesis is $\neg^\sigma p$.

(7) Add: $\vdash^\tau_{\mathcal{A}} \mathcal{A} \vee^\tau \mathcal{B}$

1.	$\neg^\tau \mathcal{A}$	(target hypothesis)
2.	$\neg^\tau \mathcal{A} \vee \neg^\tau \mathcal{B}$	(Disjunction Introduction)
3.	$\neg(\mathcal{A} \wedge \mathcal{B})$	(If $\tau = -$: De Morgan)

(8) Delete: $\vdash^\tau_{\mathcal{A}\wedge^\tau\mathcal{B}} \mathcal{B}$

1.	$\neg^\tau(\mathcal{A} \wedge^\tau \mathcal{B})$	(target hypothesis)
2.	$\neg\mathcal{A} \wedge \neg\mathcal{B}$	(If $\tau = -$: De Morgan)
3.	$\neg^\tau \mathcal{B}$	(Conjunction Elimination)

(9) Copy: $\mathcal{A}^{\sigma} \vdash^{\tau}_{\mathcal{B}} \mathcal{B} \wedge^{\tau} \neg^{\sigma\tau} \mathcal{A}$

1. $\neg^{\tau} \mathcal{B}$ (target hypothesis)
2. $\neg^{\sigma} \mathcal{A}$ (premise hypothesis)
3. $\neg^{\tau} \mathcal{B} \wedge \neg^{\sigma} \mathcal{A}$ (Conjunction Introduction)
4. $\neg \mathcal{B} \wedge \neg\neg \mathcal{A}$ (If $\tau = -$ and $\sigma = +$: Double Negation Introduction)
5. $\neg(\mathcal{B} \vee \neg^{-\sigma} \mathcal{A})$ (If $\tau = -$: De Morgan)

We are now ready to turn to Prune. Prune is neither a canonical inline rule, nor a union of such rules, because the two roles of the target are divorced. We have one constituent which is p-scoped over by the premise (we continue to call this constituent the target), and we have another constituent which gets eliminated: the local e-disjunct of the target. Canonical inline rules are simply not equipped to deal with this situation.

Prune

$\mathcal{A}^{-\tau} \vdash^{\tau}_{\mathcal{A}}$ eliminate the local e-disjunct of the target

Some extra work is required to prove the validity of Prune. There are at least two ways to go about it. One way would be to show that Prune has the same effect as a sequence of some canonical inline rules—in particular, Prune can be derived using Copy, Add, and the inline versions of *ex falso quodlibet*, $\mathcal{A}^{-\tau} \vdash^{\tau}_{\mathcal{A}} \mathcal{B}$, and idempotency of disjunction, $\vdash^{\tau}_{\mathcal{A} \vee^{\tau} \mathcal{A}} \mathcal{A}$ (which rules the sequence contains, and in what order, can in principle depend on polarity of the target and p-scope polarities of the premises). We will, however, opt for a simpler approach, which amends the generic proof of validity from this section.

Here, the idea is to modify the ascent from the target in subsection A.2.5 by skipping the part of the ascent up to the local e-disjunct of the target. Why may we do this? First, nothing forces us to apply points (B1) (negations), (B2) (e-conjunctions), and (B4) (e-universal quantifiers) of the ascent. Second, observe that there are no e-disjunctions in the part of the ascent until the local e-disjunct, because the local disjunct is by definition the child of the closest e-disjunction; in effect, no Deduction Theorem subderivations were opened in this part of the tree during descent, so there are none to close in the ascent, i.e. point (B3) would not be applied even in the regular ascent. What about (B5), the point which applies at e-existential quantifiers and whose function is to close Existential Instantiation subderivations? Here, the idea is to choose a good replacement. If we choose to replace the target by the sister of the local e-disjunct (call it S), then this S (whatever its form) does not contain any constants introduced by EI between the local e-disjunct and the target. In effect, the EI-subderivations of the skipped part are already closed. The first step in our modified ascent is therefore the closure of the Deduction Theorem of the closest e-disjunction. Given S as the replacement, this step produces $S \vee S$, which we can then simplify to S by Idempotency. In effect, we have eliminated the closest e-disjunct of the target, which is of course precisely what Prune does.

A.3 Directional Entailingness and Polarity

Despite the fact that the relation between directional entailingness (aka monotonicity) and polarity is well known, there seems to be no result (and even less a proof thereof) aimed specifically toward directional reasoning in predicate logic(s). Van Benthem (1984, p. 462, 1986, p. 38) attributes the observation that Directional Entailingness Theorem is a simple consequence of Lyndon's Theorem to Kit Fine, but nothing seems to have been published on the subject. So let us see how we get from the latter to the former. (We only deal with the semantics-to-syntax direction here; the other (easier) direction follows from the soundness of DDS anyway.)

It makes sense to start with Craig's Interpolation Theorem. This theorem, first proven by Craig (1957b) for first-order predicate logic, is one of the essential results in mathematical logic. Roughly speaking, it asserts that whenever φ entails ψ, there is intermediate formula θ (in the sense that φ entails θ and θ entails ψ), called an *interpolant*, which uses only the symbols common to both φ and ψ. The theorem holds both for sentences and for open formulas. Furthermore, it holds for predicate logic with or without equality.

Theorem 4 (Craig's Interpolation Theorem)**.** *Let φ and ψ be arbitrary formulas such that at least one non-logical[7] predicate symbol occurs in both φ and ψ. If $\varphi \models \psi$, then there is a formula θ such that:*

1. *$\varphi \models \vartheta$ and $\vartheta \models \psi$, and*
2. *if a non-logical symbol[8] occurs in ϑ, then it occurs in φ and ψ as well.*

If φ and ψ share no non-logical predicate symbol, then either $\models \neg\varphi$ or $\models \psi$.

Our concern, however, is not directly with Craig's Theorem, but rather with its extension, Lyndon's Interpolation Theorem, also known as Craig-Lyndon's Theorem. This is so because Lyndon's version imposes polarity restrictions on the interpolant. In the remainder of the section, we therefore assume to be working with a language of predicate (or propositional) logic which admits conjunction, disjunction, and negation as the only propositional connectives. Other connectives can be defined as abbreviations.

Craig's and/or Lyndon's Interpolation Theorem have been proven for many logics, and their original formulations have been extended as well. There are simply too many works on the theorems to list them all here, so we merely provide a few pointers:

- the original formulation and proof: Lyndon (1959);
- first-order predicate logic corrections and extensions: Henkin (1963), Oberschelp (1968), Motohashi (1984), and Gherardi et al. (2019);
- extension to infinitary logic $L_{\omega_1\omega}$: Lopez-Escobar (1965) and Makkai (1969);
- an overview of applicability to various logics (mostly for Craig's Theorem): Väänänen (2008).

Finally, we present the theorem itself. Note that we will only use the first three points of the theorem in our proof.

Theorem 5 (Lyndon's Interpolation Theorem)**.** *Let φ and ψ be arbitrary formulas such that $\varphi \models \psi$.[9] There is then a formula ϑ such that:*

1. *$\varphi \models \vartheta$ and $\vartheta \models \psi$;*
2. *if a relation symbol positively occurs in ϑ, then it positively occurs in φ and ψ as well;*

[7] This excludes equality.
[8] Including individual constants and function symbols in general.
[9] The formulations of Lyndon's Theorem in the literature omit the assumption of Craig's Theorem that at least one non-logical predicate symbol occurs in both φ and ψ. Strictly speaking, the omission is possible only when there are ways to represent true (false) without using any predicate symbols, for example as an independently defined zero-place connective or by allowing empty conjunctions (or disjunctions); this option is chosen by Lopez-Escobar (1965, p. 256).

3. *if a relation symbol negatively occurs in ϑ, then it negatively occurs in φ and ψ as well;*
4. *if the equality symbol positively occurs in ϑ, then it positively occurs in φ;*
5. *if the equality symbol negatively occurs in ϑ, then it negatively occurs in φ;*
6. *if an individual constant or function symbol occurs in ϑ, then it occurs in φ and ψ as well; and*
7. *if a variable occurs freely in ϑ, then it occurs freely in φ and ψ as well.*

Having laid out the framework, we can proceed with the formulation and proof of the Directional Entailingness Theorem. We start by developing a technical aid, the notion of predicate substitution.

Definition 39. Let C be some set of constituents of formula φ, and $<$ a linear order of variable symbols occurring in φ. A mapping $f\colon C \to \mathbb{P}$ is a PREDICATE SUBSTITUTION on $(\varphi, <)$ iff for every $c \in C$, the arity of $f(c)$ matches the number of free variables in $\varphi(c)$. Then define $g\colon C \to \mathbb{F}$ by $g(c) := f(c)(x_1, \ldots, x_k)$, where $x_1, \ldots, x_k$ are the free variables of $\varphi(c)$ such that $x_1 < \ldots < x_k$. Finally, set $\varphi(f) := \varphi(g)$.

The idea behind predicate substitution is to define a simple notation for replacing (possibly complex) constituents of a formula by an atomic formula. To do so, we need a way to refer to constituents: for example, $\langle 2, 1\rangle$ below refers to the constituent of form $A(x) \wedge D(x)$ in φ—the universal quantifier is the second argument of the existential quantifier, while its restrictor is its first argument; for details, see subsection A.2.1. Next, the substitution, defined simply as "replacement by P," must pick out the free variables of the constituent (in some fixed order, thus the linear order $<$) and use them as the arguments of the predicate P: observe below that Q receives e and y (in that order) as its arguments.

(10) a. Every angry dog loudly barks at some cat.
b. $\varphi :\equiv \exists e\,[B(e)]\,\forall x\,[A(x) \wedge D(x)]\,\mathrm{Ag}(x, e) \wedge \exists y\,\big[C(y)\big]\,\mathrm{Th}(y, e) \wedge L(e)$
c. $f(\langle 2, 1\rangle) = P, f(\langle 2, 2, 2, 2\rangle) = Q, e < x < y$
d. $\varphi(f) \equiv \exists e\,[B(e)]\,\forall x\,[P(x)]\,\mathrm{Ag}(x, e) \wedge \exists y\,\big[P(y)\big]\,Q(e, y)$

The notion of directional entailingness is really bound to positions in a formula. This is made explicit in the definition below. First, for a position to be upward/downward entailing, the substitution for *any* hypernym/hyponym must be possible.

Second, while it is clear that repeated application of appropriate substitutions in different directionally entailing positions results in entailment, it is not trivially true (although it is true, as we shall prove) that the success of a simultaneous substitution in these positions is always powered by the fact that we are substituting for hypernyms and hyponyms. (This fact is even less trivial in the light of the third point below.) This is why the definition below is complicated by allowing for multiple simultaneous substitutions.

Third, and most importantly, a successful substitution for a hypernym/hyponym does not count as a case of directional entailing if it is achievable without that assumption. In (11a), the first premise is false, so anything follows without any extra assumptions anyway, as shown in (11b). Similarly in (12a), the first premise is a tautology, so A may be replaced by anything, again with no additional assumptions (as long as we replace both occurrences of A with the same formula). The fact that both conclusions result from substituting B for A in both cases does not yet make them instances of directional entailing. The notion of minimally directionally entailing positions defined in the second point of the definition excludes such anomalous cases.

(11) a. $A \wedge (\neg C \wedge C), A \Rightarrow B \therefore B \wedge (\neg C \wedge C)$
b. $A \wedge (\neg C \wedge C) \therefore B \wedge (\neg C \wedge C)$

(12) a. $\neg A \vee A, A \Rightarrow B, B \Rightarrow A \therefore \neg B \vee B$
b. $\neg A \vee A \therefore \neg B \vee B$

Definition 40. For a formula φ, a linear order $<$ of variables in φ, constituent c of φ and predicate P whose arity matches the number of free variables in $\varphi(c)$, let

$$\gamma(c, P) := \forall x_1 \ldots \forall x_k (\neg\varphi(c) \vee P(x_1, \ldots, x_k)),$$
$$\delta(c, P) := \forall x_1 \ldots \forall x_k (\neg P(x_1, \ldots, x_k) \vee \varphi(c)),$$

where $x_1, \ldots, x_k$ are the free variables of $\varphi(c)$ such that $x_1 < \ldots < x_k$.

1. Let φ be an arbitrary formula, and U and D disjoint sets of constituents of φ. (U, D) are DIRECTIONALLY ENTAILING POSITIONS of φ iff

$$\bigwedge_{c \in U} \gamma(c, u) \wedge \bigwedge_{c \in D} \delta(c, d) \wedge \varphi \vDash \varphi(u \cup d)$$

for some linear order $<$ of variables in φ, and any predicate substitutions $u\colon U \to \mathbb{P}$ and $d\colon D \to \mathbb{P}$ on $(\varphi, <)$.[10]
2. Directionally entailing positions (U, D) of φ are MINIMAL iff $U' = U$ and $D' = D$ holds for any directionally entailing positions (U', D') of φ such that $U' \subset U$ and $D' \subset D$.

We are now finally ready to formulate and prove the theorem.

Theorem 6 (Directional Entailingness). *Let U (for "up") and D (for "down") be sets of constituents of some formula φ. Assume that all members of U and D are atomic formulas and that each predicate heading a member of U or D has a single occurrence in φ.*[11]

[10] Note that the above conjunction can be formed in infinitary logic as well, in particular in $L_{\omega_1\omega}$. On one hand, sets U and D might be infinite; but on the other hand, we may form infinite conjunctions. Note also that as we can characterize directional entailingness in terms of a conjunction as opposed to a set of premises, the fact that Lyndon's Theorem does not hold for sets of formulas (Lopez-Escobar 1965, p. 254) is irrelevant for our purposes.

[11] The assumptions on U and D merely facilitate the statement of the conclusion. The point is, ϑ may contain some constituent which is not a "reflex" of any constituent of U or D, but accidentally has the same form as one of these constituents. Of course, nothing guarantees that this constituent will have the polarity required by the conclusion. In absence of the initial assumptions, a formulation of the conclusion referring only to the form of the members of U and D is therefore impossible.

Of course, the formulation is not impossible in general. Taking a deductive view for a moment, the conclusion could be formulated by defining a constrained kind of deduction where decomposing the members of U and D would be disallowed, and asserting the existence of ϑ derivable from φ, and vice versa, by means of this constrained kind of deduction. Of course, such a deduction would need to keep track of the members of U and D. Presumably, the easiest way to do that would be to replace them by novel predicates and run a standard deduction, bringing us back to the sensibility of the assumptions. Cf. also the method used in Henkin's (1963) proof of his extension of Lyndon's Theorem.

Finally, note that the assumptions are only used in the final paragraph of the proof.

If (U, D) are minimally directionally entailing positions of φ, then φ is logically equivalent to formula ϑ such that:

1. *a predicate heading a member of U or D has an occurrence in ϑ,*
2. *all occurrences of a predicate heading a member of U in ϑ are positive, and*
3. *all occurrences of a predicate heading a member of D in ϑ are negative.*

Proof. Choose any linear order $<$ of variables in φ and predicate substitutions $u\colon U \to \mathbb{P}$ and $d\colon D \to \mathbb{P}$ on $(\varphi, <)$ such that $u(U)$ and $d(D)$ are disjoint and contain no predicates occurring in φ (if necessary, introduce new predicate symbols into the language).

Let $\gamma := \bigwedge_{c \in U} \gamma(c, u)$ and $\delta := \bigwedge_{c \in D} \delta(c, d)$. Then, we have $\gamma \wedge \delta \wedge \varphi \vDash \varphi(u \cup d)$. Now, by Lyndon's Theorem (plus Deduction Theorem), there is a formula ϑ such that:

- $\vDash \gamma \wedge \delta \wedge \varphi \Rightarrow \vartheta$ and $\vDash \vartheta \Rightarrow \varphi(u \cup d)$,
- if $P \in \mathbb{P}^+(\vartheta)$, then $P \in \mathbb{P}^+(\varphi)$ and $P \in \mathbb{P}^+(\varphi(u \cup d))$, and
- if $P \in \mathbb{P}^-(\vartheta)$, then $P \in \mathbb{P}^-(\varphi)$ and $P \in \mathbb{P}^-(\varphi(u \cup d))$.

Assume that for some $c \in U$, $u(c)$ has no occurrence in ϑ. We know there is a single occurrence of $u(c)$ in $\gamma \wedge \delta \wedge \varphi$, the occurrence in $\gamma(c, u)$. Replacing $u(c)$ by $\varphi(c)$ in $\vDash \gamma \wedge \delta \wedge \varphi \Rightarrow \vartheta$ thus changes only $\gamma(c, u)$, which becomes a tautology. This means we have $\vDash \gamma' \wedge \delta \wedge \varphi \Rightarrow \vartheta$, where $U' := U - \{c\}$, $u' := u|_{U'}$ and $\gamma := \bigwedge_{c \in U'} \gamma(c, u')$. As we know that $\vDash \vartheta \Rightarrow \varphi(u \cup d)$, we have $\gamma' \wedge \delta \wedge \varphi \vDash \varphi(u \cup d)$. Therefore (U', D) are directionally entailing positions of φ, which contradicts the assumption of minimality of (U, D). Summing up, all predicates from $u(U)$ occur in ϑ, and by the same reasoning, we can conclude that all predicates from $d(D)$ occur in ϑ as well.

If $u(c)$ has a negative occurrence in ϑ for some $c \in U$, it has a negative occurrence in $\gamma \wedge \delta \wedge \varphi$ as well. However, we know that $u(c)$ does not occur in δ and φ at all, and that it occurs only positively in γ. All predicates from $u(U)$ therefore occur positively in ϑ. In the same fashion, we can conclude that all predicates from $d(D)$ occur negatively in ϑ.

We have assumed $u(U)$ and $d(D)$ to be disjoint. The predicate substitution $u \cup d$ can therefore be inversed. As we have also assumed that none of the predicates in $u(U)$ and $d(D)$ occur in φ, applying $(u \cup d)^{-1}$ to $\vDash \gamma \wedge \delta \wedge \varphi \Rightarrow \vartheta$ thus does not affect φ, while transforming all the conjuncts of γ and δ into tautologies. We thus get $\vDash \varphi \Leftrightarrow \vartheta'$, where $\vartheta' :\equiv \vartheta((u \cup d)^{-1})$.

As we have required $\mathbb{H}(\varphi(U \cup D)) \subset \mathbb{P}$ and that each predicate heading a member of $\varphi(U \cup D)$ occurs only once in φ, ϑ' clearly satisfies the conditions given (for ϑ) in the conclusion of the theorem. ∎

A.4 Conservativity and Restrictedness

In the previous section we noted that the relation between directional entailingness and polarity is recognized but not explicitly proven in the literature. We observe practically the same situation in the case of conservativity and restrictedness. Again, it is van Benthem (1984, p. 462, 1986, p. 38) who attributes to Kit Fine the observation that these notions are intimately related, in particular, that the equivalence between conservativity and restrictedness follows from the work of Feferman (1968). And indeed, the semantics-to-syntax direction of our proof of Restrictedness Theorem will essentially recast Feferman's result in terms of our generalized conservativity.

The proofs of the validity of syntactic characterizations of semantic properties are typically much harder in the semantics-to-syntax direction; the characterization of conservativity in terms of restrictedness is no exception. In the mathematical literature, the simpler syntax-to-semantics direction is often omitted as trivial, but we will nevertheless explicate the proof of this direction as well. Our most general notion of restrictedness (weak restrictedness) encompasses a broader class of formulas than the usual notion (which we have dubbed atomic restrictedness). Consequently, the proof, while still straightforward, is not as trivial as in the atomic case.

The assumptions on the formal language (L) in this section are as follows. L is a language of first-order logic with equality ($=$). As in the case of the relation between polarity and directional entailingness, we limit ourselves to languages with Boolean connectives $\wedge$, $\vee$ and $\neg$. In this case, the limitation is not crucial, but it greatly facilitates the definition of weakly restricted formulas. For simplicity, we assume that L is relational, i.e. it may contain arbitrary non-logical predicates and individual constants, but no operation symbols; any language can be translated to an equi-interpretable relational language anyway. Finally, L may be either a finitary language, or an infinitary language with countable conjunctions and disjunctions and finite strings of quantifiers, i.e. $L_{\omega_1\omega}$;[12] this flexibility does not add an iota of complexity to our account, as all the heavy lifting is done by Feferman's (1968) theorems used in the proofs.

In the main text, we have defined restrictedness and conservativity with respect to a set of monadic non-logical predicates, i.e. the parameter κ was a set consisting of monadic non-logical predicates. Below, we expand on that setup by defining these notions with respect to individual constants as well, i.e. we allow these symbols to be members of κ as well. The idea is that just as $D(x)$ can function as a restrictor, limiting the "sensible" values of x to (collections of) dogs, so can e.g. $x = j$ and $x \subset a$, limiting the value of x to John and collections of Americans. Clearly, not only the constant but also the (logical) predicate plays a role in such a restrictor. We cannot expect a restrictor containing an individual constant to be based on just any predicate; for example, $\#(5, x)$ of L** (for simplicity taking 5 to be an individual constant referring to the number five here) does not restrict the sensible values of x to some subset of the domain. Also note that while $=$ is symmetric, when it comes to $\subset$ the location of the constant matters; for example, $a \subset x$ again does not restrict x to some subset of the domain. To deal with this issue flexibly—in particular, covering the formal languages with or without $\subset$, or more generally, covering both the singular and the plural languages—we introduce a parameter into the definitions of restrictedness and conservativity, taking these notions to be relative to two designated sets of binary predicates of L: the set of LEFTWARD RESTRICTING predicates $\mathcal{R}_1$ (the canonical members being $=$ and $\subset$, yielding restrictors such as $x = j$ and $x \subset a$) and the set of RIGHTWARD RESTRICTING predicates $\mathcal{R}_2$ (the canonical members being $=$ and $\supset$, yielding restrictors such as $j = x$ and $a \subset x$). Note that either or both of $\mathcal{R}_1$ and $\mathcal{R}_2$ may be empty, and that the sets may overlap (for example, $=$ may be the member of both sets). For L**, we assume that $\mathcal{R}_1 = \{=, \subset\}$ and $\mathcal{R}_2 = \{=\}$.

One reason for allowing individual constants as members of κ is simply to expand the reach of restrictedness and conservativity (without any serious complications in the notation and the proofs). More importantly, however, this move hints at a crucial feature of the semantics-to-syntax direction of the proof of Restrictedness Theorem. For the purposes of that proof, restrictedness and conservativity with respect to monadic non-logical predicates will be translated into restrictedness and conservativity with respect to

[12] The results hold for some other infinitary languages as well, see Feferman (1968) for details.

individual constants. Furthermore, we will see that this direction of the proof imposes some (fairly general) constraints on the theory. Specifically, it requires that all the axioms of the theory are either universal or predicative. As we will elaborate below, L^{**} satisfies this requirement.

The roadmap for the section is as follows. We start on the syntactic side by defining our most general notion of restrictedness, the weak restrictedness. We then turn to the semantic side, laying out the notion of conservativity in model-theoretic terms. The two sides come together in the formulation of the Restrictedness Theorem, which asserts that any (κ-)conservative formula is logically equivalent to some (κ-)restricted formula. The rest of the section is dedicated to proving this theorem. We tackle the easier, restrictedness-to-conservativity direction first, and then use Feferman's results to prove the converse direction as well.

We begin the syntactic definitions by introducing the notions which facilitate the relativization of restrictedness (and later, conservativity) to both monadic predicates and individual constants. Given the definitions of c-signature and potential restrictor below, the class of potential restrictors, limited to $D(x)$ (being a dog), $C(x)$ (being a cat), etc., in the main text, now expands to contain expressions such as $x = j$ and $j = x$ (being John) or $x \subset a$ and $x \supset x$ (being an American). Remember that $\mathcal{R}_1$ and $\mathcal{R}_2$ are two designated sets of binary predicates of L. We will see that, as a consequence of Definition 46, only certain logical predicates, like equality, subset, and superset make sense as members of these sets.[13]

Definition 41. A CONSERVATIVITY SIGNATURE (C-SIGNATURE) is a set consisting of monadic non-logical predicate symbols and individual constant symbols.

Definition 42. Let κ be a c-signature and x a variable. An atomic formula is a POTENTIAL κ-RESTRICTOR FOR x iff it is of form

- $P(x)$ for some predicate symbol $P \in \kappa$,
- $x \subset c$ for some constant $c \in \kappa$ and some predicate $\subset \in \mathcal{R}_1$, or
- $c \supset x$ for some constant $c \in \kappa$ and some predicate $\supset \in \mathcal{R}_2$.

The following definition forms the very heart of our most general notion of restrictedness. It tells us what form a formula (φ) may have so that quantifying over a certain variable (x) will produce an instance of weakly restricted quantification. The parameter τ in the definition distinguishes between formulas which are existentially ($\tau = +$) restrictedly quantifiable and formulas which are universally ($\tau = -$) restrictedly quantifiable. Furthermore, the notion is relative with respect to the given c-signature κ, which determines the set of potential restrictors via the definitions given above.

Definition 43. Let κ be a c-signature, x a variable and τ a polarity. Formula φ is κ-RESTRICTEDLY τ-QUANTIFIABLE FOR x iff

1. $\tau = +$ and φ is a potential κ-restrictor for x,
2. $\varphi \equiv \neg\varphi'$ and φ' is κ-restrictedly $-\tau$-quantifiable for x,

[13] We are not intending to make any ontological commitments by referring to the subset and superset predicates. One can safely imagine subcollection, the "among" relation or such in their stead. We only choose to highlight the set-based predicates as the model-theoretic proofs we rely on are based on the set theory anyway.

3. $\varphi \equiv \wedge^{\tau}_{i\in I}\varphi_i$ and φ_i is κ-restrictedly τ-quantifiable for x for some $i \in I$,[14]
4. $\varphi \equiv \vee^{\tau}_{i\in I}\varphi_i$ and φ_i is κ-restrictedly τ-quantifiable for x for every $i \in I$, or
5. $\varphi \equiv \exists y\, \varphi'$ or $\varphi \equiv \forall y\, \varphi'$, where y is a variable distinct from x, and φ' is κ-restrictedly τ-quantifiable for x.

In Definition 44 below, the notion of a restrictedly quantifiable formula is used to define our central notion of weak restrictedness. Note that if we omitted clause 4 on e-disjunctions from Definition 43, Definition 44 would yield the strong (p-scope-based) restrictedness. And to yield atomic restrictedness, one needs to omit from Definition 43 all the clauses except clause 1 on atomic formulas and clause 2 on negation (allowing even the latter to "apply" only once). Clearly then, all atomically restricted formulas are strongly restricted, and all strongly restricted formulas are weakly restricted in turn.

Definition 44.

(a) Let κ be a c-signature. Formula φ is WEAKLY κ-RESTRICTED iff for every subformula of φ of form $\exists^{\tau}x\, \varphi'$, formula φ' is κ-restrictedly τ-quantifiable for x. (Note that every subformula of a weakly κ-restricted formula is clearly weakly κ-restricted itself.)
(b) Formula φ is WEAKLY RESTRICTED iff there is a c-signature κ such that φ is weakly κ-restricted.

We have based the discussion in the main text on the notions of restrictor, restricted quantifier, restriction, and nuclear scope. These notions play no role in the mathematical treatment in this section. This is why we will provide no technical definitions of these terms, but it is easy to see how they connect to the terminology of this section.

The set of RESTRICTORS of a given formula consists of the potential restrictors reached as the recursive basis (point 1) of the formula in Definition 43. As we were only dealing with strong restrictedness in the main text, each formula was always restricted by a single atomic formula. Here, however, we allow for e-disjunctions (point 4), so it can happen that a formula is not restricted by a single atomic formula, but only jointly by some set of potential restrictors. For example, $D(x) \vee C(x)$ is restricted neither by $D(x)$ nor by $C(x)$, but only jointly by both $D(x)$ and $C(x)$—a consequence of clause 4 of Definition 43, which requires that all members of an (e-)disjunction are restrictedly quantifiable.

We can rephrase the definition of a RESTRICTED QUANTIFIER from the main text by saying that it is a formula of form $\exists^{\tau}x\colon\ \varphi \wedge^{\tau} \psi$ such that some potential restrictor for x in φ positively p-scopes over ψ. (τ is the polarity determining the quantification type, existential for $\tau = +$ and universal for $\tau = -$.) Setting the singularity of the set of restrictors of the p-scope-based definition aside, another difference between the strong and the weak restrictedness is that the former, but not the latter, requires that quantifier $\exists^{\tau}$ immediately dominates junction $\wedge^{\tau}$—in a sense, a restricted quantifier is the fusion of an unrestricted quantifier and a junction of the appropriate type. Only in this setting, it makes sense to talk about the RESTRICTION (above, φ) as the member of the junction which contains the restrictor(s), and the NUCLEAR SCOPE (above, ψ) as the other member.

Take sentence 'a dog barks' as an example. This sentence receives the logical form $\exists x\colon\ D(x) \wedge B(x)$. Let us consider the quantifier-free part of this formula, $D(x) \wedge B(x)$. It is built by point 3 of Definition 43, so only one member of the conjunction must be restrictedly

[14] If the formal language allows only binary conjunctions and disjunctions, the current and the next point simplify to $\varphi \equiv \varphi_1 \wedge^{\tau} \varphi_2$ and $\varphi \equiv \varphi_1 \vee^{\tau} \varphi_2$, respectively.

quantifiable. This is of course $D(x)$, the restriction (and at the same time, the restrictor); the other member, $B(x)$, is the nuclear scope. Now consider a logically equivalent formula, $\neg(\neg D(x) \vee \neg B(x))$. Here, the topmost connective is not a junction at all, so it is impossible to break down the entire formula into two parts, one containing the restrictor $D(x)$ and the other not. Or consider another logically equivalent formula, $(D(x) \wedge B(x)) \vee (D(x) \wedge B(x))$ (we have simply doubled the entire formula into a disjunction). Again, the immediate division into the restrictor and the nuclear scope is impossible. (Clearly, this formula is not strongly restricted, but it is weakly restricted; each member of the topmost disjunction is D-restrictedly existentially quantifiable for x, so the entire formula is such as well.)

However, possibly the most important feature of our definition of restricted quantifiability (and thus weak restrictedness), a feature shared with the deployment of strong restrictedness in a quantifierless formal language (see section 9.3), is that a single formula may be restrictedly quantifiable for more than one variable—or in terms from the main text, that several restricted quantifiers may share the restriction. This feature was crucial for our analysis of donkey anaphora in section 11.1. Consider (13). The quantifier-free part of this formula is $\{F, D\}$-restrictedly quantifiable for both x and y, and this is what makes it possible to top it by both quantifiers (while yielding a weakly restricted formula). For one, clause 5 of Definition 43 makes sure that once we quantify over y, the result remains restrictedly quantifiable for x. But even more importantly, the entire setup ensures that both quantifiers must be of the same type, if we are to produce a restricted formula. This is expressed by the lemma below.

(13) a. Every farmer who owns a donkey likes it.
b. $\forall x \forall y\colon\ \neg\big(F(x) \wedge (D(y) \wedge O(x,y))\big) \vee L(x,y)$

Lemma 4. *Assume that φ is κ-restrictedly π-quantifiable for x and κ-restrictedly ρ-quantifiable for y; x and y may be the same variable. Then, $\pi = \rho$.*

Proof. Let S be the set of subformulas of φ with the property that they are κ-restrictedly π-quantifiable for x and κ-restrictedly ρ-quantifiable for y. S is non-empty, as $\varphi \in S$. Choose some ψ from S such that no subformula of ψ is in S (let us say that ψ is minimal in this sense) and assume that $\pi = -\rho$. Let us look at the form of ψ.

- ψ is atomic. Then, $\pi = \rho = +$ by Definition 43, a contradiction to the assumption $\pi = -\rho$.
- $\psi \equiv \neg\psi'$. Then, ψ' is κ-restrictedly $-\pi$-quantifiable for x and κ-restrictedly $-\rho$-quantifiable for y, a contradiction to the minimality of ψ.
- $\psi \equiv \exists y\, \psi'$ or $\psi \equiv \forall y\, \psi'$. Then, ψ' is κ-restrictedly π-quantifiable for x and κ-restrictedly ρ-quantifiable for y, a contradiction to the minimality of ψ.
- $\psi \equiv \bigwedge^{\pi}_{i\in I}\psi_i$. As $\pi = -\rho$, $\psi \equiv \bigvee^{\rho}_{i\in I}\psi_i$. By Definition 43, there is a $i \in I$ such that ψ_i is κ-restrictedly π-quantifiable for x. But by Definition 43, ψ_j is also κ-restrictedly ρ-quantifiable for y for every $j \in I$. Thus, ψ_i is both κ-restrictedly π-quantifiable for x and κ-restrictedly ρ-quantifiable for y, a contradiction to the minimality of ψ.
- $\psi \equiv \bigvee^{\pi}_{i\in I}\psi_i$. As above, with $\wedge^{\rho}$ and $\vee^{\pi}$ substituted for $\wedge^{\pi}$ and $\vee^{\rho}$.

■

We now turn to model-theoretic definitions. In the remainder of the section, $\mathfrak{U}$ and $\mathfrak{U}'$ will be arbitrary structures with domains U and U', respectively. (The notion of structure is defined as usual, as a non-empty domain of individuals equipped with an interpretation, i.e. a function assigning meanings to predicate, constant, and operation symbols of the language.) We start by rehearsing the standard definition of a substructure and then build on this notion to define the central condition on models used in the definition of conservativity.

Definition 45. Structure $\mathfrak{U}$ is a SUBSTRUCTURE of $\mathfrak{U}'$ (and $\mathfrak{U}'$ is a SUPERSTRUCTURE of $\mathfrak{U}$), in symbols $\mathfrak{U} \subset \mathfrak{U}'$, iff $U \subset U'$ and for any atomic formula α and any assignment $s\colon \text{Var} \to U$, we have $\mathfrak{U} \vDash \alpha[s]$ iff $\mathfrak{U}' \vDash \alpha[s]$.

Definition 46. Let κ be a c-signature. $\mathfrak{U}'$ is a κ-EXTENSION of $\mathfrak{U}$, in symbols $\mathfrak{U} \subset_\kappa \mathfrak{U}'$, iff $\mathfrak{U} \subset \mathfrak{U}'$ and for every variable x,[15] every atomic formula α which is a potential κ-restrictor for x, and every assignment $s\colon \mathbb{V} \to U'$, $\mathfrak{U}' \vDash \alpha[s]$ entails $s(x) \in U$.

Definition 47. A sentence φ is κ-CONSERVATIVE for theory T if and only if for any models $\mathfrak{U}$ and $\mathfrak{U}'$ of T such that $\mathfrak{U} \subset_\kappa \mathfrak{U}'$, $\mathfrak{U} \vDash \varphi$ iff $\mathfrak{U}' \vDash \varphi$.

Let us slowly walk through these definitions. Starting at the bottom, Definition 47 says that a sentence (note that we are talking about sentences here, i.e. formulas with no free variables) is κ-conservative when it receives the same truth value in a model and any κ-extension of the model, for any model; in other words, the interpretation of a κ-conservative sentence does not change when going from a model to its κ-extension, and this holds for any model and any κ-extension of this model.

If the condition on a superstructure is that it must match the substructure on the domain of the latter, being a κ-extension places some additional conditions on how the superstructure looks like outside that domain. For a potential restrictor α for x built on a monadic predicate P, the requirement that $\mathfrak{U}' \vDash P(x)[s]$ entails $s(x) \in U$ really means that P may be satisfied by no new individuals in the κ-extension. That is, if κ contains predicate D 'dog', a κ-extension of a model cannot bring in any new dogs.

For potential restrictors built from a constant, the requirement that $\mathfrak{U}' \vDash \alpha[s]$ must entail $s(x) \in U$ actually says more about what kind of binary predicates make sense as members among the restricting predicates $\mathcal{R}_1$ and $\mathcal{R}_2$ than about the constant itself. For example, $x \subset c$ and $c \supset x$ always entail $s(x) \in U$ (as the interpretation of any constant is always a member of U by the definition of substructure), and so do $x = c$ and $c = x$, but $c \subset x$ and $x \supset c$ only do so when the κ-extension $\mathfrak{U}'$ equals the original structure $\mathfrak{U}$, which is of course a trivial situation. Therefore, while $=$ leads to non-trivial situations both as a member of $\mathcal{R}_1$ and $\mathcal{R}_2$, $\subset$ only makes sense in $\mathcal{R}_1$, and $\supset$ only makes sense in $\mathcal{R}_2$.

For some restricting predicates, however, the constant plays a role in the condition as well. Imagine an arbitrary parthood relation $\sqsubseteq$. The requirement that $x \sqsubseteq c$ entail $s(x) \in U$ means that an object may not "grow" any new parts in the κ-extension, or "group" the old parts into some new, intermediate part. For example, if Fido's tail is not conceptualized as a separate part of Fido in the original structure, the same should hold in an $\{f\}$-extension; or if we imagine a parthood relation between geopolitical units, if a country is divided into municipalities which are not grouped into regions, a κ-extension cannot introduce this grouping.

We are finally ready to present the central theorem of this section, which states that restrictedness is a syntactic reflex of conservativity.

[15] Clearly, if the condition holds for any variable, it holds for every variable.

Theorem 7. *Let κ be c-signature and φ an arbitrary sentence of language L with theory T. Then φ is κ-conservative (for theory T) if and only if it is equivalent (in theory T) to some atomically/strongly/weakly κ-restricted sentence.*

The rest of the section is dedicated to the proof of this theorem. As all atomically restricted formulas are strongly restricted, and all strongly restricted formulas are in turn weakly restricted, it suffices to prove the syntax-to-semantics direction for weakly restricted sentences, and the reverse direction for atomically restricted sentences.

We start with the simpler syntax-to-semantics direction. Clearly, it suffices to prove that we can conclude $\mathfrak{U}' \vDash \vartheta$ from $\mathfrak{U} \subset_\kappa \mathfrak{U}'$ and $\mathfrak{U} \vDash \vartheta$ for any weakly κ-restricted sentence ϑ; note that this statement is independent of theory T. The proof is by induction on the complexity of ϑ, presented below, the induction hypothesis being that for any weakly κ-restricted *formula* ϑ and for any assignment $s\colon \mathrm{Var} \to U$, $\mathfrak{U} \vDash \vartheta[s]$ entails $\mathfrak{U}' \vDash \vartheta[s]$. The desired result then follows from the fact that a sentence contains no free variables, which makes its truth value independent of the assignment. Furthermore note that we may assume, without the loss of generality, that ϑ is of negation normal form.

The only part of the proof relying on $\mathfrak{U}'$ being a κ-extension of $\mathfrak{U}$ is the final point dealing with universal quantification (the existential quantification case only relies on $\mathfrak{U}'$ being a superstructure of $\mathfrak{U}$), which uses this fact as a prerequisite for the application of Lemma 5, which we present after the main proof—the lemma essentially says that the truth/falsity of a formula in the κ-extension $\mathfrak{U}'$ (relative to some assignment of individuals to the free variables of the formula) which is κ-restrictedly existentially/universally quantifiable for x implies that x was assigned an element inside the substructure $\mathfrak{U}$.

- ϑ is an atomic formula. By Definition 45 of substructure, $\mathfrak{U} \vDash \vartheta[s]$ iff $\mathfrak{U}' \vDash \vartheta[s]$, so $\mathfrak{U} \vDash \vartheta[s]$ entails $\mathfrak{U}' \vDash \vartheta[s]$.
- $\vartheta \equiv \neg\vartheta'$. Clearly, ϑ' is weakly κ-restricted, and as ϑ is in negation normal form, ϑ' is atomic, so $\mathfrak{U} \vDash \vartheta'[s]$ iff $\mathfrak{U}' \vDash \vartheta'[s]$. Therefore, $\mathfrak{U} \vDash \neg\vartheta[s]$ entails $\mathfrak{U}' \vDash \vartheta[s]$.
- $\vartheta \equiv \wedge_{i\in I}\vartheta_i$. As ϑ is a conjunction of ϑ_i, assuming $\mathfrak{U} \vDash \vartheta[s]$ gives us $\mathfrak{U} \vDash \vartheta_i[s]$ for every $i \in I$. As all ϑ_i are weakly κ-restricted, $\mathfrak{U} \vDash \vartheta_i[s]$ entails $\mathfrak{U}' \vDash \vartheta_i[s]$ for every $i \in I$ by the inductive hypothesis. Conjoining these conclusions, we arrive at $\mathfrak{U}' \vDash \vartheta[s]$.
- $\vartheta \equiv \vee_{i\in I}\vartheta_i$. As ϑ is a disjunction of ϑ_i, assuming $\mathfrak{U} \vDash \vartheta[s]$ gives us $\mathfrak{U} \vDash \vartheta_i[s]$ for some $j \in I$. ϑ_j is weakly κ-restricted (in fact, all ϑ_i are), so we have $\mathfrak{U}' \vDash \vartheta_j[s]$ by the inductive hypothesis. By addition, it follows that $\mathfrak{U}' \vDash \vartheta[s]$.
- $\vartheta \equiv \exists x\, \vartheta'$ for some variable x. Assuming $\mathfrak{U} \vDash (\exists x\, \vartheta')[s]$, we have $\mathfrak{U} \vDash \vartheta'[s(u/x)]$ for some $u \in U$. By the induction hypothesis, it follows that $\mathfrak{U}' \vDash \vartheta'[s(u/x)]$. As $u \in U \subset U'$, $\mathfrak{U}' \vDash (\exists x\, \vartheta')[s]$.
- $\vartheta \equiv \forall x\, \vartheta'$ for some variable x. Assuming $\mathfrak{U} \vDash (\forall x\, \vartheta')[s]$, we have $\mathfrak{U} \vDash \vartheta'[s(u/x)]$ for every $u \in U$. ϑ' is weakly κ-restricted, so by the induction hypothesis, $\mathfrak{U}' \vDash \vartheta'[s(u/x)]$ follows for every $u \in U$.

  By Definition 44, ϑ' is κ-restrictedly universally quantifiable for x. By Lemma 5 (for $\tau = -$), $\mathfrak{U}' \vDash \neg\vartheta'[s(u/x)]$ entails $s(x) \in U$; therefore, $\mathfrak{U}' \vDash \vartheta'[s(u/x)]$ for every $u \in U' - U$.

  It follows that $\mathfrak{U}' \vDash \vartheta'[s(u/x)]$ for every $u \in U'$, which entails that $\mathfrak{U}' \vDash (\forall x\, \vartheta')[s]$.

Lemma 5. *Let κ be a c-signature, x a variable and τ a polarity, and ϑ a formula in negation normal form which is κ-restrictedly τ-quantifiable for x. Let $\mathfrak{U}$ and $\mathfrak{U}'$ be arbitrary structures and $s\colon \mathbb{V} \to U'$ any assignment. Assume $\mathfrak{U} \subset_\kappa \mathfrak{U}'$. Then $\mathfrak{U}' \vDash (\neg^\tau\vartheta)[s]$ entails $s(x) \in U$.*

Proof. The proof is by induction on the complexity of ϑ.

- ϑ is atomic. By Definition 43, $\tau = +$ and ϑ is a potential κ-restrictor for x. Assuming $\mathfrak{U}' \models (\neg^\tau \vartheta)[s]$, $s(x) \in U$ follows by the Definition 46 of κ-extension.
- $\vartheta \equiv \neg\vartheta'$. As ϑ is in negation normal form, ϑ' is atomic; by Definition 43, τ must then be negative. Assuming $\mathfrak{U}' \models (\neg^\tau \neg\vartheta')[s]$ is then equivalent to $\mathfrak{U}' \models \vartheta'[s]$, which entails $s(x) \in U$ by the induction hypothesis (for the atomic ϑ' and $\tau = +$).
- $\vartheta \equiv \bigwedge^\tau_{i\in I}\vartheta_i$. Formula $\neg^\tau \bigwedge^\tau_{i\in I} \vartheta_i$ is (identical or) logically equivalent to $\bigwedge_{i\in I}\neg^\tau\vartheta_i$. Assuming $\mathfrak{U}' \models (\neg^\tau \vartheta)[s]$ thus entails $\mathfrak{U}' \models (\neg^\tau \vartheta_i)[s]$ for every $i \in I$. By Definition 43, there is a $j \in I$ such that ϑ_j is κ-restrictedly τ-quantifiable for x. $s(x) \in U$ then follows by the induction hypothesis (for ϑ_j).
- $\vartheta \equiv \bigvee^\tau_{i\in I}\vartheta_i$. Formula $\neg^\tau \bigvee^\tau_{i\in I} \vartheta_i$ is (identical or) logically equivalent to $\bigvee_{i\in I}\neg^\tau\vartheta_i$. Assuming $\mathfrak{U}' \models (\neg^\tau \vartheta)[s]$ thus entails $\mathfrak{U}' \models (\neg^\tau \vartheta_j)[s]$ for some $j \in I$. By Definition 43, ϑ_i is κ-restrictedly τ-quantifiable for x for every $i \in I$, and thus also for $i = j$. $s(x) \in U$ then follows by the induction hypothesis (for ϑ_j).
- $\vartheta \equiv \exists^\pi y\, \vartheta'$. By Definition 43, y is distinct from x and ϑ' is κ-restrictedly τ-quantifiable for x. Assume that $\mathfrak{U}' \models (\neg^\tau \exists^\pi y\, \vartheta')[s]$. As $\neg^\tau \exists^\pi y\, \vartheta'$ is (identical or) logically equivalent to $\exists^{\pi\tau} y \neg^\tau \vartheta'$, which resolves to $\exists y \neg^\tau \vartheta'$ if $\pi = \tau$ and to $\forall y \neg^\tau \vartheta'$ if $\pi \neq \tau$.[16] In the existential case, it follows that: there is an $u \in U'$ such that $\mathfrak{U}' \models (\neg^\tau \vartheta')[s(u/y)]$. In the universal case, $\mathfrak{U}' \models (\neg^\tau \vartheta')[s(v/y)]$ holds for every $v \in U'$, and assuming that the domain is non-empty, also for some specific $u \in U'$. Let $s' := s(u/y)$. By the induction hypothesis, $s'(x) \in U$, but as x and y are distinct, we also have $s(x) \in U$.

■

We continue with the proof of the semantics-to-syntax direction of Theorem 7. As announced before, we will rely on Feferman (1968), so let us start by briefly presenting his system. We omit many technical details such as the precise definition of many-sorted structures and rather focus on the parallels between our and Feferman's notions. We also modify his notation where necessary to better fit our presentation.

For Feferman, restrictedness is defined with reference to a single relation, $\preceq$. His restricted quantifiers take the form $\exists x \preceq t\, \varphi$ and $\forall x \preceq t\, \varphi$, for a term t in which x does not occur, and are defined to be equivalent to $\exists x\, (x \preceq t \wedge \varphi)$ and $\forall x\, (x \preceq t \Rightarrow \varphi)$, which corresponds to our atomic restrictedness.

Our notion of κ-extension corresponds to Feferman's notion of outer extension: given a formal language with a binary relation $\preceq$, structure $\mathfrak{U}'$ is an OUTER EXTENSION of structure $\mathfrak{U}$ iff $\mathfrak{U}$ is a substructure of $\mathfrak{U}'$ and for any assignment $s\colon \mathbb{V} \to U'$ such that $s(y) \in U$, $\mathfrak{U}' \models (x \preceq y)[s]$ entails $x \in U$. (In our system, $\preceq$ would belong to the set of leftward restricting predicates $\mathcal{R}_1$.)

Our κ-conservativity is then completely analogous to Feferman's invariance, but with κ-extension substituted for outer extension.

To translate our problem to Feferman's system, we need to restate the κ-restrictedness using a single predicate, $\preceq$. We will therefore need to extend the formal language by this predicate. However, this alone of course does not suffice: the language will need some other additional symbols, and we will also have to extend the theory. Let us write down the

[16] The situation $\pi \neq \tau$ cannot arise in the setting of Theorem 7. Besides being κ-restrictedly τ-quantifiable for x, ϑ is weakly κ-restricted there as well. This makes ϑ' both κ-restrictedly τ-quantifiable for x (by Definition 43) and κ-restrictedly π-quantifiable for y (by Definition 44), so by Lemma 4, $\tau = \pi$.

additional symbols of the extended language and the additional axioms of the theory and then explain their significance. We extend the formal language L into L^+ by introducing

- a new binary relation $\preceq$,
- a new individual constant symbol c_P for each (monadic) predicate $P \in \kappa$, and
- new individual constant symbols $c_\subset$ and $c_\supset$ for each individual constant $c \in \kappa$ and each predicate $\subset \in \mathcal{R}_1$ and $\supset \in \mathcal{R}_2$.

and add the following axiom and axiom schemes to the original theory T to yield T^+:

(T1) $x \preceq y \Leftrightarrow \bigvee_{P \in \kappa_1}(y = c_P \wedge P(x)) \vee \bigvee_{\subset \in \mathcal{R}_1}(x \subset y) \vee \bigvee_{\supset \in \mathcal{R}_2}(y \supset x)$,
where κ_1 is the set of monadic predicates from κ,[17, 18]

(T2) $c_P \neq c_Q$ for each distinct $P, Q \in \kappa_1$,

(T3) $\neg\alpha$, for every atomic formula α which contains some non-logical predicate from L and the constant c_P for some $P \in \kappa_1$, and

(T4) $c = c_\subset$ and $c = c_\supset$ for each individual constant $c \in \kappa$ and each predicate $\subset \in \mathcal{R}_1$ and $\supset \in \mathcal{R}_2$.

The additional individual constants c_P will help us translate restrictors of form $P(x)$ to $x \preceq c_P$. The new constants of form $c_\subset$ and $c_\supset$ are a technical detail which will help us keep track of the original "source" of restrictedness; as we see from the final additional axiom of the theory, their interpretation matches the interpretation of the original constant c.

Given the new axioms of theory T^+ with respect to T, it is easy to see that there is a bijective correspondence between models of T and models of T^+. The EXTENDED MODEL $\mathfrak{U}^+$ of T^+ corresponding to model $\mathfrak{U}$ of T is a superstructure of $\mathfrak{U}$ with additional (and distinct) elements interpreting the constants c_P. Axiom scheme (T3) ensures that the new elements of the extended model are unrelated to anything via the non-logical predicates of the language; axiom scheme (T2) ensures a uniform size difference between the original and the extended model, yielding a bijection.

But the heart of the additional axioms is clearly (T1). The first disjunct of the right hand-side of the equivalence makes sure that we can restate restrictors of form $P(x)$, with $P \in \kappa$, by $x \preceq c_P$; whenever $P(x)$ of the original language holds in the original model, $x \preceq c_P$ of the extended language will hold in the extended model. The second and the third disjunct ensure a similar entailment for restrictors containing a constant from κ, which are based on a restricting predicate from $\mathcal{R}_1$ or $\mathcal{R}_2$. For the translation from $x \preceq t$ (for some term t) back to the original language, the axiom ensures that $P(x)$, $x \subset c$ and $c \supset x$ are the only possible original language "sources" of the extended language restrictor; as we will see below, the form of the term determines which option applies.

Formally, we define translation $+$ from L to L^+ and translation $-$ from L^+ to L as follows. For any formula φ of L, let φ^+ be the formula we get by substituting

- $x \preceq c_P$ for every occurrence of $P(x)$, where $P \in \kappa_1$ and x is a variable,
- $x \preceq c_\subset$ for every occurrence of $x \subset c$, where $c \in \kappa_2$ and $\subset \in \mathcal{R}_1$, and
- $x \preceq c_\supset$ for every occurrence of $c \supset x$, where $c \in \kappa_2$ and $\supset \in \mathcal{R}_2$,

and for a formula φ of L^+, let φ^- be the formula we get by substituting

[17] We assume that κ_1, $\mathcal{R}_1$, and $\mathcal{R}_2$ are finite if L is finitary—an innocuous assumption, as a finitary formula can only be restricted with respect to a finite number of predicates (or constants).

[18] Clearly, the constructions using the subscripted $\vee$ should be seen as meta-logical.

- for every $P \in \kappa_1$:
  - $P(x)$ for every occurrence of $x \preceq c_P$, where x is a variable, and
  - $\bot$ for all other atomic formulas containing c_P, and
- for every $c \in \kappa_2$, $\subset \in \mathcal{R}_1$ and $\supset \in \mathcal{R}_2$:
  - $x \subset c$ for every occurrence of $x \preceq c_{\subset}$,
  - $c \supset x$ for every occurrence of $x \preceq c_{\supset}$, and
  - c for all other occurrences of $c_{\subset}$ and $c_{\supset}$.

The additional axioms of T^+ ensure that $\mathfrak{U} \models \varphi$ entails $\mathfrak{U}^+ \models \varphi^+$, for any φ of L, and that $\mathfrak{U}^+ \models \varphi$ entailing $\mathfrak{U} \models \varphi^-$ for any φ of L^+. Note that not every φ of language L^+ is a translation ψ^+ of some ψ from L; the final subclause of both clauses of the reverse translation $-$ is designed to deal with any occurrences of the additional symbols of L^+ outside restrictors.

The path paved, we can now outline the proof itself. Taking a sentence φ of language L, κ-conservative in theory T, we translate it to sentence φ^+ of L^+ and see that φ^+ is invariant for theory T^+. By Feferman's Theorem 4.2, φ^+ is then logically equivalent to some restricted (in Feferman's sense) sentence ϑ. The translation back to the original language then yields a κ-restricted ϑ^- of the original language.

Given the definitions in this section, most of the details of the proof are transparent. For example, the above-mentioned fact that the translations from and to the original language are truth-preserving (in the sense that $\mathfrak{U} \models \varphi$ entails $\mathfrak{U}^+ \models \varphi^+$, and that $\mathfrak{U}^+ \models \vartheta$ entails $\mathfrak{U} \models \vartheta^-$), yields the result that ϑ^- is logically equivalent to φ. Or, a careful comparison of the definitions of κ-extension and outer extension, facilitated by the additional axioms of T^+, reveals that whenever $\mathfrak{U}'$ is a κ-extension of $\mathfrak{U}$, $\mathfrak{U}'^+$ is an outer extension of $\mathfrak{U}^+$, and vice versa. Two issues, however, are worth mentioning in detail.

First, it is crucial that the original and the extended models are in bijective correspondence. To apply Feferman's theorem, we need to know that $\mathfrak{U}^+ \models_T \varphi^+$ iff $\mathfrak{U}'^+ \models_T \varphi^+$ (the condition on conservativity/invariance) holds for *any* pair of models $\mathfrak{U}^+$ and $\mathfrak{U}'^+$ such that $\mathfrak{U}'^+$ is an outer extension of $\mathfrak{U}^+$. This can only be ensured with a bijection.

Second, certain conditions on the theory must hold for Feferman's theorem to be applicable. This is reasonable, as the theory carves out a set of structures as models, so if the theory is too restricted, it might not have "enough" models to allow the inference from the conservativity/invariance of φ ($\mathfrak{U}^+ \models_T \varphi^+$ iff $\mathfrak{U}'^+ \models_T \varphi^+$ holds for *any* pair of models $\mathfrak{U}^+$ and $\mathfrak{U}'^+$ such that $\mathfrak{U}'^+$ is an outer extension of $\mathfrak{U}^+$) to the existence of an T^+-equivalent restricted ϑ.

Feferman's theorem is applicable for theories consisting of axioms which are either universal or predicative (Feferman 1968, pp. 47–48).[19] In the latter case, the relation interpreting $\preceq$ must satisfy two additional constraints. First, the transitive closure of this relation must be definable in the theory. Second, all models of the theory must be well-founded and extensional.

Clearly, we cannot check the conditions on the theory in general—except to see that all the additional axioms of T^+ are clearly universal—but we can see that the conditions hold for L**. Taking a look at the mereological underpinning of L** (see Chapter 5), we see that the mereological axioms are universal, with the important exception of the existence of sum, which turns out to be predicative. However, this is unproblematic: the subcollection relation is transitive (and thus its own closure), and as non-logical monadic predicates of

[19] See also Feferman and Kreisel (1966) and Feferman (1966). For an in-depth discussion on predicativity, see Feferman (2005).

L^{**} are assumed to be distributive (i.e. $x \subset y$ and $P(y)$ entail $P(x)$, see Chapter 5), this holds for $\preceq$ as well; furthermore, most mereologies aim to be well-founded and extensional. The situation seems clear with regard to our rudimentary counting theory as well: we have proposed two universal and one predicative axiom and assumed the existence of natural numbers; Feferman's theorem remains applicable, as we can take the latter to be a stationary domain (see Feferman 1968, pp. 30–31, 34).

This concludes the proof of the claim that restrictedness is the syntactic characterization of conservativity, but note that this section is not a final word on restrictedness and conservativity. For example, Theorem 7 could be formulated as a more general interpolation theorem (see section A.3), or even more importantly, the definitions of characterization of conservativity could be extended (in parallel) in several ways, with Theorem 7 remaining valid. For instance, we could take a separate look at existentially and universally restricted formulas; in fact, Feferman (1968) does that. But more importantly, perhaps, the notion of conservativity signature κ could be extended to include not only monadic predicates. This would allow us to recognize as restricted also the formulas such as $\exists x\colon\ V(s,x) \wedge \neg M(x)$—we have used this formula as the logical form of the bound reading of sentence 'Sascha didn't visit Montmartre' in subsection 10.3.2—and to realize that restrictors may be "chained," as in $\exists y\colon\ M(y) \wedge C(x,y)$, which could be seen as an approximation of the logical form of 'mouse-chaser'. As it is not yet entirely clear to us if this extension of the notion would in fact be applicable to such cases, and as extending the notion of conservativity in this manner leads to a somewhat more elaborate proof of the syntax-to-semantics direction of Theorem 7, we leave the matter to further research.

References

Abney, Steven P. (1987). “The English Noun Phrase in its Sentential Aspect”. PhD thesis. MIT.

Aboh, Enoch O. (2004). *The Morphosyntax of Complement-Head Sequences.* Oxford and New York: Oxford University Press.

Abzianidze, Lasha (2015). “A tableau prover for natural logic and language”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* pp. 2492–2502.

Acquaviva, Paolo (1994). “The representation of operator-variable dependencies in sentential negation”. In: *Studia Linguistica* 48.2, pp. 91–132.

Adger, David (2003). *Core Syntax: A Minimalist Approach.* Oxford and New York: Oxford University Press.

Angeli, Gabor and Christopher D. Manning (2014). “Naturalli: Natural logic inference for common sense reasoning”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* pp. 534–545.

Antony, Louise M. and Norbert Hornstein, eds. (2008). *Chomsky and his Critics*. Philosophers and their Critics. Oxford: Blackwell.

Ashworth, E. J. (1974). *Logic and Language in the Post-Medieval Period.* Dordrecht: D. Reidel Publishing Company.

Baker, C. L. (1970). “Double negatives”. In: *Linguistic Inquiry* 1, pp. 169–186.

Baker, Mark (1985). “The mirror principle and morphosyntactic explanation”. In: *Linguistic Inquiry* 16, pp. 373–416.

Barker, Chris (2018). “Negative polarity as scope marking”. In: *Linguistics and Philosophy* 41.5, pp. 483–510.

Barwise, Jon (1989). *The Situation in Logic.* Stanford: CSLI Publications.

Barwise, Jon and Robin Cooper (1981). “Generalized quantifiers and natural language”. In: *Linguistics and Philosophy* 4, pp. 159–219.

Beghelli, Filippo (1995). “The Phrase Structure of Quantifier Scope”. PhD thesis. UCLA.

Beghelli, Filippo and Tim Stowell (1997). “Distributivity and negation: The syntax of *each* and *every*”. In: *Ways of Scope Taking.* Ed. by Anna Szabolcsi. Vol. 65. Studies in Linguistics and Philosophy. Dordrecht: Kluwer Academic Publishers, pp. 71–107.

Belleti, Adriana, ed. (2004). *Structures and Beyond: The Cartography of Syntactic Structures.* Vol. 3. Oxford Studies in Comparative Syntax. Oxford and New York: Oxford University Press.

Benacerraf, Paul (1973). “Mathematical truth”. In: *The Journal of Philosophy* 70, pp. 661–679.

Bernardi, Raffaella A. (2002). “Reasoning with Polarity in Categorial Type Logic”. PhD thesis. Utrecht University.

Bernath, Jeffrey L. (2009). “Pinning down articles in American sign language”. In: *ESSLI 2009 Workshop: Formal Approaches to Sign Languages.*

Bogen, James and James Woodward (1988). “Saving the phenomena”. In: *The Philosophical Review* 97.3, pp. 303–352.

Boolos, George S. [1984] (1998). "To be is to be a value of a variable (or some values of some variables)". In: *Logic, Logic, and Logic*. Ed. by Richard Jeffrey. With a forew. by John P. Burgess. Cambridge, MA, and London: Harvard University Press. Repr. of "To be is to be a value of a variable (or some values of some variables)". In: *Journal of Philosophy* 81.8, pp. 43–449.

Boolos, George S., John P. Burgess, and Richard C. Jeffrey (2002). *Computability and Logic*. 4th ed. Cambridge: Cambridge University Press.

Bowman, Samuel R. et al. (2015). "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Boyd, R. (1981). "Scientific realism and naturalistic epistemology". In: *PSA 1980*. Ed. by P. Asquith and R. Giere. Vol. 2. East Lansing, MI: Philosophy of Science Association, pp. 613–662.

Boyd, R. (1984). "The current status of scientific realism". In: *Scientific Realism*. Ed. by J. Leplin. Berkeley, CA: University of California Press, pp. 41–82.

Brandom, Robert B. (1998). *Making It Explicit: Reasoning, Representing, and Discursive Commitment*. Cambridge, MA, and London: Harvard University Press.

Brugè, Laura (2002). "The positions of demonstratives in the extended nominal projection". In: *Functional Structure in DP and IP: The Cartography of Syntactic Structures*. Ed. by Guglielmo Cinque. Vol. 1. Oxford Studies in Comparative Syntax. Oxford and New York: Oxford University Press, pp. 15–53.

Brünnler, Kai (2003). "Atomic cut elimination for classical logic". In: *Computer Science Logic (CSL)*. Ed. by M. Baaz and J. A. Makowsky. Vol. 2803. Lecture Notes in Computer Science. Springer-Verlag, pp. 86–97.

Brünnler, Kai (2004). *Deep Inference and Symmetry in Classical Proofs*. Berlin: Logos Verlag.

Brünnler, Kai (2006a). "Cut elimination inside a deep inference system for classical predicate logic". In: *Studia Logica* 82.1, pp. 51–71.

Brünnler, Kai (2006b). "Deep inference and its normal form of derivations". In: *Logical Approaches to Computational Barriers—2nd Conference on Computability in Europe*. Ed. by Arnold Beckmann et al. Vol. 3988. Lecture Notes in Computer Science. Springer-Verlag, pp. 65–74.

Brünnler, Kai (2006c). "Locality for classical logic". In: *Notre Dame Journal of Formal Logic* 47.4, pp. 557–580.

Brünnler, Kai (2010). "Nested Sequents". Habilitation Thesis. URL: http://arxiv.org/pdf/1004.1845.pdf

Brünnler, Kai and Alessio Guglielmi (2002). *Consistency Without Cut Elimination*. Tech. rep. WV-0216. Technische Universitüt Dresden.

Brünnler, Kai and Alessio Guglielmi (2004). "A first order system with finite choice of premises". In: *First-Order Logic Revisited*. Ed. by Vincent Hendricks et al. Logische Philosophie. Logos Verlag, pp. 59–74.

Brünnler, Kai and Richard McKinley (2008). "An algorithmic interpretation of a deep inference system". In: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*. Ed. by Iliano Cervesato, Helmut Veith, and Andrei Voronkov. Vol. 5330. Lecture Notes in Computer Science. Springer-Verlag, pp. 482–496.

Brünnler, Kai and Alwen F. Tiu (2001). "A local system for classical logic". In: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*. Ed. by R. Nieuwenhuis and Andrei Voronkov. Vol. 2250. Lecture Notes in Computer Science. Springer-Verlag, pp. 347–361.

Burley, Walter [14th century] (1955). *Walter Burleigh: De puritate artis logicae tractatus longior, with a Revised Edition of the Tractatus brevior.* Ed. by Philotheus Boehner. St. Bonaventure, New York: The Franciscan Institute. Translated in Spade 2000.

Burley, Walter [14th century] (1972). "De Suppositionibus". In: *Walter Burleigh's Treatise De Suppositionibus and its Influence on William of Ockham.* Ed. by Stephen Brown. Franciscan Studies 32, pp. 15–64. Translated in Spade 1997.

Cann, Ronnie (1993). *Formal Semantics: An Introduction.* Cambridge textbooks in linguistics. Cambridge: Cambridge University Press.

Carlson, Greg (1981). "Distribution of free-choice any". In: *Papers from the Seventeenth Regional Meeting of the Chicago Linguistics Society*. Ed. by R. Hendrick, C. Masek, and M. Miller. Chicago: Chicago Linguistic Society.

Chierchia, Gennaro (1995). *Dynamics of Meaning: Anaphora, Presupposition and the Theory of Grammar*. Chicago and London: The University of Chicago Press.

Chierchia, Gennaro and Sally McConnell-Ginet (1993). *Meaning and Grammar: An Introduction to Semantics.* Cambridge, MA: The MIT Press.

Chomsky, Noam (1957). *Syntactic Structures.* Haag: Mouton.

Chomsky, Noam (1965). *Aspects of the Theory of Syntax.* Cambridge, MA: The MIT Press.

Chomsky, Noam (1973). "Conditions on transformations". In: *A Festschrift for Morris Halle.* Ed. by Stephen Anderson and Paul Kiparsky. New York: Holt, Rinehart & Winston, pp. 232–286.

Chomsky, Noam (1975). *The Logical Structure of Linguistic Theory.* Dordrecht: Springer.

Chomsky, Noam (1980). *Rules and Representations.* Columbia University Press.

Chomsky, Noam (1986). *Barriers.* Cambridge, MA: The MIT Press.

Chomsky, Noam (1995). *The Minimalist Program.* Cambridge, MA: The MIT Press.

Chomsky, Noam (2002). "Language and the brain". In: *On Nature and Language.* Ed. by Adriana Belleti and Luigi Rizzi. Cambridge: Cambridge University Press.

Chomsky, Noam (2004). "Beyond explanatory adequacy". In: *Structures and Beyond: The Cartography of Syntactic Structures.* Ed. by Adriana Belleti. Vol. 3. Oxford Studies in Comparative Syntax. Oxford and New York: Oxford University Press, pp. 104–131.

Chomsky, Noam (2008). "Reply to Ludlow". In: *Chomsky and his Critics*. Ed. by Louise M. Antony and Norbert Hornstein. Philosophers and their Critics. Oxford: Blackwell, pp. 287–295.

Church, Alonzo (1936). "A note on the Entscheidungsproblem". In: *Journal of Symbolic Logic* 1, pp. 4–41.

Cinque, Guglielmo (1994). "On the evidence for partial N-movement in the romance DP". In: *Paths towards Universal Grammar.* Ed. by Guglielmo Cinque et al. Washington, DC: Georgetown University Press.

Cinque, Guglielmo (1999). *Adverbs and Functional Heads.* Oxford Studies in Comparative Syntax. Oxford and New York: Oxford University Press.

Cinque, Guglielmo, ed. (2002). *Functional Structure in DP and IP: The Cartography of Syntactic Structures.* Vol. 1. Oxford Studies in Comparative Syntax. Oxford and New York: Oxford University Press.

Cinque, Guglielmo and Luigi Rizzi (2008). "The cartography of syntactic structures". In: *The Oxford Handbook of Linguistic Analysis.* Ed. by Bernd Heine and Heiko Narrog. Oxford University Press, pp. 65–78.

Coppock, Elizabeth, Elizabeth Bogal-Allbritten, and Golsa Nouri-Hosseini (2020). "Universals in superlative semantics". In: *Language* 96.3, pp. 471–506.

Coppock, Elizabeth and Lucas Champollion (2021). "Invitation to formal semantics". URL: http://eecoppock.info/bootcamp/semantics-boot-camp.pdf (visited on 09/10/2021). Draft of August 20, 2021.

Corbett, Greville G. (2000). *Number.* Cambridge: Cambridge University Press.

Corneli, Joe, ed. (Mar. 22, 2013). *Examples of Primitive Recursive Functions*. URL: https://www.planetmath.org/examplesofprimitiverecursivefunctions (visited on 09/10/2021).

Craig, William (1957a). "Linear reasoning: A new form of the Herbrand-Gentzen theorem". In: *The Journal of Symbolic Logic* 22.3, pp. 269–285.

Craig, William (1957b). "Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory". In: *The Journal of Symbolic Logic* 22.3, pp. 269–285.

D'Alfonso, Duilio (2011). "Generalized quantifiers: Logic and language". In: *Logic and Philosophy of Science* IX.1, pp. 85–94.

Das, Anupam (2014). "The Complexity of Propositional Proofs in Deep Inference". PhD thesis. University of Bath.

Davidson, Donald (1967). "The logical form of action sentences". In: *The Logic of Decision and Action.* Ed. by Nicholar Rescher. Pittsburgh: University of Pittsburgh Press. Chap. III.

Davidson, Donald and Gilbert Harman, eds. (1972). *Semantics of Natural Language.* Dordrecht and Boston: D. Reidel Publishing Company.

De Clercq, Karen (2013). "A Unified Syntax of Negation". PhD thesis. Universiteit Gent.

de Mey, Sjaak (1991). "'Only' as a determiner and as a generalized quantifier". In: *Journal of Semantics* 8.1–2, pp. 91–106.

De Morgan, Augustus (1847). *Formal Logic: or, The Calculus of Inference, Necessary and Probable.* London: Taylor and Walton.

Dever, Josh (2008). "Compositionality". In: *The Oxford Handbook of Philosophy of Language.* Ed. by Ernest Lepore and Barry C. Smith. Oxford University Press.

Diesing, Molly (1992). *Indefinites.* Linguistic Inquiry Monographs 20. Cambridge, MA: MIT Press.

Dowty, David (1994). "The role of negative polarity and concord marking in natural language reasoning". In: *Proceedings of SALT IV.* Ed. by Mandy Harvey and Lynn Santelmann. Cornell University.

Dummett, Michael (1991). *The Logical Basis of Metaphysics.* Cambridge, MA: Harvard University Press.

Dutilh Novaes, Catarina (2011). Quantification In: *Encyclopedia of Medieval Philosophy: Philosophy Between 500 and 1500.* Ed. by Henrik Lagerlund. Dordrecht: Springer Science+Business Media B.V., pp. 1093–1095.

Egli, Urs (1979). "The Stoic concept of anaphora". In: *Semantics from Different Points of View.* Ed. by Rainer Bauerle, Urs Egli, and Arnim von Stechow. Berlin and Heidelberg: Springer-Verlag, pp. 266–283.

Evans, Gareth (1977). "Pronouns, quantifiers and relative clauses". In: *Canadian Journal of Philosophy* 7.3, pp. 467–536.

Evans, Gareth (1980). "Pronouns". In: *Linguistic Inquiry* 11.2, pp. 337–362.

Everett, Daniel L. (2005). "Cultural constraints on grammar and cognition in Pirahã: Another look at the design features of human language". In: *Current Anthropology* 46.4, pp. 621–646.

Feferman, Solomon (1966). "Predicative provability in set theory". In: *Bulletin of the American Mathematical Society* 72.3, pp. 486–489.

Feferman, Solomon (1968). "Persistent and invariant formulas for outer extensions". In: *Compositio Mathematica* 20, pp. 29–52.

Feferman, Solomon (2005). "Predicativity". In: *The Oxford Handbook of Philosophy of Mathematics and Logic*. Ed. by Stewart Shapiro. Oxford and New York: Oxford University Press, pp. 59–624.

Feferman, Solomon and Georg Kreisel (1966). "Persistent and invariant formulas relative to theories of higher order". In: *Bulletin of the American Mathematical Society* 72.3, pp. 48–485.

Field, Hartry (1980). *Science without Numbers: A Defence of Nominalism*. Princeton, NJ: Princeton University Press.

Fodor, Jerry A. (1980). "Methodological solipsism considered as a research strategy in cognitive psychology". In: *Behavioral and Brain Sciences* 3.1, pp. 63–73.

Fodor, Jerry A. and Ernest Lepore (2002). *The Compositionality Papers*. Oxford and New York: Oxford University Press.

Frank, Michael C. et al. (2008). "Number as a cognitive technology: Evidence from Pirahã language". In: *Cognition* 108, pp. 819–824.

Gallistel, Charles R. and Adam P. King (2010). *Memory and the Computational Brain: Why Cognitive Science Will Transform Neuroscience*. Oxford: Wiley-Blackwell.

Gazdar, Gerald (1979). "A solution to the projection problem". In: *Syntax and Semantics*. Vol. 11: *Presupposition*. Ed. by Oh Choon-Kyu and David A. Dinneen. New York, San Francisco, and London: Academic Press, pp. 57–89.

Geach, Peter T. (1962). *Reference and Generality: An Examination of Some Medieval and Modern Theories*. Ithaca, NY, and London: Cornell University Press.

Gentzen, Gerhard Karl E. (1934). "Untersuchungen über das logische Schließen. I". In: *Mathematische Zeitschrift* 39.2, pp. 176–210. English translation in Szabo 1969, #3.

George, Alexander (1989). "How not to become confused about linguistics". In: *Reflections on Chomsky*. Ed. by Alexander L. George. Oxford: Blackwell, pp. 9–110.

Geurts, Bart (2003). "Reasoning with quantifiers". In: *Cognition* 86, pp. 223–251.

Geurts, Bart (2006). "Take 'five': The meaning and use of a number word". In: *Non-definiteness and Plurality*. Ed. by Svetlana Vogeleer and Liliane Tasmowski. Amsterdam: John Benjamins Publishing, pp. 311–329.

Geurts, Bart and Frans van der Slik (2005). "Monotonicity and processing load". In: *Journal of Semantics* 22.1, pp. 97–117.

Gherardi, Guido, Paolo Maffezioli, and Eugenio Orlandelli (2019). "Interpolation in extensions of first-order logic". In: *Studia Logica*.

Giusti, Giuliana (2002). "The Functional Structure of Noun Phrases". In: *Functional Structure in DP and IP: The Cartography of Syntactic Structures*. Ed. by Guglielmo Cinque. Vol. 1. Oxford Studies in Comparative Syntax. Oxford and New York: Oxford University Press, pp. 54–90.

Gödel, Kurt (1931). "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I." In: *Monatshefte für Mathematik und Physik* 38, pp. 173–198. English translation: Gödel 1992.

Gödel, Kurt (1992). *On Formally Undecidable Propositions Of Principia Mathematica and Related Systems*. Trans. by B. Meltzer. With an intro. by R. B. Braithwaite. New York: Dover Publications. Republication of Gödel 1931.

Grice, Herbert P. (1989). *Studies in the Way of Words*. Cambridge, MA, and New York: Harvard University Press.

Groenendijk, Jeroen and Martin Stokhof (1991). "Dynamic predicate logic". In: *Linguistics and Philosophy* 14, pp. 39–100.

Guglielmi, Alessio (2007). "A system of interaction and structure". In: *ACM Transactions on Computational Logic* 8.1, 1:1–64.

Guglielmi, Alessio (2015). "Deep inference". In: *All About Proofs, Proofs for All*. Ed. by David Delahaye and Bruno Woltzenlogel Paleo. Vol. 55. Mathematical Logic and Foundations. London: College Publications, pp. 164–172.

Guglielmi, Alessio, Tom Gundersen, and Michel Parigot (2010). "A proof calculus which reduces syntactic bureaucracy". In: *21st International Conference on Rewriting Techniques and Applications (RTA)*. Ed. by Christopher Lynch. Vol. 6. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 135–150.

Guglielmi, Alessio, Tom Gundersen, and Lutz Straßburger (2010). "Breaking paths in atomic flows for classical logic". In: *25th Annual IEEE Symposium on Logic in Computer Science (LICS)*. Ed. by Jean-Pierre Jouannaud. IEEE, pp. 284–293.

Guglielmi, Alessio and Lutz Straßburger (2001). "Non-commutativity and MELL in the calculus of structures". In: *Computer Science Logic (CSL)*. Ed. by L. Fribourg. Vol. 2142. Lecture Notes in Computer Science. Springer-Verlag, pp. 54–68.

Gundersen, Tom (2009). "A General View of Normalisation Through Atomic Flows". PhD thesis. University of Bath.

Haegeman, Liliane (1994). *Introduction to Government and Binding Theory*. Oxford: Blackwell Publishers.

Halle, Morris and Alec Marantz (1993). "Distributed morphology and the pieces of inflection". In: *The View from Building 20: Essays in Honor of Sylvain Bromberger*. Ed. by Kenneth Hale and S. Jay Keyser. Cambridge, MA: The MIT Press, pp. 111–176.

Harley, Heidi and Elisabeth Ritter (2002). "Person and number in pronouns: A feature-geometric approach". In: *Language* 78.3, pp. 482–526.

Harman, Gilbert (1965). "The inference to the best explanation". In: *Philosophical Review* 74, pp. 88–95.

Harré, Rom (1986). *Varieties of Realism*. Oxford: Blackwell.

Harré, Rom (1988). "Realism and ontology". In: *Philosophia Naturalis* 25, pp. 386–398.

Heim, Irene (1982). "The Semantics of Definite and Indefinite Noun Phrases". PhD thesis. University of Massachusetts, Amherst.

Heim, Irene (1984). "A note on negative polarity and downward entailingness". In: *Proceedings of NELS*. Ed. by C. Jones and P. Sells. Vol. 14. 1983. Amherst: GLSA, pp. 98–107.

Heim, Irene (1990). "E-type pronouns and donkey anaphora". In: *Linguistics and Philosophy* 13.2, pp. 137–177.

Heim, Irene (2002). "On the projection problem for presuppositions". In: *Formal Semantics: The Essential Readings*. Ed. by Paul Portner and Barbara H. Partee. Cambridge, MA: Blackwell Publishing, pp. 249–260.

Heim, Irene and Angelika Kratzer (1998). *Semantics in Generative Grammar*. Oxford: Blackwell Publishers.

Hellman, Geoffrey (2017). "Philosophy of mathematics". In: *Handbook of Mereology*. Ed. by Hans Burkhardt et al. München: Philosophia Verlag, pp. 412–424.

Henkin, Leon A. (1950). "Completeness in the theory of types". In: *Journal of Symbolic Logic* 15, pp. 81–91.

Henkin, Leon A. (1963). "An extension of the Craig-Lyndon interpolation theorem". In: *The Journal of Symbolic Logic* 28.3, pp. 201–216.

Henkin, Leon A., James D. Monk, and Alfred Tarski (1971). *Cylindric Algebras: Part I*. Vol. 64. Studies in Logic and the Foundations of Mathematics. Amsterdam and London: North Holland Publishing Company.

Herburger, Elena (2000). *What Counts: Focus and Quantification*. Vol. 36. Linguistic Inquiry Monographs. Cambridge, MA: The MIT Press.

Higginbotham, James (1985). "On semantics". In: *Linguistic Inquiry* 16.4, pp. 547–593.

Higginbotham, James and Robert May (1981). "Questions, quantifiers and crossing". In: *The Linguistic Review* 1, pp. 41–80.

Hodges, Wilfrid (1998). "Compositionality is not the problem". In: *Logic and Logical Philosophy* 6, pp. 7–33.

Hodges, Wilfrid (2001). "Formal features of compositionality". In: *Journal of Logic, Language, and Information* 10, pp. 7–28.

Hoeksema, Jack (1986). "Monotonicity phenomena in natural language". In: *Linguistic Analysis* 16.1–2, pp. 25–40.

Hornstein, Norbert (1995). *Logical Form: From GB to Minimalism*. Oxford: Wiley-Blackwell.

Hornstein, Norbert (2001). *Move! A Minimalist Theory of Construal*. Oxford: Blackwell.

Hovda, Paul (2009). "What is classical mereology?" In: *Journal of Philosophical Logic* 38, pp. 55–82.

Icard, Thomas F. (2012). "Inclusion and exclusion in natural language". In: *Studia Logica* 100.4, pp. 705–725.

Icard, Thomas F. and Lawrence S. Moss (2013). "A complete calculus of monotone and antitone higher-order functions". In: *Topology, Algebra, and Categories in Logic (TACL 2013)*, pp. 96–99.

Icard, Thomas F., Lawrence S. Moss, and William Tune (2017). "A monotonicity calculus and its completeness". In: *Proceedings of the 15th Meeting on the Mathematics of Language*. Association for Computational Linguistics, pp. 75–87.

Ihsane, Tabea and Genoveva Puskás (2001). "Specific is not definite". In: *Generative Grammar in Geneva* 2, pp. 39–54.

Ionin, Tania and Ora Matushansky (2006). "The composition of complex cardinals". In: *Journal of Semantics* 23.4, pp. 315–360.

Jacobson, Pauline (2014). *Compositional Semantics: An Introduction to the Syntax/Semantics Interface*. Oxford: Oxford University Press.

Janssen, Theo (1983). "Foundations and Applications of Montague Grammar". PhD thesis. University of Amsterdam.

Janssen, Theo (2011). "Compositionality". In: *Handbook of Logic and Language*. Ed. by Johan van Benthem and Alice ter Meulen. 2nd ed. Elsevier, pp. 495–541.

Kadmon, Nirit (2001). *Formal Pragmatics*. Oxford: Blackwell Publishers.

Kamp, Hans (1981). "A theory of truth and semantic representation". In: *Formal Methods in the Study of Language: Part 1*. Ed. by J. Groenendijk, T. Janssen, and M. Stokhof. Amsterdam: Mathematisch Centrum, pp. 277–322.

Kant, Immanuel [1787] (1929). *Critique of Pure Reason*. Trans. by Norman Kemp Smith. 2nd ed. Basingstoke and New York: Palgrave Macmillan.

Karlsson, Fred (1999). *Finnish: An Essential Grammar*. Trans. by Andrew Chesterman. 2nd ed. London and New York: Routledge.

Karttunen, Lauri (1969). "Pronouns and variables". In: *Papers from the Fifth Regional Meeting of the Chicago Linguistic Society*. Ed. by R. I. Binnick et al. Chicago, IL: University of Chicago, pp. 108–116.

Katz, Jerrold J. and Jerry A. Fodor (1963). "The structure of a semantic theory". In: *Language* 39.2, pp. 17–210.

Katz, Jerrold J. and Paul M. Postal (1964). *An Integrated Theory of Linguistic Descriptions*. Cambridge, MA: The MIT Press.

Kayne, Richard S. (2005). "Some notes on comparative syntax, with special reference to English and French". In: *The Oxford Handbook of Comparative Syntax.* Ed. by Guglielmo Cinque and Richard S. Kayne. Oxford: Oxford University Press, pp. 3–69.

Keenan, Edward L. and Leonard M. Faltz (1985). *Boolean Semantics for Natural Language.* Vol. 23. Synthese Language Library. Dordrecht, Boston, and Lancaster: D. Reidel Publishing Company.

Keenan, Edward L. and Jonathan Stavi (1986). "A semantic characterization of natural language determiners". In: *Linguistics and Philosophy* 9, pp. 253–326.

Klima, Edward (1964). "Negation in English". In: *The Structure of Language.* Ed. by Jerry A. Fodor and Jerrold J. Katz. Englewood Cliffs, NJ: Prentice Hall, pp. 246–32.

Kratzer, Angelika (1989). "An investigation of the lumps of thought". In: *Linguistics and philosophy* 12, pp. 607–653.

Kratzer, Angelika (2002). "The event argument and the semantics of verbs". URL: https://semanticsarchive.net/Archive/GU1NWM4Z/

Kreisel, Georg (1952). "On the concepts of completeness and interpretation of formal systems". In: *Fundamenta Mathematicae* 39.1, pp. 103–127.

Kripke, Saul (1980). *Naming and Necessity*. Cambridge, MA: Harvard University Press.

Kripke, Saul (1982). *Wittgenstein on Rules and Private Language: An Elementary Exposition.* Cambridge, MA: Harvard University Press.

Kroeger, Paul R. (2019). *Analyzing Meaning: An Introduction to Semantics and Pragmatics.* 2nd corrected and slightly revised ed. Textbooks in Language Sciences 5. Berlin: Language Science Press.

Ladusaw, William (1979). "Polarity Sensitivity as Inherent Scope Relations". PhD thesis. University of Texas at Austin.

Lakoff, George (1970). "Linguistics and natural logic". In: *Synthese* 22, pp. 151–271.

Lakoff, George (1972). "Linguistics and natural logic". In: *Semantics of Natural Language.* Ed. by Donald Davidson and Gilbert Harman. Dordrecht and Boston: D. Reidel Publishing Company, pp. 545–665. Reprint of Lakoff 1970.

Landman, Fred (1996). "Plurality". In: *The Handbook of Contemporary Semantic Theory.* Ed. by Shalom Lappin. Oxford: Blackwell Publishers, pp. 425–457.

Larson, Richard (1988). "On the double object construction". In: *Linguistic Inquiry* 19, pp. 335–391.

Larson, Richard (1995). "Semantics". In: *Invitation to Cognitive Science.* Vol. 1: *Language.* Ed. by Lila R. Gleitman and Mark Liberman. General editor Danied N. Osherson. 2nd ed. Cambridge, MA, and London: The MIT Press.

Larson, Richard and Gabriel Segal (1995). *Knowledge of Meaning.* Cambridge, MA: The MIT Press.

Lasersohn, Peter (1995). *Plurality, Conjunction and Events.* Vol. 55. Studies in Linguistics and Philosophy. Dordrecht: Springer Science+Business Media.

Laury, Ritva (1997). *Demonstratives in Interaction: The Emergence of a Definite Article in Finnish.* Amsterdam and Philadelphia: John Benjamins Publishing Company.

Law, David and Peter Ludlow (1985). "Quantification without cardinality". In: *Proceedings of NELS 15.* Ed. by S. Berman, J-W. Choe, and J. McDonough. Brown University. GLSA.

Lewis, David (1975). "Languages and language". In: *Language, Mind, and Knowledge.* Ed. by Keith Gunderson. Vol. 7. Minnesota Studies in the Pholosophy of Science. Minneapolis: University of Minnesota Press, pp. 3–35.

Lewis, David (1991). *Parts of Classes.* Oxford and Cambridge, MA: Basil Blackwell.

Linebarger, Marcia C. (1987). "Negative polarity and grammatical representation". In: *Linguistics and Philosophy* 10, pp. 325–387.

Link, Godehard (1983). "The logical analysis of plurals and mass terms: A lattice-theoretical approach". In: *Formal Semantics: The Essential Readings.* Ed. by Paul Portner and Barbara H. Partee. Cambridge, MA: Blackwell Publishing, pp. 127–146.

Lipton, Peter (2004). *Inference to the Best Explanation.* 2nd ed. London: Routledge.

Lopez-Escobar, E. G. K. (1965). "An interpolation theorem for denumerably long formulas". In: *Fundamenta Mathematicae* LVII, pp. 253–272.

Ludlow, Peter (1994). "Conditionals, events, and unbound pronouns". In: *Lingua e stile* 29.2, pp. 165–183.

Ludlow, Peter (1995). "The logical form of determiners". In: *Journal of Philosophical Logic* 24, pp. 47–69.

Ludlow, Peter (2002). "LF and natural logic". In: *Logical Form and Language.* Ed. by Georg Peter and Gerhard Preyer. Oxford and New York: Oxford University Press, pp. 132–168.

Ludlow, Peter (2008). "Referential semantics for I-languages?" In: *Chomsky and his Critics*. Ed. by Louise M. Antony and Norbert Hornstein. Philosophers and their Critics. Oxford: Blackwell. Chap. 6, pp. 140–161.

Ludlow, Peter (2011). *The Philosophy of Generative Linguistics.* Oxford and New York: Oxford University Press.

Ludlow, Peter (2018). "Descriptions". In: *The Stanford Encyclopedia of Philosophy.* Ed. by Edward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University. URL: https://plato.stanford.edu/archives/fall2018/entries/descriptions/

Ludlow, Peter (2019). *Interperspectival Content.* Oxford: Oxford University Press.

Lycan, William G. (1984). "A syntactically motivated theory of conditionals". In: *Midwest Studies In Philosophy* 9.1, pp. 437–455.

Lyndon, Roger C. (1959). "Properties preserved under homomorphism". In: *Pacific Journal of Mathematics* 9.1, pp. 129–142.

MacCartney, Bill and Christopher D. Manning (2009). "An extended model of natural logic". In: *Proceedings of the Eighth International Conference on Computational Semantics*. Tilburg: Association for Computational Linguistics, pp. 14–156.

Makkai, Michael (1969). "On the model theory of denumerably long formulas with finite strings of quantifiers". In: *The Journal of Symbolic Logic* 34.3, pp. 437–459.

Martin, John N. (2013). "Distributive terms, truth, and the Port Royal logic". In: *History and Philosophy of Logic* 34.2, pp. 133–154.

May, Robert (1977). "The Grammar of Quantification". PhD thesis. MIT.

May, Robert (1985). *Logical Form: Its Structure and Derivation.* Cambridge, MA: MIT Press.

McCawley, James D. (1972). "A program for logic". In: *Semantics of Natural Language.* Ed. by Donald Davidson and Gilbert Harman. Dordrecht and Boston: D. Reidel Publishing Company, pp. 498–544.

McCawley, James D. (1981). *Everything that Linguists Have Always Wanted, to Know about Logic ... But Were Ashamed to Ask.* Chicago and London: The University of Chicago Press.

McKay, Thomas J. (2006). *Plural Predication.* Oxford and New York: Oxford University Press.

McMullin, E. (1992). *The Inference that Makes Science.* Milwaukee, WI: Marquette University Press.

Mendelson, Elliot (1997). *Introduction to Mathematical Logic.* New York: Chapman & Hall.

Mitrović, Moreno (2021). *Superparticles: A Microsemantic Theory, Typology, and History of Logical Atoms*. Studies in Natural Language & Linguistic Theory 98. Dordrecht: Springer.

Montague, Richard (1970). "English as a formal language". In: *Linguaggi Nella Societá e Nella Tecnica.* Ed. by B. Visentini. Milan: Edizioni di Communita, pp. 188–221.

Montague, Richard (1973). “The proper treatment of quantification in ordinary English”. In: *Approaches to Natural Language.* Ed. by Kaarlo Jaakko J. Hintikka, Julius M. E. Moravcsik, and Patrick Suppes. Vol. 49. Synthese Library (Monographs on Epistemology, Logic, Methodology, Philosophy of Science, Sociology of Science and of Knowledge, and on the Mathematical Methods of Social and Behavioral Sciences). Dordrecht: Springer.

Moss, Lawrence S. (2005). “Completeness in natural logic: What and why?” Workshop on Proof Theory at the Syntax/Semantics Interface. Harvard University Press. July 8–10.

Moss, Lawrence S. (2012). “The soundness of internalized polarity marking”. In: *Studia Logica,* 100.4, pp. 683–704.

Mostowski, Andrzej (1957). “On a generalization of quantifiers”. In: *Fundamenta Mathematical* 44.1, pp. 12–36.

Motohashi, Nobuyoshi (1984). “Equality and Lyndon’s interpolation theorem”. In: *The Journal of Symbolic Logic* 49.1, pp. 123–128.

Muskens, Reinhard (2010). “An analytic tableau system for natural logic”. In: *Logic, Language and Meaning.* Ed. by Maria Aloni et al. Berlin and Heidelberg: Springer, pp. 104–113.

Neale, Stephen (1990). *Descriptions.* Cambridge, MA: The MIT Press.

Newell, A., J. C. Shaw, and H. A. Simon (1957). “Empirical explorations of the logic theory machine: A case study in heuristic”. In: *Papers Presented at the February 2628, 1957, Western Joint Computer Conference: Techniques for Reliability.* IRE-AIEE-ACM ’57 (Western). Los Angeles, CA: Association for Computing Machinery, pp. 218–230.

Oberschelp, Arnold (1968). “On the Craig-Lyndon interpolation theorem”. In: *The Journal of Symbolic Logic* 33.2, pp. 271–274.

Ockham, William (1954). *Summa Logicae: Pars Secunda et Tertiae Prima.* Ed. by Philotheus Boehner. St. Bonaventure, NY: Franciscan Institute. English translation available at http://www.logicmuseum.com/wiki/Authors/Ockham/Summa_Logicae

Oliver, Alex and Timothy Smiley (2006). “A modest logic of plurals”. In: *Journal of Philosophical Logic* 35.3, pp. 317–348.

Parsons, Charles (1983). *Mathematics in Philosophy: Selected Essays.* Ithaca, NY: Cornell University Press.

Parsons, Terence (1978). “Pronouns as paraphrases”. Manuscript. URL: https://philosophy.ucla.edu/person/terence-parsons

Parsons, Terence (1990). *Events in the Semantics of English.* Current Studies in Linguistics 19. Cambridge, MA, and London: The MIT Press.

Parsons, Terence (1995). “Thematic relations and arguments”. In: *Linguistic Inquiry* 26.4, pp. 635–662.

Parsons, Terence (2008). “The development of supposition theory in the later 12th through 14th centuries”. In: *Mediaeval and Renaissance Logic*. Ed. By Dov M. Gabbay and John H. Woods. Vol. 2. Handbook of the History of Logic. Amsterdam: Elsevier, pp. 157–280.

Partee, Barbara H. (2004). *Compositionality in Formal Semantics: Selected papers by Barbara H. Partee.* Oxford: Blackwell Publishing.

Partee, Barbara H. (2005). *Supplement to Logic Unit: Logical Structure in Natural Language.* URL: http://people.umass.edu/partee/409/Logic_and_Natural_Language.pdf (visited on 12/21/2019).

Partee, Barbara H. (2016). “Formal semantics”. In: *The Cambridge Handbook of Formal Semantics.* Ed. by Maria Aloni and Paul Dekker. Cambridge: Cambridge University Press. Chap. 1, pp. 3–32.

Partee, Barbara H. and Mats E. Rooth (1983). "Generalized conjunction and type ambiguity". In: *Meaning, Use and the Interpretation of Language.* Ed. by Rainer Bäuerle, Christoph Schwarze, and Arnim von Stechow. Berlin: Walter de Gruyter & Co., pp. 361–393.

Paul of Venice (2002). *Logica Parva.* Ed., trans., and comm. by Alan R. Perreiah. Translation of the 1472 Edition. Leiden: Brill.

Pelletier, Francis J. (2004). "The principle of semantic compositionality". In: *Semantics: A Reader.* Ed. by Steven Davis and Brendan S. Gillon. Oxford and New York: Oxford University Press, pp. 133–156.

Pereltsvaig, Asya (2015). "The functional structure of N". In: *Contemporary Linguistic Parameters.* Ed. by Antonio Fábregas, Jaume Mateu, and Michael T. Putnam. Contemporary Studies in Linguistics. London: Bloomsbury Academic, pp. 303–331.

Pietroski, Paul (2003). "Quantification and second-order monadicity". In: *Philosophical Perspectives* 17, pp. 259–298.

Pietroski, Paul (2018). *Conjoining Meanings: Semantics without Truth Values.* Oxford: Oxford University Press.

Pinker, Steven (1994). *The Language Instinct: How the Mind Creates Language.* New York: HarperCollins.

Po-Ching, Yip and Don Rimmington (2004). *Chinese: A Comprehensive Grammar*. London and New York: Routledge.

Pollock, Jean Y. (1989). "Verb movements, universal grammar and the structure of IP". In: *Linguistic Inquiry* 20.3, pp. 365–424.

Portner, Paul and Barbara H. Partee, eds. (2002). *Formal Semantics: The Essential Readings.* Cambridge, MA: Blackwell Publishing.

Postal, Paul M. (1972). "The best theory". In: *Goals of Linguistic Theory.* Ed. by Stanley Peters. Englewood Cliffs, NJ: Prentice-Hall, pp. 131–179.

Progovac, Ljiljana (1990). "Free-choice Bilo in Serbo-Croatian: Existential or universal?" In: *Linguistic Inquiry* 21.1, pp. 13–135.

Psillos, S. (1999). *Scientific Realism: How Science Tracks Truth.* London: Routledge.

Putnam, Hilary (1975). "The meaning of 'meaning'". In: *Language, Mind, and Knowledge.* Ed. by K. Gunderson. Vol. 7. Minnesota Studies in the Philosophy of Science. Minneapolis, MN: University of Minnesota Press.

Putnam, Hilary (1988). *Representation and Reality.* Cambridge, MA, and London: The MIT Press.

Quine, Willard Van O. (1981). *Mathematical Logic.* Harvard University Press.

Ramchand, Gillian (2007). "Events in syntax: Modification and predication". In: *Language and Linguistics Compass* 1.5, pp. 476–497.

Reinhart, Tanya (1976). "The Syntactic Domain of Anaphora". PhD thesis. MIT.

Rips, Lance J. (1983). "Cognitive processes in propositional reasoning". In: *Psychological review* 90.1, pp. 38–71.

Rips, Lance J. (1994). *The Psychology of Proof: Deductive Reasoning in Human Thinking.* Cambridge, MA, and London: The MIT Press.

Rips, Lance J. (2008). "Logical approaches to human deductive reasoning". In: *Reasoning: Studies of Human Inference and Its Foundations.* Ed. by Jonathan E. Adler and Lance J. Rips. Cambridge and New York: Cambridge University Press, pp. 187–205.

Rips, Lance J. and F. G. Conrad (1983). "Individual differences in deduction". In: *Cognition and Brain Theory* 6, pp. 259–285.

Rizzi, Luigi (1990). *Relativized Minimality.* Cambridge, MA: The MIT Press.

Rizzi, Luigi (1997). "The fine structure of the left periphery". In: *Elements of Grammar.* Ed. by Liliane Haegeman. Dordrecht: Kluwer, pp. 281–337.

Rizzi, Luigi (2001). "Relativized minimality effects". In: *The Handbook of Contemporary Syntactic Theory.* Ed. by Mark Baltin and Chris Collins. Cambridge, MA, and Oxford: Blackwell Publishers, pp. 89–110.

Rizzi, Luigi (2004). "Locality and left periphery". In: *Structures and Beyond: The Cartography of Syntactic Structures.* Ed. by Adriana Belleti. Vol. 3. Oxford Studies in Comparative Syntax. Oxford and New York: Oxford University Press, pp. 223–251.

Roberts, Craige (1989). "Modal subordination and pronominal anaphora in discourse". In: *Linguistics and Philosophy* 12.6, pp. 683–721.

Rooth, Mats E. (1985). "Association with Focus". PhD thesis. University of Massachusetts.

Rothschild, Daniel (2011). "Explaining presupposition projection". In: *Semantics and Pragmatics* 4, pp. 1–43.

Rothstein, Susan (2017). *Semantics for Counting and Measuring.* Cambridge: Cambridge University Press.

Russell, Bertrand (1905). "On denoting". In: *Mind* 14.56. Reprinted in Russell 1956, pp. 479–493.

Russell, Bertrand (1956). *Logic and Knowledge: Essays 1901–1950.* Ed. by R. C. Marsh. London: Allen & Unwin.

Saeed, John I. (2009). *Semantics.* 3rd ed. Oxford: Wiley-Blackwell.

Sánchez Valencia, Víctor (1991). "Studies on Natural Logic and Categorial Grammar". PhD thesis. University of Amsterdam.

Sánchez Valencia, Victor (1994). "Monotonicity in medieval logic". In: *Language and Cognition* 4, pp. 161–174.

Sánchez Valencia, Victor (1995). "Natural logic: Parsing-driven inference". In: *Linguistic Analysis* 25, pp. 258–285.

Scha, Remko J. H. (1984). "Distributive, collective and cumulative quantification". In: *Truth, Interpretation and Information.* Ed. by Jeroen Groenendijk. Groningen-Amsterdam studies in semantics. Dordrecht: Foris Publications, pp. 131–158.

Schein, Barry (1993). *Plurals and Events.* Cambridge, MA: MIT Press.

Schein, Barry (2017). *'And': Conjunction Reduction Redux.* Cambridge, MA, and London: The MIT Press.

Schlenker, Philippe (2006). "Ontological symmetry in language: A brief manifesto". In: *Mind & Language* 21.4, pp. 504–539.

Sells, Peter (1985). *Restrictive and Non-restrictive Modification.* Tech. rep. CSLI-8528. Center for the Study of Language and Information, Stanford University.

Shannon, Claude E. and Warren Weaver (1949). *The Mathematical Theory of Communication*. Urbana, IL: The University of Illinois Press.

Shapiro, Stewart (1991). *Foundations without Foundationalism: A Case for Second-order Logic.* New York: Oxford University Press.

Sharvy, Richard (1980). "A more general theory of definite descriptions". In: *The Philosophical Review* 89, pp. 607–623.

Sher, Gila (1991). *The Bounds of Logic: A Generalized Viewpoint.* Cambridge, MA, and London: The MIT Press.

Shlonsky, Ur (2010). "The cartographic enterprise in syntax". In: *Language and Linguistics Compass* 4.6, pp. 417–429.

Simons, Peter (1987). *Parts: A Study in Ontology.* Oxford University Press.

Sommers, Fred (1970). "The calculus of terms". In: *Mind* 79, pp. 1–39.

Spade, Paul V. (1997). "Translation of the beginning of Walter Burley's Treatise on the Kinds of Supposition (De Suppositionibus), translated from Stephen Brown, "Walter Burleigh's Treatise De Suppositionibus and Its Influence on William of Ockham"". In: *Franciscan Studies* 32, pp. 15–64. Translation of Burley [14th century] 1972.

Spade, Paul V. (2000). *On the Purity of the Art of Logic: The Shorter and the Longer Treatises.* New Haven: Yale University Press. Translation of Burley [14th century] 1955.

Spector, Benjamin (2007). "Aspects of the pragmatics of plural morphology: On higher-order implicatures". In: *Presupposition and Implicature in Compositional Semantics.* Ed. by Uli Sauerland and Penka Stateva. Basingstoke and New York: Palgrave Macmillan, pp. 243–281.

Spector, Benjamin (2013). "Bare numerals and scalar implicatures". In: *Language and Linguistics Compass* 7.5, pp. 273–294.

Starke, Michal (2001). "Move Dissolves into Merge: a Theory of Locality". PhD thesis. University of Geneva.

Starke, Michal (2004). "On the inexistence of specifiers and the nature of heads". In: *Structures and Beyond: The Cartography of Syntactic Structures*. Ed. by Adriana Belleti. Vol. 3. Oxford Studies in Comparative Syntax. Oxford and New York: Oxford University Press, pp. 252–268.

Starke, Michal (2009). "Nanosyntax: A short primer to a new approach to language". In: *Nordlyd* 36.1, pp. 1–6

Stojanovic, Milan N., Tiffany E. Mitchell, and Darko Stefanovic (2002). "Deoxyribozyme-based logic gates". In: *Journal of the American Chemical Society* 124.14, pp. 3555–3561.

Straßburger, Lutz (2011). "Towards a Theory of Proofs of Classical Logic". Habilitation Thesis. URL: http://www.lix.polytechnique.fr/~lutz/papers/HDR.pdf

Sulkala, Helena and Merja Karjalainen (1992). *Finnish.* Descriptive Grammar Series. London: Routledge.

Suppes, Patrick (1979). "logical inference in English". In: *Studia Logica* 38, pp. 375–391.

Syed, Saurov (2017). "The structure of noun phrases in Bengali: What it tells us about phases and the universality of DPs". PhD thesis. University of Southern California.

Syed, Saurov and Andrew Simpson (2017). "On the DP/NP status of nominal projections in Bangla: Consequences for the theory of phases". In: *Glossa: A Journal of General Linguistics* 2.1, pp. 1–24.

Szabo, M. E., ed. and trans. (1969). *The Collected Papers of Gerhard Gentzen.* Amsterdam and London: North-Holland Publishing Company.

Szabó, Zoltán G. (2000). "Compositionality as supervenience". In: *Linguistics and Philosophy* 23.5, pp. 475–505.

Szabolcsi, Anna (1994). "Quantifiers in pair-list readings and the non-uniformity of quantification". In: *Proceedings of the Ninth Amsterdam Colloquium.* Ed. by Paul Dekker and Martin Stokhof. ILLC/Department of Philosophy. University of Amsterdam, pp. 646–664.

Szabolcsi, Anna (1996). "Strategies for scope taking". In: *Ways of Scope Taking.* Ed. by Anna Szabolcsi. Vol. 65. Studies in Linguistics and Philosophy. Dordrecht: Kluwer Academic Publishers, pp. 109–154.

Szabolcsi, Anna, ed. (1997). *Ways of Scope Taking.* Vol. 65. Studies in Linguistics and Philosophy. Dordrecht: Kluwer Academic Publishers.

Szabolcsi, Anna (2010). *Quantification.* Cambridge: Cambridge University Press.

Szabolcsi, Anna (2015). "What do quantifier particles do?" In: *Linguistics and Philosophy* 38.2, pp. 159–204.

Szabolcsi, Anna, Lewis Bott, and Brian McElree (2008). "The effect of negative polarity items on inference verification". In: *Journal of Semantics* 25.4, pp. 411–450.

Thomason, Richmond H., ed. (1974). *Formal Philosophy: Selected Papers of Richard Montague.* New Haven and London: Yale University Press.

Tubella, Andrea A. and Lutz Straßburger (2019). "Introduction to deep inference". Lecture notes for ESSLLI'19. University of Latvia. URL: https://www.lix.polytechnique.fr/~lutz/papers/ESSLLI19notes.pdf

Tune, William (2016). "A Lambda Calculus for Monotonicity Reasoning". PhD thesis. Indiana University.

Vaänänen, Jouko (2008). "The Craig Interpolation Theorem in abstract model theory". In: *Synthese* 164, pp. 401–420.

van Benthem, Johan (1983). "Determiners and logic". In: *Linguistics and Philosophy* 6.4, pp. 447–478.

van Benthem, Johan (1984). "Questions about quantifiers". In: *The Journal of Symbolic Logic* 49.2, pp. 443–466.

van Benthem, Johan (1986). *Essays in Logical Semantics.* Dordrecht: D. Reidel Publishing Company.

van Benthem, Johan (1991). *Logic in Action: Categories, Lambdas and Dynamic Logic.* Amsterdam: North-Holland.

van Benthem, Johan (2008). "A brief history of natural logic". In: *Logic, NavyaNyâya & Applications: Homage to Bimal Krishna Matilal.* Ed. by M. K. Chakraborty et al. London: College Publications, pp. 21–42.

van der Sandt, Rob (1992). "Presupposition projection as anaphora resolution". In: *Journal of Semantics* 9, pp. 333–377.

van Eijck, Jan (2007). "Natural logic for natural language". In: *Logic, Language, and Computation.* 6th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2005 (Batumi, GA, Sept. 12–16, 2005). Ed. by Balder D. ten Cate and Henk W. Zeevat. Berlin and Heidelberg: Springer-Verlag, pp. 216–230.

Vaught, Robert L. (1995). *Set Theory: An Introduction.* 2nd ed. Boston, Basel, and Berlin: Birkhäuser.

von Fintel, Kai (1994). "Restrictions on Quantifier Domains". PhD thesis. University of Massachusetts.

Wągiel, Marcin (2018). "Subatomic Quantification". PhD thesis. Masaryk University.

Wason, Peter C. and Philip N. Johnson-Laird (1972). *Psychology of Reasoning: Structure and Content.* Cambridge, MA, and London: Harvard University Press.

Wasow, Thomas (1972). "Anaphoric Relations in English". PhD thesis. MIT.

Westerståhl, Dag (1985). "Logical constants in quantifier languages". In: *Linguistics and Philosophy* 8, pp. 387–413.

Wiggins, David (1980). "'Most' and 'all': Some comments on a familiar programme, and on the logical form of quantified sentences". In: *Reference, Truth, and Reality: Essays on the Philosophy of Language.* Ed. by Mark de Bretton Platts. London: Routledge & Kegan Paul, pp. 318–346.

Williamson, Timothy (2004). "Philosphical 'intuitions' and scepticism about judgement". In: *Dialectica* 58.1, pp. 109–153.

Williamson, Timothy (2010). "Review of Robert Brandom's *Reason in Philosophy: Animating Ideas*". In: *Times Literary Supplement* 5579, pp. 22–23.

Williamson, Timothy (2017). "Semantic paradoxes and abductive methodology". In: *The Revenge of the Liar.* Ed. by B. Armour-Garb. Oxford: Oxford University Press, pp. 326–346.

Wittgenstein, Ludwig (1913). "Review: The science of logic". In: *The Cambridge Review* 34.853, p. 351. URL: http://fair-use.org/the-cambridge-review/1913/03/06/reviews/the-science-of-logic

Wittgenstein, Ludwig (1922). *Tractatus Logico-Philosophicus.* London: Kegan Paul.

Wittgenstein, Ludwig [1953] (2009). *Philosophical Investigations.* Ed. by P. M. S. Hacker and Joachim Schulte. Trans. by G. E. M. Anscombe, P. M. S. Hacker, and Joachim Schulte. 4th ed. Oxford: Wiley-Blackwell.

Zamansky, Anna, Nissim Francez, and Yoad Winter (2006). "A 'natural logic' inference system using the Lambek calculus". In: *Journal of Logic, Language and Information* 15.3, pp. 273–295.

Zeijlstra, Hedde (2004). "Sentential Negation and Negative Concord". PhD thesis. University of Amsterdam.

Zeman, J. Jay (1967). "A system of implicit quantification". In: *The Journal of Symbolic Logic* 32.4, pp. 48–504.

Živanović, Sašo (2002). "Kvantifikacija v naravnem človeškem jeziku [Quantification in the Natural Language]". BA thesis. University of Ljubljana, Faculty of Mathematics and Physics.

Živanović, Sašo (2007). "Kvantifikacijski vidiki logične oblike v minimalistični teoriji jezika [Quantificational Aspects of LF]". A translation of the official thesis. PhD thesis. University of Ljubljana, Faculty of Arts.

Živanović, Sašo (2008). "Varieties of most: On different readings of superlative determiners". In: *Studies in Formal Slavic Linguistics: Contributions from Formal Description of Slavic Languages 6.5.* Ed. by Franc Marušič and Rok Žaucer. Linguistik international 19. Frankfurt am Main: Peter Lang, pp. 337–354.

Zuber, Richard and Edward L. Keenan (June 2019). "A note on conservativity". In: *Journal of Semantics* 36, pp. 573–582.

Zwarts, Frans (1981). "Negatief polaire uitdrukkingen I". In: *GLOT* 4.1, pp. 35–132.

Name Index

Subject Index