UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI FILOSOFIA
“PIERO MARTINETTI”

Doctoral Programme in Philosophy and Human Sciences

Cycle XXXIV

Tractable depth-bounded approximations to some propositional logics

Towards more realistic models of logical agents

PhD Thesis

PhD candidate:
Alejandro Solares-Rojas

Supervisors:
Prof. Marcello D’Agostino
Prof. Anna Zamansky

Programme coordinator:
Prof. Andrea Pinotti

A.A. 2021/21

To my parents and siblings, whose love and support have no boundaries.

To Isa, Paolo, Pietro and Víctor, who are my Italian–South American family.

To Marcello, for believing in me and sharing with me his knowledge and wisdom.

Abstract

The depth-bounded approach seeks to provide realistic models of reasoners. Recognizing that most useful logics are idealizations in that they are either undecidable or likely to be intractable, the approach accounts for how they can be approximated in practice by resource-bounded agents. The approach has been applied to Classical Propositional Logic (**CPL**), yielding a hierarchy of tractable depth-bounded approximations to that logic, which in turn has been based on a *KE*/*KI* system.

This Thesis shows that the approach can be naturally extended to useful non-classical logics such as First-Degree Entailment (**FDE**), the Logic of Paradox (**LP**), Strong Kleene Logic ($\mathbf{K_3}$) and Intuitionistic Propositional Logic (**IPL**). To do this, we introduce a *KE*/*KI*-style system for each of those logics such that: is formulated via signed formulae, consist of linear operational rules and branching structural rule(s), can be used as a direct-proof and a refutation method, and is interesting independently of the approach in that it has an exponential speed-up on its tableau system counterpart. The latter given that we introduce a new class of examples which we prove to be hard for all tableau systems sharing the $\vee/\wedge$ rules with the classical one, but easy for their analogous *KE*-style systems. Then we focus on showing that each of our *KE*/*KI*-style systems naturally yields a hierarchy of tractable depth-bounded approximations to the respective logic, in terms of the maximum number of allowed nested applications of the branching rule(s). The rule(s) express(es) a generalized rule of bivalence, is (are) essentially cut rule(s) and govern(s) the manipulation of virtual information, which is information that an agent does not hold but she temporarily assumes as if she held it. Intuitively, the more virtual information needs to be invoked via the branching rule(s), the harder the inference is for the agent. So, the nested application the branching rule(s) provides a sensible measure of inferential depth. We also show that each hierarchy approximating **FDE**, **LP**, and $\mathbf{K_3}$, admits of a 5-valued non-deterministic semantics; whereas, paving the way for a semantical characterization of the hierarchy approximating **IPL**, we provide a 3-valued non-deterministic semantics for the full logic that fixes the meaning of the connectives without appealing to "structural" conditions.

Moreover, we show a super-polynomial lower bound for the strongest possible version of clausal tableaux on the well-known class of "truly fat" expressions (which are easy for *KE*), settling a problem left open in the literature. Further, we investigate a hierarchy of tractable depth-bounded approximations to **CPL** based only on *KE*. Finally, we propose a refinement of the p-simulation relation which is adequate to establish positive results about the superiority of a system over another with respect to proof-search.

Sommario

L'approccio a profondità limitata mira a fornire modelli realistici di agenti razionali. Constatando che le logiche più utili sono idealizzazioni in quanto sono indecidibili o probabilmente intrattabili, l'approccio mostra come possano essere approssimate in pratica da agenti con risorse limitate. L'approccio è stato applicato alla Logica Proposizionale Classica (**CPL**), fornendo una gerarchia di approssimazioni a tale logica, ognuna delle quali è a profondità limitata e trattabile, che a sua volta si è basata su un sistema *KE*/*KI*.

Questa Tesi mostra che l'approccio a profondità limitata può essere esteso naturalmente a utili logiche non classiche, come First-Degree Entailment (**FDE**), la Logica del Paradosso (**LP**), la Logica di Kleene Forte ($\mathbf{K_3}$) e la Logica Proposizionale Intuizionistica (**IPL**). A questo scopo introduciamo, per ognuna di tali logiche, un sistema in stile *KE*/*KI*, con le seguenti caratteristiche: è formulato attraverso formule segnate, consiste di regole operazionali lineari e regole strutturali di ramificazione, può essere usato come metodo di dimostrazione diretta o di refutazione, è interessante indipendentemente dall'approccio in quanto è esponenzialmente più veloce della sua controparte in termini di tableau. Mostriamo tale superiorità introducendo una classe di esempi che sono difficili per tutti i metodi basati sui tableau con le regole classiche per i connettivi $\vee$ e $\wedge$, ma facili per i corrispondenti sistemi in stile *KE*. Poi ci concentriamo sul mostrare che ognuno dei nostri sistemi in stile *KE*/*KI* produce naturalmente una gerarchia di approssimazioni alla rispettiva logica che sono a profondità limitata e trattabili, in termini del numero massimo consentito di applicazioni annidate della regola (o delle regole) di ramificazione. La regola (le regole) esprime (esprimono) una regola generalizzata di bivalenza, è (sono) essenzialmente una regola (regole) di taglio e governa (governano) la manipolazione di informazione virtuale, ovvero informazione che l'agente non possiede, ma assume temporaneamente come se ne fosse in possesso. Intuitivamente, più informazione virtuale deve essere utilizzata attraverso la regola (le regole) di ramificazione, più difficile è l'inferenza per l'agente. Dunque, l'applicazione ripetuta delle regole di ramificazione fornisce una misura della profondità dell'inferenza. Inoltre mostriamo che ogni gerarchia che approssima **FDE**, **LP** e $\mathbf{K_3}$ ammette una semantica non-deterministica a 5 valori; al contempo, prepariamo la strada per una caratterizzazione semantica della gerarchia che approssima **IPL**, forniamo una semantica non-deterministica a 3 valori che fissa il significato dei connettivi senza fare appello a condizioni "strutturali".

Inoltre, mostriamo un limite inferiore super-polinomiale per la versione più forte possibile dei tableaux clausali sulla ben nota classe di espressioni "veramente grasse" (che sono facili per *KE*), risolvendo un problema lasciato aperto in letteratura.

Dopodichè, studiamo una gerarchia di approssimazioni di **CPL** trattabili a profondità limitata basata solo su *KE*. Infine, proponiamo un raffinamento della relazione di p-simulazione che è adeguato a stabilire risultati positivi riguardo alla superiorità di un sistema rispetto ad un altro in relazione al problema della dimostrazione meccanica.

Acknowledgements

I wish to thank my supervisors, Marcello D'Agostino and Anna Zamansky, for their encouragement, advice, and guidance during the preparation of this thesis. Thanks, Anna, for your academic and personal support, particularly during the beginning of the COVID outbreak; for inviting me and giving me financial support to visit the Information Systems Department at the University of Haifa—even if it ended up being only virtually given the outbreak. Thanks, Marcello, for making me feel at home in Milan; for being always willing to help me despite your busy agenda; for believing in me and sharing with me your knowledge and wisdom; for everything I learned by working with you. I will never be able to properly express how much I am indebted to you.

Thanks are also due to João Marcos and Walter Carnielli for accepting to act as external reviewers despite their busy agendas; for very useful comments and suggestions, and for pointing out interesting future directions of research. I am also indebted to Luis Estrada-González, who encouraged and supported me to apply for the PhD Programme in the first place; particularly, thanks to his guidance, my project came out to be interesting for the admission committee. I am also grateful to Axel Barceló-Aspeitia and Carlos César Jiménez because my broad conception of logical phenomena was greatly influenced by casual discussions with them about situated and embodied cognition. (Some ideas of Barceló on these questions are published in [111, 28].) I have also benefited from conversations with Giuseppe Primiero, Hykel Hosni, other members of the LUCI Group at UniMi, and Miguel Pérez-Gaspar. Besides, I received very useful comments from brilliant researchers in the following events: 51st National Congress of the Mexican Mathematical Society (SMM), XIX International Philosophy Congress of the Philosophical Association of Mexico (AFM), First Bilateral Workshop UNAM-UniCa on Philosophical Logic, Logic Workshops–Milano Logic Group (now LUCI), Logica 2021, and the 54th National Congress of the SMM.

My deepest gratitude goes to my parents and siblings. Without your unconditional love and support, nothing of this would have been possible. Thanks, mom and dad, for supporting every single step that has led me to this achievement; for encouraging me to have a curious mind and heart; for teaching me to be humble and pursue what I love. Thanks, brother and sister, for stimulating my curiosity and confidence; for paving my way to pursue studies abroad; for making me feel that I was not that far from home.

I wish also to thank Paolo Baldi for his invaluable personal and academic support, and contagious positive attitude; for sheltering me in the most difficult part of the pandemic; for all his unfailing comments and suggestions regarding my research.

Thanks are also due to Pietro Casati, for sharing with me his atypical passion for Logic; for the wonderful discussions over topics beyond research during our long hikes and spontaneously at our home; for letting me be part of his family and for his invaluable academic and personal support. I am also grateful to Víctor Carranza-Pinedo for helping me enormously to deal with bureaucracy given my poor command of Italian; for interesting philosophical discussions and a sincere friendship. Special thanks are due to my beloved Isadora Tenório-de Araujo for all the beauty we shared, in good and bad times; for taking care of me and for letting me take care of her; for lighting up even the darkest days; for loving me just the way I am. I will never be able to properly express how much I am grateful to you four.

I also want to thank Carlos Prieto-Mendoza for being a supportive friend and colleague, and for stimulating philosophical discussions. Thanks also to María Martínez-Ordaz for her advice and for inviting me to contribute to the amazing project LógicaMX with a tutorial. Many thanks also to Valerio Buonomo for a warm welcome at UniMi and Milan in general. Thanks also to Gabriela Martínez-García for her unfailing support and wise advice along many years of friendship. Last but not least, thanks to Alma Aguilar-Rosales for stimulating discussions, sharing with me her interesting ideas on embodied cognition, and for her support during the last months of the preparation of this Thesis.

This research was funded by the Department of Philosophy "Piero Martinetti" of the University of Milan under the Project "Departments of Excellence 2018-2022" awarded by the Ministry of Education, University and Research (MIUR). Part of the research was also supported by the Israel Science Foundation (grant 550/19).

Contents

Chapter 1

Introduction

Nowadays there is a plethora of phenomena regarded as falling in the domain of Logic. The advent of non-classical logics, logical pluralism, and logical dynamics have attracted the attention of researchers from diverse disciplines into the agenda in Logic. These days, the computer scientist, the philosopher, the linguist, the mathematician, and the cognitive scientist—to name some—find a common ground in Logic, having a rather welcome and fruitful interaction, and expanding and diversifying the area in doing so. One crucial result of this interdisciplinary interaction is a tendency to conceive logical or rational phenomena as operations performed by *embodied* and *situated agents*; that is, (commonly *goal-oriented* and *interconnected*) agents endowed with a physical structure, situated in an exploitable environment, and so *resource-bounded*. According to that tendency, a logic—broadly and intuitively conceived—"is a formal and somewhat idealized description of a logical agent" [87, p. 144]. The latter being an individual (human or artificial), a team formed by individuals, an institution, a corporation, or the like.[1] Such a tendency (i) appears explicitly already in the work of Barwise and Perry [29], and Cherniak [44]; (ii) has been advocated and explored by, e.g., van Benthem [150, 151], Gabbay and Woods [87, 88], Morado and Savion [121, 133], and Aliseda [1, 2]; and (iii) is ultimately reminiscent of Aristotle's pursue for profiting from the notion of inference in the theory of argument and reasoning.[2]

[1] Yet, "institutions rather than individuals are the embodiment of [for instance] inductive logics. Much the same can be said for classical systems of deductive logic" [87, p. 156].

[2] By *inference* we shall mean the general process of drawing conclusions from a previously given body of information. In contrast, even thought there is no consensus about what counts as a (logical) *consequence relation*, we shall be working with an specific and narrow definition of it. Naturally, according to such a definition, all those relations shall be counted as inferences, but not vice versa. In particular, unlike consequence relations, inferences can be non-monotonic (e.g., abduction and induction) and even non-reflexive.

Given that tendency, the centrality in Logic of notions such as truth-preservation, logical consequence, and even inference has been questioned. Presumably, the notion of *information* has gained more importance, serving as a sort of umbrella term in the interdisciplinary investigations of logical phenomena. In turn, approaching logics from an informational perspective have been put forward in a variety of ways. Great part of this variety may be due to the, in turn, variety of ways in which the bidirectional relation between logics and information have been accounted [see 152, 113]. Throughout this Thesis we will take that relation from information to logics. However, beware that the motivation, formalization and justification in each account of why and how logics can or should be approached in informational terms varies considerably, as the very concepts of information and logics are intertwined.

1.1 The import of informational approaches in Logic

Some informational approaches to logics are well known. The notion of information played a crucial role already in Hintikka's approach to first-order logic; e.g., in his theory of *surface information* and *depth information*, and in his game-theoretic semantics [97, 98]. However, perhaps the most crystal-clear examples of informational approaches are present in the semantics of some non-classical logics. Among the latter, a widely known example is present in the informational semantics for intuitionistic logic as formulated by Beth [33] and Kripke [107]. The models associated with these semantics are intended to represent informational processes where an agent (or a group of agents) progressively gains more information about a current information state. Another notable example is the one constituted by the informational semantics for relevance logics as given by Urquhart [146]. According to this semantics, validity is defined in terms of certain valuations on a semilattice of possible pieces of information. A third popular example is Belnap-Dunn intuitive informational interpretation of *First Degree Entailment* (**FDE**, also known as Belnap-Dunn logic) as a 4-valued logic in which "a computer should think" [7, 70, 30, 31]. This logic is paraconsistent and, under Belnap-Dunn interpretation, is associated with restricting the effect of inconsistencies that might be contained in databases.

Certainly, as specially pointed out by Dunn [71], informational approaches in Logic have been pivotal in the very development of non-classical logics. In fact, at least in great part, those approaches have become worthy of attention due to the advent of non-classical logics as those mentioned in the previous paragraph, in which the orthodox truth-theoretical and inferentialist approaches have resulted less suitable when providing corresponding semantics. Moreover, it has been argued, for instance by Allo and Mares [6], that informational approaches to logics count as genuine alternatives to the truth-theoretical and inferentialist orthodox approaches, and that the former is preferable to the latter in that it allows us to provide a particularly attractive account of non-classical logics, logical rivalry and logical pluralism. As we shall suggest below, allowing us to account for those three aspects of contemporary Logic is a virtue we may expect from an informational approach.

Now, there is a highly-active and fast-growing entire field of research, founded by van Benthem [150] where the centrality of the notion of inference is in question: logical dynamics. According to this fast-expanding field, inference is one among various informational actions equally important within Logic. This field has been particularly focused on the formal study of epistemic and doxastic phenomena. Roughly, the hallmark of it is the study of how the attitudes of knowledge and belief modify in virtue of new information. Yet, in its most updated version, most of the field is focused on the study of how the distribution of knowledge and belief in multi-agent settings evolves in virtue of how those agents exchange information. Clearly, this field is pretty ambitious: it incorporates ampliative inference, belief revision, defeasible and non-monotonic reasoning, informational actions other than inference (e.g., questions, observations and dialogue), multi-agent and distributed systems, communication, and so on [see 151]. In fact, apart from questioning the centrality of the notion of inference, in this field, the corresponding direction of the relation between logic and information seems more inclined to be from the former to the latter. That is, *prima facie*, this field provides a logical account of informational dynamics, rather than an informational account of logical dynamics. Nonetheless, it can be argued that it does both and, more importantly, that narrower informational approaches—as those restricted to inference—really engage with it [see 3, 5]. We shall suggest that another virtue we may expect from an informational approach is for it to be able to embrace logical dynamics.

In parallel, informationally oriented approaches have provided a suitable conceptual framework for a fast-growing trend to think of logical phenomena as operations performed by embodied and situated agents; which can be traced back at least to the work of Barwise and Perry [29], and Cherniak [44]. This "naturalizing" or "pragmatizing" treatment of logical phenomena has been particularly adopted and fruitful in Artificial Intelligence, Formal Epistemology, Cognitive Science, Game Theory, and similar fields. Under such a conception of logical phenomena, accounting for resource-boundedness of agents has acquired crucial importance when building models that aim to be more realistic: agents, no matter whether human or artificial, have bounded cognitive and computational resources (time, memory, information, attention, energy, etc.).[3] Thus, we may also expect from an informational approach

[3]This is naturally related to a (correspondingly, programmed or instinctive) "economical" be-

that it accounts for the cost of reasoning or computing—e.g., of making an inference or a decision. In the literature various informational approaches accounting for resource-boundedness of agents have been proposed; some examples addressing inference are: an approach based on *awareness structures* and on the distinction between *explicit* and *implicit* knowledge was proposed by Levesque [112], and Fagin and Halpern [74]; an approach drew on bringing closer logical inference and *heuristics* via bridging notions such as *cognitive economy* was advanced by Morado and Savion [121, 133]; an approach directly based on computational complexity aspects—namely, on *proof size*—was put forward by Artemov and Kuznetz [11, 12, 13]; an approach that draws on the tradition of *subjective probability* was suggested by Parikh [123]; an approach based on the distinction between *actual* and *virtual* information that an agent may entertain is the "depth-bounded approach", introduced by D'Agostino and co-authors [61, 59, 56, 57, 58, 60, 63].[4]

Needless to say, some of those approaches are not incompatible with respect to each other; however, it is not obvious whether some—let alone all—of them admit of a single conceptual framework either.[5] Leaving for future work a comparison with other commensurable approaches, in this Thesis we shall restrict our analysis to D'Agostino's et al. depth-bounded approach. As argued throughout the Thesis, the reason for this is that such an approach seems promising in that it offers a natural and elegant account of resource-boundedness of agents and has shown flexible enough to be applied to a wide range of reasoning phenomena and, ultimately, as to comply with several virtues we may expect from an informational approach.

As we shall argue in the next section, accounting for logical phenomena from an informational approach is not just a formal artifact for, e.g., providing non-classical logics with suitable semantics. Rather, informational approaches allow for a better philosophical account of the wide variety of contemporary practices in Logic [see 5]. The advent of non-classical logics, logical pluralism, and the dynamic turn in Logic, together with the need of less idealized models of logical phenomena, call into

havior that agents tend to follow when performing an operation: agents tend to choose performances that require less resource-consumption—resources currently or forseeably at their disposal, of course. Game-theoretic settings offer a particularly suitable conceptual framework to account for this latter aspect. See [87, 88, 121, 133].

[4]An example of a more ambitious approach addressing not only inference, but also observation, memory and communication, as performed by resource-bounded agents, has been proposed by Solaki [139]. As might be expected, this approach draws on tools commonly used in dynamic epistemic logic such as *plausibility models* and *impossible-worlds semantics*.

[5]Presumably, a general semantic framework where various approaches can be usefully articulated is that of *awareness structures*, which is based on the distinction *explicit-implicit* knowledge, to the effect that an agent may implicitly know that a sentence follows from a set of assumptions, without being *aware* of it. See [137].

question orthodox approaches to Logic. Having arisen in the very context of that advent and need, informational approaches generally constitute natural accounts of logical phenomena as nowadays conceived and investigated.

1.2 Unraveling some virtues of informational approaches

Something labeled as an "informational approach" should have as its most basic notion that of information. Accordingly, we may expect that an informational approach to logics should investigate the properties of logical phenomena using a conceptual framework based on informational notions. In particular, when it comes to inference, such an approach should provide its own conceptual framework and not trace it upon truth-theoretic or inferentialist approaches. Thus, regardless the nature of the formalism through which that approach may be formulated—be it model-theoretic, proof-theoretic, game-theoretic, algebraic, or whichever else—such an approach should be based on informational notions. The latter clearly presupposes that a particular approach does not need to be tied to a single formalism. For instance, truth-theoretic or inferentialist approaches do not need to be tied to, respectively, model-theoretic or proof-theoretic formalisms. This does not seem controversial since it is only an emphasis on a clear distinction between formalisms and their interpretations. Specifically, a distinction between formal and intuitive conceptions of inference; "something that roughly corresponds to the interaction between pure and applied semantics where the former formalises the latter or the latter interprets the former" [p. 6, 169].

In this sense, (at least) the basis of an informational approach should be neutral regarding considerations about truth-conditions as well as inferential-conditions. As a consequence, such an approach should allow us to naturally accommodate classical as well as non-classical logics within it. What is more, beyond being a conceptual tool to deal with a variety of logics, an informational approach should provide means to account for the advent of non-classical logics, logical rivalry, and logical pluralism.

The starting point of a description of what virtues we may expect from an informational approach to logics is to spell out what the notion of "information" means within this context. We shall not attempt an in-depth analysis of this primitive and intuitive notion except for the following remarks. As stated in the relevant literature, this a highly polymorphic notion, much as the way it relates to logics. For instance, Floridi [84] has claimed that (factual semantic) information must be true. This is called by him "veridicality thesis", which basically says that an agent x is informed that p only if p is true: $I_x p \to p$. As Dunn [69] has pointed out, Floridi's claim has more to do with technical considerations about dealing with what Floridi himself called the *Bar-Hillel-Carnap* (*BHC*) *paradox* than with natural language considerations. Further, Dunn argued—and we agree with him—that it is part of the pragmatics of the concept "information" to expect information to be true, but it is not part of its semantics, i.e., of the literal meaning of the concept.

Dunn [69, 71] thinks of information as "what is left from knowledge when you subtract, justification, truth, belief, and [...] the thinker" [69, p. 589]. Roughly speaking, Dunn conceives information as any kind of semantic content. Taking a step forward in the degree of generality, Allo [3, 4, 5] thinks of information as the outcome of successful modelling (in particular, in the design of logical theories), where a model is the result of adopting a level of abstraction (LoA, for short). Remarkably, Allo proposes a relational conception of information: it depends both on the environment and on the kind of agents under consideration. Subscribing with this relational conception, we think of information as independent of justification, truth, belief, but not the agent. Specifically, we conceive *actual* information as content which is *practically* (meaning *feasibly*, as we shall explain throughout the Thesis) accessible to the agent and with which she can operate; regardless whether or not that content is true, believed, or justified.

Now, as Allo and Mares point out [6, 3]—just like other approaches to the notion of inference—informational approaches are based on a platitude about what it means to "follow from"; namely, on what they called "content-nonexpansion platitude" (CN, for short):[6]

> CN: A follows from Γ iff the content of A does not exceed the combined content of all the members in Γ;

which is clearly the *explanandum* and not the *explanans*. Informational content is almost invariably associated with a certain proportion of a given logical space: the proportion that is ruled out by that information. According to the standard order of explanation, the construction of such logical space is completely based on the identification of possibility with consistency, and thus of necessity with inconsistency of negation. Consequently, the corresponding logical space contains only possibilities that are both consistent and complete; situation which is entirely in tune with truth-theoretic principles. Therefore, under such order of explanation, it is hard to construct a logical space constituted of finer distinctions without rejecting

[6] Note that in [6, 3], Allo and Mares use "logical consequence" to refer to what here we call "inference". Again, here we shall take the former notion to denote something much narrower than the latter does.

such standard principles. Nevertheless, as Allo and Mares have argued, inverting the usual direction of explanation, informational content is not defined relative to a pre-given logical space, rather it is addressed in terms of the LoA at which the agents in question access and use information.

Basically, the inversion of the usual direction of explanation consists in taking considerations about logical discrimination (i.e., how finely propositions are individuated [101]) as conceptually prior to considerations about deductive strength. In fact, this inversion occurs at two levels. First, information does not depend anymore on a prior account of meaning, rather it is used to naturalise meaning. Second, information is no longer explained in terms of the notion of logical possibility, but directly in terms of how information—that is available in an environment—is accessed and used. Since the first is familiar from ecological realism as well as from situation semantics, the novelty of the proposal of Allo and Mares lies on the second. According to their proposal—and we also follow them on this point—different logics correspondingly formalise different ways of accessing and using information, as well as lead to different ways of carving out the logical space.

In the first place, the way agents carve out contents by specifying a logical space is determined by the information that is available to them. The notion of logical discrimination captures what can be distinguished in a given logical system and, according to Allo and Mares' proposal, constitutes the main criterion for the construction of the corresponding logical space. For instance, unlike classical logic, intuitionistic logic discriminates between A and $\neg\neg A$, and paraconsistent (e.g., relevance) logics allow us to discriminate between $A \wedge \neg A$ and $B \wedge \neg B$. Put differently, those non-classical logics allow for finer distinctions than classical logic does. This latter situation is reflected by the inverse relation between deductive strength and discriminatory power: the more a logic proves, the fewer distinctions (or discriminations) it registers [101, p. 225]).[7]

Agents can tell apart contents more or less finely and what counts as the correct LoA when doing this can only be determined once their purpose is clear.[8] According to orthodoxy, there is only one "logical" way to discriminate contents: in terms of their truth-conditions. This, together with the also orthodox view that such conditions can only be assigned in accordance with the classical truth-tables, leads to a logical monism. Nonetheless, some non-classical ways of discriminating contents—as those mentioned above present in paraconsistent and intuitionistic logics—are as

[7]There are exceptions to this relation [see 101], but all the logics we shall address in the Thesis comply with it.

[8]That is, as particularly emphasized by Allo [3, 4, 5], how the logical space is carved out depends on pragmatic/extra-logical criteria; namely, on the purpose of the agents' modelling.

logical or rational as the classical way of doing so. This latter claim is supported by various considerations, some of which are directly related to the usefulness of non-classical logics, and others to how agents choose a logical system. When adopting a logical system to model or assess an argument, the agents' choice is between deductively strong classical logic—alternatively, a contra-classical logic [see 100, 73]—and deductively weaker sub-classical logics—respectively, a corresponding subsystem. If the former has undesired consequences, the agents retreat to the latter. As pointed out above, by adopting a deductively weaker logic, (generally) the agents obtain some additional discriminatory power in return. Thus, the moral is that when evaluating a logic, agents must balance the inversely proportional virtues of logical discrimination and deductive strength to decide which logic is the most suitable for a given purpose.

In the second place, the notion of available information seems suitable of being analyzed in terms of how agents access and use information in their environment. The importance of the notions of access and use in the analysis of what counts as available information lies in the relational conception of information mentioned above: how agents access or fail to access certain information is tied to the distributed nature of it. Agents are not able to access all information at once and, due to that, using information often amounts to combining information which was accessed in different circumstances. We can plausibly relate the various ways in which agents access and combine information from different circumstances to different "reasoning-styles" and, in turn, relate these styles to different logical systems. For instance, as Allo and Mares put it [6, 3]:

1. When agents have global access (i.e., access all accessible information at once) to maximally consistent bodies of information, and the process of combining information is cumulative, then the corresponding reasoning-style is in tune with classical logic.

2. In case of local access with consistency driven cumulative use via incomplete informational states (e.g., retaining access to the information obtained in previous states) the resulting reasoning style is in accordance with intuitionistic logic.

3. When there is local access with non-cumulative use via the presence of (possibly incomplete and inconsistent) situations—without, thereby, always having access to the totality of the information agents used—the corresponding reasoning-style can be captured by a relevance or other substructural logic.

The corresponding types of environment on focus are respectively: possible worlds, proof-stages and situations.[9] Thus, following Allo and Mares, we can re-phrase CN in more accurate terms:

> CN*: For A to follow from Γ the information that A may not exceed the information that is accessible in any environment where the information that Γ is accessible as well [6, p. 173].

Thus, one of the virtues we may expect of an informational approach is that it accommodates the advent of non-classical logics, logical rivalry, and logical pluralism within it and, more importantly, that it also allows us to account for all of them. In fact, as Allo emphasizes, "the main virtue of this type of approach is the connection it establishes between two main concerns in logical (and other types of formal) modelling: the ability to extract information from our model (inference), and the ability to distinguish between relevant properties of the model (discrimination)" [3, p. 23]. As he also says, this latter virtue is the clue to show that informational approaches really engage with logical dynamics. The abilities of extracting information from a model and of making some distinctions in a model are concerns clearly present in the development of dynamic epistemic and doxastic logics:

> Extracting information from an epistemic or doxastic model is what we do when (a1) we assign beliefs or ascribe knowledge to an agent, and (b1) we predict the effect of certain actions on their knowledge or beliefs. Distinguishing relevant features of a model is what we do when (a2) we compare or contrast the epistemic states of different agents, and (b2) compare the effect of different types of actions [3, p. 26].

So, the inverse relation between deductive strength and discrimination underlying informational approaches to logics is compatible with logical dynamics in the following sense: when the inference at issue is powerful, the corresponding agents seem highly knowledgeable according to the model; whereas, when the underlying inference is more discriminating, intuitively distinct knowledge or belief-states are not collapsed by the model. However, although this compatibility seems plausible, it also seems quite reductive. The strength of dynamic epistemic and doxastic logics comes precisely from the wide range of informational actions that they can cover. Arguably, those logics represent cases where an increase in discriminatory power does not transparently correspond to a decrease in deductive power; thus, not being in harmony—if not in a contrived way—with the aforementioned inverse

[9] Clearly, in the classical case, these informational conditions boil down to truth-theoretical conditions.

relation. Accordingly, the dynamic turn in Logic represents a challenge for informational approaches. However, as Allo [3] points out, that challenge does not seem insurmountable: First, the centrality of CN* might have to be reconsidered. Second, the way in which deductive strength and logical discrimination are related will have to be correspondingly generalised.

Now, orthodoxy has it that logics are uninformative. The orthodox interpretation of CN, inherited to CN*, tells us that the validity of purely deductive inference depends exclusively on the condition that the information carried by the conclusion of an inference is already "contained" in the information carried by its assumptions—i.e., in the initial information explicitly possessed by the agent. That is, according to the received view, purely deductive reasoning is *analytic*. However, this view completely disregards the computational and cognitive cost or effort needed to "extract" the conclusion from the assumptions. In fact, such an interpretation of CN and CN* clashes with the fact that most interesting and useful logics are either undecidable or likely to be intractable. For instance, classical first order logic and the main systems of relevance logics (**T**, **E**, **R**) are undecidable [45, 145, 147], Intuitionistic Propositional Logic (**IPL**) is PSPACE-complete [140], and **CPL** and **FDE** are both co-NP complete [47, 148]. These computabilty and complexity results strongly suggest that many interesting logics are far from being uninformative. More specifically, they suggest that the conclusion of certain complex inferences may convey information that is not contained in the assumptions in the objective sense that there is no—and probably there will never be—feasible procedure for extracting that information from the information conveyed by the assumptions. So, in that sense, these latter inferences should be regarded as *synthetic*.[10]

Most useful and interesting logics are uninformative only for highly idealized—e.g., logically omniscient—agents. We cannot realistically assume that a rational, yet resource-bounded, agent be informed of all conclusions *in principle* obtainable from the information she explicitly possesses. Put differently, we cannot expect a real agent—no matter whether human or artificial—to be always able to recognize in *practice* that a certain conclusion follows from a given set of assumptions. This is a source of major difficulties in research areas—such as Economics, AI, Cognitive Science and Philosophy—that are in need of less idealized, yet theoretically principled, models of logical agents with bounded cognitive and computational resources. We suggest that another virtue we may expect from an informational approach is that it provides means to model realistic resource-bounded agents, from both the cog-

[10]Incidentally, this defies the persistent dogma of empiricism according to which all logical inferences should be analytic. See [57, 110].

nitive and computational viewpoints.[11] That is, we may expect that informational approaches allow us to drop highly idealized assumptions about agents' inferential capabilities in a theoretically principled way. In particular, these approaches may provide means to solve the well known anomalies that the orthodoxy has raised; namely, the BHC paradox, the enduring scandal of deduction, and the problem of logical omniscience.[12]

As any formal model, logics involve a good deal of idealized assumptions and, thus, they are not intended to faithfully represent the actual inferential power of real agents. Their theoretical character lies on a purpose-based equilibrium between their normative component—which allows us to disregard, e.g., cognitive biases or distortions on the agents' behavior or performance—and their descriptive component—whose level of faithfulness depends on the corresponding modelling goal. However, even from a prescriptive viewpoint, the requirements that logics impose on the agents are too strong. Thus, from this perspective, logics are seen as limiting normative models to which *approximating* models should converge. To put it with D'Agostino: the appeal to an "idealized reasoner" has usually the effect of sweeping under the rug a good deal of interesting questions, including how idealized such a reasoner should be. Idealization may well be a matter of degree [56, p. 19]. Thereby, we may expect that—regardless being informational or not—an approach to logics accounting for resource-bounded agents provides means to define a hierarchy of increasingly idealized logical agents, in terms of correspondingly stronger consequence relations. In turn, such a hierarchy of approximations to the "perfect reasoner"—classical or not—should provide all the flexibility required by a suitable model of practical rationality.

1.3 The basics and origin of the depth-bounded approach

The depth-bounded approach constitutes an informational approach to Logic that seeks to provide more realistic, but still theoretically principled, models of resource-bounded agents. The approach departs from the fact that, as mentioned above, most interesting logical systems are either undecidable or likely to be intractable. Based on the idea that those systems are idealizations, the approach aims to define how those systems can be *approximated* in practice by realistic resource-bounded agents.

[11] Of course, logical models involve kinds of idealizations other than computational or cognitive, but we are addressing only the latter here.

[12] See [56] for a recapitulation of these anomalies.

Extending some ideas of Gabbay [85, 87, 86] and related to some research in Computer Science and AI (see Section 2.4 for references), D'Agostino and co-authors [61, 59, 56, 57, 58, 60, 63] have pointed out that the normative ideal that the orthodoxy imposes over logical agents can only be *approximated* in practice. In D'Agostino words: From this point of view, it makes sense to require as in [86] that a logical system consist not only in an algorithmic or semantic characterization of a logic **L**, but also in a definition of how this logic **L** can be *approximated in practice* by realistic agents, no matter whether human or artificial [58, p. 80]. Thus, D'Agostino's and co-authors introduced the depth-bounded approach as applied to **CPL**, which provides an account of how this logic can be approximated in practice by resource-bounded agents. This was done in two moves: (i) by providing a semantic and proof-theoretic characterization of a tractable 0-depth approximation and (ii) by defining an infinite hierarchy of tractable k-depth approximations, which can be naturally related to a hierarchy of realistic resource-bounded agents, and admits of an elegant proof-theoretic characterization.

A key idea underlying the depth-bounded approach to **CPL** is that the meaning of a connective is specified solely in terms of the information that is *actually* possessed by an agent, i.e., information practically accessible to her and with which she can operate. This kind of information is called *actual*, and we shall use the verb "to *hold*" as synonymous with "to actually possess". The semantics is ultimately based on intuitive, albeit non-deterministic, 3-valued tables that were first put forward by W.V.O. Quine [129] to capture the "primitive" meaning of the logical constants. The values have a natural informational interpretation ("accept", "reject", "abstain"). The proof-theoretic characterization given in [59, 58] is based on introduction and elimination (intelim) rules that, unlike those of Gentzen-style natural deduction, involve no "discharge" of hypotheses. The *0-depth approximation* consists of the consequence relation associated with the intelim rules only, is computationally easy (tractable) and corresponds to Quine's non-deterministic semantics. The *depth* of **CPL**-inferences is measured in terms of the maximum number of nested applications of a single *branching* rule, which is a Classical Dilemma rule called *PB* ("Principle of Bivalence"). *PB* governs the manipulation of *virtual* information, i.e., hypothetical information that an agent does not hold, but she temporarily *assumes as if* she held it. Intuitively, the more times such *virtual* information needs to be invoked via *PB*, the harder the corresponding inference is for any agent who is able to perform at least 0-depth inferences, both from the computational and the cognitive point of view. Thus, the nested applications of that rule provide a sensible measure of inferential depth. In essence, each k-depth logic corresponds to a limited capability of manipulating virtual information. The underlying intuition is that we cannot expect that an agent using a *limiting* logic **L** effectively perform all the valid inferences of **L**, but only those that are within the reach of its limited computational and cognitive resources.

Within the approach, the inferential depth associated with an agent is not intended to be interpreted as an upper bound on the inferential power of her. Rather, it is understood as the maximum depth for which it is guaranteed that, if she holds the information explicitly carried by the assumptions, she holds the information explicitly carried by the conclusion; the latter already carried by the assumptions but (in general) only implicitly. It is worth noting that an agent may not be aware even of easy consequence of her assumptions and there is still a difference between implicit and explicit information. Besides, even implicit information that can be feasibly extracted from explicit one requires consumption of resources. Nevertheless, the depth-bounded approach is not focused on the distinction between explicit and implicit information, but on the distinction between two kinds of implicit information. One kind is what we call *actual* information which the agent can feasibly extract by using only information that she holds. The other kind is what we call *virtual* information that, in turn, essentially requires the simulation of potential information that the agent does not hold. As recalled in the next Chapter, it turns out that virtual information may also be feasibly extracted whenever, precisely, the nested use of virtual information is limited.

Now, the proof-theoretic characterization of the depth-bounded approach to **CPL** is interesting in its own right. It is half-way between a *classical* version of natural deduction—which mirrors the classical meaning of the connectives and not their intuitionistic meaning as Gentzen's original rules—and the method of semantic tableaux. Concretely, it results from combining the classical proof systems *KE* and *KI*, which were introduced in [117, 118, 119], and were shown in [53, 54, 65] to be computationally and proof-theoretically advantageous—especially with respect to standard cut-free refutation systems such as analytic tableaux [e.g., 138]. The hallmark of *KE* and *KI*—as well as the system resulting from combining them—is that they reduce the amount of branching to a minimum by making all branches mutually exclusive. Accordingly, they have a single branching rule—the rule *PB* mentioned above—and the rest of their rules have all a linear format. While the latter rules are operational, *PB* is structural in that does not involve any specific logical operator; besides, *PB* is essentially a non-eliminable *cut* rule. Thus, it turns out to be quite natural to investigate the subsystems that result from limiting the applications of that single structural rule. So, in [59, 58], the system resulting from combining *KE* and *KI*—henceforth dubbed *intelim method*—is used to provide a proof-theoretic characterization of a hierarchy of tractable depth-bounded approximations to **CPL**.

Namely, the operational rules are taken as fixing the meaning of the connectives solely in terms of actual information, and require no "discharge" of temporary assumptions. In turn, the depth of inferences is measured in terms of the maximum number of nested applications of PB that are allowed.[13]

The basic, 0-depth, logic of the hierarchy of approximations to **CPL** yielded by the approach is identified with the logic of the inferences that can be drawn by using only actual information, and whose validity can be determined on the sole basis of the informational meaning of the connectives. In turn, the k-depth logics are the logics associated with inferences whose validity cannot be justified solely by the meaning of the connectives and requires, if only temporarily, the introduction of virtual information allowed up to a number a number of times k. This latter information is not even implicitly contained in the information carried by the assumptions. Thereby, only the 0-depth logic complies with CN* above, in that its valid inferences are *analytic* in the strict sense of being justified only by virtue of the way in which the language is immediately used. In fact, the inferences associated with such a logic convey no information at all by definition, and so are in tune with the tenet that analytic inferences are utterly uninformative: the conclusion is "contained" in the assumptions. However, the valid inferences of the 0-depth logic are only a subclass of the valid inferences of **CPL** and, therefore, the sense of "analytic" they characterize is stricter than the general one regarding *full* **CPL**. While such a full logic is most likely intractable, the 0-depth logic is tractable; i.e., the valid inferences of the latter can be recognized in feasible (polynomial) time. Put differently, the problem of deciding if the conclusion is "contained" in the assumptions—indeed, the discovery of a strictly analytic proof of the conclusion from the assumptions—is tractable, as we would expect from any sensible notion of "containment".

By contrast, the inferences of the k-depth logics are *synthetic* and so do not comply with CN*. This in the twofold sense that their validity does not depend solely on the meaning of the connectives and their conclusion conveys information that is not even implicitly contained in the information carried by the assumptions—in an objective, not merely psychological sense. Accordingly, besides being related to increasingly levels of cognitive effort or computational cost, the levels of the hierarchy of approximations yielded by the approach can be naturally associated with degrees of syntheticity of inferences. Still, crucially, each k-depth logic (for fixed k) inherits the tractability of the basic logic—yet the complexity of the decision proce-

[13] By contrast, in Gentzen-style proof systems, some of the "discharge" rules of natural deduction, as well as their counterparts in the sequent calculus, make essential use of virtual information. Given that in Gentzen-style systems cut is eliminable, no hierarchy of approximations can be defined by limiting the application of the cut rule.

dure grows with k. Thereby, the depth-bounded approach provides means to model resource-bounded agents, whose inferential power is related to increasingly stronger consequence relations which are all tractable—while their limit remains, of course, intractable.

The hierarchy of tractable approximations to **CPL** yielded by the approach—as well as ideas thereof—have been further investigated and applied in recent works. For example: in [52, 56] it has been outlined how the approach provides means to solve the BHC paradox, the enduring scandal of deduction, and the problem of logical omniscience; the hierarchy has served as a basis for defining approximations of belief functions [25], probabilities [27] and their qualitative counterpart [26]; a multiagent setting for the approach started to be explored in [46]; and a concrete application of the approach in the context of answer set programming was introduced in [24]. There is also ongoing research exploring extensions of the approach to classical first-order logic and epistemic logics [64, 110].

1.4 The Thesis contribution

The depth-bounded approach to **CPL**, as remarked in [58], is the first step of a more general research program that aims to define similar approximations to first-order logic and to a variety of non-classical logics. A preliminary step of the first order case can be found in [64]. In this Thesis we first reassess the depth-bounded approach to **CPL** focusing on its proof-theoretical basis, and then show that the approach can be naturally extended to cover (at least some) propositional non-classical logics.

The Thesis starts by succinctly recalling the main notions and motivations underlying the approach. After recalling some basic working definitions, non-deterministic semantics are motivated via Quine's theory of the "primitive" meaning of the (classical) connectives [129], and formal definitions, together with few basic properties thereof, are recalled as stated by Avron and co-authors [e.g., 22]. General aspects of the theory of relative complexity of proof systems are briefly recalled in turn. This is done in terms of the usual tool for comparing the power of those systems, i.e., the p-simulation relation introduced by Cook and Reckhow [49]. Besides, the notion of *approximation system* is motivated and defined in terms of accounting for the resource-boundedness of realistic agents.

The rest of Part I of the Thesis is devoted to recall and reassess the depth-bounded approach to **CPL**. We do this with a special focus on the proof-theoretic basis of the approach which, again, is a proof system resulting from the combination of the systems *KE* and *KI*, here dubbed *intelim method*. Our main reason to pay special attention to the proof-theoretic basis is that it is interesting independently of the approach. Namely, it was shown by D'Agostino and Mondadori [53, 54, 65] that the intelim method is computationally and proof-theoretically advantageous, especially with respect to standard cut-free refutation systems such as analytic tableaux. We recall the main results regarding the computational advantages of the intelim method in terms of the *p*-simulation relation and focused on the subsystem *KE* of the method, as those results were originally stated in [53, 54, 65]. (*KE* alone is a well-known proof system for **CPL** and has been widely and fruitfully studied in the area of automated reasoning [e.g., 75, 76, 42].)

Then, we prove new lower bounds on analytic tableaux, which strengthen and extend the results given by by D'Agostino and Mondadori about the superiority of *KE* over analytic tableaux, and settle a problem hitherto left open in the literature. Namely, we introduce a class of examples in the pure disjunction-conjunction fragment of the language, and prove a super-polynomial lower bound on that class of examples for all tableau methods sharing the $\vee$ and $\wedge$ rules with classical tableaux. These include known tableau methods for a variety of logics; for instance, tableaux for First Degree Entailment (**FDE**) [e.g., 53, 81], the Logic of Paradox (**LP**) and Strong Kleene Logic ($\mathbf{K_3}$) [e.g., 82, 127, 15], and Intuitionistic Propositional Logic (**IPL**) [e.g., 79, 80]. On the other hand, our new examples are easy for *KE*-style (and so *KE*/*KI*-style) variants of those tableau systems. Moreover, we show a super-polynomial lower bound for the strongest (in terms of the *p*-simulation relation) possible version of clausal tableaux on the class of "truly fat" expressions used by D'Agostino in [54] to state a super-polynomial lower bound for simple clausal tableaux (by contrast, as it is well-known, the "truly fat" expressions are easy for truth-tables and *KE*). As explained in Section 3.3, this settles a problem left open in [115, 116, 9].

Part I continues by recalling how the depth-bounded approach was applied to **CPL** by D'Agostino and co-authors [e.g., 61, 56, 57, 58]. Although our presentation of such an application is mostly focused on the proof-theoretic underpinning provided by the intelim method, we also recall the semantical characterization of the hierarchy of approximations yielded by the approach. Namely, a 3-valued non-deterministic semantics which ultimately stems from Quine's theory of the "primitive" meaning of the connectives. Our special focus on the proof-theoretic underpinning at that point of Part I is due to the fact that the characterization in terms of the non-deterministic semantics seems to suggest an exponential blow up. Indeed, the tractability of the approximations yielded by the approach was shown via their proof-theoretic characterization. We close Part I by investigating how the approach can be applied to **CPL** taking as a proof-theoretic basis *only KE*. Concretely, following a suggestion by D'Agostino and Mondadori [65], we define a hierarchy of depth-bounded approximations based on *KE*. Besides, we provide a semantical characterization of this latter hierarchy in terms of a corresponding 3-valued non-deterministic semantics. Although arguably less natural than the analogous hierarchy based on the intelim method, the hierarchy based on *KE* may be still preferred for potential uses in automated reasoning.[14]

Part II is devoted to show that the depth-bounded approach can be naturally extended to (at least some) propositional non-classical logics. Concretely, we extend the depth-bounded approach to **FDE**, **LP**, $\mathbf{K_3}$ and **IPL**. Despite all these logics admit of intuitive informational interpretations, they are all likely to be computationally intractable. The first three (many-valued) logics are closely related to each other and are all co-NP complete; while **IPL** is PSPACE-complete. We approach each of these four logics by means of a corresponding *KE*/*KI*-style system such that: (i) is formulated in terms of signed formulae, where the signs have an intuitive informational interpretation; (ii) has linear introduction and elimination rules, which fix the meaning of the connectives; (iii) has branching structural rule(s) expressing a generalized rule of bivalence; (iv) can be used as both a direct-proof and a refutation method; (v) obeys the subformula property. Given the new lower bounds proven in Section 3.3, these systems are interesting independently of the approach mainly because they have an exponential speed-up on their tableau counterparts.

Then we focus on showing that each of our *KE*/*KI*-style systems naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations to the corresponding logic, in terms of the maximum number of nested applications that are allowed of the branching rule(s).[15] The latter is (are) essentially cut rule(s) which intuitively govern(s) the manipulation of virtual information, as opposed to the introduction and elimination rules that intuitively govern the use of actual information. The key intuition is that the more virtual information needs to be invoked via the branching rule(s), the harder the inference is for the agent, both from the cognitive and computational viewpoints. Thus, the nested application of those rules provides a sensible measure of inferential depth, and so the levels of the corresponding hierarchy can be naturally related to the inferential power of agents.

Furthermore, we show that, in the case of the many-valued logics, each hierarchy admits of a 5-valued non-deterministic semantics. Regarding **IPL**, we pave the way for a non-deterministic semantics suitable for the resulting hierarchy by providing an alternative 3-valued non-deterministic semantics for full **IPL** which specifies the

[14] *KE* with limited bivalence was also investigated by Finger and Gabbay [e.g., 75, 76], but their approach is different to ours.

[15] In the case of the many-valued logics, tractability of the approximations is proven; whereas, in the case of **IPL**, it is (for now) only conjectured.

meaning of the connectives without appealing to any "structural" condition.[16]

In Part III, we come back to the general issue of the relative complexity of proof systems. We argue that the *p*-simulation relation, although certainly adequate to establish negative results, is misleading to establish positive results about the superiority of a proof system over another with respect to the *problem of mechanical proof*. In a nutshell, we are usually interested in knowing not only that a system admits short proofs in cases in which another does not, but also how hard it is to find such short proofs. A natural way of measuring the difficulty of finding the solution of a problem within a given formal system is in terms of the amount of information required to obtain it. This amount of information is, in turn, inversely related to the probability of finding the required solution "by chance", using the rules of the formal system completely "at random". So, it seems natural to measure the relative difficulty of finding short proofs within two proof systems in terms of the relative frequency with which such short proofs are found when we apply the rules of the system "blindly".

Based on these intuitions, we propose a way of enhancing the *p*-simulation relation and produce results which are more relevant to the problem of mechanical proof. Specifically, we define a preorder relation, called *p-emulation*, which is more adequate than *p*-simulation to capture the intuitive meaning of "more efficient" when referred to non-deterministic algorithms. In fact, we show that "S_2 *p*-emulates S_1 but not viceversa" is a good rendering of the intuitive notion of "S_2 is a refinement of S_1", thus allowing for relevant positive results about the relative efficiency of logical systems. Moreover, we also show how such results can be made stable via a stronger relation that we call *monotonic p-refinement*. Further, we test our definitions and results thereof with a case study: *KE* vs. Smullyan's (binary) tableaux.

[16]Part of our results on the hierarchy of approximations to **FDE** has been presented at Logica 2021 and submitted for publication in the proceedings as a joint paper with Prof. D'Agostino.

Part I

Depth-bounded classical logic and complexity issues

Chapter 2

Preliminaries

2.1 Basic working definitions

Let $\mathcal{L}$ be some *propositional language*, each of its *connectives* having a specific natural number as its arity. 0-ary connectives are called *propositional constants.* In turn, let $F(\mathcal{L})$ and $At(\mathcal{L})$ respectively be the set of well-formed and atomic formulae of $\mathcal{L}$. We use $p, q, r, ...$, possibly with subscripts, as metalinguistic variables for atomic $\mathcal{L}$-formulae; $A, B, C, ...$, possibly with subscripts, for arbitrary $\mathcal{L}$-formulae; and $\Gamma, \Delta, \Lambda, ...$, possibly with subscripts, to vary over sets of $\mathcal{L}$-formulae.[1] We shall assume that all propositional languages share the same set of atomic variables and, so, we shall identify a language $\mathcal{L}$ with $\mathcal{C}(\mathcal{L})$. Now, by a *literal* we mean, as usual, a formula which is either an atomic variable (*positive* literal) or the negation of an atomic variable (*negative* literal). In turn, a *clause* is a disjunction of literals.[2] Finally, the *complement* of a formula A, is equal to $\neg B$ if $A = B$ and to B if $A = \neg B$.

Definition 2.1.1. For every formula A:

- A *subformula* of A is defined inductively as follows:

  1. A is a subformula of A.
  2. If $\neg B$ is a subformula of A, then so is B.
  3. For every binary operator $\circ$, if $B \circ C$ is a subformula of A, then so are B and C.

[1]For readability, when $\mathcal{L}$ is clear by the context, we will omit the prefix $\mathcal{L}$ in "$\mathcal{L}$-formula(e)".

[2]Given that classical disjunction is commutative and idempotent, in all logics with a classical disjunction, a clause can be regarded as a finite set of literals and denoted simply by listing its elements.

4. Nothing else is a subformula of A.

- A *proper subformula* of A is any subformula of A that is different from A.
- An *immediate subformula* of A is any proper subformula of A that is not a proper subformula of any proper subformula of A.

We denote by sub the function that maps any given set Γ of formulae to the set of all its subformulae, and by at the function that maps any given Γ to the set of its atomic subformulae. Moreover, we define the *degree* of a $\mathcal{L}$-formula A as the number of occurrences of connectives in A.

Now—as the plethora of phenomena counted as "logical" witnesses—there is no universally accepted definition of (logical) *consequence* nor of *logic* or *logical system* (see [86] for a comprehensive definition). As a working definition, in the context of this Thesis we shall call *consequence relation* on a language $\mathcal{L}$ any relation $\mid\!\sim \subseteq 2^{F(\mathcal{L})} \times F(\mathcal{L})$, satisfying the following conditions:

Reflexivity: If $A \in \Gamma$, then $\Gamma \mid\!\sim A$.

Monotonicity: If $\Gamma \mid\!\sim A$, then $\Gamma \cup \Delta \mid\!\sim A$.

In turn, a *Tarskian consequence relation* (Tcr for short) on $\mathcal{L}$ is a consequence relation on $\mathcal{L}$ satisfying the following additional condition:

Cut for sets: If $\Gamma \mid\!\sim A$ for every $A \in \Delta$ and $\Gamma \cup \Delta \mid\!\sim B$, then $\Gamma \mid\!\sim B$.

An $\mathcal{L}$-*substitution* is a function $\sigma : F(\mathcal{L}) \longrightarrow F(\mathcal{L})$ such that for every n-ary connective $\diamond$ and formulae $A_1, ..., A_n$,

$$\sigma(\diamond(A_1, ..., A_n)) = \begin{cases} \diamond(\sigma(A_1), ..., \sigma(A_n)) & \text{if } n > 0 \\ \diamond & \text{if } n = 0 \end{cases}$$

A *Tarskian propositional logic* is a pair $\mathbf{L}=\langle \mathcal{L}, \mid\!\sim_{\mathbf{L}} \rangle$, where $\mathcal{L}$ is a propositional language and $\mid\!\sim_{\mathbf{L}}$ is a Tcr on $\mathcal{L}$ satisfying the following additional condition:

Structurality: If $\Gamma \mid\!\sim_{\mathbf{L}} A$, then $\sigma(\Gamma) \mid\!\sim_{\mathbf{L}} \sigma(A)$ for every $\mathcal{L}$-substitution σ.[3]

[3]Where $\sigma(\Gamma)$ is short for $\{\sigma(A) \mid A \in \Gamma\}$.

Definitions 2.1.2. Let $\mid\!\sim$ be a Tcr for $\mathcal{L}$. $\mid\!\sim$ is *finitary* if for every Γ and every A such that $\Gamma\mid\!\sim A$, there is a finite $\Delta \subseteq \Gamma$ such that $\Delta\mid\!\sim A$. In turn, a Tarskian propositional logic $\langle \mathcal{L}, \mid\!\sim_{\mathbf{L}} \rangle$ is *finitary* if so is $\mid\!\sim_{\mathbf{L}}$.

In a finitary Tarskian propositional logic, the following "restricted" version of transitivity suffices:

Cut (*Transitivity*): If $\Gamma\mid\!\sim A$ and $\Gamma \cup \{A\}\mid\!\sim B$, then $\Gamma\mid\!\sim B$.

Since finitariness is essential for *practical* reasoning—where a conclusion is always derived from a finite set of assumptions—throughout the thesis we are interested only in finitary logics.

2.2 Background on non-deterministic semantics

2.2.1 Motivation and key idea

An early case of non-deterministic semantics—closely related to the depth-bounded approach to **CPL**—can be found in Quine's *The Roots of Reference* [129].[4] There he outlined a dispositional theory of what he called the "primitive" meaning of the connectives and observed that this semantics fails to be truth-functional. More generally, he suggested that a sentence is *analytic* for the native speakers of a language when: (i) they learn its truth in the very process of learning how to use the words occurring in it, or (ii) when they can obtain it from such basic analytic truths via inference rules whose validity is also learned in the same process of learning the words. In particular, Quine argued that we learn the meaning of the logical words by "finding connections of dispositions" [129, p. 78]. While the governing circumstances that fix the meaning of negation are simple—namely, we learn to assent (respectively dissent) to $\neg A$ exactly when we dissent (respectively assent) to A—the cases of conjunction and disjunction are not that straightforward:

> A governing circumstance that goes far towards fixing its meaning is that a conjunction commands assent when and only when each component does. [...] It is in dissent that the rub comes. [...] The circumstances of dissent from a conjunction have to be mastered independently of the excessively simple rule of assent. Still, one of the rules of dissent is simple enough: the conjunction commands dissent whenever a component does.

[4] See [22, 19] for a comprehensive list of references on non-deterministic semantics.

p \ q	assent	abstain	dissent
assent	assent	abstain	dissent
abstain	abstain	?	dissent
dissent	dissent	dissent	dissent

p \ q	assent	abstain	dissent
assent	assent	assent	assent
abstain	assent	?	abstain
dissent	assent	abstain	dissent

Table 2.1: Quine's incomplete 3-valued tables for conjunction (left) and disjunction (right)

> [...] Conjunction has its blind spot, however, when neither component commands assent or dissent. There is no direct way of mastering this quarter. In some such cases the conjunction commands dissent and in others it commands nothing. This sector is mastered only later, in theory-laden ways. Where the components are "it is a mouse" and "it is a chipmunk", and neither is affirmed nor denied, the conjunction will still be denied. But where the components are "it is a mouse" and "it is in the kitchen", and neither is affirmed nor denied, the conjunction will perhaps be left in abeyance. [...]
> Alternation [disjunction], like conjunction, has its blind quarter where neither component commands assent or dissent. We might assent to the alternation of "it is a mouse" and "it is chipmunk" or we might abstain [129, p. 76-77].

Quine's proposal calls for a 3-valued logic that fails to be fully truth-functional, in that the truth-tables for conjunction and disjunction are incomplete.[5] The latter are reproduced in Tab. 2.1 [129, p. 77].

According to Quine's proposal, these incomplete tables yield conjunction and disjunction operators which are "more primitive than the genuine truth-functional conjunction and disjunction, in that they can be learned by induction from observation of verdictive behaviour" [129, p. 78]. According to the tables, some logical laws may qualify as analytic and some other may not. For instance—concerning the intuitionist—Quine observes that the law of excluded middle is not bound up with the very learning of "or" and "not", as described by the tables, and indeed cannot be derived from them. Thus, "it lies rather in the blind quarter of alter-

[5]In fact, Quine refers to the corresponding tables for the connectives as "verdict tables"; the "verdicts" being assent, dissent and abstention.

nation. Perhaps [...] should be seen as synthetic" [129, p. 80]. Quine relates this situation—according to which some logical laws are not bound up with the learning of the logical words—to the enduring disagreement on them.[6]

The semantics introduced by Quine was independently re-proposed—with no apparent connection with the intuitive interpretation given by Quine—by Crawford and Etherington [51]. Specifically, they re-introduced such a 3-valued non-deterministic semantics for investigating tractable inference and claimed (without proof) that it provides a characterization of (an extension of) unit resolution. As we shall explain in Chapter 3, this 3-valued non-deterministic truth-tables and ideas thereof are perfectly in tune with the depth-bounded approach as applied to **CPL**. In this section, however, we focus on the fact that Quine's and Crawford-Etherington's proposals are particular cases of the key idea underlying non-deterministic semantics. Namely, for instance, in Quine's tables the entries in which both arguments have the value "abstain" yield two alternative possible values, meaning that the value of the compound sentence is not uniquely determined by the values of its immediate subformulae but can be either of the two possible values.

The general theory of non-deterministic semantics has been articulated and extensively investigated by Avron and co-authors [e.g. 20, 21, 16, 17, 18, 22, 19], who have used those semantics particularly for studying proof-theoretic properties of Gentzen-style sequent calculi. The primary notion of such a theory is that of *non-deterministic matrix* (*Nmatrix*) which, in turn, is a natural generalization of the notion of ordinary many-valued matrix. The principle of truth-functionality (also known as, compositionality) is basic in many-valued logic in general, and in classical logic in particular. This principle dictates that the truth-value of a complex formula is uniquely determined by the truth-values of its subformulae. Nevertheless, as Avron and Zamansky say—and as exemplified by Quine's considerations above—"real-world information is inescapably incomplete, uncertain, vague, imprecise or inconsistent, and these phenomena are in an obvious conflict with the principle of truth-functionality" [22, p. 227]. The notion of Nmatrix provides a possible solution to this problem by relaxing the principle of truth-functionality. Namely, in Nmatrices, the truth-value of a complex formula can be chosen non-deterministically out of some non-empty set of options. Thereby, non-deterministic semantics is non-truth-functional, as opposed to deterministic semantics. However, as shown by Avron and co-authors, the semantics of Nmatrices shares with the semantics of ordinary (deterministic) matrices important properties such as compactness, decidability (in the finite case) and, more

[6]It is well know that Quine changed his mind considerably on the *analytic-synthetic distinction* along his career. On this point and for a philosophical criticism of Quine's proposal in [129], see [92].

importantly, analyticity.

2.2.2 Basic notions and properties

In this subsection we briefly recall formal definitions of some basic notions of non-deterministic semantics as stated by Avron and co-authors.

Definition 2.2.1. A *non-deterministic matrix* (*Nmatrix*) for $\mathcal{L}$ is a triple $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where:

- $\mathcal{V}$ is a non-empty set of truth-values;
- $\mathcal{D}$ is a non-empty proper subset of $\mathcal{V}$ (whose elements are called the *designated* elements of $\mathcal{V}$);
- $\mathcal{O}$ is a function that associates an n-ary function $\widetilde{\diamond} : \mathcal{V}^n \longrightarrow 2^{\mathcal{V}} \setminus \{\emptyset\}$ with every n-ary connective $\diamond$ of $\mathcal{L}$.

We say that $\mathcal{M}$ is *(in)finite* if so is $\mathcal{V}$.

Definitions 2.2.2. Let $\mathcal{M}$ be an Nmatrix for $\mathcal{L}$.

- An n-ary connective $\diamond$ of $\mathcal{L}$ is *non-deterministic* in $\mathcal{M}$, if there are some $x_1, ..., x_n \in \mathcal{V}$, such that $\widetilde{\diamond}(x_1, ..., x_n)$ is not a singleton.
- $\mathcal{M}$ is a *proper* Nmatrix if at least one of the connectives of $\mathcal{L}$ is non-deterministic in $\mathcal{M}$. In turn, $\mathcal{M}$ is *strictly proper* if that connective has arity $n > 0$. $\mathcal{M}$ is *deterministic* if it is not proper.

Definitions 2.2.3. Let $\mathcal{M}$ be an Nmatrix for $\mathcal{L}$.

- A *partial $\mathcal{M}$-valuation* for $\mathcal{L}$ is a function $v : F(\mathcal{L})^* \longrightarrow \mathcal{V}$ for some $F(\mathcal{L})^* \subseteq F(\mathcal{L})$ satisfying the following conditions:
  - The set $F(\mathcal{L})^*$ is closed under subformulae; i.e., $\mathsf{sub}(F(\mathcal{L})^*) = F(\mathcal{L})^*$.
  - For each n-ary connective $\diamond$ of $\mathcal{L}$, the following holds for all $A_1, ..., A_n \in F(\mathcal{L})^*$:
  $$v(\diamond(A_1, ..., A_n)) \in \widetilde{\diamond}(v(A_1), ..., v(A_n)). \qquad (2.1)$$
- A partial $\mathcal{M}$-valuation is a (full) $\mathcal{M}$-*valuation* if its domain is $F(\mathcal{L})$.

Remark 1. Note that the only difference between Nmatrices and ordinary (deterministic) matrices is that in the case of the latter we have '=' rather than '∈' in condition (2.1). Indeed, when taking an Nmatrix as a generalization of a matrix, the latter is viewed as a special type of an Nmatrix in which each $\tilde{\diamond}$ always returns a singleton. In such a case, each $\tilde{\diamond}$ can be treated as a function $\tilde{\diamond} : \mathcal{V}^n \longrightarrow \mathcal{V}$. Thus, when there is no risk of confusion, we shall identify singletons of truth-values with the truth-values themselves.

Remark 2. As in many-valued deterministic semantics, in non-deterministic semantics, each formula has a defined truth-value. That is why $\emptyset$ is excluded from being a value of $\tilde{\diamond}$. However—as exemplified in the depth-bounded approach below—the absence of any defined truth-value for a formula can still be simulated within the non-deterministic formalism by introducing a special truth-value $\bot$ representing exactly this case.

Proposition 2.2.4 (Analyticity, [18]). *Let $\mathcal{M}$ be an Nmatrix for $\mathcal{L}$, and v be a* partial $\mathcal{M}$-valuation*. Then v can be extended to a (full) $\mathcal{M}$-valuation.*

Definitions 2.2.5. Let $\mathcal{M}$ be an Nmatrix for $\mathcal{L}$. The *Tcr induced by* $\mathcal{M}$, $\vDash_{\mathcal{M}}$, is defined by: $\Gamma \vDash_{\mathcal{M}} A$ if for every partial $\mathcal{M}$-valuation v, if ($v(B)$ is defined and) $v(B) \in \mathcal{D}$ for all $B \in \Gamma$, then ($v(A)$ is defined and) $v(A) \in \mathcal{D}$. We denote by $\mathbf{L}_{\mathcal{M}} = \langle \mathcal{L}, \vDash_{\mathcal{M}} \rangle$ the *Tarskian propositional logic induced by* $\mathcal{M}$.[7]

Theorem 2.2.6 (Compactness, [21]). *For every propositional language $\mathcal{L}$ and any finite Nmatrix $\mathcal{M}$ for $\mathcal{L}$, the Tcr induced by $\mathcal{M}$, $\vDash_{\mathcal{M}}$, is finitary.*

Definitions 2.2.7. Given a Tarskian propositional logic $\mathbf{L} = \langle \mathcal{L}, \mid\!\sim_{\mathbf{L}} \rangle$ and a Nmatrix $\mathcal{M}$ for $\mathcal{L}$, we say that:

- $\mathbf{L}$ is *sound* for $\mathcal{M}$ iff $\Gamma \mid\!\sim_{\mathbf{L}} A$ implies $\Gamma \vDash_{\mathcal{M}} A$;
- $\mathbf{L}$ is *complete* for $\mathcal{M}$ iff $\Gamma \vDash_{\mathcal{M}} A$ implies $\Gamma \mid\!\sim_{\mathbf{L}} A$;
- $\mathbf{L}$ is *characterized* by $\mathcal{M}$ iff $\mathbf{L}$ is both sound and complete for $\mathcal{M}$.

Theorem 2.2.8 (Decidability, see [19]). *Let Γ be a finite set of formulae, A a formula, and $\mathcal{M}$ a finite Nmatrix for $\mathcal{L}$. Then the question whether $\Gamma \vDash_{\mathcal{M}} A$ is decidable.*

[7]This and the following definition can be analogously and respectively formulated w.r.t. (multiple-conclusion) *Scott consequence relation* and *Scott propositional logic*. On this point see the references to Avron et al.

Remark 3. Because of analyticity $\Gamma \vDash_{\mathcal{M}} A$ is decidable whenever Γ and $\mathcal{M}$ are finite. Namely, when assesing whether or not $\Gamma \vDash_{\mathcal{M}} A$, analyticity allow the search to be restricted to partial $\mathcal{M}$-valuations whose domain is $\mathsf{sub}(\Gamma \cup \{A\})$. This, together with the finiteness of Γ and $\mathcal{M}$, assures that the search space is also finite and, thus, that the corresponding algorithm always terminates.

Notation 2.2.9. To simplify reading, in what follows we shall omit the prefix or subscript '$\mathcal{M}$' in the notions above.

2.3 Background on computational complexity[8]

2.3.1 Absolute and relative complexity

Computational Complexity Theory can be seen as a refinement of Computability Theory. While the latter theory studies which problems can or cannot be decided (or computed, or solved) *in principle*—meaning, within a finite, yet unbounded, amount of resources such as time and space—the former theory studies which problems can or cannot be decided within a *bounded* amount of resources. In Complexity Theory, there is a working assumption—known as the *Cobham–Edmonds thesis*—according to which the class of *tractable* (or feasible, or practically solvable) problems is identified with the complexity class P of the problems that are decidable by a conventional (deterministic) Turing machine within polynomial time; i.e., whitin a number of steps bounded above by some fixed polynomial in the length of the input.

Roughly, the justification underlying the Cobham–Edmonds thesis is that, as the length of the input grows, exponential time algorithms require resources that quickly go beyond any practical constraint. Needless to say, an exponential time algorithm may be preferable in practice to a polynomial time algorithm with running time, say, $(10n)^{1000}$. Nonetheless, the notion of polynomial time computability is theoretically robust and useful, mainly because it is invariant under any reasonable model of computation. In fact, this latter aspect is backed up by an analogue of the Church-Turing thesis in the context of complexity. Namely, the *Invariance Thesis*, saying that a Turing machine can simulate any "reasonable" model of computation with at most a polynomial increase in time and space. Moreover, polynomial time computability is invariant under any reasonable selection of "encoding scheme" for the problem at issue. Finally, there is increasing evidence—particularly coming from the theory of NP-completeness—according to which, problems encountered in practice that are not purposely constructed to defy the power of our computational devices, tend to be

[8]This section is based on Chapter 4 of [53].

either intractable or decidable within time bounded by a polynomial of reasonably low degree.

Now, more specifically, the elements of complexity classes—such as P—are languages. The vast majority of computational problems can be taken as language-recognition problems. Namely, problems asking whether or not a word over a given alphabet belongs to some distinguished set of words. An important example taken from Logic consists in that the problem of determining whether a propositional formula is a tautology (i.e., valid) in **CPL** can be identified with the set TAUT of all the words over the alphabet of propositional calculus which express classical tautologies. So, an algorithm solving that problem is one deciding for every given word over the alphabet, whether or not it belongs to TAUT. In this sense, P can be defined as the class of the languages which can be recognized within polynomial time by some (deterministic) algorithm. Now, when non-deterministic models of computation are considered, the analogue of P is the class NP. The notation 'NP' stands for "non-deterministic polynomial time", since such a class was originally defined as consisting of the problems decidable within polynomial time by some non-deterministic algorithm. However, nowadays it is customary to (equivalently) define NP as the class of all languages L such that, for every word $w \in L$, there is a proof of its membership in L which is bounded above by some polynomial function of the length of w [see 48].

As it is well known, the main role played by propositional logic in Complexity Theory lies on the following results due to Cook and Reckhow:

Theorem 2.3.1 ([47]). *There is a deterministic polynomial time algorithm for the classical tautology problem iff P = NP.*

Theorem 2.3.2 ([49]). *There is a non-deterministic polynomial time algorithm for the classical tautology problem iff NP is closed under complementation; i.e., iff P = co-NP.*

Regarding the first result, most researchers in Complexity Theory conjecture that P $\neq$ NP. That conjecture implies that no *proof procedure* can be uniformly tractable for the whole class of classical tautologies—although, a proof procedure may be tractable for certain infinite subclasses of such a class, of course. As for the second result, it involves the notion of *proof system* rather than that of proof procedure. The following definitions are adapted from Cook and Rechow's [49, 50]: let Σ be a finite alphabet. With Σ^* we denote the set of all finite strings or "words" over the alphabet Σ. A *language* L is defined as a subset of Σ^*; i.e., a set of strings over a fixed alphabet Σ. The length of a string x is denoted as $|x|$.

Definition 2.3.3. If Σ_1 and Σ_2 are finite alphabets, a function $f : \Sigma_1^* \longrightarrow \Sigma_2^*$ is in $\mathcal{F}$ if it can be computed by a deterministic Turing machine in time bounded by a polynomial in the length of the input.

The class $\mathcal{F}$ of functions computable in polynomial time allow us to make precise the vague notion of "feasibly computable function".

Definition 2.3.4. If $L \subseteq \Sigma^*$, a *proof system* for L is a function $f : \Sigma_1^* \longrightarrow L$ for some alphabet Σ_1, where $f \in \mathcal{F}$ and f is onto.

That $f \in \mathcal{F}$ ensures that, when given an alleged proof—i.e., a string over Σ_1—there is a tractable method of checking whether or not it really is a proof, and if so, of what it is a proof. For instance, a proof system S is associated with a function f such that $f(x) = A$ whenever x is a string of symbols standing for a legitimate proof of A in S. When x does not stand for a proof in S, then $f(x)$ is taken to denote some fixed tautology in L; say, in the case of **CPL**, $p \vee \neg p$.

Definition 2.3.5. A proof system f is *polynomially bounded* if there is a polynomial $p(n)$ such that for all $y \in L$, there is an $x \in \Sigma_1^*$ such that $y = f(x)$ and $|x| \leq p(|y|)$.

In the above definition, $f(x) = y$ is to hold if x is a proof of y. So, the definition characterizes a proof system such that, for every element of L, there is a "short"—i.e., polynomially bounded—proof of its membership in L. On the one hand, that a proof system is polynomially bounded does not imply—unless P=NP—that there is a proof procedure based on it—namely, a deterministic version—that is also so. Still, that a proof system is not polynomially bounded does imply that there is no polynomially bounded proof procedure based on it.

A key observation about polynomially bounded proof systems, due to Cook and Reckhow, is as follows:

Theorem 2.3.6 ([49]). *There is a polynomially bounded proof system for the classical tautology problem iff NP = co-NP.*

The question of whether a proof system is polynomially bounded or not concerns its *absolute* complexity. It has been shown that, for example, most conventional proof systems for **CPL**—such as analytic tableaux (and so Gentzen-style cut-free systems in tree form) and resolution—are not polynomially bounded, by presenting for each system some infinite class of *hard examples* that have no polynomial-size proofs.[9] Those results have the following crucial consequence for the use of proof

[9] More specifically, they are called "hard" in the sense that any S-proof showing that $x \in L$ must be infeasibly long relative to the size of x.

systems for automated deduction: a complete proof system should not be expected to be tractable. Rather, either completeness is given up for attaining tractability—by, e.g., restricting the language—or heuristics guiding the proofs are adopted. In fact, the results regarding most conventional proof systems imply that, when we are seeking for proofs expressible as formal derivations in some of those systems, the option of heuristics alone is not sufficient. Thus, we should do both, give up completeness and be prepared to use heuristics.

Now, although the existence of a polynomially bounded proof system for **CPL** is regarded as highly improbable—since it is strongly suspected that NP $\neq$ co-NP—the importance of the complexity analysis of proof systems is by no means restricted to questions of absolute complexity. There are many interesting questions regarding their *relative* complexity, which are computationally significant even when the systems have been proven intractable. Differences between proof systems do not boil down to those caused by applied heuristics, but crucially include questions of relative efficiency between them. In fact, as far as automated deduction is concerned, when choosing an appropriate system to start with, considerations of relative complexity are prior to any heuristic one. For instance, considerations of relative efficiency are important when intractable proof systems for **CPL** considerably differ with respect to the extension and type of the subsets of TAUT for which they are polynomially bounded.

2.3.2 Relative complexity and simulations

Although not directly related to problems such as NP$\stackrel{?}{=}$co-NP, questions of relative complexity of proof systems have some relevance to the more practical problems of investigating mechanized proof-search strategies and constructing efficient automatic theorem provers. In fact, solving questions such as NP$\stackrel{?}{=}$co-NP seems still distant. By contrast, significant progress has been made in classifying the relative complexity of well known proof systems, as well as in proving lower bounds for restricted systems. Now, when studying the relative complexity of proof systems, some basic qualitative notions require to be given quantitative versions. Let S be a proof system for a propositional logic. With

$$\Gamma \stackrel{n}{\vdash}_S A$$

we denote that there is a proof π of A from Γ in system S such that $|\pi| \leq n$; where, as usual, $|\pi|$ denotes the *length* of π intended as a string of symbols over the alphabet of S.

Now, suppose that, given two systems S and S', there is a function g such that for all Γ, A,

$$\Gamma \overset{n}{\vdash}_{S'} A \text{ implies } \Gamma \overset{g(n)}{\vdash}_{S} A. \tag{2.2}$$

In relative complexity, we are interested in the rate of growth of g for particular systems S and S'. Then, positive results concerning the relation in (2.2) are usually obtained via *simulation procedures*:

Definition 2.3.7. If $f_1 : \Sigma_1^* \longrightarrow L$ and $f_2 : \Sigma_2^* \longrightarrow L$ are proof systems for L, a *simulation* of f_1 in f_2 is a computable function $h : \Sigma_1^* \longrightarrow \Sigma_2^*$ such that $f_2(h(x)) = f_1(x)$ for all $x \in \Sigma_1^*$.[10]

In turn, negative results concerning (2.2) consist of *lower bounds* for the function g.

Now, an important case of the relation in (2.2) occurs when $g(n)$ is a polynomial in n, i.e., $g \in \mathcal{F}$. This can be shown by giving a simulation function h such that for some polynomial $p(n)$, $|h(x)| \leq p(|x|)$ for all x. Thus, in such a case, we say that S *polynomially simulates* or *p-simulates* S'. The simulation h is then a function that translates proofs in S' into proofs in S, and preserves tractability. More specifically, if follows from the definitions that:

Proposition 2.3.8. *If a proof system S' for L p-simulates a polynomially bounded proof system S for L, then S' is also polynomially bounded.*

It is easy to see, given that $\mathcal{F}$ is closed under composition, that the p-simulation relation is reflexive and transitive (i.e., a preorder), and so its symmetric closure is an equivalence relation. Thereby, proof systems can be ordered by means of p-simulation and been put into equivalence classes with respect to their relative complexity. In turn, systems which belong to the same equivalence class can be considered as having "essentially"—meaning, up to a polynomial—the same complexity or efficiency. In contrast, if S p-simulates S' but not the other way round, we can say that—as far as the length of proofs is concerned—S is essentially more efficient or powerful than S'. For example, S may be polynomially bounded for every $L \subset$ TAUT for which S' is polynomially bounded but not viceversa; in such a case, S has a larger "practical" scope than S'.

Cook and Reckhow [49, 50] started the study of the relative complexity of proof systems. Since then, some open problems have been solved while some others have been raised [see 149]. Results in the Cook's and Reckhow's tradition concern primarily questions about proof-length; so, they are not directly concerned with the very

[10] We assume that the language of both proof systems is the same. For an even more general definition not containing this assumption see [131].

important question—at least from the practical point of view—of the complexity of *proof-search*. Put differently, results stated in terms of the p-simulation relation concern the relative length of *minimal* proofs in different systems. So, those results are scarcely informative about the relative difficulty of proof-search. In a nutshell, "easy" proofs may be rather hard to find! In that sense, considerations regarding the size of the search space in which such "easy" proofs are to be found turn out to be crucial. On the one hand, sometimes it is possible to define systematic procedures to efficiently explore the corresponding search space. In such a case, a speed-up in proof-length can imply a similar speed-up in proof-search. On the other hand, such systematic procedure may not be available to us and the existence of shorter proofs may not helpful for designing more efficient proof procedures. Thereby, the importance of those results for automated deduction must be assessed case by case.

2.4 The approximation problem

As mentioned in Section 1.2, most interesting and useful logics are either undecidable or likely to be intractable. So, only highly idealized agents would be always able to recognize in practice that a certain conclusion follows from a given set of assumptions. Real resource-bounded agents, in contrast, cannot be expected to be informed of all conclusions potentially obtainable from the information they explicitly possess. Nonetheless, idealization may well be a matter of degree; that is, the capability of correctly recognizing validity or inconsistency may well vary from one agent to another. For instance, any human agent who understands the (e.g., classical) meaning of '$\rightarrow$' can recognize the validity of *modus ponens*; however, fewer agents are able to correctly make inferences involving complex case reasoning and very few are able to prove theorems from a mathematical theory.

As other types of formal modelling, a logic generally involves a big deal of idealization and, as such, is not intended to faithfully describe the actual inferential behavior of rational agents. In words of Gabbay and Woods:

> A logic is an idealization of certain sorts of real-life phenomena. By their very natures, idealizations misdescribe the behaviour of actual agents. This is to be tolerated when two conditions are met. One is that the actual behaviour of actual agents can defensibly be made out to *approximate* to the behaviour of the ideal agents of the logician's idealization. The other is the idealization's *facilitation* of the logician's discovery and demonstration of deep laws [87, p. 58].

This leads to what D'Agostino et al. [e.g., 59, 56, 87, 86] have called the *approximation problem* which, in the context of logical systems, can be concisely stated as follows:

> *Approximation problem*: Can we define *in a natural way* a hierarchy of logical systems that indefinitely approximate a given ideal [l]ogic in such a way that these approximations provide useful formal models of the logical competence of different resource-bounded agents? [56, p. 5].

As D'Agostino also points out, robust solutions to this problem are likely to have a significant practical impact in all research areas—such as Philosophy, AI, Cognitive science and Economics—that are in need of less idealized, yet theoretically principled, models of reasoners with bounded cognitive and computational resources. However, those solutions require "an imaginative re-examination of logical systems as they are usually presented in the literature" [59, p. 44]. This given that, for instance, standard proof-theoretic formalisms—such as natural deduction, tableaux, and sequent calculi—are structurally inadequate to define in a natural way a measure of the difficulty of inferences. For instance, one crucial requirement that a formalism defining such a measure should satisfy is that the meaning of the logical operators—as given by the operational rules—remains the same throughout the hierarchy of approximations to the "goal" logic at issue. Thereby, an agent may still be credited with understanding the meaning of a finite set of sentences—say, the axioms of a theory—even when she is unable to draw all the conclusions which are in principle obtainable from it.[11]

Despite its practical and theoretical importance, the approximation problem has been scarcely addressed in the logical and philosophical literature. However, approximations to (full, or fragments of) **CPL** via tractable subsystems of increasing inferential power has received some attention in Computer Science and AI [e.g., 41, 134, 114, 51, 66, 135, 75, 77, 78, 76, 109]. While these contributions are closely related to each other, they have been given in a rather scattered and differently motivated way. Logic still lacks a robust and general foundations for an approximation theory. D'Agostino's et al. depth-bounded approach constitutes a step towards such a theory, by trying to provide a single proof-theoretical and semantic framework for some of the main ideas of those contributions.

As recalled in the next Section, the approach has already been shown apt as to provide an adequate solution to the approximation problem in the case of **CPL**

[11] The kind of approximation problem stated above concerns computational idealizations typically made by logical models. Needless to say, these models involve other kinds of idealizations which, in turn, give rise to other kinds of approximation problems.

and, simultaneously, as to provide means to solve the well known anomalies that the corresponding orthodox semantics and proof-theory have raised (again, logical omniscience, BHC paradox and the scandal of deduction). More specifically, the approach leads to defining, in a natural way, a hierarchy of tractable depth-bounded consequence relations that infinitely approximate **CPL**; hierarchy that appears to be a plausible model for representing rational agents with increasing, albeit bounded, cognitive and computational resources. Accordingly, it allow us to define degrees of logical omniscience, which may be naturally related to increasingly idealized agents, in terms of correspondingly stronger consequence relations. Thus, the approach allows us to characterize in a natural and uniform way such a hierarchy of approximations to the "perfect reasoner"—which hitherto has been restricted to be the classical one—and so provides the flexibility required by a suitable model of practical rationality.

Although there is nothing apparent that prevent us to extend the depth-bounded approach—or other approaches based on approximations—to finitary *non*-Tarskian logics (and even to broader phenomena studied in logical dynamics), the thesis will be restricted to finitary Tarskian propositional logics, leaving the rest for future research. Thus, we formally define the notion of approximation with respect to that kind of logics:

Definition 2.4.1. Let **L** be a finitary Tarskian propositional logic $\langle \mathcal{L}, \mid\!\sim_{\mathbf{L}} \rangle$. An *approximation system* for **L** is a triple $\mathscr{A} = \langle P, \preceq, \{R_\alpha\}_{\alpha \in P} \rangle$, where $\langle P, \preceq \rangle$ is a directed set, called the *parameter set*, and $\{R_\alpha\}_{\alpha \in P}$ is a family of consequence relations on $\mathcal{L}$ such that:

- $\alpha \prec \beta$ implies $R_\alpha \subset R_\beta$;
- for each $\alpha \in P$, R_α is decidable in polynomial time;
- $\bigcup_{\alpha \in P} R_\alpha = \mid\!\sim_{\mathbf{L}}$

We shall call **L** the *limiting logic* of the approximation system $\mathscr{A}$. Each relation R_α is an *approximation* to **L**.

Naturally, approximation systems are of practical and theoretical interest whenever the limiting logic is known or conjectured to be intractable.

Chapter 3

Tractable depth-bounded approximations to CPL

3.1 Introduction

The depth-bounded approach was first put forward by D'Agostino and co-authors as an "informational view" of **CPL**, accounting for how this logic can be approximated in practice by resource-bounded agents. As mentioned in Chapter 1, a key idea underlying the approach when applied to **CPL** is that the meaning of a logical operator is specified solely in terms of the information that is *actually* possessed by the agent under consideration. In turn, that an agent actually possesses information means that this is information *practically* accessible to her and with which she *can* operate—e.g., in decision making. We call this kind of information *actual*, as opposed to *potential* information that is available to the agent only in principle.[1] Besides, we use the verb "to *hold*" as synonymous with "to actually possess".

As already briefly discussed in Chapter 1, the approach leads to an infinite hierarchy of tractable depth-bounded approximations to **CPL**, which can be naturally related to the inferential power of the agents. This hierarchy admits of an intuitive 3-valued non-deterministic semantics and an elegant proof-theoretic characterization.

[1] This distinction between actual and potential information is already present in the work of Hintikka. As well-known, he related what he labelled the "scandal of deduction" to the undecidability of first-order logic, which implies some inescapable uncertainty about the validity of inferences involving quantifiers. According to him, if we take seriously the "old important idea" that information consists in reducing uncertainty, then "[r]elief from this sort of uncertainty ought to be reflected by any realistic measure of the information which we actually possess (as distinguished from the information we in some sense have potentially available to us) and with which we can in fact operate" [99, p. 229].

The latter is half-way between a *classical* version of natural deduction—which mirrors the classical meaning of the connectives and not their intuitionistic meaning as Gentzen's original rules—and the method of semantic tableaux. Concretely, as mentioned in Chapter 1, it results from combining the classical proof systems *KE* and *KI*. Here we refer to the system obtained by such a combination as *intelim method*. It is this proof-system which allows us to measure the *depth* of inferences in terms of the maximum number of nested applications of a single structural rule that are allowed. This rule—called *PB* as it expresses the Principle of Bivalence—is essentially a (non-eliminable) cut rule, that governs the manipulation of *virtual* information; i.e., information that an agent does not hold, but she temporarily *assumes as if* she held it. Intuitively, the more virtual information needs to be invoked via *PB*, the harder the inference is for the agent. In this sense, the nested applications of that rule provide a sensible measure of inferential depth. This naturally leads to defining an infinite hierarchy of tractable approximations to **CPL**, where the inferential power of agents is naturally bounded by their limited capability of manipulating virtual information.

In what follows we first recall the proof-theoretic basis for applying the depth-bounded approach to **CPL**, and then we recall the application itself together with the 3-valued non-deterministic semantics associated to it. We have three reasons for such an order of presentation: (i) The proof-theoretic basis is interesting in its own right since it is more efficient—and, presumably, more natural—than other proof system for **CPL** such as analytic tableaux and Gentzen's cut-free sequent calculus. (ii) The characterization of the hierarchy of approximations in terms of the 3-valued non-deterministic semantics seems to suggest an exponential blow up. Indeed, the tractability of the approximations yielded by the approach was proven via their proof-theoretic characterization. (iii) That order of presentation matches the historical development of the approach to **CPL**.

3.2 The proof-theoretic basis

The proof systems *KE* and *KI* were introduced by Mondadori [117, 118, 119]. Later on, D'Agostino and Mondadodri [53, 65] investigated their relative complexity, showing that those systems linearly simulate each other and that both dominate clausal analytic tableaux in terms of the p-simulation relation. Specifically, they showed that even the analytic restriction of *KE* can p-simulate clausal analytic tableaux but not vice versa. Moreover, D'Agostino [54] showed that clausal analytic tableaux cannot even p-simulate the truth-tables method. As we shall explain below, *KE* is a refutation system which is a variant of analytic tableaux, but essentially more efficient.[2] Regarding *KI*, it is a direct-proof system which can be regarded as a proof-theoretic version of the truth-tables method, but essentially more efficient. Namely, *KI* can *p*-simulate the truth-tables method [see 53] but the truth-tables method cannot *p*-simulate *KI* [120]. In turn, combining *KE* and *KI* yields the proof system that here we have dubbed *intelim method*. This latter proof system (i) is half-way between a classical version of natural deduction and analytic tableaux; (ii) constitutes the proof-theoretic basis of the depth bounded approach to **CPL**; (iii) enjoys the computational advantages of its subsystems *KE* and *KI*.[3]

3.2.1 *KE*

KE has an exponential speed-up on both analytic tableaux and Gentzen's cut-free sequent calculus, even if *KE*'s *analytic restriction*—which yields only refutations with the *subformula property*—is considered. Moreover, *KE* was proposed not only as a more efficient alternative to those systems, but also as a more natural one. Namely, unlike those proof systems and via a cut rule, *KE* can represent the use of auxiliary lemmas in proofs (which are an inherent part of the mathematical activity), and properly express the bivalence of classical logic. Below we just recall briefly *KE* and its computational efficiency advantages, but these advantages are—as might be expected—closely related to its being more natural in the above sense. So, the latter is left implicit in our brief presentation.[4]

In what follows we shall use *signed formulae* (S-formulae, for short); namely, expressions of the form $\mathsf{T}\,A$ or $\mathsf{F}\,A$, where A is a formula. As recalled below, *KE* (as well as *KI* and the intelim method) can indifferently be presented in terms of S-

[2]The relative complexity results in [53, 54, 65] were originally stated disregarding distinct types of analytic tableaux. It was later on that Massacci [115, 116] gave the unexpected result that *binary* tableaux can present an exponential speed-up over *clausal* ones. We shall recall the difference between those types of tableaux, as well as Massacci's result, in Subsection 3.3.2. More importantly, in the same Subsection, we shall show that *KE* (and so both *KI* and the intelim method) dominates also binary tableaux. In fact, we shall show that *KE* dominates the strongest possible version of clausal tableaux called *unrestricted* tableaux and, somewhat surprisingly, that the latter are dominated even by the truth-tables method.

[3]Further, the normalization of the intelim method [55, 58] implies that such a method still has an exponential speed-up on analytic tableaux and on cut-free sequent proofs even when only normal intelim proofs and refutations are considered.

[4]For a detailed account of why *KE* should be regarded as a more natural proof-theoretic characterization of **CPL** see [53, 65]. See also [55, 63] for an analogous observation when comparing a "fully fledged" natural deduction version of *KE* with Gentzen-Prawitz style natural deduction systems for **CPL**.

formulae or usual formulae. However, we shall recall and consider the signed version because it relates more directly to the *KE*/*KI*-style proof systems for non-classical logics we shall introduce and use in Part II of this Thesis. Whereas in the case of **CPL** the use of S-formulae in *KE*/*KI* systems is optional, in the case of non-classical logics it is indispensable. We shall use $\varphi, \psi, \theta, ...$, as variables ranging over S-formulae, and $X, Y, Z, ...$, as variables ranging over sets of S-formulae. Besides, we continue using $A, B, C, ...$, as variables ranging over (unsigned) formulae, and Γ, Δ, Λ,..., as variables ranging over sets of formulae. So, we shall write $\mathsf{T}\,\Gamma$ to denote $\{\mathsf{T}\,A \mid A \in \Gamma\}$. Besides, let us use S as a variable ranging over $\{\mathsf{T}, \mathsf{F}\}$. In turn, we say that the *unsigned part* of an S-formula is the unsigned formula that results from it by removing its sign. Given an S-formula φ, we denote by φ^u the unsigned part of φ and by X^u the set $\{\varphi^u | \varphi \in X\}$. In the context of **CPL** intuitively, $\mathsf{T}\,A$ means "A is true" and $\mathsf{F}\,A$ means "A is false". Formally:

Definition 3.2.1. A classical (i.e., Boolean) valuation v *satisfies* an S-formula $\mathsf{T}\,A$ if $v(A) = \mathit{true}$ and an S-formula $\mathsf{F}\,A$ if $v(A) = \mathit{false}$. A set X of S-formulae is *satisfiable* if there is a classical valuation v which satisfies all its elements.

Thus, the truth-value of $\mathsf{T}\,A$ is the same as that of A, whereas the truth-value of $\mathsf{F}\,A$ is the same as that of $\neg A$. In turn, we say that the *conjugate* of $\mathsf{T}\,A$ is $\mathsf{F}\,A$ and vice versa.

Now, the hallmark of *KE* is the reduction of the amount of branching to a minimum by making all branches mutually exclusive. Accordingly, *KE* has only one *branching* rule expressing the classical Principle of Bivalence, and the rest of its rules have all a *linear* format. Indeed, *KE* is more efficient than other proof systems for **CPL** because the application of its unique branching rule allows us to avoid many redundant branchings in the corresponding trees. Specifically, *KE* includes the set of linear elimination rules displayed in Table 3.1. In these rules, we shall refer to the premise containing the connective that is to be eliminated as *major* and to the other premise as *minor*. This set of rules is not complete for **CPL**. Completeness is achieved by adding only the following branching rule, called *PB* given that it expresses the Principle of Bivalence:

$$\frac{}{\mathsf{T}\,A \mid \mathsf{F}\,A}$$

So, *PB* allows us to append both $\mathsf{T}\,A$ and $\mathsf{F}\,A$ as sibling nodes at the end of any branch of the tree, generating two new mutually exclusive branches. Essentially, *PB* is a classical *cut* rule which is *not eliminable*, but whose use—as recalled below—can be restricted so as to satisfy the analytic cut property (i.e., subformula property).

Definitions 3.2.2.

$$\begin{array}{c} \mathsf{T}\,A \vee B \\ \mathsf{F}\,A \\ \hline \mathsf{T}\,B \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \vee B \\ \mathsf{F}\,B \\ \hline \mathsf{T}\,A \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \vee B \\ \hline \mathsf{F}\,A \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \vee B \\ \hline \mathsf{F}\,B \end{array}$$

$$\begin{array}{c} \mathsf{F}\,A \wedge B \\ \mathsf{T}\,A \\ \hline \mathsf{F}\,B \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \wedge B \\ \mathsf{T}\,B \\ \hline \mathsf{F}\,A \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \wedge B \\ \hline \mathsf{T}\,A \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \wedge B \\ \hline \mathsf{T}\,B \end{array}$$

$$\begin{array}{c} \mathsf{T}\,A \rightarrow B \\ \mathsf{T}\,A \\ \hline \mathsf{T}\,B \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \rightarrow B \\ \mathsf{F}\,B \\ \hline \mathsf{F}\,A \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \rightarrow B \\ \hline \mathsf{T}\,A \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \rightarrow B \\ \hline \mathsf{F}\,B \end{array}$$

$$\begin{array}{c} \mathsf{T}\,\neg A \\ \hline \mathsf{F}\,A \end{array} \qquad \begin{array}{c} \mathsf{F}\,\neg A \\ \hline \mathsf{T}\,A \end{array}$$

Table 3.1: Elimination rules for the standard **CPL** connectives

- Let $X = \{\varphi_1, ..., \varphi_m\}$. Then $\mathcal{T}$ is a *KE-tree* for X if there exists a finite sequence $(\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n)$ such that $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $(\varphi_1, ..., \varphi_m)$, $\mathcal{T}_n = \mathcal{T}$, and for each $i < n$, $\mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of an elimination rule to preceding S-formulae in the same branch, or by an application of *PB*.
- A branch of a *KE*-tree is *closed* if it contains an S-formula and its conjugate; otherwise, it is *open*.
- A *KE*-tree is *closed* when all its branches are closed; otherwise, it is *open*.
- A *KE*-tree is a *KE-refutation* of X if $\mathcal{T}$ is a closed *KE-tree* for X.
- A *KE*-tree $\mathcal{T}$ is a *KE-proof* of A from Γ if $\mathcal{T}$ is a *KE*-refutation of $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$.
- A is *KE-provable* from Γ if there is *KE-proof* of A from Γ.

A version of *KE* for unsigned formulae is simply obtained by replacing each S-formula $\mathsf{T}\,A$ with A and each S-formula $\mathsf{F}\,A$ with $\neg A$, and modifying all definitions in the obvious way. The soundness proof of *KE* is essentially the same as that for Smullyan's tableaux [138, p. 25]. Since we shall often refer to these tableaux throughout the Thesis, we recall their rules in Table 3.2. We present the signed version for

$$\frac{\mathsf{T}\,A \vee B}{\mathsf{T}\,A \mid \mathsf{T}\,B} \qquad \frac{\mathsf{F}\,A \vee B}{\begin{array}{c}\mathsf{F}\,A\\ \mathsf{F}\,B\end{array}} \qquad \frac{\mathsf{T}\,A \wedge B}{\begin{array}{c}\mathsf{T}\,A\\ \mathsf{T}\,B\end{array}} \qquad \frac{\mathsf{F}\,A \wedge B}{\mathsf{F}\,A \mid \mathsf{F}\,B}$$

$$\frac{\mathsf{T}\,A \to B}{\mathsf{F}\,A \mid \mathsf{T}\,B} \qquad \frac{\mathsf{F}\,A \to B}{\begin{array}{c}\mathsf{T}\,A\\ \mathsf{F}\,B\end{array}} \qquad \frac{\mathsf{T}\,\neg A}{\mathsf{F}\,A} \qquad \frac{\mathsf{F}\,\neg A}{\mathsf{T}\,A}$$

Table 3.2: Smullyan's tableaux rules

easier comparison but, analogously, an unsigned version obtains by replacing each S-formula $\mathsf{T}\,A$ with A and each S-formula $\mathsf{F}\,A$ with $\neg A$.[5]

Proposition 3.2.3 (Soundness). *If there is a closed KE-tree for X, then X is unsatisfiable.*

At some parts of the Thesis, we shall use Smullyan's unifying notation [138] to reduce the number of cases. This notation is summarized in the tables displayed in Table 3.3. So, the elimination rules of *KE* can be "packed" into the following four types of rules (where β_i', $i = 1, 2$ denotes the conjugate of β_i):

$$\text{Rule A1} \quad \frac{\alpha}{\alpha_1} \qquad\qquad \text{Rule A2} \quad \frac{\alpha}{\alpha_2}$$

$$\text{Rule B1} \quad \frac{\begin{array}{c}\beta\\ \beta_1'\end{array}}{\beta_2} \qquad\qquad \text{Rule B2} \quad \frac{\begin{array}{c}\beta\\ \beta_2'\end{array}}{\beta_1}$$

Under this notation, we say that a branch b of a *KE*-tree is *E-complete* if (i) for every S-formula of type α occurring in b, both α_1 and α_2 occur in b; and (ii) for every S-formula of type β occurring in b: (iia) if β_1' occurs in b, then β_2 occurs in b; (iib) if β_2' occurs in b, then β_1 occurs in b. In turn, we say that b is *β-complete* if for every S-formula of type β occurring in b and some $i = 1, 2$, either β_i or β_i' occurs in b. So, b is *complete* if it is E-complete and β-complete. Finally, a *KE*-tree $\mathcal{T}$ is *completed* if every branch of $\mathcal{T}$ is either closed or complete.[6]

[5]Moreover, signed versions of tableaux relate more directly to sequent calculi in that T-signed formulae play the role of left-side formulae with respect to a sequent arrow, while F-signed formulae play the role of right-side ones.

[6]Recall that the notion of Smullyan's *completed tableau* is defined analogously in [138].

α	α_1	α_2
$\mathsf{T}\, A \wedge B$	$\mathsf{T}\, A$	$\mathsf{T}\, B$
$\mathsf{F}\, A \vee B$	$\mathsf{F}\, A$	$\mathsf{F}\, B$
$\mathsf{F}\, A \to B$	$\mathsf{T}\, A$	$\mathsf{F}\, B$
$\mathsf{T}\, \neg A$	$\mathsf{F}\, A$	$\mathsf{F}\, A$
$\mathsf{F}\, \neg A$	$\mathsf{T}\, A$	$\mathsf{T}\, A$

β	β_1	β_2
$\mathsf{F}\, A \wedge B$	$\mathsf{F}\, A$	$\mathsf{F}\, B$
$\mathsf{T}\, A \vee B$	$\mathsf{T}\, A$	$\mathsf{T}\, B$
$\mathsf{T}\, A \to B$	$\mathsf{F}\, A$	$\mathsf{T}\, B$

Table 3.3: Smullyan's unifying notation

Now, the completeness of *KE* can be shown in several ways. However, proving it by modifying the traditional completeness proof for the tableau method [see 138] allows also to show that *KE* remains complete even when its applications of *PB* are restricted to be *analytic* ones. Namely, a *KE*-tree $\mathcal{T}$ for X is said to be *analytic* if *PB* is applied in $\mathcal{T}$ only to (proper) subformulae of formulae in X^u.[7] So, the *analytic restriction of KE* is the system in which the applications of *PB* are restricted to subformulae of the formulae occurring above in the same branch. A proof modifying the traditional completeness proof for tableaux is given in [53, 65]:[8]

Theorem 3.2.4 (Completeness, [53, 65]). *If $\Gamma \vDash_{\boldsymbol{CPL}} A$, then there is a closed KE-tree for $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\, A\}$.*

Such a proof yields a *subformula principle* in the following form:

Corollary 3.2.5 (Analytic cut property). *If X is unsatisfiable, then there is a closed KE-tree $\mathcal{T}'$ for X such that all the applications of PB are analytic (i.e., preserve the subformula property).*

The above corollary says that the analytic restriction of *KE* is complete. A constructive proof of this subformula principle, which yields a procedure for transforming any *KE*-proof in an equivalent *KE*-proof which enjoys the subformula property, is given in [118]. The subformula property is a key property of logical systems in that it allows us to search for refutations (or proofs) by *analytic* methods, i.e., by considering solely deduction steps involving formulae that are "contained" in the assumptions (or also the conclusion in the case of proofs). This implies a drastic reduction of the search space which is crucial for the purpose of automated deduction. When it comes

[7] The elimination rules are obviously analytic in that the conclusion of their application is always a signed subformula of the major premise.

[8] See [117] for a proof *à la* Kalmar.

$$\frac{\mathsf{T}\,A}{\mathsf{T}\,A \vee B} \qquad \frac{\mathsf{T}\,B}{\mathsf{T}\,A \vee B} \qquad \frac{\begin{array}{c}\mathsf{F}\,A\\ \mathsf{F}\,B\end{array}}{\mathsf{F}\,A \vee B}$$

$$\frac{\mathsf{F}\,A}{\mathsf{F}\,A \wedge B} \qquad \frac{\mathsf{F}\,B}{\mathsf{F}\,A \wedge B} \qquad \frac{\begin{array}{c}\mathsf{T}\,A\\ \mathsf{T}\,B\end{array}}{\mathsf{T}\,A \wedge B}$$

$$\frac{\mathsf{F}\,A}{\mathsf{T}\,A \to B} \qquad \frac{\mathsf{T}\,B}{\mathsf{T}\,A \to B} \qquad \frac{\begin{array}{c}\mathsf{T}\,A\\ \mathsf{F}\,B\end{array}}{\mathsf{F}\,A \to B}$$

$$\frac{\mathsf{T}\,A}{\mathsf{F}\,\neg A} \qquad \frac{\mathsf{F}\,A}{\mathsf{T}\,\neg A}$$

Table 3.4: Introduction rules for the standard **CPL** connectives

to propositional logics, this search space is finite for each putative inference, paving the way for decision procedures. In the case of *KE*, the subformula property assures that a bound can be imposed on the applications of *PB*—which could potentially be applied to arbitrary formulae—with no loss of deductive power.

3.2.2 *KI*

A direct-proof system for **CPL** can be obtained if instead of considering the analytic elimination rules of *KE*, we consider the *synthetic* introduction rules displayed in Tab. 3.4. Analogously to the analytic rules, the set of synthetic rules is not complete for **CPL**. Again, completeness is obtained by adding an unique branching rule: *PB*. The introduction rules together with *PB* constitute the system *KI* and yield definitions of *KI-tree*, as well as of closed and open branches and trees, which are analogous to those in Def. 3.2.2. In turn:

Definitions 3.2.6.

A *KI*-tree $\mathcal{T}$ is a *KI*-proof of A from Γ if $\mathcal{T}$ is a *KI*-tree for $\mathsf{T}\,\Gamma$ such that $\mathsf{T}\,A$ occurs in every open branch.

A is *KI-provable* from Γ if there is a *KI*-proof of A from Γ.

A version for unsigned formulae is obtained, as before, by changing all F's into $\neg$, and deleting all T's. The soundness of the rules constituting *KI* can be immediately verified. Besides, *KI* is a complete proof system for **CPL** and remains complete if the applications of the unique branching rule *PB* are restricted to atomic formulae occurring either in the assumptions or in the conclusion. Moreover, the applications of the introduction rules can be restricted, without loss of completeness, to subformulae of the assumptions or of the conclusion. For this completeness results and and an overview of *KI* see [53]. Thus, *KI* also enjoys the subformula property.

3.2.3 *KE*/*KI*: the intelim method

Combining *KE* and *KI* yields a system that is half-way between a classical version of natural deduction—which mirrors the classical meaning of the conectives and not their intuitionistic meaning as Gentzen's original rules—and the method of semantic tableaux [see 55, 58]. As we shall recall in the next section, that system constitutes the proof-theoretical basis for applying the depth-bounded approach to **CPL**. However, the system is interesting in its own right and so we recall it independently and before of such an application.

We shall refer to *KI*'s introduction rules together with *KE*'s elimination rules as *intelim* rules. This combination of rules is what makes that the system defined on their basis resembles natural deduction and—as shown below—can be used as a direct-proof method as well as a refutation method. Namely, the intelim rules generate *intelim sequences*; i.e., finite sequences $(\varphi_1, ..., \varphi_n)$ of S-formulae such that, for every $i = 0, ..., n$, either φ_i is an assumption or the conclusion of the application of an intelim rule to preceding S-formulae. Of course, intelim rules are not complete for **CPL**. Completeness is obtained by adding solely the *branching* rule common to *KE* and *KI*: *PB*. By adding *PB* to the stock of rules, proofs and refutations are represented by downward-growing trees—which brings the method somewhat closer to tableaux.

Definitions 3.2.7.

- Let $X = \{\varphi_1, ..., \varphi_m\}$. Then $\mathcal{T}$ is an *intelim tree for X* if there is a finite sequence $(\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n)$ such that $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $(\varphi_1, ..., \varphi_m)$, $\mathcal{T}_n = \mathcal{T}$, and for each $i < n$, $\mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of an intelim rule to preceding S-formulae in the same branch, or by an application of *PB*.
- A branch of an intelim tree is *closed* if it contains both $\mathsf{T}\,A$ and $\mathsf{F}\,A$ for some A; otherwise, it is *open*.

- Ani ntelim tree is *closed* if all its branches are closed; otherwise, it is *open*.
- An *intelim proof of* φ *from* X is an intelim tree $\mathcal{T}$ for X such that φ occurs in all open branches of $\mathcal{T}$.
- A is *intelim-provable* from Γ if there is an intelim-proof of $\mathsf{T}\,A$ from $\mathsf{T}\,\Gamma$.
- An *intelim refutation* of X is a closed intelim tree $\mathcal{T}$ for X.
- Γ is *intelim-refutable* if there is an intelim refutation of $\mathsf{T}\,\Gamma$.

Observe that, according to the above definitions, every refutation of X is, simultaneously, a proof of φ from X, for every φ.

Then, some features of the method of intelim trees are: (i) like in natural deduction, it has both introduction and elimination rules; (ii) intelim trees can be used as a direct proof method as well as a refutation method; (iii) all the operational rules are linear and there is only one branching rule corresponding to the Principle of Bivalence. In turn, the soundness and completeness of the method of intelim trees trivially follows from the soundness and completeness of its subsystems, *KE* and *KI*. Moreover:

Definition 3.2.8. We say that an intelim proof $\mathcal{T}$ of φ from X (an intelim refutation of X) has the *subformula property* (SFP) if, for every S-formula ψ occurring in $\mathcal{T}$, $\psi^u \in \mathsf{sub}(X^u \cup \{\varphi^u\})$ ($\psi^u \in \mathsf{sub}(X^u)$).

Given the properties of its subsystems *KE* and *KI*, it is far from surprising that the intelim method enjoys the SFP:

Proposition 3.2.9 ([55]). *Every intelim proof of* φ *from* X *(intelim refutation of* X*) can be transformed into an intelim proof of* φ *from* X *(an intelim refutation of* X*) with the SFP.*

As mentioned above, the SFP allows us to search for proofs or refutations by taking into account solely inference steps involving formulae that are "contained" in the assumptions—or also the conclusion in the case of proofs. Particularly, in the method of intelim trees, the SFP ensures that we can impose a bound on the applications of *PB*, which could potentially be applied to arbitrary formulae, without loss of completeness. Similarly, a bound can be imposed on sensible applications of introduction rules, which could potentially be applied *ad infinitum* producing ever more complex formulae. [See 55, 59, 58]. Apart from the SFP, normalization of proofs for the system of intelim trees is shown in [55, 58], and normalization of proofs of a ("fully fledged") natural deduction variant is shown in [63].

3.3 Relative complexity issues

In this Section we first recall that *KE*, *KI*, and the intelim method can linearly simulate each other. Then we recall some results, given by D'Agostino and Mondadori, comparing the efficiency of those systems with that of other proof systems for **CPL**. Most of these results are recalled with respect to *KE*, as they were originally stated. However, it is implied that analogous results hold for *KI* and the intelim method (given the linear simulation just mentioned). More importantly, we prove new lower bounds on analytic tableaux which strengthen D'Agostino's and Mondadori's previous results, and settle a problem left open in the literature.

3.3.1 Some previous results

Let us begin by recalling few working concepts. The *length* of a proof x, denoted by $|x|$, is the total number of symbols occurring in x (intended as a string). Now, the λ-*complexity* of x, $\lambda(x)$, is the *number of lines* in the proof x; each "line" being a formula, a sequent, or any other expression related to an inference step, depending on the system under consideration. Besides, the ρ-*complexity* of x, $\rho(x)$, is the length (correspondingly, total number of symbols) of a line of maximal length occurring in x. These complexity measures are connected by the relation $|x| \leq \lambda(x) \cdot \rho(x)$. The λ-measure suffices to settle negative results about the p-simulation relation, but it does not generally suffices for positive results. However, it may be adequate also for positive results whenever the $\rho(x)$-measure is not to significantly increased by the simulation procedure at issue. Since the results we recall below involve only procedures of that kind, in what follows we shall neglect the $\rho(x)$-measure and identify the complexity of a proof system with its $\lambda(x)$-measure.

The following was proven by D'Agostino:

Theorem 3.3.1 ([53]). *KE and KI can linearly simulate each other. Moreover, the simulation preserves the subformula property.*

Since the intelim method is constituted by *KE* and *KI*, and nothing else, the above immediately implies:

Corollary 3.3.2. *Both KE and KI can linearly simulate the intelim method and vice versa. Moreover, the simulation preserves the subformula property.*

As mentioned above, most of the results below are recalled in terms of *KE*, as they were originally stated. However, the reader should bear in mind that analogous results hold for *KI* and the intelim method, according to Theorem 3.3.1 and Corollary 3.3.2.

Now, the complexity of the truth-table method for a given formula A is often measured by the number of rows in the complete truth-table for that formula; i.e., 2^k, where k is the number of distinct atomic variables in A. However, a more accurate measure of the complexity of truth-tables takes into account also the length of the formula to be tested. More specifically, given a decidable logic $\mathbf{L}$ that can be characterized by means of m-valued truth-tables, the complexity of the decision procedure for $\mathbf{L}$ based on those tables is essentially $O(k \cdot n \cdot m^k)$; where n is the length of the input formula and k is the number of distinct variables in it. This is an upper bound which stems immediately from a semantic characterization of $\mathbf{L}$. In the case of **CPL**, where $m = 2$, this upper bound is essentially $O(k \cdot n \cdot 2^k)$. Since the truth-table procedure can plausibly be regarded as the most basic semantical and computational characterization of **CPL**, $O(k \cdot n \cdot 2^k)$ can also plausibly be taken as a natural upper bound on the classical tautology problem. So, let us say that a proof system for **CPL** is *standard* if its complexity is $O(n^c \cdot 2^k)$, where n is the length of the input formula, c is a fixed constant, and k the number of distinct variables occurring in the formula. Equivalently, we may say that a proof system is standard if it p-simulates the truth-tables method.

As we shall see below, analytic tableaux, and their equivalent cut-free sequent calculus in tree form, are not standard. By contrast:

Theorem 3.3.3 ([53, 65]). *The analytic restriction of KE is a standard proof system. In fact, for every tautology A of length n and containing k distinct variables, there is an analytic KE-refutation $\mathcal{T}$ of $\emptyset \cup \{\mathsf{F}\, A\}$ with $\lambda(\mathcal{T}) = O(\lambda(\mathcal{T})) = O(n \cdot 2^k)$.*

Where, of course, $\lambda(\mathcal{T})$ denotes the number of nodes in a *KE*-tree $\mathcal{T}$.

Now, *KE* is more efficient than analytic tableaux, even in the domain of analytic deduction. Namely:

Theorem 3.3.4 ([53, 65]). *The analytic restriction of KE p-simulates binary analytic tableaux. In fact, if there is a binary analytic tableaux proof $\mathcal{T}$ of A from Γ, then there is analytic KE-proof $\mathcal{T}'$ of A from Γ such that $\lambda(\mathcal{T}') \leq 2\lambda(\mathcal{T})$.*

Theorem 3.3.5 ([53, 65]). *Clausal analytic tableaux cannot p-simulate the analytic restriction of KE.*

As we shall recall in the next Subsection, the latter result was obtained by showing a class of examples which are "hard" for clausal analytic tableaux but "easy" for *KE*. In fact, the same class of examples is also easy for the truth-tables method and thus:

Theorem 3.3.6 ([54]). *Clausal analytic tableaux cannot p-simulate the truth-tables method (i.e., clausal analytic tableaux are not standard).*

In turn, another class of examples due to Cook and Reckhow [49]—also recalled in the next Subsection—are hard for the truth-tables method but easy for *KE*. Thereby:

Theorem 3.3.7 ([53, 65]). *The truth-table method cannot p-simulate KE.*[9]

Now, a couple of results comparing *KE* with refinements of analytic tableaux are:

Theorem 3.3.8 ([65]). *The analytic restriction of KE linearly simulates binary analytic tableaux with merging.*

Theorem 3.3.9 ([65]). *The analytic restriction of KE linearly simulates binary analytic tableaux with lemma generation.*

In turn, given the close relation between analytic tableaux and Gentzen's cut-free sequent calculus, is far from surprising that:

Theorem 3.3.10 ([65]). *KE can p-simulate the cut-free Gentzen sequent calculus, but not vice versa.*

Finally, *KE* can linearly simulate natural deduction (in tree form). Besides, the simulation procedure preserves the subformula property:

Theorem 3.3.11 ([53, 65]). *If there is a natural deduction proof $\mathcal{T}$ of A from Γ, then there is a KE-proof $\mathcal{T}'$ of A from Γ such that $\lambda(\mathcal{T}') \leq 3\lambda(\mathcal{T})$ and $\mathcal{T}'$ contains only formulae A such that A occurs in $\mathcal{T}$.*

3.3.2 Three forms of analytic tableaux

Relative complexity results are often obtained by showing a class of examples that are "hard" for one of the systems being compared but not for the other. As we shall recall below, that is the case of Theorems 3.3.5–3.3.7 above. In Subsection 2.3.2

[9]The analogous result for *KI* is straightforward in that *KI* can be viewed as a uniform improvement of the truth-table method. Namely, *KI* remains complete when the applications *PB* are restricted to atomic formulae. Thus, it is clear that, when testing a formula A for tautologyhood, *KI* can be used as a straightforward simulation in tree form of the method of truth-tables. However, if the atomic applications of *PB* are postponed until no further application of an introduction rule (over the set of subformulae of A) is possible, it may well be the case that we stop expanding a branch before *PB* has been applied to all the atomic formulae. This kind of procedure is equivalent to a "lazy evaluation" of Boolean formulae via partial truth-assignments.

we explained that the usual tool for comparing proof systems is the p-simulation relation. Informally, we say that two proof systems are equally powerful (from the complexity viewpoint) if they can p-simulate each other. Accordingly, we say that a proof system S' is more powerful than other proof system S whenever S' can p-simulate S but not vice versa; i.e., if we can map every S-proof of a formula A into an S'-proof of A by means of a polynomial function (in the size of the S-proof) but not vice versa.[10] Of course, if S' can p-simulate S, then for every formula A, the length of the shortest S'-proof of A must be bounded above by a polynomial function of the length of the shortest S-proof of A. Thus, whenever there is an infinite class H of formulae with S'-proofs of size $O(f(n))$, and it can be proven that the length of their shortest S-proofs cannot be bounded above by any polynomial function of $f(n)$, then it follows that S cannot p-simulate S'. In other words, if for a class of formulae H there is a superpolynomial lower bound (in the size of the formulae) on every S-proof, whereas there are "short" polynomial S'-proofs, then we can conclude that S' is more powerful than S. In such a case, we say that the formulae in H are "hard" for S but "easy" for S'. Therefore, that a system is more powerful than another is often proven by exhibiting a class of formulae that are hard for the latter but easy for the former.

In [49], Cook and Reckhow introduced a class of examples that they conjectured to be hard for analytic tableaux, but later proven by Massacci [115, 116] to be easy for them, provided *binary* and not *clausal* analytic tableaux are considered. More specifically, Cook and Reckhow conjectured those examples to have only exponential size analytic tableaux refutations, but Massacci showed them to have quasi-polynomial size binary analytic tableaux refutations. Below we shall briefly recall the examples as well as Massacci's result, but let us first recall the difference between clausal and binary analytic tableaux underlying such a result.

Clausal tableaux are a particular case of *n-ary* tableaux. In turn, the latter differ from binary tableaux in their branching rules. In order to recall the rules of n-ary tableaux, we first enrich the language to include expressions of the form $A_1 \vee ... \vee A_n$ and $A_1 \wedge ... \wedge A_n$, where $A_1, ..., A_n$ are all formulae. Then, the rules are displayed in Table 3.5. Now, the rules of binary tableaux are the standard ones given by Smullyan [138] (Table 3.2 above), which we conveniently consider in its unsigned version at this point. We recall the branching binary rules in Table 3.6, formulated in a way that makes their difference with the n-ary rules crystal clear. On the basis of the rules in Tabs. 3.5 and 3.6, the respective notions of n-ary or binary analytic tableau, closed and open branch and tableau, refutation, and proof can be defined in the standard

[10] In some cases below, we discuss these notions and results thereof in terms of refutations instead of proofs.

$$\frac{A_1 \vee ... \vee A_n}{A_1 \mid ... \mid A_n} \qquad \frac{\neg(A_1 \vee ... \vee A_n)}{\begin{array}{c}\neg A_1 \\ \vdots \\ \neg A_n\end{array}} \qquad \frac{\neg\neg A}{A}$$

$$\frac{\neg(A_1 \wedge ... \wedge A_n)}{\neg A_1 \mid ... \mid \neg A_n} \qquad \frac{A_1 \wedge ... \wedge A_n}{\begin{array}{c}A_1 \\ \vdots \\ A_n\end{array}}$$

Table 3.5: n-ary tableaux rules

$$\frac{A_1 \vee ... \vee A_n}{A_i | A_1 \vee ... \vee A_{i-1} \vee A_{i+1} \vee ... \vee A_n} \qquad \frac{\neg(A_1 \wedge ... \wedge A_n)}{\neg A_i | \neg A_1 \vee ... \vee \neg A_{i-1} \vee \neg A_{i+1} \vee ... \vee \neg A_n}$$

Table 3.6: Branching binary tableaux rules

way [e.g., 138].

Now, clausal tableaux are simply n-ary tableaux where all formulae are clauses (disjunctions of literals). The key observation is that, when the input formulae are all clauses, each application of the $\vee$-elimination rule of n-ary (specifically, clausal) tableaux requires the clause to be decomposed into its component literals in a single step; whereas, the $\vee$-elimination of binary tableaux requires the clause to be decomposed into its immediate subformulae. So, when the set of input formulae is only made of clauses, all nodes of an n-ary tableau—except for the initial ones storing the input set of clauses—are labelled by literals; whereas, the nodes of a binary tableau may also be clauses. Thereby, the difference between clausal and binary analytic tableaux boils down to the way in which clauses are represented, and to the exact form of the decomposition rules that are used. In clausal tableaux, clauses are represented as finite sets of literals and the corresponding rule requires a clause to be decomposed into its component literals in one step. In binary tableaux, clauses are represented as disjunctions built up from literals using a binary disjunction connective, and the tableau rule is such that a formula $A \vee B$ is decomposed into its immediate subformulae A and B.

Let us now recall the class of examples introduced by Cook and Reckhow, as well as Masacci's result. This class consists of sets of clauses H_n associated with labelled binary trees of depth n as follows: consider a binary tree $\mathcal{T}$ where each node, except the root, is labelled with a distinct literal, and sibling nodes are labelled with complementary literals. Then, it is stipulated that, for each pair of sibling nodes, the left-side node is labelled with a positive literal and the right-side one is labelled with the corresponding negative literal. Now, each branch b is associated with the clause containing exactly the literals in b. Thereby, the whole tree $\mathcal{T}$ is in turn associated with the set $H(\mathcal{T})$ of the clauses associated with $\mathcal{T}$'s branches. Finally, let $H_n = H(\mathcal{T}_n)$ denote the set of clauses associated with the complete binary tree of depth n. The tree with one node, T_0, is associated to the empty clause. We can represented H_n as follows [53, 65, 115, 116]:

$$H_n = \bigcup\{\pm A^1 \vee \pm A^2_{\pm} \vee \pm A^3_{\pm\pm} \vee \ldots \vee A^n_{\pm\ldots\pm}\},$$

where $+A$ means A and $-A$ means $\neg A$, and the subscript of A^i is a string of $i-1$ '+'s or '−'s corresponding to the sequence of signs of the preceding A^j, $j < i$. Thus, H_n contains 2^n clauses and $2^n - 1$ distinct atomic formulae. For example:

$$H_1 = \{A, \neg A\},$$
$$H_2 = \{A^1 \vee A^2_+, A^1 \vee \neg A^2_+, \neg A^1 \vee A^2_-, \neg A^1 \vee \neg A^2_-\}.$$

Each set H_n is, of course, unsatisfiable. Cook and Reckhow conjectured a lower bound on the number of nodes of a closed tableau for the conjunction of all disjunctions in H_n which is exponential in the size of H_n. Namely, they conjectured that any closed tableau for such a conjunction has at least $2^{2^{cn}}$ nodes, where $c > 0$ is some constant independent of n. Nevertheless, Massacci exhibited quasi-polynomial size binary tableaux refutations for H_n, and pointed out that the lower bound conjectured by Cook and Reckhow holds only for clausal tableaux. A consequence of Massacci's result is that binary tableaux can present an exponential speed-up over clausal ones; equivalently, the former can p-simulate the latter but not vice versa. Specifically, Massacci's showed that:

Theorem 3.3.12 ([115, 116]). *The proof complexity of binary analytic tableaux for H_n is bounded from above by $O(2^{n^2})$.*

In turn, the following Theorem was proven by Cook and Urquhart, which, given Massacci's result above, implies a restricted version of Cook and Reckhow conjectured lower bound:[11]

[11] The first published proof of an exponential lower bound for clausal tableaux was given by Murray and Rosenthal [122]. Furthermore, an improved version of the proof published in [149] is given in [9].

Theorem 3.3.13 ([149]). *The proof complexity of clausal analytic tableaux for H_n is bounded from below by $2^{\Omega(2^n)}$.*

Moreover, Massacci observed that clausal tableaux can be simulated by binary tableaux, and the resulting simulations are only twice as big as the original clausal tableau proofs. So, combining this observation with the last couple of Theorems, it follows that:

Theorem 3.3.14 ([115, 116]). *Binary tableaux can p-simulate clausal tableaux but not vice versa.*

Now, D'Agostino and Mondadori [53, 65] pointed out that Cook and Reckhow examples H_n are in fact hard for the truth-tables method since they involve $2^n - 1$ distinct atomic formulae. D'Agostino and Mondadori also showed that, by contrast, there are easy analytic *KE*-refutations of H_n which contain $2n + 2^n n - 2$ nodes. They describe the form of those *KE*-refutations as follows:

> [...] start with H_n. This will be a set containing $m(= 2^n)$ disjunctions of which $m/2$ start with A^1 and the remaining $m/2$ with its negation. Then apply *PB* to $\neg A^1$. This creates a branching with $\neg A^1$ in one branch and $\neg\neg A^1$ in the other. Now, on the first branch, by means of $m/2$ applications of the rule E∨1 we obtain a set of formulae which is of the same form as H_{n-1}. Similarly on the second branch we obtain another set of the same form as H_{n-1}. By reiterating the same procedure we eventually produce a closed tree for the original set H_n [65, pp. 307-308].

This shows that the truth-table method cannot *p*-simulate (the analytic restriction of) *KE* in "non-trivial" cases (Theorem 3.3.7 above); i.e., cases where the exponential blow up in the truth-table method depends on the logical form of the expressions and not only on the large number of variables. Remarkably, as pointed out by D'Agostino and Mondadori [65, p. 308], this class of examples illustrates an interesting phenomenon: while the complexity of the corresponding *KE*-refutations is not affected by the order in which the elimination rules are applied, in certain cases it may be highly sensitive to the choice of formulae to which we apply *PB*; let us refer to the latter as *PB-formulae*. "Wrong" choices can increase the size of the refutations up to an exponential factor, when "short" refutations are available if different choices are made. In the examples H_n, if *PB* is always applied to the last atomic variable in each clause, it is not difficult to see that the size of the refutation trees becomes exponential. So, D'Agostino and Mondadori suggested an *heuristic principle* to avoid this phenomenon (at least for the case where the input formulae are in clausal form):

> Let ϕ be a branch to which none of the linear *KE*-rules is applicable. Let S_ϕ be the set of clauses occurring in the branch ϕ, and let $p_i, ..., p_k$ be the list of all the atomic formulae occurring in S_ϕ. Let N_{p_I} be the number of clauses C such that p_i or $\neg p_i$ occurs in C. Then apply *PB* to an atom p_i such that N_{p_I} is maximal [65, p. 308].

When this principle is followed, "right" choices for the application of *PB* are guaranteed. Naturally, as far as proof-search in *KE* is concerned, more sophisticated criteria are needed to guide the choice of the *PB*-formulae. However, regardless how such a choice is made, (the analytic restriction of) *KE* can never perform significantly worse than the tableau method. The latter given that the simulation of the binary tableau rules via *KE*-rules given in [53, 65] is independent of the choice of *PB*-formulae. Nonetheless, good choices can sometimes be determinant for obtaining essentially shorter proofs than those obtained by the tableau method. Still, there are cases where how *PB*-formulae are chosen is unimportant. One of these cases is provided by the well-known class of "truly fat" expressions, used in [54] to show that clausal tableaux cannot p-simulate truth-tables (Theorem 3.3.6 above), and in [53, 65] to show that clausal tableaux cannot p-simulate the analytic restriction of *KE* (Theorem 3.3.5 above). Let us denote this class by H^c_k.[12] So, the examples in H^c_k are easy for (the analytic restriction of) *KE* as well as for the truth-table method, but hard for clausal tableaux. In [53, 54, 65], H^c_k was defined and used as follows: Let $p_1, ..., p_k$ be a sequence of k atomic variables. Then consider all the possible clauses containing as members, for each $i = 1, 2, ..., k$, either p_i or $\neg p_i$ and no other member. There are 2^k of such clauses. Now, let H^c_k denote the set containing these 2^k clauses. Thus, the expression $\bigwedge H^c_k$ is unsatisfiable. For example, $\bigwedge H^c_2$ is the following expression in conjunctive normal form:

$$(p_1 \vee p_2) \wedge (p_1 \vee \neg p_2) \wedge (\neg p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2).$$

So, this class of formulae owes its rather fancy name to the fact that the length of a "truly fat" expression is large compared to the number of distinct variables in it. Now, the key observation to compare the efficiency of clausal tableaux and truth-tables by means of such a class is that while the complexity of the tableaux refutations depends essentially on the length of the formula to be tested, the complexity of truth-tables depends only on the number of distinct propositional variables that occur in

[12]The superindex 'c' stands for "classical" since in the next Subsection we shall introduce a related class of examples which concerns both classical and non-classical proof systems.

it. In fact, the complexity of the truth-tables method is not always exponential in the length of the formula, but it is so only when the number of distinct variables approaches the length of the formula. In particular, observe that for the class H_k^c, truth-tables contain as many rows as clauses in the expressions, i.e., 2^k. Thereby, this class is not hard for the truth-tables. By contrast, a superpolynomial lower bound for clausal tableau refutations of this class was proven by D'Agostino:

Theorem 3.3.15 ([54]). *Every closed clausal tableau for the set H_k^c contains more than $k!$ distinct branches.*

Given that $k!$ grows faster than any polynomial function of 2^k, and there are 2^k rows in the truth-tables test of H_k^c, this implies that clausal tableaux cannot p-simulate the truth-tables method (Theorem 3.3.6 above).

Such a combinatorial explosion of clausal tableaux is not just an oddity due to a careful choice of artificial examples but the result of a fundamental inadequacy: the elimination of bivalence from the tableau method! The latter has the undesired consequence that the possible cases enumerated in the tableaux test are not mutually exclusive. When applying the tableaux branching rules for expressions in clausal form, it is crystal clear that the branches do not stand for mutually inconsistent alternatives. Consequently, when expanding the tree, we may (and very often do) end up considering more cases than necessary. In contrast, bivalence is clearly incorporated in the truth-tables method where all the possible assignments are mutually exclusive.

As recalled in Section 3.2, given that *KE* properly expresses the Principle of Bivalence via *PB*, distinct branches of a *KE*-tree define mutually exclusive alternatives; i.e., contain mutually inconsistent sets of formulae. Consequently, the expressions in H_k^c are easy also for (the analytic restriction of) *KE*. In fact, as shown in [53, 65], the number of branches in a *KE*-tree for H_k^c is exactly 2^{k-1}, that is the number of clauses in the expression divided by 2, and the refutation trees have size $O(k \cdot 2^k)$. This, together with the combinatorial explosion of clausal tableaux stated by Theorem 3.3.15, implies that analytic tableaux cannot p-simulate (the analytic restriction of) *KE* (Theorem 3.3.5 above).

Now, Massacci's result that binary tableaux dominate clausal ones in terms of the p-simulation relation opened the question whether or not the factorial lower bound for clausal tableaux with respect to the class H_k^c given by D'Agostino extends to binary tableaux. Hence, it also opened the question whether or not Theorems 3.3.6 and 3.3.5 also hold for binary tableaux. In the next Subsection we shall show that such a lower bound is indeed robust and extends not only to binary tableaux, but to the strongest (in terms of the p-simulation relation) possible version of analytic tableaux in the domain of clauses as unique input formulae. These strongest possible clausal tableaux were introduced by Arai, Pitassi and Urquhart [9], who called them *unrestricted* clausal tableaux. We shall show below that, in fact, unrestricted clausal tableaux cannot even p-simulate the truth-tables method.

Arai et al. [9] pointed out that Massacci's result recalled above strongly depends on the way in which clauses are parenthesized. Namely, the result depends on the assumption that '$\vee$' associates to the right. With that order of bracketing quasi-polynomial size binary tableaux refutations of H_n can be obtained, as shown by Massacci. However, Arai et al. proved that on the assumption that '$\vee$' associates to the left, exponential size binary tableaux refutations are required, much as in the case of (simple) clausal tableaux. This exponential separation in terms of bracketing led Arai et al. to introduce a generalized version of clausal tableaux in which such a separation does not hold, and called them *unrestricted* clausal tableaux. The difference between these generalized clausal tableaux with binary and clausal ones again boils down to the way in which clauses are represented, and to the exact form of the decomposition rules that are used. In the case of unrestricted clausal tableaux, clauses are again taken as finite sets of literals but instead of insisting on decomposing a clause in one step, arbitrary partitions of clauses are allowed. Formally, an *unrestricted* clausal tableau for a finite set Γ of clauses is a tree whose root is labelled with Γ and whose nodes of depth > 0 are labelled with clauses. The latter are generated, starting from the clauses in Γ, by means of the following liberalized decomposition rule:

$$\frac{C}{C_1 \mid ... \mid C_n}$$

where $C_1, ..., C_n$ is any partition of C. Applying the decomposition rule *at* a node d with premise C means that C labels either d or a node occurring above d in the same branch, and that the children of d are labelled with the clauses $C_1, ..., C_n$ in accordance with the rule. In this context we say that C is *associated* with d, although d itself may be labelled with another formula. Again, the notions of closed branch and tableau, and refutation are defined in the standard way.

Clearly, the liberalized decomposition rule of unrestricted clausal tableaux is more powerful and flexible than either the clausal or the binary rule. In fact, Arai et al. [9] showed that unrestricted clausal tableaux p-simulate both binary and clausal tableaux but the p-simulation does not hold in the reverse direction for either system, so that unrestricted tableaux are the most powerful variant of clausal tableaux as far as the length of proofs is concerned. Besides, they proved a super-polynomial lower bound for unrestricted clausal tableaux on Cook and Reckhow's class of examples H_n which, as we mentioned above, are hard also for the truth-tables method (but

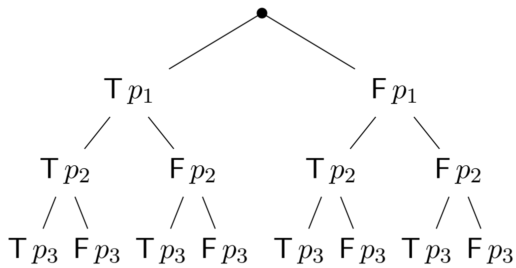


Figure 3.1: $\mathcal{T}_3$: each branch corresponds to a total assignment to $p_1, ..., p_k$.

easy for *KE*). In the next Subsection we shall show a super-polynomial lower bound for unrestricted clausal tableaux on the class of examples H_k^c which, again, are easy for the truth-tables and *KE*. Thus, we shall show that even the strongest possible version of clausal tableaux cannot p-simulate the truth-tables method and, thus, that it is quite departed from the standard semantics of classical logic.

3.3.3 New lower bounds on analytic tableaux

Let $p_1, p_2, p_3, ...$ be any enumeration of the atomic variables of a standard propositional language. For each given $k \in \mathbb{N}$, let $\mathcal{T}_k$ be the tree of signed atomic formulae that represents all possible assignments to the first k atomic variables. For example, $\mathcal{T}_3$ is the tree in Figure 3.1. Consider now the first $2k-1$ atomic variables and let $\mathcal{T}_k^*$ be the tree obtained from $\mathcal{T}_{2k-1}$ by truncating each branch as soon as it contains either k atomic variables signed with T or k atomic variables signed with F. For example, $\mathcal{T}_3^*$ is the tree in Figure 3.2. Note that, since $2k-1$ is odd, each branch contains k atomic variables signed with T or, otherwise, k atomic variables signed with F. $\mathcal{T}_k^*$ contains $O(c^k)$ branches (with $c = 4$).

For each branch b of $\mathcal{T}_k^*$ let the signed formula φ_b be defined as follows:

$$\varphi_b = \begin{cases} \mathsf{F} \bigwedge\{p_i \mid \mathsf{T}\, p_i \text{ occurs in } b\} & \text{if } b \text{ contains } k \text{ atomic formulae signed with } \mathsf{T} \\ \mathsf{T} \bigvee\{p_i \mid \mathsf{F}\, p_i \text{ occurs in } b\} & \text{otherwise} \end{cases}$$

Now, let $H_k = \{\varphi_b \mid b \text{ is in } \mathcal{T}_k^*\}$. For example:

$$H_2 = \{\mathsf{F}\, p_1 \wedge p_2, \mathsf{F}\, p_1 \wedge p_{3,} \mathsf{T}\, p_2 \vee p_3, \mathsf{F}\, p_2 \wedge p_3, \mathsf{T}\, p_1 \vee p_3, \mathsf{T}\, p_1 \vee p_2\}.$$

The number of formulae in H_k is equal to the number of branches in $\mathcal{T}_k^*$ and therefore is $O(c^k)$. Moreover each formula contains k atomic variables.

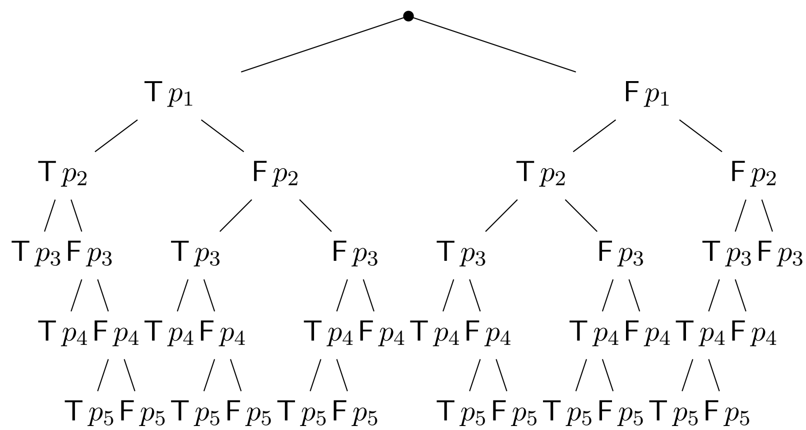


Figure 3.2: $\mathcal{T}_3^*$: each branch corresponds to a partial assignment to $p_1, ..., p_{2k-1}$.

Consider the "quasi-clausal" tableau method based on the following n-ary expansion rules (for every $n \in \mathbb{N}$):

$$\frac{\mathsf{T}\, p_1^{\pm} \vee \cdots, \vee p_n^{\pm}}{\mathsf{T}\, p_1^{\pm} \mid \cdots \mid \mathsf{T}\, p_n^{\pm}} \qquad \frac{\mathsf{F}\, p_1^{\pm} \wedge \cdots \wedge p_n^{\pm}}{\mathsf{F}\, p_1^{\pm} \mid \cdots \mid \mathsf{F}\, p_n^{\pm}} \tag{3.1}$$

Where $p_i^{\pm}$ is either the literal p_i or the literal $\neg p_i$. (Note that the signed formulae in H_k do not contain negative literals.)

Theorem 3.3.16. *Any closed quasi-clausal tableau for H_k contains more than $k!$ branches.*

Proof. Let $h(d)$ be the number of nodes resulting from applications of the expansion rules that occur in the path to the node d (i.e., not counting the nodes labelled with the initial signed formulae). Let $g(d)$ be the number of children of d that do not generate immediately closed paths. Then, if d's children result from the expansion of a T-formula (an F-formula), $g(d) = k - m$, where m is the number of atomic F-formulae (T-formulae) occurring in the path to d. Hence:

$$g(d) \geq k - h(d)$$

Thus, any closed quasi-clausal tableau contains more than $\prod_{i=0}^{k-1} k - i = k!$ branches. □

Since $k!$ cannot be bounded above by any polynomial in c^k, this implies that the set of all H_k is intractable for tableaux.

In turn, showing the inconsistency of H_k by means of the complete truth-tables would require checking the 2^{2k-1} branches of $\mathcal{T}_k^*$ and so the set of all H_k for $k \in \mathbb{N}$ is tractable for the truth-tables. Thus, this is yet another class of examples showing the mismatch between classical tableaux (or corresponding cut-free Gentzen systems) and the standard semantics of **CPL**. The main interest of this specific class of examples is that it involves only the logical operators $\wedge$ and $\vee$. Hence, these examples are hard for any non-classical tableau method that adopts the $\wedge$ and $\vee$ rules in (3.1).

As we recalled above, in [115, 116], Massacci showed that in the context of classical logic, clausal tableaux, in which every clause is decomposed in one go as in our quasi-clausal tableaux, cannot p-simulate standard binary tableaux. This result appears to put in question the significance of Theorem 3.3.16. However, we shall show that the lower bound is robust and applies to binary tableaux as well as to quasi-clausal ones, independently of how the atomic variables are associated to form binary disjunctions and conjunctions and of the order in which they occur.

Let us say that a tableau $\mathcal{T}$ is *non-redundant* if (i) no branch of $\mathcal{T}$ contains two or more nodes labelled with the same clause and (ii) for every node d, its labelling formula is used at least once as premise of a rule application in the subtree generated by d. It is straightforward to verify that:

Lemma 3.3.17. *Every closed tableau $\mathcal{T}$ for a set Γ of formulae can be transformed into a closed non-redundant tableau $\mathcal{T}'$ for Γ such that $|\mathcal{T}'| \leq |\mathcal{T}|$.*

Therefore we can assume without loss of generality that tableaux are non-redundant.

Theorem 3.3.18. *Any closed binary tableau for H_k contains more than $k!$ branches.*

Proof. Let $\mathcal{T}$ be a closed binary tableau for H_k. In the context of this proof it is convenient to view a tableau for H_k as a tree whose root is labelled with H_k and whose nodes of depth > 0 are labelled with signed formulae generated, starting from those in H_k, by means of the usual (binary) tableau rules for signed formulae. Applying a tableau rule *at* a node d *with premise* φ means that a β-formula φ—using Smullyan's notation—labels either d or a node occurring above d in the same branch, and that the children of d are labelled with its components β_1 and β_2. (Note that only β-formulae can occur in a tableau for H_k.) In this context we say that the premise φ of the rule application is *associated* with d, although φ may be the labelling signed formula of another node occurring above d. For each node d of a tableau, we denote by $\varphi(d)$ the labelling formula of d.

We now define a sequence $\mathcal{T}_{\sigma_1}, ..., \mathcal{T}_{\sigma_k}$ based on $\mathcal{T}$.

- Base: $\mathcal{T}_{\sigma_1}$. Let $\varphi \in H_k$ be the signed formula associated with the root of $\mathcal{T}$, i.e., the two nodes of depth 1 result from applying a tableau rule to φ. Given an arbitrary atomic variable σ_1 occurring in φ. Let d_{σ_1} be the node that contains such occurrence of σ_1. Assign this atomic variable as *distinguished atom* to this node and to every node in $\mathcal{T}$ resulting from applying the relevant tableau rule to a node whose distinguished atom is σ_1. $\mathcal{T}_{\sigma_1}$ is the subtree generated by d_{σ_1} with σ_1 assigned to all nodes that descend from d_{σ_1} as explained above.

- Step: $\mathcal{T}_{\sigma_{i+1}}$. Given $\mathcal{T}_{\sigma_i}$ ($i < k$), let d be the highest node with an assigned distinguished atom such that neither of its children is assigned a distinguished atom. This means that the children of d result from applying the relevant tableau rule to a new signed formula φ in H_k. Let σ_{i+1} be an arbitrary atomic variable occurring in φ and distinct from all $\sigma_1, ..., \sigma_i$ (for $i < k$ such an atomic variable always exists), and $d_{\sigma_{i+1}}$ the child of d that contains such occurrence of σ_{i+1}. Assign σ_{i+1} as distinguished atom to this node and to every node in $\mathcal{T}_{\sigma_i}$ that results from applying the relevant tableau rule to a node whose distinguished atom is σ_{i+1}. $\mathcal{T}_{\sigma_{i+1}}$ is the subtree generated by d_{σ_1} with the new distinguished atom σ_{i+1} assigned to all the nodes that descend from $d_{\sigma_{i+1}}$.

Assuming that $\mathcal{T}$ is non-redundant, there is at least a branch of $\mathcal{T}_{\sigma_i}$ that closes on σ_i, i.e., the closure rule is applied to nodes labelled with $\mathsf{S}\,\sigma_i$ and its conjugate, $\mathsf{S}\,\sigma_i$ descends from d_{σ_i}. Let us call such a branch a σ_i*-branch*. Consider $\mathcal{T}_{\sigma_k}$ and observe that different choices of σ_i lead to distinct σ_i-branches. In general, the σ_i-branches are all distinct from all the σ'_i-branches whenever $\sigma_i \neq \sigma'_i$. Let σ denote a sequence of choices for σ_i ($i \leq k$). Hence, the difference between the set

$$B_\sigma = \{\sigma_i\text{-branches} \mid \sigma_i \in \sigma, i \leq k\}$$

and the analogous set of a distinct sequence σ' of choices is always non-empty. Since each signed formula in H_k contains k atomic variables there are at least $k!$ different choice sequences and, therefore, at least $k!$ distinct sets B_σ. It follows that there are at least $k!$ branches in $\mathcal{T}$. □

The main interest of the class of examples H_k is that it involves only the logical operators $\wedge$ and $\vee$, with no restriction on the way in which β-formulae are formed, provided they contain the same set of atomic variables. Hence, these examples are hard for all non-classical tableau methods that share the $\wedge$ and $\vee$ rules with classical tableaux. These include known tableau methods for a variety of logics, for instance, tableaux for **FDE** [e.g., 53, 81], **LP** and $\mathbf{K_3}$ [e.g., 82, 127, 15], and **IPL** [e.g., 79, 80].

Note that the proof makes no reference either to the way in which the atomic variables are associated, or to the order in which they occur in disjunctions and conjunctions.

On the other hand, it can be verified that these examples are easy for *KE* (and so for the intelim method). To see this, just observe that the very tree $\mathcal{T}_k^*$ used to define the class H_k can be easily expanded into a closed *KE*-tree for H_k that contains the same number of branches as $\mathcal{T}_k^*$ and so these examples have polynomial size *KE*-refutations. Hence, this latter fact together with Theorem 3.3.18 imply that *KE*-style systems can p-simulate the tableau methods which share the $\wedge$ and $\vee$ rules with classical (Smullyan-style) tableaux but the p-simulation does not hold in the reverse direction.

Now, the same approach can be used to prove new lower bounds for classically refutable sets of clauses using unrestricted clausal tableaux [9]. Recall that a *literal* is either an atomic variable or the negation of an atomic variable and that (in all logics with a classical disjunction) a *clause* can be taken as a finite set of literals. To help readibility, In what follows we shall represent negative literals as $\overline{p}$ with p an atomic variable. Moreover, we shall represent a clause by simply listing its literals with no separators, e.g., $p_1\overline{p}_2p_3$ will represent the clause $\{p_1, \overline{p}_2, p_3\}$.

Let us go back to the tree $\mathcal{T}_k$ defined at the beginning of this Subsection (see Figure 3.1 for $\mathcal{T}_3$). For each branch b of $\mathcal{T}_k$ let

$$C_b = \{p_i \mid \mathsf{T}\, p_i \text{ occurs in } b\} \cup \{\overline{p}_i \mid \mathsf{F}\, p_i \text{ occurs in } b\}$$

and

$$H_k^c = \{C_b \mid b \text{ is a branch of } \mathcal{T}_k\}.$$

Now, within this context, clausal tableaux are tableaux that use the following decomposition rule:

$$\frac{L_1 \vee ... \vee L_n}{L_1 \mid ... \mid L_n} \tag{3.2}$$

where each L_i is a literal. So, clauses are decomposed in one go. Recall that unrestricted clausal tableaux [9], on the other hand, are governed by the following liberalized decomposition rule:

$$\frac{C}{C_1 \mid ... \mid C_n}$$

where $C_1, ..., C_n$ is any partition of C. Recall also that: (i) a factorial lower bound for clausal tableau refutations of H_k^c was proven by D'Agostino in [54]; (ii) in [115, 116], Massacci showed that clausal tableaux cannot p-simulate binary tableaux; (iii) in

[9] Arai et al. showed that unrestricted clausal tableaux can p-simulate both binary and clausal tableaux but the p-simulation does not hold in the reverse direction for either system. It turns out that the factorial lower bound in [54] holds also for unrestricted tableaux on the same class H_k^c of examples, so settling a problem left open in [115, 116, 9].

Theorem 3.3.19. *Any unrestricted clausal tableau refutation of H_k^c contains at least $k!$ distinct branches.*

We omit the proof that can be adapted from the proof of Theorem 3.3.18. In turn, since H_k^c is easy for the truth-tables:

Corollary 3.3.20. *Unrestricted clausal tableaux cannot p-simulate the truth-tables.*

The above Corollary is further evidence for the claim, put forward by D'Agostino and Mondadori in [53, 65], that analytic tableaux and cut-free Genzen systems, despite their widely acknowledged merits, exhibit startling anomalies both from the computational and the semantic viewpoints, in that they cannot p-simulate the most straightforward algorithm based on the standard semantics of **CPL**.

3.4 Tractable depth-bounded deduction

In *KE*, *KI*, and the intelim method, the operational rules are all linear and the only branching rule is the structural rule *PB*. Thus, it is quite natural to investigate the subsystems that result from bounding the applications of that single structural rule. A suggestion in this direction for *KE* appears already in [65]. However, *KE* with limited bivalence was more thoroughly investigated only until [e.g., 75, 76]. The same idea applied to the intelim method started to be investigated in [55], and led to what here we have called the "depth-bounded approach" to **CPL** [e.g., 61, 59, 56, 57, 58, 63].

In the intelim method, either the introduction or the elimination rules may be regarded as redundant. This in the sense that both *KE* and *KI* are complete for **CPL**—and, besides, they can linearly simulate each other. Nevertheless, there are two reasons for using both introduction and elimination rules: (a) it allows for more natural and shorter proofs, although not essentially shorter because the corresponding simulation is polynomial; (b) it reduces the number of applications of *PB* that, as we shall explain below, is key to define the notion of *depth* in the depth-bounded approach to **CPL**. Accordingly, the latter was originally investigated as having the intelim method as proof-theoretical basis, but it can be straightforwardly adapted to be based on *KE* or *KI* alone. Below we first recall the approach to **CPL** as it was originally investigated, and then we briefly discuss how a *KE*-basis for it looks like.

3.4.1 Depth-bounded intelim deduction

As mentioned at the beginning of this Chapter, a key idea underlying the depth-bounded approach to **CPL** is that the *informational* meaning of a logical operator is specified solely in terms of the information that is held (actually possessed) by the agent under consideration. Again, that an agent holds information means that this is information *practically* accessible to her and with which she *can* operate. As we shall recall in this section, the intelim method can be used as a natural basis for applying the depth-bounded approach to **CPL** because: (i) the intelim rules can be interpreted in a way in which they fix the meaning of the classical connectives only in terms of the information which the agent holds; (ii) accordingly, the logic characterized by the intelim rules—the basic (0-depth) logic of approximations to **CPL**—is tractable; (iii) the nested applications of *PB* provide a sensible measure of inferential *depth*, which naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations to **CPL**; (iv) such approximations can be naturally related to the inferential power of the agent and admit of an intuitive, albeit non-deterministic, 3-valued semantics that was first put forward by W.V.O. Quine [129] and whose values have a natural informational interpretation.

Given that a key idea underlying the approach is that the meaning of the logical operators is fixed exclusively in terms of the information that agents hold, the approach crucially depends on the understanding of the notion of "holding information". Thus, within the context of **CPL**, some basic remarks regarding such a notion are [see 57]: first, such a notion is understood in a "weak" ordinary sense, according to which it may well be that the agent holds the information that A is true even if A is in fact false, or vice versa. So, in turn, it is not assumed a "strong" notion of information complying with, e.g., Floridi's "veridicality thesis", according to which information must be truthful [84]. Second, the notion of "holding the information that a sentence is true" is treated symmetrically to the notion of "holding the information that a sentence is false"; however, the two notions are not complementary, even when the underlying notions of truth and falsity are. Namely, not holding the information that A is true is clearly not equivalent to holding the information that A is false, even if we identify falsity with the lack of truth. Accordingly, it may well be that for a given A, the agent neither holds the information that A is true nor the information that A is false. On the other hand, when applying the depth-bounded approach to **CPL**, it is assumed that no agent can hold both the information that A

is true and the information that A is false, as that is taken to amount to holding no definite information about A.[13] Third, that an agent holds information means that this is information practically accessible to her and with which she can operate. This can be made more precise by requiring that the notion at issue satisfies the following condition:

> *Strong Manifestability.* If an agent a grasps the informational meaning of a sentence A, then a should be able to tell, in practice and not only in principle, whether or not (s)he actually possesses the information that A is true, or the information that A is false or neither of them [57, p. 412].

As recalled below, "in practice" in the condition above is to be interpreted in the sense that the agent has a *feasible* procedure to decide whether or not she holds the information that A is true or false, on the sole basis of the information that she *explicitly* holds and of the meaning of the logical operators occurring in A.

We mentioned above that the rules of the intelim method make the latter to resemble natural deduction—with no discharge rules. As it is customary in natural deduction, those rules can be seen as *definitions* of the classical connectives if the meaning of the latter is identified with the role they play in basic inferences—yielding a proof-theoretic semantics. Indeed, those rules satisfy a form of the inversion principle [55, 59], which plays a crucial role in the admissibility of a set of natural deduction rules as definitions of the connectives. Such a principle essentially says that no information can be obtained from applying an elimination rule to a sentence A that would not have already been available if A had been obtained by means of an introduction rule. In that sense, the introduction rules can be taken as sufficient for fixing the meaning of the connectives, while the elimination rules can be taken ultimately as "justified" in terms of the former. Further, observe that the intelim rules for disjunction and conjunction are dual of each other, and that a sentence and its negation are treated in a symmetric way.

Now, we can re-interpret the S-formulae in the intelim rules in informational terms as follows: the intended meaning of $\mathsf{T}\,A$ is "A is informationally true" or "the agent holds the information that A is true"; whereas the intended meaning of $\mathsf{F}\,A$ is "A is informationally false" or "the agent holds the information that A is false".[14]

[13] As recalled below, this assumption does not rule out the possibility of "hidden" inconsistencies in an agent's *information state*, but only of inconsistencies which can be feasibly detected by the agent. Moreover, we shall drop such an assumption when we apply the depth-bounded approach to paraconsistent logics in Ch. 6 below.

[14] As pointed out in [58], in this context, 'T' and 'F' serve as propositional attitudes and so, in a multi-agent setting, could be indexed by symbols '$x, y, z, ...$' denoting different agents. Thus, $\mathsf{T}_x A$

Thus, we re-interpret the intelim rules as follows: for all sentences A and B, any agent holding the information about A and B above the line, thereby holds also the information about A and B below the line. Alternatively—in terms of Quine's dispositional theory of the "primitive" meaning of the connectives mentioned in the previous chapter—we might use the signs T and F to respectively mean assent and dissent, based on the information the agent holds. In turn, we shall keep using $\varphi, \psi, \theta, ...$, as variables ranging over S-formulae, and $X, Y, Z, ...$, as variables ranging over sets of S-formulae. Also, recall that the unsigned part of an S-formula is the unsigned formula that results from it by removing the sign T or F. Given an S-formula φ, we denote by φ^u the unsigned part of φ and by X^u the set $\{\varphi^u \mid \varphi \in X\}$. Finally, we say that the conjugate of $\mathsf{T}\,A$ is $\mathsf{F}\,A$ and vice versa.

The basic (0-depth) logic of the hierarchy of depth-bounded approximations to **CPL** is characterized by means of the intelim rules. In turn, such a basic logic is identified with the logic that characterizes exactly all the classical logical inferences that can be drawn by using only actual information—via the meaning of the connectives. The intelim rules generate *intelim sequences*; i.e., finite sequences $(\varphi_1, ..., \varphi_n)$ of formulae such that, for every $i = 0, ..., n$, either φ_i is an assumption or it is the conclusion of the application of an intelim rule to preceding S-formulae. Again, the intelim rules are not complete for **CPL** but—as we shall recall below— they are only so for the basic (0-depth) logic. Completeness for full **CPL** is obtained by adding the unique branching rule *PB*, which allows us to append both $\mathsf{T}\,A$ and $\mathsf{F}\,A$ as sibling nodes at the end of any intelim sequence, generating two new (mutually exclusive) branches. Thus, by adding *PB* to the stock of rules, proofs and refutations are represented by downward-growing trees—bringing the method somewhat closer to tableaux.

The intuitive informational meaning of *PB* is that one of the two cases must obtain even if the agent has no actual information about which is the case. In this sense, we call the information expressed by each conjugate S-formula *virtual* information; i.e., hypothetical information that an agent does not hold, but she temporarily *assumes as if* she held it. So, each application of *PB* stands for the introduction of virtual information about the truth or falsity of a formula A. We call the S-formulae $\mathsf{T}\,A$ and $\mathsf{F}\,A$ introduced by an application of *PB virtual assumptions*. Besides, as recalled above, *PB* is essentially a cut rule which is not eliminable, but its application can be restricted so as to satisfy the subformula property. Furthermore, from the infor-

and $\mathsf{F}_x A$ would respectively mean "agent x holds the information that A is true (false)". Even though in this thesis we will be dealing with different *sources* informing an agent, we will not deal with multi-agent settings of the depth-bounded approach. However, a multi-agent setting for the approach started to be explored in [46].

mational viewpoint of the depth-bounded approach, the main conceptual advantage of this proof-theoretical characterization of **CPL** consists in that it clearly separates the rules that fix the meaning of the connectives in terms of the information that the agent holds (the intelim rules) from the single structural rule that introduces virtual information (the *PB* rule).[15] Intuitively, the more virtual information needs to be invoked via *PB*, the harder the inference is for the agent, both from the computational and the cognitive viewpoint. In this sense, the nested applications of PB provide a sensible measure of inferential *depth*. This naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations to **CPL** in terms of the maximum number of nested applications of PB that are allowed.

Thus, the 0-depth logic is simply the logic of the inferences that can be drawn by using only actual information, and whose validity can be determined on the sole basis of the informational meaning of the connectives. The latter is reminiscent of Quine's dispositional theory in that those inferences are justified only by virtue of the way in which the language is immediately used in inference, without any reference to metaphysical assumptions about the "world". By contrast, the k-depth logics, $k > 0$, are the logics of the classical inferences whose validity requires the introduction of virtual information about the truth or falsity of formulae. That is, the latter inferences require to make reference to the information-transcendent notions of truth and falsity. These notions, in turn, are assumed to obey the classical Principle of Bivalence: an "external reality" makes any sentence determinately either true or false regardless of the agent's holding any information about it. That Principle is thus regarded as a metaphysical assumption which plays an indirect inferential role in justifying inferences, and cannot be justified by means of the meaning of the connectives. Thus, the Principle is not really a logical rule but a *structural* assumption concerning the relationship between language and "world".

Moreover, Bivalence is not the only metaphysical-structural assumption governing the notions of classical truth and falsity. Another such an assumption is expressed by the Principle of Non-contradiction: the "world" is such that no sentence can be simultaneously true and false. In **CPL**, the negation of A means that A is false and the falsity of any sentence, as its truth, is a relation between the sentence itself and the postulated "external reality". Therefore, in **CPL** Non-contradiction is, like Bivalence, a structural assumption about the relationship between language and

[15]Cf. Gentzen-style proof systems, where the "discharge rules" of natural deduction, as well as their counterparts in the sequent calculus, make essential use of virtual information. Given that in Gentzen-style systems cut is eliminable, no hierarchy of approximations can be defined by controlling the application of the cut rule.

"world".[16] Non-contradiction is regarded as an structural assumption in the intelim method and expressed by the closure of a branch [see 55]. Thereby, according to the depth-bounded approach, the semantics of **CPL** is not completely characterized by the meaning of the connectives as it can be defined via an inferential approach. In order for that semantics to be fully characterized the reference to the metaphysical, information-transcendent, notions of truth and falsity is required.

Based on the definitions of Subsection 3.2.3, let us now recall the main definitions and results expressing these intuitions.

Definitions 3.4.1. The *depth* of an intelim tree $\mathcal{T}$ is the maximum number of virtual assumptions occurring in a branch of $\mathcal{T}$. An intelim tree $\mathcal{T}$ is a *k-depth intelim proof* of φ from X (a *k-depth refutation* of X) if $\mathcal{T}$ is an intelim proof of φ from X (a intelim refutation of X) and $\mathcal{T}$ is of depth k.

Note that a 0-depth intelim tree is nothing but an intelim sequence. Thereby:

Definitions 3.4.2. For all φ, X, φ is *0-depth deducible* from X, denoted by $X \vdash_0 \varphi$, if there is a 0-depth proof of φ from X. In turn, X is *0-depth refutable*, denoted by $X \vdash_0$, if there is a 0-depth refutation of X.

Remark 4. We write $X \vdash_0$ to mean that $X \vdash_0 \varphi$ for all φ. Below we use similar notations with analogous intended meanings. Besides, as we shall discuss in the next section, a stronger 0-depth deducibility relation is defined as follows: A is *0-depth deducible from* Γ if there is a 0-depth refutation of $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$. However, as explained in the next section, such a stronger relation fails to satisfy cut and, thus, to be a Tarskian cosequence relation. This stronger relation may still be preferred if cut is not taken as a necessary condition for a logical consequence relation.

Notation 3.4.3. Let us use the same relation symbol '$\vdash_0$' to denote 0-depth deducibility and 0-depth refutability.

Proposition 3.4.4 ([59, 58]). *The relation* $\vdash_0$ *is a structural Tarskian consequence relation (Tcr).*

[16] By contrast, Non-contradiction is also an intuitionistic principle, but in intuionistic terms such a Principle refers rather to the internal consistency of a mathematician's mental representations. More formally, in intuitionistic logic, Non-contradiction stems immediately from the meaning of the negation connective: $\neg A$ can be legitimately asserted only if one is able to provide a refutation of A. Instead, classically, $\neg A$ is true iff A is false. Hence, the meaning of $\neg A$ is intertwined with the metaphysical notion of falsity, and so is not amenable to epistemic characterizations.

Besides, $\vdash_0$ has no tautologies. This might be expected since a tautology is usually described as "a conclusion of the empty set of assumptions". There is no way of extracting information from the empty set of assumptions without introducing virtual information. Accordingly, tautologies appear only at depths $k > 0$, when the use of virtual information is allowed, and the set of provable tautologies increases with k. More importantly, on the basis that the intelim method enjoys the subformula property (SFP) it can be shown that the 0-depth logic is tractable—just as we should expect since it intends to be the logic of actual information. In fact, given the structure of intelim sequences—which involve no branching nor "case reasoning"—the SFP ensures a straightforward feasible decision procedure for $\vdash_0$, and thus the corresponding logic is in accordance with the Strong Manifestability condition above. Namely, let $|\varphi|$ denote the *size* of an S-formula φ (i.e., the total number of occurrences of symbols in φ), and the *size* of a finite set of S-formulae Y be defined as $\sum_{\varphi \in Y} |\varphi|$ and denoted by $|Y|$. Then:

Theorem 3.4.5 ([59]). *Whether or not $X \vdash_0 \varphi$ ($X \vdash_0$) can be decided in time $O(n^2)$, where $n = |X \cup \{\varphi\}|$ ($n = |X|$).*[17]

Although $\vdash_0$ is *explosive* (i.e., when X is 0-depth refutable, $X \vdash_0 B$ for any B), 0-depth refutability is stricter than classical refutability, for a set X may well be 0-depth non-refutable but classically refutable. More importantly, we can feasibly detect that a set X of assumptions is 0-depth refutable and, therefore, we may as well abstain of drawing bizarre conclusions on its basis. Thus, we always have feasible means to ensure that a set X is 0-depth non-refutable—in which case $\vdash_0$ may be regarded as not explosive—even if X is classically refutable. In informal terms—and reminiscent of Quine's dispositional theory of the "primitive" meaning of the connectives—$\vdash_0$ stands for the "easy" inferences that (nearly) every agent learns to make correctly in the very process of learning the meaning of the (classical) connectives.

The 0-depth logic is simply the system of deductive reasoning with no virtual information. For those classical inferences that cannot be justified solely by the meaning of the connectives, we need to incorporate information that is not even implicitly contained in the assumptions. Concretely, the k-depth logics are defined in terms of the nested use of virtual information, allowed up to a number of times k. Here the underlying intuition is that the more times virtual information is required, the most difficult the corresponding inference is for the agent.

Now, the notion of k-depth deducibility depends not only on the depth at which the use of virtual information is recursively allowed, but also on the subset of $F(\mathcal{L})$ on

[17] See also [55].

which the introduction of virtual information is allowed. We call this subset *virtual space* and define it as a function f of the set $\Gamma \cup \{A\}$ consisting of the premises Γ and of the conclusion A of the given inference. Specifically, let $\mathcal{F}$ be the set of all functions f on the finite subsets of $F(\mathcal{L})$ such that: (i) for all Δ, $\mathsf{at}(\Delta) \subseteq f(\Delta)$; (ii) $f(\Delta)$ is closed under subformulae, i.e., $\mathsf{sub}(f(\Delta)) = f(\Delta)$; (iii) the *size* of $f(\Delta)$ is bounded above by a polynomial in the size of Δ, i.e., $|f(\Delta)| \leq p(|\Delta|)$ for some fixed polynomial p.[18] Distinguished examples of functions in $\mathcal{F}$ are the identity function $f(\Delta) = \Delta$, sub and at. Note that, in general, $f(\Delta)$ may contain formulae that are not in sub. For instance, the operation f that maps Δ to the set of all formulae of bounded degree that can be built out of sub and at is also in $\mathcal{F}$.

Put differently, the set of formulae that can be used as *PB*-formulae may be bounded in a variety of ways without loss of completeness. The strictest option is allowing as *PB*-formulae only atomic formulae that occur in $\Gamma \cup \{A\}$. A more liberal option is allowing only subformulae of the formulae in $\Gamma \cup \{A\}$. Shorter deductions can be obtained by further liberalizing the restrictions on the virtual space allowing for deductions that do not enjoy the SFP (simply by permitting applications of *PB* to formulae that are not subformulae either of the premises or of the conclusion), but in which the virtual space is still bounded. The choice of an specific function to yield suitable values of the virtual space for each particular deduction problem is the result of decisions that are conveniently made by the system designer, depending on the intended application. Such decisions affect the deductive power of each given k-depth deducibility relation, and so the "speed" at which the approximation process converges to the limiting logic at issue. In turn, the functions in $\mathcal{F}$ are partially ordered by the relation $\trianglelefteq$ such that $f_1 \trianglelefteq f_2$ iff, for every finite Δ, $f_1(\Delta) \subseteq f_2(\Delta)$. Thus, finally:

Definitions 3.4.6. For all X, φ, and for all $f \in \mathcal{F}$,

- $X \vdash_0^f \varphi$ iff $X \vdash_0 \varphi$;
- for $k > 0$, $X \vdash_k^f \varphi$ iff $X \cup \{\mathsf{T}\,A\} \vdash_{k-1}^f \varphi$ and $X \cup \{\mathsf{F}\,A\} \vdash_{k-1}^f \varphi$ for some $A \in f(X^u \cup \{\varphi^u\})$.

When $X \vdash_k^f \varphi$, we say that φ is *deducible at depth k from X over the f-bounded virtual space*. Note that the above definitions cover also the case of k-depth refutability by assuming $X \vdash_k^f$ as equivalent to $X \vdash_k^f \varphi$ for all φ. Thus:

- $X \vdash_0^f$ iff $X \vdash_0$;

[18] Requirement (iii) is essential in order to define a hierarchy of *tractable* approximations to the limiting logic under consideration; in this case, **CPL**.

- for $k > 0$, $X \vdash^f_k$ iff $X \cup \{\mathsf{T}\,A\} \vdash^f_{k-1}$ and $X \cup \{\mathsf{F}\,A\} \vdash^f_{k-1}$ for some $A \in f(X^u)$.

When $X \vdash^f_k$, we say that X is *refutable at depth k over the f-bounded virtual space.*

Now, it follows immediately from Def. 3.4.1 and 3.4.6 that:

Proposition 3.4.7. *For all X, φ and all $f \in \mathcal{F}$, $X \vdash^f_k \varphi$ ($X \vdash^f_k$) iff there is a k-depth intelim proof of φ from X (a k-depth intelim refutation of X) such that all its PB-formulae are in $f(X^u \cup \{\varphi^u\})$ ($f(X^u)$).*

Notation 3.4.8. We shall abuse of the same relation symbol $\vdash^f_k$ to denote both k-depth deducibility and k-depth refutability.

In the above Definitions 3.4.6, the transition from $\vdash^f_{k-1}$ to $\vdash^f_k$ is determined by introducing virtual information about the truth or falsity of some A in the f-bounded virtual space, and checking that in either case φ is deducible at the immediately lower depth. Unlike $\vdash_0$, the relations $\vdash^f_k$ are not Tarskian. However, they get very close to being such, for they satisfy reflexivity, monotonicity, and the following version of cut:[19]

Depth-bounded cut: If $X \vdash^f_j \varphi$ and $X \cup \{\varphi\} \vdash^f_k \psi$, then $X \vdash^f_{j+k} \psi$.

Moreover, the relations $\vdash^f_k$ may not be structural in that structurality depends on the function f that defines the virtual space. For example, $\vdash^{\mathsf{sub}}_k$ is structural, while $\vdash^{\mathsf{at}}_k$ is not. In general, structurality can be imposed by restricting the operations in $\mathcal{F}$ to those such that, for all σ and all Δ, $\sigma(f(\Delta)) \subseteq f(\sigma(\Delta))$. This is not satisfied if $f = \mathsf{at}$, but it is satisfied if $f(\Delta) = \mathsf{sub}(\Delta)$, or $f(\Delta)$ is the set of all formulae of given bounded degree that can be built out of $\mathsf{sub}(\Delta)$.

Now, in [58] the following is proven:

Theorem 3.4.9 (Generalized SFP, [58]). *For every $f \in \mathcal{F}$, every intelim proof of φ from X (every intelim refutation of X), $\mathcal{T}$, can be transformed into an intelim proof of φ from X (an intelim refutation of X), $\mathcal{T}'$, such that for every S-formula ψ occurring in $\mathcal{T}'$,*

$$\psi^u \in f(X^u \cup \{\varphi^u\}) \cup \mathsf{sub}(X^u \cup \{\varphi^u\})$$

if $\mathcal{T}'$ is a proof of φ from X, or

[19] In the terminology of [59], the depth-bounded deducibility relations that can be characterized by simply limiting the nested applications of *PB* in the combined *KE*/*KI* system—that here we have called "intelim method"—are the *weak* k-depth consequence relations.

$$\psi^u \in f(X^u) \cup \mathsf{sub}(X^u)$$

if $\mathcal{T}'$ is a refutation of X.[20]

Of course, the above Theorem implies the usual SFP of the intelim method when $f = \mathsf{sub}$. So, the Theorem immediately suggests a decision procedure for k-depth deducibility: to determine if φ is k-depth deducible from a finite set X apply the intelim rules, together with PB up to a number k of times, in all possible ways starting from X and restricting to applications which preserve the SFP. If the resulting intelim tree is closed or φ occurs at the end of all its open branches, φ is k-depth deducible from X, otherwise it is not. On the basis of this, it can be shown that each $\vdash_k^f$ inherits the tractability of $\vdash_0$, yet the complexity of the decision procedure grows with k—and with the degree of the polynomial p that bounds the size of the virtual space defined by f. Specifically:

Theorem 3.4.10 ([59, 58]). *For each $f \in \mathcal{F}$ and each $k \in \mathbb{N}$, whether or not $X \vdash_k^f \varphi$ ($X \vdash_k^f$) can be decided in polynomial time.*

More specifically, when $f \trianglelefteq \mathsf{sub}$, the complexity of the decision problem is $O(n^{k+2})$, where $n = |X \cup \{\varphi\}|$ ($n = |X|$). In general, the complexity is $O(p(n)^{k+2})$ where p is a polynomial depending on f.[21] Thus, for each fixed k, $\vdash_k^f$ admits of a feasible decision procedure and thus in accordance with the Strong Manifestability condition. Remarkably, tractability here is a sort of "by-product" of the approach's informational interpretation of the connectives which, in turn, makes no direct reference to computational complexity questions. More specifically, the tractability of each approximation results from a notion of depth that applies to *single* proofs and refutations. The measure of depth does not hinges on computational complexity, but on the distinction between actual and virtual information; and so the tractability of k-depth consequence is derivative. A consequence of this is, for example, that depth-bounded deduction provides means to solve the problem of logical omniscience that seem to overcome Parikh's main objection against complexity-based approaches, according to which, "[t]he issue of computational complexity can only make sense for an infinite family of questions, whose answers may be undecidable or at least not in polytime. But for individual questions whose answers we do not know, the appeal to computational complexity misses the issue" [123, p. 462].

Now, the following result takes us to the issue of the semantics of the depth-bounded approximations to **CPL**:

[20] Clearly, when $f = \mathsf{sub}$, we obtain the usual SFP.

[21] Recall that, by definition, the virtual space is polynomially bounded.

Proposition 3.4.11 ([59]). *The 0-depth logic, $\langle \mathcal{L}, \vdash_0 \rangle$, cannot be characterized by any finite-valued deterministic matrix.*

However, as recalled below, it can be shown that the 0-depth logic can be characterized by a 3-valued non-deterministic matrix (Nmatrix).[22]

The classical meaning of the connectives is usually specified by the standard truth-functional semantics that fix the conditions under which a sentence is true or false in terms of the truth or falsity of its immediate constituents. In turn, as we recalled above, the classical *information-transcendent* notions of truth and falsity are assumed to obey Bivalence and Non-Contradiction. Such a way of fixing the meaning of a connective is in tune with the traditional view of inference as a truth-transmission device, but it is at odds with the view of logical inference as an information-processing device.[23] As argued in the introduction of the thesis, in order to comply with the latter view, a semantics based on informational notions is required. Moreover, in order to obtain models of less idealized (e.g., not logically omniscient) agents, it is required a semantics based on the notion of actual information, where the underlying notion of information is neutral w.r.t. truth and falsity. Thus, the primary notions of such a semantics are not classical truth and falsity, but *informational truth* and *informational falsity*; namely, *holding* the information that a sentence is true, respectively false. These notions do not satisfy the informational version of Bivalence: it certainly cannot be assumed that, for any A, either the agent holds the information that A is true or holds the information that A is false. On the other hand, when the depth-bounded approach is applied to **CPL**, it may be reasonably assumed that Non-contradiction lifts to the informational level: no agent can hold both the information that A is true and the information that A is false, as that would be deemed to amount to possessing no definite information about A.[24]

In turn, the truth-values 1 and 0 are used to respectively denote informational truth and falsity. When a formula A takes neither of these two defined truth-values,

[22]In [61, 59] the 0-depth logic is characterized in terms of other semantics; namely and respectively, *constraint-based semantics* and *modular semantics*. In [59], the latter is shown to be equivalent to the 3-valued non-deterministic semantics to be recalled below; whereas, in [56] it is pointed out that the former is essentially equivalent to the other two. We found non-deterministic semantics to be particularly suitable to carry out our investigation in this Thesis.

[23]In fact, the classical meaning of the connectives is overdetermined by the standard truth-functional semantics and this fact is probably the reason why standard proof-theoretic semantics for **CPL** are rather contrived. As recalled above, classical inferences are better construed as arising from the interplay between a weaker basic semantics for the connectives—provided by the intelim rules—and the purely structural Principles of Bivalence and Non-Contradiction.

[24]As mentioned in footnote 10, we shall drop this assumption when we apply the depth-bounded approach to paraconsistent logics.

$\tilde{\vee}$	1	0	$\bot$
1	$\{1\}$	$\{1\}$	$\{1\}$
0	$\{1\}$	$\{0\}$	$\{\bot\}$
$\bot$	$\{1\}$	$\{\bot\}$	$\{1, \bot\}$

$\tilde{\wedge}$	1	0	$\bot$
1	$\{1\}$	$\{0\}$	$\{\bot\}$
0	$\{0\}$	$\{0\}$	$\{0\}$
$\bot$	$\{\bot\}$	$\{0\}$	$\{0, \bot\}$

$\tilde{\neg}$	
1	0
0	1
$\bot$	$\bot$

$\tilde{\to}$	1	0	$\bot$
1	$\{1\}$	$\{0\}$	$\{\bot\}$
0	$\{1\}$	$\{1\}$	$\{1\}$
$\bot$	$\{1\}$	$\{\bot\}$	$\{1, \bot\}$

Table 3.7: 3N-tables

we say that its truth-value is *unknown*.[25] Accordingly, a *partial valuation* v for $\mathcal{L} = \{\vee, \wedge, \neg, \to\}$ is a partial function $v : F(\mathcal{L}) \longrightarrow \{0, 1\}$. We denote by $v(A) = \bot$ whenever v is *undefined* for A. It is technically convenient to treat $\bot$ as a third truth-value and so we interpret it as denoting a third primary notion: *ignorance*.[26] Thereby, we take the three truth-values as partially ordered by the relation $\preceq$, such that $x \preceq y$ (read "x is less defined than, or equal to, y") iff $x = \bot$ or $x = y$ for $x, y \in \{0, 1, \bot\}$. Thus, a *3-valuation* v for $\mathcal{L}$ is a (total) function $v : F(\mathcal{L}) \longrightarrow \{0, 1, \bot\}$. Now, we pick out from the set of all 3-valuations those which agree with the intended meaning of the connectives. We do this through the following Nmatrix, which conservatively extends the standard matrix of **CPL**:

Definition 3.4.12. Let $\mathcal{M}_3$ be an Nmatrix for $\mathcal{L}$, where $\mathcal{V} = \{1, 0, \bot\}$, $\mathcal{D} = \{1\}$ and the functions in $\mathcal{O}$ are defined by the 3N-tables in Tab. 3.7.

Remark 5. The part of each 3N-table which involves solely defined truth-values—i.e., 1 or 0—is respectively identical with a standard truth-table of **CPL**; let us refer to this part of each 3N-table as its *defined kernel*. Now, in each 3N-table, the entries involving $\bot$ are established by the corresponding defined kernel as follows: In case one component or two components of a complex formula are (respectively, is) $\bot$, we are left with a case of *underspecification* where all possible truth-values are allowed. Accordingly, when one or both arguments in a $\mathcal{M}_3$-function are (respectively, is) $\bot$,

[25] Originally, D'Agostino [58] referred to the case where a formula does not have a defined truth-value as a case of "informational indeterminacy". However, here we find the term "unknown" more appropriate.

[26] Related to the previous footnote, the reference to this third primary notion is also mentioned here for the first time.

the value of the function must include all the truth-values that are *compatible* with the corresponding defined kernel. Now, a truth-value—w.r.t. the value of a $\mathcal{M}_3$-function involving $\bot$ in its arguments—is *compatible* with the corresponding defined kernel if, on the basis of the very defined kernel, such a truth-value does not imply that one particular argument of those which are $\bot$ had a defined truth-value instead. In turn, that a $\mathcal{M}_3$-function has as value not a singleton—but a set of alternative truth-values—means that the truth-value of the compound formula is not uniquely determined by the truth-values of its immediate subformulae, but can be either of the truth-values shown.

To illustrate these ideas take the truth-table for $\vee$: If one of the disjuncts is 1, then—regardless of the truth-value of the other disjunct, including $\bot$—the disjuntion is (uniquely) 1. Now, if disjunct A is 0 and disjunct B is $\bot$, the disjunction cannot be 1 nor 0 because that would imply that disjunct B respectively was already 1 or 0; yet, the disjunction may well be $\bot$. In turn, when both disjuncts are $\bot$, the only excluded truth-value for the disjunction is 0 as it would imply that both disjuncts were already 0; however, 1 and $\bot$ are both admissible truth-values for the disjunction. The latter case amounts to the situation where the disjunction may be either 1 or $\bot$ depending on whether or not the agent holds the *additional* information that at least one disjunct must be true.[27]

Remark 6. The close connection between Quine's dispositional theory of the "primitive" meaning of the connectives and the notions underlying $\mathcal{M}_3$—in turn expressing their informational meaning—should be apparent. That the agent holds the information that A is true (false) is plausibly tantamount to the agent being in the disposition to assent (respectively, dissent) to A. Indeed, the 3N-tables associated with $\mathcal{M}_3$ can be obtained from Quine's incomplete 3-valued tables as follows: the truth-value 1 (standing for informational truth) corresponds to Quine's "assent", 0 (representing informational falsity) corresponds to Quine's "dissent", and $\bot$ (standing for unknown) corresponds to Quine's "abstain". In turn, Quine's "blind spots" correspond to the entries where there are two alternative possible truth-values, indicating that the truth-value of the compound sentence is not uniquely determined by the truth-values of its component sentences but can be either of the two truth-values shown; which is in tune with Quine's considerations. Finally, the material conditional can be defined as usual, $A \to B := \neg A \vee B$.

Now, as mentioned in Chapter 2, Quine's semantics was independently re-proposed by Crawford and Etherington, who used it for investigating tractable inference and

[27]For a concrete example of the latter case, think of your computer telling you that the user or the password is wrong; you may do not know which is wrong, but you do know that at least one is so.

for providing—without proof—a semantic characterization of an extension of unit-resolution. Crawford's and Etherington's account of the inferences justified only by the corresponding truth-tables is restricted to formulae in negation normal form; whereas their semantics for classical valid inferences that are not justified solely by those tables is based on a classical *reductio ad absurdum* that applies only to formulae in clausal form. As recalled in what follows, the depth-bounded approach supports Crawford's and Etherington's intuition that their 3-valued non-deterministic semantics may become the basic foundational tool for a general theory of tractable approximations to **CPL**, but also show that the scope of that semantics is much wider than what they envisaged, in that it is relevant to any logical formalism with no syntactic restrictions.

Definition 3.4.13. A *3N-valuation* is a 3-valuation v s.t. for all A, B:

1. $v(\neg A) = \tilde{\neg}(v(A))$;
2. $v(A \circ B) \in \tilde{\circ}(v(A), v(B))$.

Where $\circ$ is $\vee$, $\wedge$ or $\rightarrow$.

Remark 7. A 3N-valuation can be seen as describing an *information state* that is closed under the implicit information that depends only on the informational meaning of the connectives.[28] This is information that the agent holds and with which she can operate, in the precise sense that she has a feasible procedure to decide, for every A, whether the information that A is true, or the information that A is false, or neither of them actually belongs to her information state.

In what follows we keep using S-formulae for the sake of uniformity, but recall that, in the case of **CPL**, such an use is optional. We say that a 3N-valuation v *satisfies* an S-formula $\mathsf{T}\,A$ if $v(A) = 1$ and an S-formula $\mathsf{F}\,A$ if $v(A) = 0$. Thus:

Definition 3.4.14. For all X and φ,

- φ is a *0-depth consequence* of X, $X \vDash_0 \varphi$, if for every 3N-valuation v, v satisfies φ whenever v satisfies all the S-formulae in X.
- X is *0-depth inconsistent*, $X \vDash_0$, if there is no 3N-valuation v such that v satisfies all the S-formulae in X.

Proposition 3.4.15 ([59, 58]). *For every X, φ,*

[28] Intuitively, an information state represents the total information that an agent holds, either explicitly or implicitly, on the basis of the intended meaning of the connectives.

$$X \vDash_0 \varphi \textit{ iff } X \vdash_0 \varphi.$$

The 0-depth consequence relation $\vDash_0$ is a subsystem of **CPL** obtained by replacing the notion of *possible world* with our weaker notion of *information state*; the latter described by a 3N-valuation. Put differently, given that 3N-valuations are intended to describe information states, rather than possible worlds, they are usually partial: within a given information state some formulae may be assigned neither 1 nor 0, representing the agent's ignorance about their truth-value. Again, the 0-depth logic is simply the system of deductive reasoning with no virtual information. The classical inferences that cannot be justified solely by the meaning of the connectives require the incorporation of information that is not even implicitly contained in the current information state. Concretely, the k-depth logics, $k > 0$, require the simulation of *virtual extensions* of the current information state; extensions that are formally defined through the notion of *refinement*:

Definition 3.4.16. Let v, w be 3N-valuations. Then, w is a *refinement* of v, $v \sqsubseteq w$, if $v(A) \preceq w(A)$ for all A.

Thus, 3N-valuations are partially ordered by the usual *refinement relation*, $\sqsubseteq$.

Analogously to k-depth deducibility, the notion of k-depth consequence depends not only on the depth at which the use of virtual information is recursively allowed, but also on the virtual space. Thus the following definitions mimic Defs. 3.4.6:

Definitions 3.4.17. For all X, φ, and for all $f \in \mathcal{F}$,

- $X \vDash_0^f \varphi$ iff $X \vDash_0 \varphi$;
- for $k > 0$, $X \vDash_k^f \varphi$ iff $X \cup \{\mathsf{T}\,A\} \vDash_{k-1}^f \varphi$ and $X \cup \{\mathsf{F}\,A\} \vdash_{k-1}^f \varphi$ for some $A \in f(X^u \cup \{\varphi^u\})$.

Analogously to k-depth refutability, the above definitions cover also the case of k-depth inconsistency by assuming $X \vDash_k^f$ as equivalent to $X \vDash_k^f \varphi$ for all φ. When $X \vDash_k^f \varphi$ ($X \vDash_k^f$) we say that φ is a *k-depth consequence* of X (X is *k-depth inconsistent*) *over the f-bounded virtual space.*

In the above Definitions, the transition from $\vDash_k^f$ to $\vDash_{k+1}^f$ is determined by simulating refinements of the current information state in which the truth-value of some A in the f-bounded virtual space is defined, and checking that in either case φ is satisfied at the immediately lower depth. That use of a defined truth-value for A, which is not even potentially contained in the current information state, is what we call virtual information.

Now, given Prop. 3.4.15 and the close correspondence between Defs. 3.4.6 and 3.4.17, it is far from surprising that:

Proposition 3.4.18. *For all X, φ and all $f \in \mathcal{F}$,*

$$X \vDash_k^f \varphi \text{ iff } X \vdash_k^f \varphi.$$

Now, we shall abuse of the same relation symbols $\vDash_0$ and $\vDash_j^k$ to respectively denote both 0-depth consequence and inconsistency, and k-depth consequence and inconsistency (over the f-bounded virtual space). Given that $\vDash_0$ is monotonic, $\vDash_j^f \subseteq \vDash_k^f$ whenever $j \leq k$. The transition from $\vDash_{k-1}^f$ to $\vDash_k^f$ corresponds to an increase in the depth at which the nested use of virtual information—restricted to formulae in the virtual space defined by f—is allowed. Besides, note that $\vDash_j^{f_1} \subseteq \vDash_k^{f_2}$ whenever $f_1 \trianglelefteq f_2$. Then, it is not difficult to show that each relation $\vDash_k^f$ is an approximation to full **CPL** in that the latter is the limit of the sequence of relations $\vDash_k^f$ as $k \longrightarrow \infty$:

Proposition 3.4.19 ([59, 58]). *For every f, the relation $\vDash_\infty^f = \bigcup_{k \in \mathbb{N}} \vDash_k^f$ is the consequence relation of **CPL**.*

3.4.2 Depth-bounded *KE* deduction

Given that both *KE* and *KI* are complete for **CPL**, and that in both the operational rules are all linear and the only branching rule is *PB*, the application of the depth-bounded approach to **CPL**—in terms of bounding the applications of *PB*—can be based on either of them. In what follows, we briefly discuss how the approach to **CPL** can be based on *KE*; the case of *KI* being analogous.

Although arguably less natural than the proof-theoretical basis constituted by the intelim method, the basis constituted by *KE* allows us to define an analogous hierarchy of approximations to **CPL**. So, the intuitions and notions of the approach can be, respectively, related and defined according to such a hierarchy. To begin with, we take the elimination rules in Tab. 3.1 as fixing the meaning of the connectives only in terms of actual information. Specifically, we interpret S-formulae and those rules in the same informational terms we did for the intelim method. Thus, we interpret the elimination rules as involving only actual information and so they characterize a corresponding 0-depth logic. For recall that the latter is intended to be the logic of the inferences that can be drawn by using only actual information, and whose validity can be determined on the sole basis of the informational meaning of the connectives. Accordingly, as we shall recall below, such a logic is tractable.

The elimination rules generate *eliminative sequences*. Namely, finite sequences $(\varphi_1, ..., \varphi_n)$ of formulae such that, for every $i = 0, ..., n$, either φ_i is an assumption or it is the conclusion of the application of an elimination rule to preceding S-formulae. The elimination rules are not complete for **CPL** but precisely, as we shall show below, just for the 0-depth (basic) logic in the hierarchy of approximations. As before, completeness for full **CPL** is obtained by adding only *PB*. With the addition of *PB*, deductions are represented by downward-growing trees. Naturally, we interpret *PB* in the same informational terms as above. So, each application of *PB* stands for the introduction of virtual information about the truth or falsity of a formula, and the S-formulae $\mathsf{T}\,A$ and $\mathsf{F}\,A$ introduced by an application of *PB* are called *virtual assumptions*. As before, the nested applications of *PB* leads to defining an infinite hierarchy of tractable depth-bounded approximations to **CPL**. Sepcifically, the k-depth logics, $k > 0$, are the logics of the classical inferences whose validity requires the introduction of virtual information. As before, a key intuition is that the more virtual information needs to be invoked via *PB*, the harder the inference is for the agent. The nested applications of *PB* provide a sensible measure of inferential depth. Thus, the obtained hierarchy can be naturally related to the inferential power of the agents and, as we shall show below, admit of a 3-valued non-deterministic "informational" semantics.

Now, *KE* allows us to define a direct as well as an indirect notion of deducibility; the latter via refutability.

Definitions 3.4.20. For all X, φ,

- A *direct KE-proof of* φ *from* X is a *KE*-tree $\mathcal{T}$ for X such that φ occurs in all open branches of $\mathcal{T}$.
- A *KE-refutation* of X is a *KE*-tree $\mathcal{T}$ for X such that every branch of $\mathcal{T}$ is closed.

Note that, according to the above definitions, every *KE*-refutation of X is, simultaneously, a direct *KE*-proof of φ from X, for every φ. This is so because there are no open branches and the condition that φ occurs at the end of all open branches is vacuously satisfied.

Thus, we say that:

Definitions 3.4.21. The *depth* of a *KE*-tree $\mathcal{T}$ is the maximum number of virtual assumptions occurring in a branch of $\mathcal{T}$. A *KE*-tree $\mathcal{T}$ is a *k-depth direct KE-proof of* φ *from* X (a *k-depth KE-refutation of* X) if $\mathcal{T}$ is a direct *KE*-proof of φ from X (*KE*-refutation of X) and $\mathcal{T}$ is of depth k.

Note that a 0-depth *KE*-tree is nothing but an eliminative sequence. Thus:

Definition 3.4.22. For all X, φ and Γ, A,

- φ is *0-depth directly KE-deducible* from X, $X \vdash_0 \varphi$, if there is a 0-depth direct *KE*-proof of φ from X.

- X is *0-depth KE-refutable*, $X \vdash_0$, if there is a 0-depth *KE*-refutation of X.

- A is *0-depth KE-deducible* from Γ, $\Gamma \vdash_{KE(0)} A$, if $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\} \vdash_0$.

Notation 3.4.23. Along the thesis we shall abuse of the same relation symbols $\vdash_k$, $k \geq 0$, to denote different, yet analogous, relations. However, the context will always rule out any ambiguity and make clear in what sense those relations are analogous. In the present context, we use the same relation symbol $\vdash_0$ to denote both 0-depth direct *KE*-deducibility and 0-depth *KE*-refutability. Further, since the notation *KE*(k) was introduced in [65] to denote each subsystem of *KE* obtained by allowing at most k nested analytic applications of *PB*, and the corresponding deducibility relations correspond exactly to what here we call "k-depth *KE*-deducibility", we shall use the subindex *KE*(k) in the corresponding relation symbols $\vdash_{KE(k)}$, $k \geq 0$.

Given that the set of elimination rules is a subset of the intelim rules, it is far from surprising that:

Proposition 3.4.24. *The relation* $\vdash_0$ *based on KE is a structural Tcr.*

Besides, it is not surprising that $\vdash_0$ has no tautologies. As expected—analogously to the approximations to **CPL** defined via the intelim method—tautologies appear only at depths $k > 0$, when the use of virtual information is allowed, and the set of provable tautologies increases with k.

Now, it follows from Defs. 3.4.20, 3.4.21 and 3.4.22, that if $\mathsf{T}\,\Gamma \vdash_0 \mathsf{T}\,A$, then $\Gamma \vdash_{KE(0)} A$. Nonetheless, the converse of the latter does not hold. This because the relation $\vdash_{KE(0)}$ is stronger than the relation $\vdash_0$ (regardless whether based on *KE* or on the intelim method). For example, $\mathsf{T}\,\{p \vee q, p \rightarrow r, q \rightarrow r\} \nvdash_0 \mathsf{T}\,r$ (based on either *KE* or the intelim method), but $\{p \vee q, p \rightarrow r, q \rightarrow r\} \vdash_{KE(0)} r$. Indeed, unlike $\vdash_0$ (based on *KE* or the intelim method), $\vdash_{KE(0)}$ *does* has tautologies; for instance, $\emptyset \vdash_{KE(0)} A \vee \neg A$. However, as expected, not all classical tautologies are provable under $\vdash_{KE(0)}$; for example, $\emptyset \nvdash_{KE(0)} (A \vee (B \wedge C)) \rightarrow ((A \vee B) \wedge (A \vee C))$. Again, the set of provable tautologies increases with k.

Now, remarkably, unlike the relation $\vdash_0$ (again, based on either *KE* or the intelim method), $\vdash_{KE(0)}$ is not a Tcr:

Proposition 3.4.25. *The relation* $\vdash_{KE(0)}$ *satisfies reflexivity, monotonicity but not cut.*

Proof. For reflexivity, suppose that $A \in \Gamma$. Then, there is a 0-depth *KE*-refutation of $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$ and so $\Gamma \vdash_{KE(0)} A$. For monotonicity, suppose $\Gamma \vdash_{KE(0)} A$. Then, there is a 0-depth *KE*-refutation of $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$, and so there is also a 0-depth *KE*-refutation of $\mathsf{T}\,\Gamma \cup Y \cup \{\mathsf{F}\,A\}$. Therefore, $\Gamma \cup Y^u \vdash_{KE(0)} A$. On the other hand, $\vdash_{KE(0)}$ fails to satisfy cut since it may well be that there are 0-depth *KE*-refutations of $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$ and $\mathsf{T}\,\Gamma \cup \{\mathsf{T}\,A, \mathsf{F}\,B\}$, but there is no 0-depth *KE*-refutation of $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,B\}$. For example, take $\Gamma := \{p \vee q, p \to r, q \to r, r \to s, r \to t\}$, $A := r$ and $B := s \wedge t$. □

On the other hand, it is easy to see that $\vdash_{KE(0)}$ is structural.

Now, the tractability of the logics $\langle \mathcal{L} \vdash_0 \rangle$ and $\langle \mathcal{L}, \vdash_{KE(0)} \rangle$ trivially follows from the tractability of the 0-depth logic based on the intelim method (Theorem 3.4.5). In fact, the following was early shown:

Theorem 3.4.26 ([65]). *Let b be a branch of a KE-tree containing m nodes, each of which is an occurence of a formula of degree d_m. The task of saturating b can be performed in time $O(n^2)$, where $n = \Sigma_{m \in b} d_m$.*[29]

Then, as above, let $|\varphi|$ denote the size of an S-formula φ (i.e., the total number of occurrences of symbols in φ), and the size of a finite set of S-formulae Y be defined as $\Sigma_{\varphi \in Y} |\varphi|$ and denoted by $|Y|$. So we have:

Corollary 3.4.27. *Whether or not $X \vdash_0 \varphi$ ($\Gamma \vdash_{KE(0)} A$) can be decided in time $O(n^2)$, where $n = |X \cup \{\varphi\}|$ ($n = |\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}|$).*

As mentioned in [65], the set for which $\vdash_{KE(0)}$ is complete includes the Horn-clause fragment of **CPL**. However, $\vdash_{KE(0)}$ is not restricted to clausal form formulae.

Now, as above, the k-depth logics are defined in terms of the nested use of virtual information—via the applications of *PB*—allowed up to a number of times k. Again, these logics depend not only on the depth at which the use of virtual information is recursively allowed, but also on the virtual space; which is defined exactly as before. Accordingly, the set $\mathcal{F}$ of all functions f on the finite subsets of $F(\mathcal{L})$ is also defined exactly as above. Thus,

Definitions 3.4.28. For all X, φ, Γ, A, and for all $f \in \mathcal{F}$,

- $X \vdash_0^f \varphi$ iff $X \vdash_0 \varphi$;
- for $k > 0$, $X \vdash_k^f \varphi$ iff $X \cup \{\mathsf{T}\,A\} \vdash_{k-1}^f \varphi$ and $X \cup \{\mathsf{F}\,A\} \vdash_{k-1}^f \varphi$ for some $A \in f(X^u \cup \{\varphi^u\})$;

[29] It is said that a branch b of a *KE*-tree is *saturated* if, whenever the premise (or the premises) of an elimination rule belongs (belong) to b, then its conclusion also belongs to b. Moreover, recall that in the case of **CPL** the difference between the signed and unsigned versions of *KE* is immaterial.

- $\Gamma \vdash^f_{KE(0)} A$ iff $\Gamma \vdash_{KE(0)} A$;
- for $k > 0$, $\Gamma \vdash^f_{KE(k)} A$ iff $\Gamma \cup \{B\} \vdash^f_{KE(k-1)} A$ and $\Gamma \cup \{\neg B\} \vdash^f_{KE(k-1)} A$ for some $B \in f(\Gamma \cup \{A\})$.

When $X \vdash^f_k \varphi$, we say that φ is *directly KE-deducible at depth k from X over the f-bounded virtual space*. Note that the above definitions cover also the case of k-depth *KE*-refutability by assuming $X \vdash^f_k$ as equivalent to $X \vdash^f_k \varphi$ for all φ. When $X \vdash^f_k$, we say that X is *KE-refutable at depth k over the f-bounded virtual space*. In turn, when $\Gamma \vdash^f_{KE(k)} A$, we say that A is *KE-deducible at depth k from* Γ *over the f-bounded virtual space.*

It follows immediately from Defs. 3.4.21 and 3.4.28 that:

Proposition 3.4.29. *For all X, φ, Γ, A, and for all $f \in \mathcal{F}$,*

- *$X \vdash^f_k \varphi$ iff there is a k-depth direct KE-proof of φ from X such that all its PB-formulae are in $f(X^u \cup \{\varphi^u\})$;*
- *$X \vdash^f_k$ iff there is a k-depth KE-refutation of X such that all its PB-formulae are in $f(X^u)$;*
- *$\Gamma \vdash^f_{KE(k)} A$ iff there is a k-depth KE-refutation of $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$ such that all its PB-formulae are in $f(\Gamma \cup \{A\})$.*

Notation 3.4.30. We shall abuse of the same relation symbol $\vdash^f_k$ to denote both k-depth direct *KE*-deducibility and k-depth *KE*-refutability.

Given that the set of elimination rules together with *PB* is a subset of the intelim rules together with *PB*, it is far from surprising that, unlike $\vdash_0$ (based on *KE* or on the intelim method), the relations $\vdash^f_k$ are not Tarskian. However, they satisfy reflexivity, monotonicity, and the following version of cut:

Depth-bounded cut: If $X \vdash^f_j \varphi$ and $X \cup \{\varphi\} \vdash^f_k \psi$, then $X \vdash^f_{j+k} \psi$.

Besides, as the relations $\vdash^f_k$ based on the intelim method, the analogous relations based on *KE* may not be structural in that structurality depends on the function f that defines the virtual space. Again, structurality can be imposed by restricting the operations in $\mathcal{F}$ to those such that, for all σ and all Δ, $\sigma(f(\Delta)) \subseteq f(\sigma(\Delta))$.

In turn, it easy to check that the relations $\vdash^f_{KE(k)}$ are reflexive and monotonic. However, of course, they do not satisfy cut (since we observed that $\vdash_{KE(0)}$ already fails cut). Further, the relations $\vdash^f_{KE(k)}$ may not be structural in that, again, structurality depends on the function f that defines the virtual space. Once more, structurality can be imposed by restricting the operations in $\mathcal{F}$.

As expected, the tractability of the k-depth bounded logics based on *KE* follows from the tractability of the analogous logics based on the intelim method (Theorem 3.4.10). So, as before, each $\vdash_k^f$ and $\vdash_{KE(k)}^f$ respectively inherits the tractability of $\vdash_0$ and $\vdash_{KE(0)}$. However, the complexity of the decision procedure grows with k—and with the degree of the polynomial p bounding the size of the virtual space defined by f.

Theorem 3.4.31 ([65]). *For each $f \in \mathcal{F}$ and each $k \in \mathbb{N}$, whether or not $X \vdash_k^f \varphi$ ($\Gamma \vdash_{KE(k)}^f A$) can be decided in polynomial time.*

We refer to [65] in the above Theorem because it was essentially stated there, although only for $f = \mathsf{sub}$. Given Theorem 3.4.9, the above holds for every f.

Clearly, the set of tautologies for which $\vdash_{KE(k)}^f$ and $\vdash_k^f$ are complete tends to the set of all classical tautologies, TAUT, as $k \longrightarrow \infty$. As remarked in [65], the crucial point is that low-degree k-depth logics cover large fragments of **CPL**, and are powerful enough for a wide range of applications. Besides, the source of the complexity of proving a tautology in the systems is clearly identified.

Now, semantics for the logics $\langle \mathcal{L}, \vdash_k^f \rangle$, $k \geq 0$, based on *KE* can be obtained along the lines of the semantics of the analogous logics based on the intelim method. However, semantics for the logics $\langle \mathcal{L}, \vdash_{KE(k)}^f \rangle$, $k \geq 0$, can be given only indirectly via inconsistency. As before, the primary notions of the semantics are *informational truth*, *informational falsity* and *ignorance*; interpreted as above and respectively denoted by the truth-values 1, 0, and $\bot$. We take these three truth-values as partially ordered by the relation $\preceq$, such that $x \preceq y$ (read "x is less defined than, or equal to, y") iff $x = \bot$ or $x = y$ for $x, y \in \{0, 1, \bot\}$. Thereby, a *3-valuation* v for $\mathcal{L}$ is a function $v : F(\mathcal{L}) \longrightarrow \{0, 1, \bot\}$. In turn, we pick out from the set of all 3-valuations those which agree with the intended meaning of the connectives via the following Nmatrix:

Definition 3.4.32. Let $\mathcal{M}'_3$ be an Nmatrix for $\mathcal{L}$, where $\mathcal{V} = \{1, 0, \bot\}$, $\mathcal{D} = \{1\}$ and the functions in $\mathcal{O}$ are defined by the 3N'-tables in Tab. 3.8.

Remark 8. Unlike the 3N-tables, the 3N'-tables above do not have a "proper" *defined* kernel. Namely, the yielded values of the $\mathcal{M}'_3$-functions when the argument(s) are defined do not include only defined truth-values. This corresponds to the fact of our informational interpretation of *KE* according to which, for example, if an agent holds both the information that A is true and that B is true ($\mathsf{T}\,A$ and $\mathsf{T}\,B$), it *may* well

$\tilde{\vee}$	1	0	$\bot$
1	$\{1,\bot\}$	$\{1,\bot\}$	$\{1,\bot\}$
0	$\{1,\bot\}$	$\{0,\bot\}$	$\{\bot\}$
$\bot$	$\{1,\bot\}$	$\{\bot\}$	$\{1,\bot\}$

$\tilde{\wedge}$	1	0	$\bot$
1	$\{1,\bot\}$	$\{0,\bot\}$	$\{\bot\}$
0	$\{0,\bot\}$	$\{0,\bot\}$	$\{0,\bot\}$
$\bot$	$\{\bot\}$	$\{0,\bot\}$	$\{0,\bot\}$

$\tilde{\neg}$	
1	$\{0,\bot\}$
0	$\{1,\bot\}$
$\bot$	$\{\bot\}$

$\tilde{\to}$	1	0	$\bot$
1	$\{1,\bot\}$	$\{0,\bot\}$	$\{\bot\}$
0	$\{1,\bot\}$	$\{1,\bot\}$	$\{1,\bot\}$
$\bot$	$\{1,\bot\}$	$\{\bot\}$	$\{1,\bot\}$

Table 3.8: 3N'-tables

be that she either does not hold information about the truth-value of $A \wedge B$ (recall that *KE* has no introduction rules), or holds that it is also true ($\mathsf{T}\, A \wedge B$). However, when holding both that A is true and that B is true, it cannot be the case that she holds the information that $A \wedge B$ is false ($\mathsf{F}\, A \wedge B$).

It is easy to see that the other entries of the 3N'-tables involving only defined truth-values as arguments are justified in a similar way. Let us refer to the part consisting of those entries in each 3N'-table as its *kernel*. Then, the entries in the tables involving $\bot$ are established in an entirely analogous way to that in which they were established for the 3N-tables. Namely, if one or two arguments are (is) $\bot$, we are left with a case of underspecification and, thus, the value of the function must include all the truth-values that are compatible with the corresponding kernel. As before, a truth-value—w.r.t. the value of a $\mathcal{M}_3'$-function involving $\bot$ in its arguments—is compatible with the corresponding kernel if, on the basis of the very kernel, such a truth-value does not imply that one particular argument of those which are $\bot$ had a defined truth-value instead.

Definition 3.4.33. A *3N'-valuation* is a 3-valuation v s.t. for all A, B:

1. $v(\neg A) \in \tilde{\neg}(v(A))$;
2. $v(A \circ B) \in \tilde{\circ}(v(A), v(B))$.

Where $\circ$ is $\vee$, $\wedge$ or $\to$.

Remark 9. As before, a 3N'-valuation can be seen as describing an *information state* that is closed under the implicit information that depends only on the informational meaning of the connectives. This is information that the agent holds and with which she can operate in that, as recalled below, she has a feasible procedure to decide, for every A, whether the information that A is true, or the information that A is false, or neither of them actually belongs to her information state.

Again, we keep using S-formulae but recall that such an use is optional in the case of **CPL**. We say that a 3N'-valuation v *satisfies* an S-formula $\mathsf{T}\,A$ if $v(A) = 1$ and an S-formula $\mathsf{F}\,A$ if $v(A) = 0$. Thus:

Definition 3.4.34. For all X, φ, Γ, A,

- φ is a *0-depth direct consequence of* X, $X \vDash_0 \varphi$, if for every 3N'-valuation v, v satisfies φ whenever v satisfies all the S-formulae in X;
- X is *0-depth inconsistent*, $X \vDash_0$, if there is no 3N'-valuation v satisfying all the S-formulae in X;
- A is a *0-depth consequence* of Γ, $\Gamma \vDash_{KE(0)} A$, if $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\} \vDash_0$.

Proposition 3.4.35. *For all X,*

$$X \vDash_0 \textit{ iff } X \vdash_0.$$

Proof. Obviously, any set of S-formulae for which the closure condition obtains is unsatisfiable. Besides, soundness of the elimination rules can be immediately verified by inspection of the 3N'-tables. For example, if an agent holds the information that $A \wedge B$ is false (the truth-value of $A \wedge B$ is 0) and the information that A is true (the truth-value of A is 1), then she holds the information that B is false, since the other possible two truth-values are ruled out by the table for $\wedge$.

As for completeness, suppose that $X \nvdash_0$, i.e., X is not 0-depth *KE*-refutable. We show that X is 0-depth consistent. Now, consider the set $X^* = \{\psi \mid X \vdash_0 \psi\}$. (Note that X^* is finite, because conclusions of the elimination rules are always less complex than the corresponding major premise.) Since X is not 0-depth *KE*-refutable, for no formula A, $\mathsf{T}\,A$ and $\mathsf{F}\,A$ are both in X^*. Then, it is easy to verifty that the function v defined as follows:

$$v(A) = \begin{cases} 1 & \text{if } \mathsf{T}\,A \in X^* \\ 0 & \text{if } \mathsf{F}\,A \in X^* \\ \bot & \text{otherwise} \end{cases}$$

is a 3N'-valuation, i.e., it agrees with the 3N'-tables. Here we just outline three typical cases.

1. Suppose $v(A) = v(B) = \bot$. Then $\mathsf{F}\, A \vee B \notin X^*$. Otherwise, if $\mathsf{F}\, A \vee B \in X^*$, then by definition of X^* and elimination rules for $\vee$, $\mathsf{F}\, A$ and $\mathsf{F}\, B$ should also be in X^*; therefore, by definition of v, $v(A) = v(B) = 0$, against our assumption. Hence $v(A \vee B) \neq 0$. Moreover, $\mathsf{T}\, A \vee B$, may or may not belong to X^*, and so $v(A \vee B) = 1$ or $V(A \vee B) = \bot$.

2. Suppose $v(A) = v(B) = 0$. Then, by definition of v, $\mathsf{F}\, A$ and $\mathsf{F}\, B$ are both in X^*. So, $\mathsf{T}\, A \vee B$ does not belong to X^*; otherwise by the corresponding elimination rules for $\vee$, each of the pairs $\{\mathsf{T}\, A, \mathsf{F}\, A\}$ and $\{\mathsf{T}\, B, \mathsf{F}\, B\}$ should be in X^*, against the hypothesis that X is not 0-depth *KE*-refutable (recall again that *KE* has no introduction rules). On the other hand, $\mathsf{F}\, A \vee B$ may or may not belong to X^*. So, $v(A \vee B) = 0$ or $v(A \vee B) = \bot$.

3. Suppose $v(A) = \bot$ and $v(B) = 0$. Then, by definition of v, $\mathsf{F}\, B$ is in X^*. So, $\mathsf{T}\, A \vee B \notin X^*$. Otherwise, if $\mathsf{T}\, A \vee B \in X^*$, by definition of X^* and one elimination rule for $\vee$, $\mathsf{T}\, A$ should be in X^*; therefore, by definition of v, $v(A) = 1$, against our assumption. Moreover, $\mathsf{F}\, A \vee B \notin X^*$. Otherwise, if $\mathsf{F}\, A \vee B \in X^*$, by definition of X^* and one elimination rule for $\vee$, $\mathsf{F}\, A$ should be in X^*; therefore, by definition of v, $v(A) = 0$, against our assumption. Hence, $v(A \vee B) = \bot$.

Now, observe that: (i) $\psi \in X^*$ for all $\psi \in X$ and so, by definition of v, v satisfies all $\psi \in X$. This is shown by induction on the degree of A such that $\mathsf{S}A \in X^*$, where S is a variable ranging over $\{\mathsf{T}, \mathsf{F}\}$.

- Base case: Suppose $A := p$. Then, since for no A, $\mathsf{T}\, A$ and $\mathsf{F}\, A$ are both in X^*, and by definition of v, v satisfies $\mathsf{S}p$.

- Inductive hypothesis: Suppose that if A has degree n, then v realizes $\mathsf{S}A$.

- Inductive step: Let A have degree $n+1$, then we have several cases depending on whether $\mathsf{S} = \mathsf{T}$ or $\mathsf{S} = \mathsf{F}$, and on the logical form of A. We discuss only the case $\mathsf{F}\, A$, $A := B \wedge C$, the other cases being similar. Thus, if $\mathsf{T}\, B$ is also in X^* then, by definition of X^* and the corresponding elimination rule for $\wedge$, $\mathsf{F}\, C \in X^*$. Hence, by inductive hypothesis, v satisfies both $\mathsf{T}\, B$ and $\mathsf{F}\, C$. This, together with $\mathsf{F}\, B \wedge C \in X^*$, imply that v satisfies $\mathsf{F}\, B \wedge C$.

Therefore, there is a 3N'-valuation that satisfies all the S-formulae in X and so X is not 0-depth inconsistent, $X \nvDash_0$. □

Corollary 3.4.36. *For all X, φ,*

$$X \vDash_0 \varphi \text{ iff } X \vdash_0 \varphi.$$

Proof. Suppose $X \nvdash_0 \varphi$. Then X is not 0-depth refutable. Now, consider the set $X^* = \{\psi \mid X \vdash_0 \psi\}$. Proceed exactly as in the previous proof. Finally, observe that: (i) $\psi \in X^*$ for all $\psi \in X$ and so, by definition of v, v satisfies all $\psi \in X$; (ii) by hypothesis that $X \nvdash_0 \varphi$, $\varphi \notin X^*$ and so v does not satisfy φ. Hence $X \nvDash_0 \varphi$. □

The logic $\langle \mathcal{L}, \vDash_0 \rangle$ based on *KE* and the logic $\langle \mathcal{L}, \vDash_{KE(0)} \rangle$ are systems of deductive reasoning with no virtual information. For the classical inferences that cannot be justified solely by the meaning of the connectives, we need to incorporate information that is not even implicitly contained in the current information state. So, as before, the k-depth logics, $k > 0$ require the simulation of virtual extensions of the current information state. This extensions are formalized through the following notion:

Definition 3.4.37. Let v, w be 3N'-valuations. Then, w is a *refinement* of v, $v \sqsubseteq w$, if $v(A) \preceq w(A)$ for all A.

So, 3N'-valuations are partially ordered by the usual *refinement relation*, $\sqsubseteq$.

Analogously to k-depth (direct and indirect) *KE*-deducibility, $k > 0$, the notions of k-depth consequence depends not only on the depth at which the use of virtual information is recursively allowed, but also on the virtual space. Thereby, the following definitions mimic Defs. 3.4.28:

Definitions 3.4.38. For all X, φ, Γ, A, and for all $f \in \mathcal{F}$,

- $X \vDash_0^f \varphi$ iff $X \vDash_0 \varphi$;
- for $k > 0$, $X \vDash_k^f \varphi$ iff $X \cup \{\mathsf{T}\,A\} \vDash_{k-1}^f \varphi$ and $X \cup \{\mathsf{F}\,A\} \vDash_{k-1}^f \varphi$ for some $A \in f(X^u \cup \{\varphi^u\})$;
- $\Gamma \vDash_{KE(0)}^f A$ iff $\Gamma \vDash_{KE(0)} A$;
- for $k > 0$, $\Gamma \vdash_{KE(k)}^f A$ iff $\Gamma \cup \{B\} \vdash_{KE(k-1)}^f A$ and $\Gamma \cup \{\neg B\} \vdash_{KE(k-1)}^f A$ for some $B \in f(\Gamma \cup \{A\})$.

When $X \vDash_k^f \varphi$, we say that φ is a *direct k-depth consequence of X over the f-bounded virtual space*. Observe that the above definitions cover also the case of k-depth inconsistency by assuming $X \vDash_k^f$ as equivalent to $X \vDash_k^f \varphi$ for all φ. When $X \vDash_k^f$, we say that X is *k-depth inconsistent over the f-bounded virtual space*. In turn, when $\Gamma \vDash_{KE(k)}^f A$, we say that A is a *k-depth consequence of Γ over the f-bounded virtual space*.

Now, the next proposition follows from the fact that *KE* with unbounded k—i.e., *KE* with an arbitrary number of applications of PB—is sound and complete for full **CPL**:

Proposition 3.4.39. *For all X, φ, and all $f \in \mathcal{F}$,*

$$X \vDash_k^f \varphi \textit{ iff } X \vdash_k^f \varphi.$$

The depth-bounded approximations based on *KE* are arguably less natural than the analogous approximations based on the intelim method in at least three respects: (i) Using both introduction and elimination rules allows for more natural and shorter proofs—although not essentially shorter because *KE* and *KI* can linearly simulate each other. (ii) Using both types of rules reduces the number of applications of *PB* that, as stated above, is key for our meausre of the depth of an inference. (iii) The non-deterministic semantics for the basic (0-depth) logics based on *KE* is clearly less intuitive than the analogous semantics for the basic logic based on the intelim method—the latter being in line with Quine's dispositional theory of the "primitive" meaning of the connectives. On the other hand, depth-bounded *KE* may be still preferred for potential uses in automated reasoning.

Part II

Depth-bounded non-classical logics

Chapter 4

Tractable depth-bounded approximations to FDE, LP and K_3

4.1 Introduction

First-Degree Entailment (**FDE**, also known as Belnap-Dunn logic) [7, 70, 30, 31], the Logic of Paradox (**LP**) [14, 126], and Strong Kleene Logic ($\mathbf{K_3}$) [105] are closely related to each other and admit of an intuitive informational interpretation as, respectively, a 4-valued logic (in which "a computer *should* think") and 3-valued logics. Specifically, **LP** is a paraconsistent logic, which makes it possible to draw non-trivial deductions from possibly inconsistent pieces of information; whereas, $\mathbf{K_3}$ is a paracomplete logic, that makes it possible to draw deductions from information that might be partial. One of the simplest approaches to paraconsistent and paracomplete reasoning is based on many-valued semantics. In such an approach, the set of truth-values is extended by including new elements other than the two classical ones. So, in turn, the simplest way to implement this approach is using 3 truth-values as in, respectively, **LP** and $\mathbf{K_3}$. Thus, 3 truth-values can be used to handle either inconsistency or partiality of information, one at a time. However, for a logic to handle information that might be both inconsistent and partial, at least 4 different truth-values are required [see 19]. A famous 4-valued logic handling both incosistency and partiality of information is **FDE**, which is therefore paraconsistent and paracomplete.

The informational interpretation of **FDE** is motivated from the use of deductive reasoning as a basic tool in the area of "intelligent" database management or question-answering systems. Databases have a great propensity to be incomplete and become inconsistent: what is stored in a database is usually obtained from different sources which may provide only partial information and may well conflict with each other. So, the values are interpreted as four possible ways in which an atom p can belong to the present state of information of a computer's database, which is fed by a set of equally "reliable" sources. That is, such values record information of the kind "there is a (no) source assenting to p" and "there is a (no) source dissenting to p". As explained below, the truth-values of **LP** and $\mathbf{K_3}$ can plausibly be interpreted along the same lines.

Now, despite their informational flavor, the three logics are co-NP complete [see 148, 10], and so idealized models of how an agent *can* think. In this Chapter we show how the depth-bounded approach can be naturally extended to these many-valued logics, and so provide infinite hierarchies of tractable depth-bounded approximations to them. Under the intuitive informational interpretation admitted by the three logics described above, we identify a need for *imprecise values* such as "*at least true*", which is implicit in the choice of the set of designated values in the semantics of **FDE**. Then, inspired by [53] and [15, 81, 82], we address this question by shifting to *signed formulae*, where the signs express such imprecise values associated with two distinct bipartitions of the corresponding set of standard values.

Thereby, we provide *KE*/*KI*-style proof systems for the three logics, each of which: (i) is formulated by means of signed formulae; (ii) has linear introduction and elimination rules, which fix the meaning of the connectives; (iii) has two branching rules which express a *generalized* rule of bivalence, are structural in that they do not involve any connective, and are essentially cut rules; (iv) can be used as both a direct-proof and a refutation method; (v) obeys the subformula property. Given that the examples introduced in Subsection 3.3.3 are hard for all tableau systems sharing the $\vee/\wedge$ rules with classical tableaux but easy for their analogous *KE*-style systems, our *KE*/*KI*-style systems at issue are interesting independently of the depth-bounded approach mainly because they have an exponential speed-up on their tableau counterparts. In this Chapter, however, we focus on showing that each of our systems naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations to, respectively, **FDE**, **LP** or $\mathbf{K_3}$, in terms of the maximum number of nested applications that are allowed of the branching rules. Intuitively, in each of those systems, the introduction and elimination rules govern the use of actual information, whereas the branching structural rules govern the manipulation of virtual information (i.e., hypothetical information about the imprecise value of a formula). As in the classical case, the key intuition is that the more virtual information needs to be invoked via the branching rules, the harder the inference is for the agent. Thus, the nested applications of those rules provide a sensible measure of inferential depth, and so the levels of the corresponding hierarchy can be naturally related to the inferential power of agents.

We further show that the resulting hierarchies admit of an intuitive 5-valued non-deterministic semantics. This semantics essentially takes the signs as imprecise values (i.e., two-element sets of the standard values), and a fifth value is taken to represent the case where the agent's information is insufficient to even establish any of the imprecise values. (Part of our results regarding **FDE** have been presented at Logica 2021 and submitted for publication in the proceedings as a joint paper with Prof. D'Agostino.)

4.2 FDE interpreted informationally

First Degree Entailment (**FDE**) arose out of the study on relevance logics. It can be interpreted in two ways: (i) as the study of the validity of formulae of the form $A \to B$, where $\to$ is Anderson and Belnap's relevant implication [8] and A, B are implication-free formulae; (ii) as the study of the notion of *relevant deducibility* between standard formulae built-up from the usual connectives. In this second interpretation, **FDE** is associated with the problem of obtaining sound information from possibly inconsistent databases. On the basis of work of Dunn [e.g., 70] and an observation by Smiley (in correspondence), Belnap [30, 31] gave an interesting semantic characterization of **FDE** in terms of a 4-valued logic, and also pointed out its usefulness as the logic in which "a computer should think". This characterization has become not only the standard semantics of **FDE**, but also its standard presentation. In this section we first briefly recall Belnap's 4-valued semantics. Then, we also recall shortly the semantics justifying the intuitive reading of the 4 truth-values; namely, Dunn's 2-valued relational semantics.

4.2.1 Belnap's 4-valued semantics

Deductive reasoning—interpreted as a process of revealing "hidden" information from explicit data—is a basic tool in the area of "intelligent" database management or question-answering systems. In turn, databases have a great propensity to be incomplete and become inconsistent: what is stored in a database is usually obtained from different sources which may provide only partial information and may well conflict with each other. Besides, even if the information obtained from each source is not explicitly inconsistent, it may "hide" contradictions.

For a matrix to characterize a logic adequate for making deductions with information that might be both inconsistent and partial, at least 4 different values are needed [see 19]. The most popular and well-motivated such a 4-valued matrix is precisely Belnap-Dunn(-Smiley)'s.[1] In turn, the interpretation of the corresponding 4 truth-values suggested by Belnap is epistemic or informational. Concretely, the set of those truth-values—originally, called "told values" by Belnap to emphasize their epistemic character—is $\{\mathbf{t}, \mathbf{f}, \mathbf{b}, \mathbf{n}\}$ and is denoted by **4**. These truth-values are interpreted as four possible ways in which an atom p can belong to the present state of information of a computer's database, which in turn is fed by a set of equally "reliable" sources: **t** means that the computer is told that p is true by some source, without being told that p is false by any source; **f** means the computer is told that p is false but never told that p is true; **b** means that the computer is told that p is true by some source and that p is false by some other source (or by the same source in different times); **n** means that the computer is told nothing about the truth-value of p. As explained below, these four truth-values form two distinct lattices, depending on whether we consider the partial information ordering induced by set-inclusion (*approximation* lattice) or the partial ordering based on "closeness to the truth" (*logical* lattice). The information ordering is the one according to which the epistemic state of the computer concerning an atom can evolve over time. As Belnap points out:

> When an atomic formula is entered into the computer as either affirmed or denied, the computer modifies its current set-up by adding a "told True" or "told False" according as the formula was affirmed or denied; it does not subtract any information it already has [...] In other words, if p is affirmed, it marks p with **t** if p were previously marked with **n**, with **b** if p were previously marked with **f**; and of course leaves things alone if p was already marked either **t** or **b**. [30, p. 12]

(Warning: do not confuse the values in **4** with *true* and *false*. The latter are *local* values referring to the information coming from a source, the former are *global* values, summarizing the epistemic state of the computer with respect to all the sources.)

A *set-up* is simply an assignment to each of the atoms of exactly one of the values in **4**. The truth-values of complex formulae are obtained by means of considerations of monotony in an order, related to "Scott's thesis" about approximation lattices [30]. Mathematically, these lattices are just complete lattices; however, they also

[1] As is well known, it was T. J. Smiley who shown (in correspondence) that 4 values are sufficient to characterize the logic in question, and introduced the corresponding truth-tables below—though using numbers instead of names for the values and intending his result as merely technical. For the history of the semantics of **FDE** see [72].

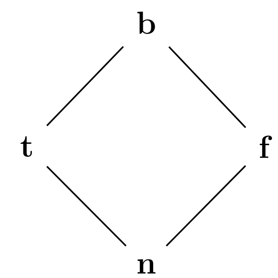


Figure 4.1: *A4*

$\tilde{\vee}$	**t**	**f**	**b**	**n**
t	**t**	**t**	**t**	**t**
f	**t**	**f**	**b**	**n**
b	**t**	**b**	**b**	**t**
n	**t**	**n**	**t**	**n**

$\tilde{\wedge}$	**t**	**f**	**b**	**n**
t	**t**	**f**	**b**	**n**
f	**f**	**f**	**f**	**f**
b	**b**	**f**	**b**	**f**
n	**n**	**f**	**f**	**n**

$\tilde{\neg}$	
t	**f**
f	**t**
b	**b**
n	**n**

Table 4.1: **FDE**-tables

satisfy a non-mathematical condition dictating that it must be appropriate to read $x \sqsubseteq y$ as "x approximates y". Besides, in the presence of these lattices, the functions between them that are taken into account are uniquely the continuous ones—since these are the only functions which respect the lattices qua approximation lattices. Furthermore, in the finite case, for a function to be continuous is just for it to be monotonic. Thus, the truth-values of complex sentences in Belnap's characterization of **FDE** are established by considering an approximation-lattice known as *A4*, and where $x \sqsubseteq_{\mathbf{4}} y$ is interpreted as "approximates the information in". This lattice can be depicted as in Figure 4.1 (where $\sqsubseteq_{\mathbf{4}}$ goes uphill). Here we do not repeat the longish considerations based on 4.1 that led Belnap to establish the truth-value of complex sentences. Instead, we just recall their result; namely, the truth-tables in Table 4.1 (which correspond to those given by Smiley, [cf. 8]):[2]

These truth-tables constitute in turn a logical lattice, which is known as *L4* and whose associated partial order we shall denote by $x \preceq_{\mathbf{4}} y$. This lattice has conjunction as meet and disjunction as join, and can be depicted as in Figure 4.2 (where $\preceq_{\mathbf{4}}$ goes uphill).

[2]The considerations at issue are presented in full detail in [30].

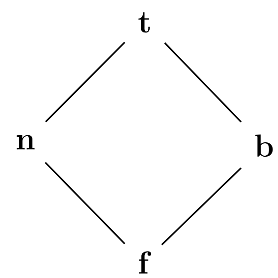


Figure 4.2: *L4*

Now, using the truth-tables, every set-up can be extended to a function of all formulae into **4** in the usual inductive way. That is, given a set-up s, we extend s to a valuation function $v : F(\mathcal{L}) \longrightarrow \mathbf{4}$, where $\mathcal{L} = \{\vee, \wedge, \neg\}$, as follows:

$$\begin{aligned} v(A \vee B) &= \tilde{\vee}(v(A), v(B)) \\ v(A \wedge B) &= \tilde{\wedge}(v(A), v(B)) \\ v(\neg A) &= \tilde{\neg}(v(A)) \end{aligned}$$

We call this function a **4**-*valuation*. It establishes how the computer is to answer questions about complex formulae based on a set-up. While answering questions on the basis of a given epistemic set up is computationally easy, we don't have a logic yet. As Belnap puts it, we "want some rules for the computer to use in generating what it implicitly knows from what it explicitly knows", i.e., we need a logic for the computer to reason. Sticking to Belnap's original presentation [30, 31], the consequence relation of this semantic characterization of **FDE** is:

Definitions 4.2.1. *A entails B*, denoted by $A \to B$, iff for every **4**-valuation v, $v(A) \preceq_{\mathbf{4}} v(B)$. In turn, Γ *entails* A, $\Gamma \vDash_{\mathbf{FDE}} A$, iff the conjunction of all formulae in Γ entails A.

Notation 4.2.2. To simplify reading, in what follows we shall omit the subscript '**FDE**' in $\vDash_{\mathbf{FDE}}$.

Nevertheless, there is an equivalent way to define such a consequence relation that fits better our approach to **FDE** below. To present that alternative definition, we need first to introduce the following terminology:

Definition 4.2.3. Given a **4**-valuation v, we say that A is:

- *at least true* under v if $v(A) = \mathbf{t}$ or $v(A) = \mathbf{b}$;

- *non-true* under v if $v(A) = \mathbf{f}$ or $v(A) = \mathbf{n}$;
- *at least false* under v if $v(A) = \mathbf{f}$ or $v(A) = \mathbf{b}$;
- *non-false* under v if $v(A) = \mathbf{t}$ or $v(A) = \mathbf{n}$.

Definition 4.2.4. $\Gamma \vDash A$ iff for every $\mathbf{4}$-valuation v, (i) if all the formulae in Γ are at least true under v, then A is at least true under v, and (ii) if all formulae in Γ are non-false under v, then A is non-false under v.

Proposition 4.2.5. *The definitions of* $\Gamma \vDash A$ *given in Def. 4.2.1 and Def. 4.2.4 are equivalent.*

Proof. By definition of $\preceq_{\mathbf{4}}$, for every $\mathbf{4}$-valuation v, $v(A_1 \wedge ... \wedge A_n) \preceq_{\mathbf{4}} v(B)$ iff the following four conditions hold for every $\mathbf{4}$-valuation v: (1) if $v(A_1 \wedge ... \wedge A_n) = \mathbf{t}$ then $v(B) = \mathbf{t}$; (2) if $v(A_1 \wedge ... \wedge A_n) = \mathbf{b}$, then $v(B) = \mathbf{t}$ or $v(B) = \mathbf{b}$; (3) if $v(A_1 \wedge ... \wedge A_n) = \mathbf{n}$, then $v(B) = \mathbf{t}$ or $v(B) = \mathbf{n}$; (4) if $v(A_1 \wedge ... \wedge A_n) = \mathbf{f}$, then $v(B) = \mathbf{t}$, or $v(B) = \mathbf{b}$, or $v(B) = \mathbf{n}$, or $v(B) = \mathbf{f}$. However, the truth-value $\mathbf{f}$ for $v(A_1 \wedge ... \wedge A_n)$ does not actually imply a condition for the truth-value of B (since, in such a case, B can take any of the truth-values), and so we can get rid of (4). Now, conditions (1)-(3) hold for every $\mathbf{4}$-valuation v iff for every such a valuation, if $v(A_1 \wedge ... \wedge A_n) = \mathbf{t}$ or $v(A_1 \wedge ... \wedge A_n) = \mathbf{b}$, then $v(B) = \mathbf{t}$ or $v(B) = \mathbf{b}$; and if $v(A_1 \wedge ... \wedge A_n) = \mathbf{t}$ or $v(A_1 \wedge ... \wedge A_n) = \mathbf{n}$, then $v(B) = \mathbf{t}$ or $v(B) = \mathbf{n}$. Finally, by the truth-table of $\wedge$, the latter holds iff for every $\mathbf{4}$-valuation v, (i) if for each A_i $v(A_i) = \mathbf{t}$ or $v(A_i) = \mathbf{b}$, then $v(B) = \mathbf{t}$ or $v(B) = \mathbf{b}$, and (ii) if for each A_i $v(A_i) = \mathbf{t}$ or $v(A_i) = \mathbf{n}$, then $v(B) = \mathbf{t}$ or $v(B) = \mathbf{n}$. □

Still, the most common is to define $\Gamma \vDash A$ based on a corresponding matrix, as any many-valued logic and in terms of preservation of designated truth-values. Thus, Belnap-Dunn's matrix, $\mathcal{M}_{\mathbf{4}}$, is defined as follows:

Definition 4.2.6. Let $\mathcal{M}_{\mathbf{4}}$ be a matrix for $\mathcal{L}$, where $\mathcal{V} = \mathbf{4}$, $\mathcal{D} = \{\mathbf{t}, \mathbf{b}\}$ and the functions in $\mathcal{O}$ are defined by the truth-tables in Tab. 4.1.

Definition 4.2.7. A $\mathbf{4}$-*valuation* is a function $v : F(\mathcal{L}) \longrightarrow \mathbf{4}$ such that for all A, B:

1. $v(\neg A) = \tilde{\neg}(v(A))$;
2. $v(A \circ B) = \tilde{\circ}(v(A), v(B))$.

Where $\circ$ is $\vee$ or $\wedge$.

The consequence relation is then defined as follows:

Definition 4.2.8. $\Gamma \vDash A$ iff for every **4**-valuation v, if $v(B) \in \{\mathbf{t}, \mathbf{b}\}$ for all $B \in \Gamma$, then $v(A) \in \{\mathbf{t}, \mathbf{b}\}$.

That it suffices to mention truth-preservation for characterizing the logical consequence in question was shown already by Dunn [70]. Specifically, he pointed out that if some inference fails to always preserve non-falsity, then it can be shown that it also fails to preserve truth. This is easy to see if one takes a set-up which switches **b** and **n** but leaves **t** and **f** alone, and observes that—owing to the logical symmetry between **b** and **n**—the truth-value of any complex formula has the same feature.

4.2.2 Dunn's 2-valued relational semantics

Dunn [70] introduced the idea of a valuation not as a function but as a *relation* from formulae to the classical truth-values *true* and *false*. This allows formulae to be related to just *true*, to just *false*, to both, or to neither. Specifically:

Definition 4.2.9. A *2-valuation* is a relation $\eta \subseteq At(\mathcal{L}) \times \{true, false\}$. Given a 2-valuation η, this is extended to a relation $\eta \subseteq F(\mathcal{L}) \times \{true, false\}$ satisfying:

$$A \vee B\, \eta\, true \text{ iff } A\, \eta\, true \text{ or } B\, \eta\, true;$$
$$A \vee B\, \eta\, false \text{ iff } A\, \eta\, false \text{ and } B\, \eta\, false;$$

$$A \wedge B\, \eta\, true \text{ iff } A\, \eta\, true \text{ and } B\, \eta\, true;$$
$$A \wedge B\, \eta\, false \text{ iff } A\, \eta\, false \text{ or } B\, \eta\, false;$$

$$\neg A\, \eta\, true \text{ iff } A\, \eta\, false;$$
$$\neg A\, \eta\, false \text{ iff } A\, \eta\, true.$$

Definition 4.2.10. *A entails B* iff for all 2-valuations η, if $A\, \eta\, true$, then $B\, \eta\, true$. In turn, Γ *entails* A iff the conjunction of all formulae in Γ entails A.

Remark 10. It is well known that 4-valued semantics and 2-valued relational semantics are equivalent. Besides, the latter justifies the intuitive reading of the 4 truth-values of the former in that the set **4** can be seen as the powerset of $\{true, false\}$; where, $\mathbf{t} = \{true\}$, $\mathbf{f} = \{false\}$, $\mathbf{b} = \{true, false\}$, and $\mathbf{n} = \emptyset$. Further, Dunn [70] also interprets a sentence being related to both or neither of the classical truth-values in an epistemic—non aletheic—sense. Namely, Dunn does not claim that there are sentences which are in fact both true and false, nor sentences which are in fact neither true nor false. He rather points out that, for instance, there are plenty of situations where agents suppose, assert, believe, etc., contradictory sentences to be true. So, sentences being both true and false may express the contradictory nature of some of the agents' beliefs, assertions and so on.

4.3 Proof-theory of FDE, LP and K_3 revisited

In this section we briefly recall some proof systems for **FDE**, **LP** and $\mathbf{K_3}$. Our exposition of these systems is given in somewhat informal terms since the formal definitions are to be found in the original references. Some of the systems, as well as ideas thereof, are closely related to the approach we shall take to defining tractable approximations to the three many-valued logics. However, as shown by the hard examples introduced in Subsection 3.3.3, a crucial difference between the tableau methods to be recalled and the intelim methods introduced below—as well as *KE*-style and *KI*-style systems—is that the latter have an exponential speed-up on the former. Roughly, the reason of this is that while the tableau methods have operational branching rules that imply a good deal of redundant branchings in the corresponding tree, the intelim methods have only structural branching rules that reduce the amount of branching to a minimum by making all branches mutually exclusive. In the overall context of the Thesis, another important difference between the tableau methods and the intelim methods—and, in fact, cut-based systems in general—is that since in the former cut is eliminable, no approximations can be defined by controlling the application of the cut rules. Further, regardless computational efficiency issues, the natural deduction systems to be recalled below do not comply with a key idea underlying depth-bounded approximations, according to which the meaning of a logical operator is fixed only in terms of actual information. This given that, in the natural deduction systems, some of the (operational) "discharge" rules make essential use of virtual information.

4.3.1 Hilbert-style system

The first characterization of **FDE** is due to Belnap [32]. Namely, *first degree entailments* are formulae of the form $A \to B$, where A and B contain at most disjunction, conjunction, and negation. Belnap provided an axiom system that exactly captures the provable first degree entailments of the Anderson-Belnap system **E** of relevant entailment [8], which we reproduce in Table 4.2.[3] Then:

Definition 4.3.1. A is *provable* from Γ if there is a finite sequence $B_1, ..., B_n, A$ ($n \geq 0$) such that for each formula in the sequence either (i) it belongs to Γ, (ii) it is an instance of an axiom schema, or (iii) it results from an application of a rule to preceding formulae in the sequence.

[3]Note that all these axioms are actually axiom schemata. That is, one can substitute arbitrary formulae for A, B, C, obtaining instances of axioms.

Axiom schemas:

- $A \to A \vee B$
- $B \to A \vee B$
- $A \wedge B \to A$
- $A \wedge B \to B$
- $A \wedge (B \vee C) \to (A \wedge B) \vee C$
- $A \to \neg\neg A$
- $\neg\neg A \to A$

Rules:

- $\dfrac{A \to B,\ B \to C}{A \to C}$
- $\dfrac{A \to C,\ B \to C}{A \vee B \to C}$
- $\dfrac{A \to B,\ A \to C}{A \to B \wedge C}$
- $\dfrac{A \to B}{\neg B \to \neg A}$

Table 4.2: Hilbert-style system for **FDE**

4.3.2 Natural deduction

In the literature there are natural deduction systems that are closely related to the systems that we shall introduce for the first time in this Chapter. For instance, the introduction and elimination rules displayed in Tab. 4.3 constitute a Gentzen-Prawitz style system for **FDE** due to Priest [128]. The vertical dots appearing in one of the rules stand for a proof of the formula below the dots depending on assumptions that may include those enclosed in square brackets. The latter are "discharged" by the application of the rule under consideration, in the sense that the conclusion no longer depends on them, but only on the yet undischarged assumptions that occur in the leaves.

Definitions 4.3.2. We say that a *proof of A depending on* Γ is a tree of occurrences of formulae constructed in accordance with the rules in Tab. 4.3, such that A occurs at the root and Γ is the set of all undischarged assumptions that occur at the leaves. In turn, A is *deducible from* Γ if there is a proof of A depending on some $\Delta \subseteq \Gamma$.

Now, in [128] Priest provided also systems for **LP** and $\mathbf{K_3}$ by simply and respectively adding, to the rules in Tab. 4.3, one of the following rules:[4]

[4]Since the meaning of $A \to B$ in these logics can be expressed by $\neg A \vee B$, rules for implication are straightforwardly obtained.

$$\frac{A}{A \vee B} \qquad \frac{B}{A \vee B} \qquad \frac{A \vee B \quad \begin{matrix}[A]\\ \vdots \\ C\end{matrix} \quad \begin{matrix}[B]\\ \vdots \\ C\end{matrix}}{C}$$

$$\frac{A \quad B}{A \wedge B} \qquad \frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B}$$

$$\frac{\neg A \wedge \neg B}{\neg(A \vee B)} \qquad \frac{\neg(A \vee B)}{\neg A \wedge \neg B} \qquad \frac{\neg A \vee \neg B}{\neg(A \wedge B)}$$

$$\frac{\neg(A \wedge B)}{\neg A \vee \neg B} \qquad \frac{A}{\neg\neg A} \qquad \frac{\neg\neg A}{A}$$

Table 4.3: Gentzen-Prawitz style natural deduction rules for **FDE**

$$\frac{}{A \vee \neg A} \qquad\qquad \frac{A \wedge \neg A}{B}$$

Additional rule for **LP** — Additional rule for $\mathbf{K_3}$

Alternative natural deduction systems for **FDE** were given by Voishvillo [153], and Tamminga and Tanaka [142]. Regarding **LP**, essentially the same system was reintroduced by Kooi and Tamminga [106]. As for $\mathbf{K_3}$, essentially the same system was reintroduced by Tamminga [141]. Besides, alternative Fitch-style systems for the three logics were given by Roy [132].

4.3.3 Tree methods and signs

The first tableaux for **FDE** are due to Dunn [70]. He introduced a direct-proof tableau system based on a modification of Jeffrey's method of (classical) "coupled trees" [103]. Dunn's system rules, recalled in Tab. 4.4, are syntactically identical to the rules of the *unsigned* version of Smullyan's classical tableaux [138]. Thereby:

Definitions 4.3.3.

- Given two formulae A and B, a pair $T = (\mathcal{T}_t, \mathcal{T}_b)$ is a *tableau* for (A, B) if there are two finite sequences $(\mathcal{T}_1, ..., \mathcal{T}_n)$, $(\mathcal{T}_1', ..., \mathcal{T}_m')$ such that $\mathcal{T}_1$ and $\mathcal{T}_1'$ are

$$\frac{A \vee B}{A \mid B} \qquad \frac{\neg(A \vee B)}{\begin{array}{c}\neg A\\ \neg B\end{array}} \qquad \frac{A \wedge B}{\begin{array}{c}A\\ B\end{array}} \qquad \frac{\neg(A \wedge B)}{\neg A \mid \neg B} \qquad \frac{\neg\neg A}{A}$$

Table 4.4: Coupled trees rules

one-point trees whose root is respectively A and B, $\mathcal{T}_n = \mathcal{T}_t$ and $\mathcal{T}'_m = \mathcal{T}_b$, and for each $i < n$ and $i < m$, $\mathcal{T}_{i+1}$ and $\mathcal{T}'_{i+1}$ results, respectively, from $\mathcal{T}_i$ or $\mathcal{T}'_i$ by an application of a rule to preceding formulae in the same branch. We refer to $\mathcal{T}_t$ as *top tree* and to $\mathcal{T}_b$ as *bottom tree*.

- T is *complete* if $\mathcal{T}_t$ and $\mathcal{T}_b$ cannot be further expanded (by applications of the rules).
- Given a tree $\mathcal{T}$, a *path* in $\mathcal{T}$ is a finite sequence of nodes such that the first node is the root of $\mathcal{T}$ and each of the subsequent nodes is an immediate successor of the previous one.
- A path a in a tree $\mathcal{T}$ *covers* a path b in a tree $\mathcal{T}'$ if every atomic formula p or negated atomic formula $\neg p$ that occurs in b occurs also in a.
- B is *provable* from A if there is a complete tableau T for (A, B) such that every path in $\mathcal{T}_t$ covers some path in $\mathcal{T}_b$.
- A is *provable* from Γ if A is provable from the conjunction of all formulae in Γ.

Now, to the best of our knowledge, the first proof systems for **FDE** based on *signed formulae* (S-formulae, for short)—in which the signs stand for sets of truth-values instead of single truth-values—are two refutation tree systems due to D'Agostino [53]. Soon after, Fitting [82] introduced a direct-proof tableau system for **FDE**, which is based on the same use of S-formulae. In fact, in the same period, a general method to use signs as sets of truth-values suitable for any finite-valued propositional logic was provided by Hähnle [93]. The key idea underlying that use is to increase the expressivity of the signs and, thus, significantly decrease the number of new branches per rule application. Later on, Avron [15] used four signs—interpreting them as intuitively corresponding to positive/negative information concerning truth/falsity—to provide tableaux for a diversity of logics; including **FDE**, all 3-valued logics, and some logics that do not have finite characteristic matrix.

Unlike Dunn's coupled trees, D'Agostino's systems [53] use one tree only. The S-formulae, in terms of which those systems are formulated, are expressions of the form $\mathsf{T}\,A$, $\mathsf{F}\,A$, $\mathsf{T}^*\,A$, $\mathsf{F}^*\,A$, where A is an (unsigned) formula. We write $\mathsf{T}\,\Gamma$ for $\{\mathsf{T}\,A | A \in \Gamma\}$. Besides, we use $\varphi, \psi, \theta, ...$, possibly with subscripts, as variables ranging over S-formulae; and $X, Y, Z, ...$, possibly with subscripts, as variables ranging over sets of S-formulae. Intuitively, we interpret: $\mathsf{T}\,A$ as "A is at least true", $\mathsf{F}\,A$ as "A is non-true", $\mathsf{T}^*\,A$ as "A is non-false", and $\mathsf{F}^*\,A$ as "A is at least false". Formally, given Def. 4.2.3:

Definition 4.3.4. A **4**-valuation v *realizes* an S-formula

- $\mathsf{T}\,A$ if A is at least true under v;
- $\mathsf{F}\,A$ if A is non-true under v;
- $\mathsf{T}^*\,A$ if A is non-false under v;
- $\mathsf{F}^*\,A$ if A is at least false under v.

A set of S-formulae X is said to be *realizable* if there is a **4**-valuation v which realizes every element of X.

So, the S-formulae are associated with two distinct bipartitions of the set of 4 truth-values of the standard interpretation of **FDE**. In turn, we say that the *conjugate* of $\mathsf{T}\,A$ is $\mathsf{F}\,A$ and vice versa, and that the conjugate of $\mathsf{T}^*\,A$ is $\mathsf{F}^*\,A$ and vice versa. Besides, we say that the *converse* of T is T^* and vice versa, and that the converse of F is F^* and vice versa.

Thereby, the first system introduced by D'Agostino has the rules in Tab. 4.5, which are formally analogous to the rules of the *signed* version of Smullyan's classical tableaux. Then:

Definitions 4.3.5.

- Given a set of S-formulae $X = \varphi_1, ..., \varphi_m$, we say that $\mathcal{T}$ is a *tableau* for X if there exists a finite sequence $(\mathcal{T}_1, ..., \mathcal{T}_n)$ such that $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $(\varphi_1, ..., \varphi_m)$, $\mathcal{T}_n = \mathcal{T}$, and for each $i < n$, $\mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of a rule to preceding S-formulae in the same branch.
- A branch of a tableau is *closed* if it contains both an S-formula and its conjugate.
- A tableau is *closed* if all its branches are closed.

$$\frac{\mathsf{T}\,A\wedge B}{\begin{array}{c}\mathsf{T}\,A\\ \mathsf{T}\,B\end{array}}\qquad \frac{\mathsf{T}^*\,A\wedge B}{\begin{array}{c}\mathsf{T}^*\,A\\ \mathsf{T}^*\,B\end{array}}\qquad \frac{\mathsf{F}\,A\vee B}{\begin{array}{c}\mathsf{F}\,A\\ \mathsf{F}\,B\end{array}}\qquad \frac{\mathsf{F}^*\,A\vee B}{\begin{array}{c}\mathsf{F}^*\,A\\ \mathsf{F}^*\,B\end{array}}$$

$$\frac{\mathsf{F}\,A\wedge B}{\mathsf{F}\,A\,|\,\mathsf{F}\,B}\qquad \frac{\mathsf{F}^*\,A\wedge B}{\mathsf{F}^*\,A\,|\,\mathsf{F}^*\,B}\qquad \frac{\mathsf{T}\,A\vee B}{\mathsf{T}\,A\,|\,\mathsf{T}\,B}\qquad \frac{\mathsf{T}^*\,A\vee B}{\mathsf{T}^*\,A\,|\,\mathsf{T}^*\,B}$$

$$\frac{\mathsf{T}\,\neg A}{\mathsf{F}^*\,A}\qquad \frac{\mathsf{F}\,\neg A}{\mathsf{T}^*\,A}\qquad \frac{\mathsf{T}^*\,\neg A}{\mathsf{F}\,A}\qquad \frac{\mathsf{F}^*\,\neg A}{\mathsf{T}\,A}$$

Table 4.5: First Smullyan-style tableaux rules

- A is *provable* from Γ if there is a closed tableau for $\mathsf{T}\,\Gamma\cup\{\mathsf{F}\,A\}$.

Now, essentially the same tableau refutation system was reintroduced by Bloesch [39], who used it for both **FDE** and **LP**. Besides, a system differing only notationally and used for **FDE**, **LP** and $\mathbf{K_3}$, was given by Priest [127]. Further, the same system was once again reintroduced by Fitting in [83], where it is used for **FDE**.

In turn, Fitting's system for **FDE** in [82], although formulated using essentially the same S-formulae above, is closer to Dunn's coupled trees method. Besides, in [82] only two signs, T and F, are used (explicitly) and a different convention on them is followed. Namely, a *signed formula* is an expression of the form $\mathsf{T}\,A$ or $\mathsf{F}\,A$, where A is a formula and which is intuitively and respectively interpreted as "A is at most true"—meaning that $v(A) = \mathbf{t}$ or $v(A) = \mathbf{n}$—and "A is at most false"—meaning that $v(A) = \mathbf{f}$ or $v(A) = \mathbf{n}$.[5] The tableau system introduced by Fitting in [82] allows us to test formulae for equivalence. More specifically, it allows us to test whether a pair of formulae have the same truth-value under any **4**-valuation. Since only the signs T and F are used, the corresponding rules, shown in Tab. 4.6, are syntactically identical to the rules of the signed version of Smullyan's classical tableaux. Thereby:

Definitions 4.3.6.

- Given two formulae A and B, we say that A *restricts* B if under any **4**-valuation, if A is at most true so is B. A *requires* B if under any **4**-valuation, if A is at least true (meaning that $v(A) = \mathbf{t}$ or $v(A) = \mathbf{b}$) so is B.

- A *tableau* for an S-formula φ is a tree of S-formulae constructed in accordance with the rules in Tab. 4.6 starting from φ.

[5]These, of course, respectively correspond to "A is non-false" and "A is non-true" above.

$$\frac{\mathsf{T}\,A \wedge B}{\begin{array}{c}\mathsf{T}\,A\\ \mathsf{T}\,B\end{array}} \qquad \frac{\mathsf{F}\,A \vee B}{\begin{array}{c}\mathsf{F}\,A\\ \mathsf{F}\,B\end{array}} \qquad \frac{\mathsf{F}\,A \wedge B}{\mathsf{F}\,A \mid \mathsf{F}\,B} \qquad \frac{\mathsf{T}\,A \vee B}{\mathsf{T}\,A \mid \mathsf{T}\,B} \qquad \frac{\mathsf{T}\,\neg A}{\mathsf{F}\,A} \qquad \frac{\mathsf{F}\,\neg A}{\mathsf{T}\,A}$$

Table 4.6: Second Smullyan-style tableaux rules

- A tableau is *complete* if every non-atomic formula occurrence has had the appropriate rule applied to it, on the corresponding branch.
- If a and b are tableau branches, a *subsumes* b if every signed atomic formula on b also occurs on a.
- If $\mathcal{T}$ and $\mathcal{T}'$ are complete tableaux, $\mathcal{T}$ *covers* $\mathcal{T}'$ if each branch of $\mathcal{T}$ subsumes some branch of $\mathcal{T}'$.

Then Fitting proved:

Proposition 4.3.7 ([82]). *Let A and B be arbitrary formulae.*

- *A restricts B iff a complete tableau for $\mathsf{T}\,A$ covers a complete tableau for $\mathsf{T}\,B$.*
- *A requires B iff a complete tableau for $\mathsf{F}\,B$ covers a complete tableau for $\mathsf{F}\,A$.*
- *A and B are equivalent iff each restricts and requires the other.*[6]

4.3.4 ***KE***-style trees: RE_{fde}

Now, we give special attention to D'Agostino's second refutation system in [53] since it constitutes part of the proof-theoretic basis for defining our hierarchy of approximations to **FDE**, **LP** and $\mathbf{K_3}$. This system was introduced as a more efficient alternative to other proof systems for propositional and first-order **FDE**, and was baptized RE_{fde} since it is based on KE. Again, the hallmark of KE—inherited by RE_{fde}—is the reduction of the amount of branching to a minimum by making all branches mutually exclusive. Accordingly, RE_{fde} has only two branching rules expressing a *generalized* rule of bivalence and the rest of its rules have all a linear format.[7] The propositional rules of RE_{fde} are recalled in Tab. 4.7.

[6] Clearly, the method can be used to check the validity of arguments by, in turn, checking whether the conjunction of all the premises and the conclusion are equivalent.

[7] Generalizations of the rule of bivalence have been fruitfully used in the context of many-valued and substructural logics [see 94, 42, 62].

$$\frac{\mathsf{F}\,A \wedge B \quad \mathsf{T}\,A}{\mathsf{F}\,B} \qquad \frac{\mathsf{F}\,A \wedge B \quad \mathsf{T}\,B}{\mathsf{F}\,A} \qquad \frac{\mathsf{F}^*\,A \wedge B \quad \mathsf{T}^*\,A}{\mathsf{F}^*\,B} \qquad \frac{\mathsf{F}^*\,A \wedge B \quad \mathsf{T}^*\,B}{\mathsf{F}^*\,A}$$

$$\frac{\mathsf{T}\,A \wedge B}{\mathsf{T}\,A} \qquad \frac{\mathsf{T}\,A \wedge B}{\mathsf{T}\,B} \qquad \frac{\mathsf{T}^*\,A \wedge B}{\mathsf{T}^*\,A} \qquad \frac{\mathsf{T}^*\,A \wedge B}{\mathsf{T}^*\,B}$$

$$\frac{\mathsf{T}\,A \vee B \quad \mathsf{F}\,A}{\mathsf{T}\,B} \qquad \frac{\mathsf{T}\,A \vee B \quad \mathsf{F}\,B}{\mathsf{T}\,A} \qquad \frac{\mathsf{T}^*\,A \vee B \quad \mathsf{F}^*\,A}{\mathsf{T}^*\,B} \qquad \frac{\mathsf{T}^*\,A \vee B \quad \mathsf{F}^*\,B}{\mathsf{T}^*\,A}$$

$$\frac{\mathsf{F}\,A \vee B}{\mathsf{F}\,A} \qquad \frac{\mathsf{F}\,A \vee B}{\mathsf{F}\,B} \qquad \frac{\mathsf{F}^*\,A \vee B}{\mathsf{F}^*\,A} \qquad \frac{\mathsf{F}^*\,A \vee B}{\mathsf{F}^*\,B}$$

$$\frac{\mathsf{T}\,\neg A}{\mathsf{F}^*\,A} \qquad \frac{\mathsf{F}\,\neg A}{\mathsf{T}^*\,A} \qquad \frac{\mathsf{T}^*\,\neg A}{\mathsf{F}\,A} \qquad \frac{\mathsf{F}^*\,\neg A}{\mathsf{T}\,A}$$

$$\frac{}{\mathsf{T}\,A \mid \mathsf{F}\,A} \qquad \frac{}{\mathsf{T}^*\,A \mid \mathsf{F}^*\,A}$$

Table 4.7: Propositional RE_{fde} rules

α	α_1	α_2
$\mathsf{T}\, A \wedge B$	$\mathsf{T}\, A$	$\mathsf{T}\, B$
$\mathsf{F}\, A \vee B$	$\mathsf{F}\, A$	$\mathsf{F}\, B$
$\mathsf{T}\, \neg A$	$\mathsf{F}^*\, A$	$\mathsf{F}^*\, A$
$\mathsf{F}\, \neg A$	$\mathsf{T}^*\, A$	$\mathsf{T}^*\, A$
$\mathsf{T}^*\, A \wedge B$	$\mathsf{T}^*\, A$	$\mathsf{T}^*\, B$
$\mathsf{F}^*\, A \vee B$	$\mathsf{F}^*\, A$	$\mathsf{F}^*\, B$
$\mathsf{T}^*\, \neg A$	$\mathsf{F}\, A$	$\mathsf{F}\, A$
$\mathsf{F}^*\, \neg A$	$\mathsf{T}\, A$	$\mathsf{T}\, A$

β	β_1	β_2
$\mathsf{F}\, A \wedge B$	$\mathsf{F}\, A$	$\mathsf{F}\, B$
$\mathsf{T}\, A \vee B$	$\mathsf{T}\, A$	$\mathsf{T}\, B$
$\mathsf{F}^*\, A \wedge B$	$\mathsf{F}^*\, A$	$\mathsf{F}^*\, B$
$\mathsf{T}^*\, A \vee B$	$\mathsf{T}^*\, A$	$\mathsf{T}^*\, B$

Table 4.8: Unifying notation

Thus, (the propositional fragment of) RE_{fde} consists of: (i) elimination rules for the connectives; (ii) two structural rules that allows us to respectively append $\mathsf{T}\, A$ and $\mathsf{F}\, A$, or $\mathsf{T}^*\, A$ and $\mathsf{F}^*\, A$ as sibling nodes at the end of any branch of a tree, generating two new (mutually exclusive) branches. The latter are essentially cut rules which are *not* eliminable, and are respectively called PB and PB^* given their relation with a generalized Principle of Bivalence. Indeed, RE_{fde} is more efficient than other proof systems because the application of PB and PB^* allow us to avoid many redundant branchings in the refutation trees. Now, for succinctness, (an extension of) Smullyan's unifying notation [138] was used in [53] as shown in the Table 4.8.

Thereby, the elimination rules can be "packed" into the following four types of rules (where β_i', $i = 1, 2$ denotes the conjugate of β_i):

Rule A1 $\quad \dfrac{\alpha}{\alpha_1} \qquad$ Rule A2 $\quad \dfrac{\alpha}{\alpha_2}$

Rule B1 $\quad \dfrac{\beta \quad \beta_1'}{\beta_2} \qquad$ Rule B2 $\quad \dfrac{\beta \quad \beta_2'}{\beta_1}$

In each application of the corresponding rules, the S-formulae α and β are called *major premises*; whereas, in each application of rules of type B1 and B2 the S-formulae β_i', $i = 1, 2$ are called minor premises (rules of type A1 and A2 have no minor premises.)

Definitions 4.3.8.

- Let $X = \{\varphi_1, ..., \varphi_m\}$. Then $\mathcal{T}$ is an RE_{fde}-*tree* for X if there exists a finite sequence $(\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n)$ such that $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $(\varphi_1, ..., \varphi_m)$, $\mathcal{T}_n = \mathcal{T}$, and for each $i < n$, $\mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of an elimination rule to preceding S-formulae in the same branch, or by an application of PB or PB^*.

- A branch of an RE_{fde}-tree is *closed* when it contains an S-formula and its conjugate; otherwise it is *open*.

- The tree itself is said to be *closed* when all its branches are closed.

- A formula A is *provable* from the set of formulae Γ iff there is a closed RE_{fde}-tree for $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$.

Now we recall the proof of the completeness of RE_{fde} yielding the subformula property of the system given in [53]. We also prove soundness stated there without proof. For soundness, Def. 4.2.3 and 4.3.4, as well as the following lemma are required:

Lemma 4.3.9. *If $\mathcal{T}$ is a closed RE_{fde}-tree for X, then the tableau $\mathcal{T}'$, obtained from $\mathcal{T}$ by replacing every (occurrence of an) S-formula with (an occurrence of) its converse, is a closed RE_{fde}-tree for the set X^* of the converses of the S-formulae in X.*

Proof. We use the notation $\begin{array}{c}\mathcal{T}\\ \varphi\end{array}$ to denote either an empty intelim tree or a non-empty intelim tree such that φ is one of its terminal nodes. Let $\mathcal{T}$ be a closed RE_{fde}-tree for X. We proceed by induction on the number of nodes in $\mathcal{T}$, denoted by $\lambda(\mathcal{T})$:

- Base case: Suppose that $\lambda(\mathcal{T}) = 2$. Then $\mathcal{T}$ consists of only two assumptions and so either $X = \{\mathsf{T}\,A, \mathsf{F}\,A\}$ or $X = \{\mathsf{T}^*\,A, \mathsf{F}^*\,A\}$, for some A. Thus, in either case, $\mathcal{T}'$ is a closed tableau for, respectively, $X^* = \{\mathsf{T}^*\,A, \mathsf{F}^*\,A\}$ or $X^* = \{\mathsf{T}\,A, \mathsf{F}\,A\}$ for some A.

- Inductive hypothesis: Suppose that if $\mathcal{T}$ is a closed RE_{fde}-tree for X and $\lambda(\mathcal{T}) = k$, with $k > 2$, then $\mathcal{T}'$ is a closed RE_{fde}-tree for X^*.

- Induction step: Let $\lambda(\mathcal{T}) = k + 1$. Thus, if $\mathcal{T}$ consists only of assumptions, then the argument is analogous to the case involving only two assumptions (clearly, if the ith and jth nodes of $\mathcal{T}$ correspond to conjugate elements in X, the ith and jth nodes of $\mathcal{T}'$ correspond to conjugate elements in X^*). Now, if

$\mathcal{T}$ contains applications of rules, then there are 22 cases depending on which rule has been applied in the last inference of $\mathcal{T}$. We consider only the case in which the rule is an elimination of disjunction, the other cases being similar. Thus, suppose that $\mathcal{T}$ has the following form:

$$\begin{array}{c} \mathcal{T}_{n-1} \\ \mathsf{F}\, A \end{array}$$

(where $\mathsf{F}\, A \vee B$ lies on the same branch that $\mathsf{F}\, A$). Now, by inductive hypothesis, if $\mathcal{T}_{n-1}$ is a closed RE_{fde}-tree for X, $\mathcal{T}'_{n-1}$ is a closed RE_{fde}-tree for X^*. In turn, by hypothesis, if $\mathcal{T}_{n-1}$ is not a closed RE_{fde}-tree for X, $\mathcal{T}$ is. Thereby, the tableau $\mathcal{T}'$ with the form:

$$\begin{array}{c} \mathcal{T}'_{n-1} \\ \mathsf{F}^*\, A \end{array}$$

(where $\mathsf{F}^*\, A \vee B$ lies on the same branch that $\mathsf{F}^*\, A$.) is a closed RE_{fde}-tree for X^*.

□

Proposition 4.3.10 (*RE*$_{fde}$**-soundness).** *If* $\mathcal{T}$ *is a closed* RE_{fde}*-tree for* $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\, A\}$ *or for* $\mathsf{T}^*\,\Gamma \cup \{\mathsf{F}^*\, A\}$*, then* $\Gamma \vDash A$.

Proof. It is easy to see that all the rules of RE_{fde} are correct in the sense that every **4**-valuation which realizes the premise of a rule of type A realizes also the conclusion of the rule, and every **4**-valuation which realizes the premises of a rule of type B realizes also the conclusions of the rule. As for PB and PB^*, any **4**-valuation realizes, of course, exactly one of the conclusions of the corresponding rule. Thereby, it follows, by an elementary inductive argument, that if a **4**-valuation v realizes all the initial S-formulae of a RE_{fde}-tree $\mathcal{T}$, then there is exactly one branch b of $\mathcal{T}$ such that v realizes all the S-formulae occurring in b. However, of course, no **4**-valuation can realize two conjugate S-formulae simultaneously. Thus, if $\mathcal{T}$ is a closed RE_{fde}-tree, no **4**-valuation can realize all the initial S-formulae of $\mathcal{T}$. Hence, if $\mathcal{T}$ is a closed RE_{fde}-tree for $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\, A\}$, it follows that for every **4**-valuation v, A is at least true in v whenever all formulae in Γ are. Moreover, it follows from lemma 4.3.9 that no **4**-valuation can realize $\mathsf{T}^*\,\Gamma \cup \{\mathsf{F}^*\, A\}$. So, for every **4**-valuation v, A is non-false in v whenever all the formulae in Γ are. Therefore, by def. 4.2.4, $\Gamma \vDash A$. □

Now, in order to prove completeness, the following notions were introduced in [53]:

Definitions 4.3.11. A branch b of a RE_{fde}-tree $\mathcal{T}$ is *complete* if (i) for every α in b both α_1 and α_2 occur in b and (ii) for every β in b at least one of β_1, β_2 occurs in b. In turn, an RE_{fde}-tree $\mathcal{T}$ is *completed* when every branch of $\mathcal{T}$ is complete.

Next, we have the following proposition:

Proposition 4.3.12. *Every complete open branch of any RE_{fde}-tree is realizable.*

This proposition follows immediately from an analog of Hintikka's lemma within our framework. So, we first define an analog of Hintikka sets within our framework:

Definition 4.3.13. A set of S-formulae X is an *R-Hintikka set* iff it satisfies the following conditions:

- H_0: No signed variable and its conjugate are both in X.
- H_1: If $\alpha \in X$, then $\alpha_1 \in X$ and $\alpha_2 \in X$.
- H_2: If $\beta \in X$, then $\beta_1 \in X$ or $\beta_2 \in X$.

It follows from the definitions above that the set of S-formulae in a complete open branch of any RE_{fde}-tree is an R-Hintikka set.

Then Proposition 4.3.12 follows immediately from the following lemma:

Lemma 4.3.14 ([53]). *Every R-Hintikka set is realizable.*

Proof. Let X be an R-Hintikka set. Let us assign to each variable p which occurs in at least an element of X a value in **4** as follows:

- If $\mathsf{T}\,p \in X$ and $\mathsf{F}^*\,p \notin X$, give p the value **t**.
- If $\mathsf{T}\,p \in X$ and $\mathsf{F}^*\,p \in X$, give p the value **b**.
- If $\mathsf{F}\,p \in X$ and $\mathsf{T}^*\,p \notin X$, give p the value **f**.
- If $\mathsf{F}\,p \in X$ and $\mathsf{T}^*\,p \in X$, give p the value **n**.
- If $\mathsf{T}^*\,p \in X$ and $\mathsf{F}\,p \notin X$, give p the value **t**.
- If $\mathsf{F}^*\,p \in X$ and $\mathsf{T}\,p \notin X$, give p the value **f**.

Now, it is obvious that the **4**-valuation induced by this assignment realizes all the signed variables occurring in X. Thus, we only need to observe that:

(i) If a **4**-valuation realizes both α_1 and α_2, then it realizes also α.

(ii) If a **4**-valuation realizes at least one of β_1 and β_2, then it realizes also β.

The lemma then follows from an easy induction on the degree of the S-formulae in X. □

We still require a couple of definitions to show the completeness of RE_{fde}. The following introduces a notion akin to the notion of R-Hintikka set:

Definition 4.3.15. A set of signed formulae X is an *R-analytic set* iff it satisfies the following conditions:

- A_0: No signed variable and its conjugate are both in X.
- A_1: If $\alpha \in X$, then $\alpha_1 \in X$ and $\alpha_2 \in X$.
- A_2: If $\beta \in X$ and $\beta'_1 \in X$, then $\beta_2 \in X$.
- A_3: If $\beta \in X$ and $\beta'_2 \in X$, then $\beta_1 \in X$.

Note that an R-analytic set differs from a R-Hintikka set in that it may be the case that for some β in the set neither β_1 nor β_2 are in the set.

Definition 4.3.16. An R-analytic set is *β-complete* if for every $\beta \in X$ either of the following two conditions is satisfied:

- either $\beta_1 \in X$ or $\beta'_1 \in X$;
- either $\beta_2 \in X$ or $\beta'_2 \in X$.

Then it is easy to verify that:

Fact 4.3.17. *If X is an R-analytic set and X is β-complete, then X is an R-Hintikka set.*

Now, completeness follows from Fact 4.3.17 and Prop. 4.3.12:

Proposition 4.3.18 (RE_{fde}-completeness). *If $\Gamma \vDash A$ then there is a closed RE_{fde}-tree for $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$.*

This proof of the completeness of RE_{fde} yields the subformula property (SFP) as a corollary.

Corollary 4.3.19 (Analytic cut property). *If there is a closed RE_{fde}-tree $\mathcal{T}$ for X, then there is a closed RE_{fde}-tree $\mathcal{T}'$ for X such that the rules PB and PB^* are applied only to subformulae of S-formulae in X.*

As pointed out in [53], in fact, the proof above shows that, when applying the branching rules, only the immediate signed subformulae of signed formulae of type β occurring above in the same branch and that have not been already "analysed" need to be considered.[8] Now, given that the elimination rules are obviously analytic, it follows that:

Corollary 4.3.20 (SFP). *If there is a closed RE_{fde}-tree $\mathcal{T}$ for X, then there is a closed RE_{fde}-tree $\mathcal{T}'$ for X such that every S-formula occurring in $\mathcal{T}'$ is a signed subformula of S-formulae in X.*

In Lemma 4.5.9 below we shall prove a more general version of the SFP by means of proof transformations.

4.4 The need for imprecise values

For the unrestricted language allowing arbitrary formulae involving $\wedge, \vee$ and $\neg$, the decision problem for **FDE**'s consequence relation is co-NP complete [see 148, 10]. This fact follows from the celebrated result by Cook [47] showing that **CPL** is co-NP complete, together with the fact that the decision problem of inconsistency in **CPL** can be reduced to the decision problem of entailment in **FDE**. The latter specifically as follows:

Proposition 4.4.1. *Γ is classically inconsistent iff $\Gamma \vDash (p_1 \wedge \neg p_1) \vee (p_2 \wedge \neg p_2) \vee ... \vee (p_n \wedge \neg p_n)$, where $p_1, ..., p_n$ are the atoms occurring in Γ.*

Proof. By definition, Γ is classically inconsistent iff there is no classical valuation, $v : F(\mathcal{L}) \longrightarrow \{true, false\}$, such that $v(A) = true$ for all $A \in \Gamma$. In turn, also by definition, this holds iff for every **4**-valuation, $v : F(\mathcal{L}) \longrightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{b}, \mathbf{n}\}$, such that $v(A) \in \{\mathbf{t}, \mathbf{b}\}$ for all $A \in \Gamma$, $v(A) = \mathbf{b}$ for some $A \in \Gamma$ and so, by the **FDE**-tables, $v(p_i) = \mathbf{b}$ for some p_i occurring in Γ. Thus, by the **FDE**-tables for $\neg$ and $\wedge$, this holds

[8] Where we extend the definition of *immediate subformula* to signed formulae in the obvious way.

iff for every **4**-valuation v such that $v(A) \in \{\mathbf{t}, \mathbf{b}\}$ for all $A \in \Gamma$, $v(p_i \wedge \neg p_i) = \mathbf{b}$ for some p_i occurring in Γ and so, by the **FDE**-table for $\vee$, $v((p_1 \wedge \neg p_1) \vee ... \vee (p_n \wedge \neg p_n)) \in \{\mathbf{t}, \mathbf{b}\}$. Hence, the latter holds iff $\Gamma \vDash (p_1 \wedge \neg p_1) \vee ... \vee (p_n \wedge \neg p_n)$. □

This situation brings us to the need for tractable approximations. In the next section we shall present a sort of natural deduction system for **FDE** based on two key observations.

First, as is implicit in the quotation from Belnap in Subsection 4.2.1, the values in **4**, except for **b**, cannot be taken as *stable*. An epistemic set up is just a snapshot of an epistemic state that evolves over time. If we want to consider the truth-values **t**, **f**, **n** as stable we need to assume complete information about the set of sources Ω. Namely, while the meaning of **b** is "*there is* at least a source assenting to p and at least a source dissenting from p" (which is information empirically accessible to x in the sense that x may *hold* this information without a complete knowledge of Ω), the meaning of **t**, **f** and **n** involves information of the kind "*there is no* source such that...", and so requires complete information about the sources in Ω, which may not be empirically accessible to x at any given time. What if the agent does not have such a complete knowledge about the sources? For instance, the agent may well be receiving information from an "open" set of sources as they become accessible (even if the information coming from each single source is assumed to be robust). In such a case, the possibility for an agent to come across a source falsifying "there is no source such that..." is always open.

The situation just described is, of course, an instance of the general issue of the conclusive verifiability and falsifiability of universally and existentially quantified empirical claims [124, 23]. In particular, there cannot be any amount of observational data that would conclusively verify a universal generalization—dually, conclusively falsify an existential generalization. For, nothing guarantees that there is a fixed number of examples, let alone that all examples are accessible to the agent. So, "there is a white raven" is not conclusively falsifiable since, after all, nothing prevents the agent to come across a white raven; and, dually, "there is no white raven" is not conclusively verifiable. On the other hand, of course, scientists dramatically make that kind of claims and work with them: although our world is an endless source of empirical data and some data may well be non-accessible, scientists *assume* the content of their non-conclusively verifiable claims for a variety of reasons. Furthermore, naturally, this situation is not exclusive of science but we can think of several and diverse settings. To mention a practical one, consider a poll raised regarding the efficacy of a brand new vaccine. An agent collecting and processing the data may well lack information from some sources—say, some sources do not answer or it is not logistically viable to ask all the sources. However, the agent might be in a position in which she has queried all, say, relevant and competent sources.

Therefore, despite their informational nature, three of the values in **4** are *information-transcendent* when interpreted as timeless. They refer to an objective state of affairs concerning the domain of all sources, that may well be inaccessible to the computer at any given time. This motivates the need for *a stable imprecise value* such as "**t** *or* **b**", which is implicit in the choice of the set of designated values by Belnap. Inspired by work of D'Agostino [53], and Fitting and Avron [81, 82, 15] (briefly recalled in Section 4.3), we shall address this question by shifting to *signed formulae*, where the signs express such imprecise values associated with two distinct bipartitions of **4**.

A *second* key observation is that, as suggested by Belnap [30, 31], there is no reason to assume that an agent is "told" about the values of *atoms only.* As we shift from objective truth and falsity to informational truth and falsity, this is a highly unrealistic restriction. In most practical contexts we may be told that a certain disjunction is true without being told which of the two disjuncts is the true one, or that a certain conjunction is false without being told which of the two conjuncts is the false one. As a simple example of the former situation, take the information that Alice and Bob are siblings (either they have the same mother or they have the same father); for the latter, take the information that Alice and Bob are not siblings, i.e., for any individual x, the conjunction "x is a parent of Bob and x is a parent of Alice" must be false, which amounts to saying that either the first or the second conjunct is false, without necessarily knowing which. In the context of **CPL**, these considerations naturally lead to a non-deterministic 3-valued semantics which was anticipated by Quine, as we explained in Chapter 2. (See [57] for further references and a discussion that includes an interesting quotation from Michael Dummett to the effect that in non-mathematical contexts our information may well be *irremediably disjunctive* in nature.)

These two observations prompt us to propose a proof-theoretical approach to depth-bounded reasoning in **FDE** that is similar to the one taken in [61, 59, 58] for **CPL**. Before addressing this issue, however, we shall provide in the next section a proof-theoretical characterization of unbounded reasoning in **FDE** that will pave the way for defining its tractable approximations.

4.5 Intelim deduction in FDE

In what follows we shall use *signed formulae* (again, S-formulae for short) as introduced by D'Agostino in [53] and recalled in Sec. 4.3. However, we intuitively re-interpret those S-formulae in terms of information that is *actually* possessed by an agent. Namely, denoting an agent with x and a **4**-valuation with v, their intended interpretation is respectively as follows: $\mathsf{T}\,A$ means "x holds that A is at least true" (expressing that $v(A) \in \{\mathbf{t}, \mathbf{b}\}$); $\mathsf{F}\,A$ means "x holds that A is non-true" (saying that $v(A) \in \{\mathbf{f}, \mathbf{n}\}$); $\mathsf{T}^*\,A$ means "x holds that A is non-false" ($v(A) \in \{\mathbf{t}, \mathbf{n}\}$); F^* means "x holds that A is at least false" ($v(A) \in \{\mathbf{f}, \mathbf{b}\}$).[9] Crucially, S-formulae of the form $\mathsf{T}\,A$ or $\mathsf{F}^*\,A$ express information that x may hold even without a complete knowledge of the set sources Ω. However, this is not the case of the other two types of S-formulae which involve complete knowledge of Ω and so can only be assumed hypothetically.

In turn, recall that the *conjugate* of $\mathsf{T}\,A$ is $\mathsf{F}\,A$ and vice versa, and that the conjugate of $\mathsf{T}^*\,A$ is $\mathsf{F}^*\,A$ and vice versa. Besides, we write $\mathsf{T}\,\Gamma$ for $\{\mathsf{T}\,A | A \in \Gamma\}$, and use $\varphi, \psi, \theta, ...$, possibly with subscripts, as variables ranging over S-formulae; and $X, Y, Z, ...$, possibly with subscripts, as variables ranging over sets of S-formulae. Moreover, let us use $\bar{\varphi}$ to denote the conjugate of φ. Furthermore, let us use S as a variable ranging over $\{\mathsf{T}, \mathsf{F}, \mathsf{T}^*, \mathsf{F}^*\}$, and with $\bar{\mathsf{S}}$ denote: F if $\mathsf{S} = \mathsf{T}$, T if $\mathsf{S} = \mathsf{F}$, F^* if $\mathsf{S} = \mathsf{T}^*$, and T^* if $\mathsf{S} = \mathsf{F}^*$. Finally, we say that the *unsigned part* of an S-formula is the unsigned formula that results from it by removing its sign. Given an S-formula φ, we denote by φ^u the unsigned part of φ and by X^u the set $\{\varphi^u | \varphi \in X\}$. Note also that, for the reasons explained in the previous section, an agent may hold the information that $\mathsf{T}\,A \vee B$, but neither the information that $\mathsf{T}\,A$ nor that $\mathsf{T}\,B$. Similarly, she may hold the information that $\mathsf{F}^*\,A \wedge B$, but neither the information that $\mathsf{F}^*\,A$ nor that $\mathsf{F}^*\,B$.

We identify the basic (*0-depth*) logic of our hierarchy of approximations with the inferences that an agent can draw without making hypotheses about the "objective" state of affairs concerning the whole of Ω. In other words, without making hypothetical assumptions that go beyond the information that she holds. We shall show that a natural proof-theoretic characterization of this basic logic is obtained by means of the set of elimination rules of RE_{fde}, together with suitable introduction rules for the connectives. We display all these rules in Table 4.9, and shall refer to them as *intelim* rules. Note that the analogous 0-depth system for **CPL** in [59, 58] (and recalled in Chapter 3) is characterized by the intelim rules obtained by removing all the starred signs, replacing them with the unstarred signs T and F, interpreted as "only true" and "only false", and eliminating duplicates. Observe also that the characterization of the basic logic bears some resemblance with natural deduction, but does not have

[9] Similar approaches to **FDE** were given by Blasio [36, 37], and Shramko and Wansing [136]. However, those approaches were extended along very different lines and used for very different purposes. Particularly, in those approaches there is no attempt to provide tractable approximations. We thank Luis Estrada-González for having pointed us at [136].

$$\mathsf{T}\,\neg(A \vee B)^{@}$$
$$\mathsf{T}\,\neg C^{@}$$
$$\mathsf{F}^*\,A \vee B$$
$$\mathsf{F}^*\,A$$
$$\mathsf{F}^*\,C$$
$$\mathsf{F}^*\,A \vee C$$
$$\mathsf{T}\,\neg(A \vee C)$$

Figure 4.3: An itelim sequence

discharge rules, since no hypothetical reasoning is involved. Besides, observe that the intelim rules for disjunction and conjunction are dual of each other, and that a sentence and its negation are treated in a symmetric way. Now, recall that in the elimination rules, we refer to the premise containing the connective that is to be eliminated as *major* and to the other premise as *minor*. In turn, given that the intelim rules have all a linear format, their application generates *intelim sequences*. Namely, finite sequences $(\varphi_1, ..., \varphi_n)$ of S-formulae such that, for every $i = 0, ..., n$, either φ_i is an assumption or it is the conclusion of the application of an intelim rule to preceding S-formulae. In Fig. 4.3 we show a simple example of an intelim sequence, where each assumption is marked with an '@'.

The intelim rules are all sound, but not complete for full **FDE**. Indeed, as we shall show below, these rules just characterize the basic logic in the hierarchy of approximations. Completeness for full **FDE** is obtained by adding only the two branching structural rules of RE_{fde}, PB and PB^*:

$$\frac{}{\mathsf{T}\,A \mid \mathsf{F}\,A} \qquad \frac{}{\mathsf{T}^*\,A \mid \mathsf{F}^*\,A}$$

Recall that according to these rules, we are allowed to: (i) append both $\mathsf{T}\,A$ and $\mathsf{F}\,A$ as sibling nodes to the last element of any intelim sequence; (ii) append both $\mathsf{T}^*\,A$ and $\mathsf{F}^*\,A$ in a similar way. The intuitive meaning of these rules is that one of the two cases must obtain considering the whole of Ω even if the agent has no actual information about which is the case. In this sense, we call the information expressed by each conjugate S-formula *virtual* information; i.e., hypothetical information that the agent does not hold, but she temporarily *assumes as if* she held it.

As recalled in Chapter 3, for **CPL** only *PB*, with T and F interpreted as "only true" and "only false", makes sense and is sufficient for completeness. With the addition of *PB* and PB^* to the stock of rules, deductions are represented by downward-growing trees, which brings the method somewhat closer to tableaux. Each appli-

$\dfrac{\mathsf{F}\,A}{\mathsf{F}\,A \wedge B}$	$\dfrac{\mathsf{F}\,B}{\mathsf{F}\,A \wedge B}$	$\dfrac{\mathsf{F}^*\,A}{\mathsf{F}^*\,A \wedge B}$	$\dfrac{\mathsf{F}^*\,B}{\mathsf{F}^*\,A \wedge B}$
$\dfrac{\mathsf{T}\,A}{\mathsf{T}\,A \vee B}$	$\dfrac{\mathsf{T}\,B}{\mathsf{T}\,A \vee B}$	$\dfrac{\mathsf{T}^*\,A}{\mathsf{T}^*\,A \vee B}$	$\dfrac{\mathsf{T}^*\,B}{\mathsf{T}^*\,A \vee B}$
$\dfrac{\begin{array}{c}\mathsf{T}\,A \\ \mathsf{T}\,B\end{array}}{\mathsf{T}\,A \wedge B}$	$\dfrac{\begin{array}{c}\mathsf{F}\,A \\ \mathsf{F}\,B\end{array}}{\mathsf{F}\,A \vee B}$	$\dfrac{\begin{array}{c}\mathsf{T}^*\,A \\ \mathsf{T}^*\,B\end{array}}{\mathsf{T}^*\,A \wedge B}$	$\dfrac{\begin{array}{c}\mathsf{F}^*\,A \\ \mathsf{F}^*\,B\end{array}}{\mathsf{F}^*\,A \vee B}$
$\dfrac{\mathsf{T}\,A}{\mathsf{F}^*\,\neg A}$	$\dfrac{\mathsf{F}\,A}{\mathsf{T}^*\,\neg A}$	$\dfrac{\mathsf{T}^*\,A}{\mathsf{F}\,\neg A}$	$\dfrac{\mathsf{F}^*\,A}{\mathsf{T}\,\neg A}$
$\dfrac{\begin{array}{c}\mathsf{F}\,A \wedge B \\ \mathsf{T}\,A\end{array}}{\mathsf{F}\,B}$	$\dfrac{\begin{array}{c}\mathsf{F}\,A \wedge B \\ \mathsf{T}\,B\end{array}}{\mathsf{F}\,A}$	$\dfrac{\begin{array}{c}\mathsf{F}^*\,A \wedge B \\ \mathsf{T}^*\,A\end{array}}{\mathsf{F}^*\,B}$	$\dfrac{\begin{array}{c}\mathsf{F}^*\,A \wedge B \\ \mathsf{T}^*\,B\end{array}}{\mathsf{F}^*\,A}$
$\dfrac{\mathsf{T}\,A \wedge B}{\mathsf{T}\,A}$	$\dfrac{\mathsf{T}\,A \wedge B}{\mathsf{T}\,B}$	$\dfrac{\mathsf{T}^*\,A \wedge B}{\mathsf{T}^*\,A}$	$\dfrac{\mathsf{T}^*\,A \wedge B}{\mathsf{T}^*\,B}$
$\dfrac{\begin{array}{c}\mathsf{T}\,A \vee B \\ \mathsf{F}\,A\end{array}}{\mathsf{T}\,B}$	$\dfrac{\begin{array}{c}\mathsf{T}\,A \vee B \\ \mathsf{F}\,B\end{array}}{\mathsf{T}\,A}$	$\dfrac{\begin{array}{c}\mathsf{T}^*\,A \vee B \\ \mathsf{F}^*\,A\end{array}}{\mathsf{T}^*\,B}$	$\dfrac{\begin{array}{c}\mathsf{T}^*\,A \vee B \\ \mathsf{F}^*\,B\end{array}}{\mathsf{T}^*\,A}$
$\dfrac{\mathsf{F}\,A \vee B}{\mathsf{F}\,A}$	$\dfrac{\mathsf{F}\,A \vee B}{\mathsf{F}\,B}$	$\dfrac{\mathsf{F}^*\,A \vee B}{\mathsf{F}^*\,A}$	$\dfrac{\mathsf{F}^*\,A \vee B}{\mathsf{F}^*\,B}$
$\dfrac{\mathsf{T}\,\neg A}{\mathsf{F}^*\,A}$	$\dfrac{\mathsf{F}\,\neg A}{\mathsf{T}^*\,A}$	$\dfrac{\mathsf{T}^*\,\neg A}{\mathsf{F}\,A}$	$\dfrac{\mathsf{F}^*\,\neg A}{\mathsf{T}\,A}$

Table 4.9: Intelim rules for the standard **FDE** connectives

cation of PB or PB^* stands for the introduction of virtual information about the imprecise value of a formula A, which we shall respectively call the *PB-formula* or *PB^*-formula*. Note once again that, whereas signed formulae of the form $\mathsf{T}\, A$ and $\mathsf{F}^*\, A$ are empirically obtainable, signed formulae of the form $\mathsf{T}^*\, A$ and $\mathsf{F}\, A$ are obtainable only by applying PB or PB^*. In turn, the S-formulae $\mathsf{T}\, A$, $\mathsf{F}\, A$, $\mathsf{T}^*\, A$ and $\mathsf{F}^*\, A$ appended via those branching rules will be all called *virtual assumptions*. Now, as mentioned in Section 4.3, PB and PB^* are essentially cut rules that may introduce formulae of arbitrary degree. However, as we will show in Lemma 4.5.9, their application can be restricted so as to satisfy the subformula property (SFP). Moreover, from our informational viewpoint, the main conceptual advantage of this proof-theoretic characterization consists in that it clearly separates the *intelim rules* that fix the meaning of the connectives in terms of the information that an agent holds from the two *structural rules* that introduce virtual information (PB and PB^*).

Intuitively, the more virtual information needs to be invoked via PB or PB^*, the harder the inference is for the agent, both from the computational and the cognitive viewpoint. In this sense, the nested applications of PB and PB^* provide a sensible measure of inferential *depth*. This naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations to **FDE** in terms of the maximum number of nested applications of PB and PB^* that are allowed. Before giving definitions and results, we remark that (i) unlike the branching rules of Smullyan-style tableaux, our branching rules are *structural* in that they do not involve any specific logical operator; (ii) as explained in Section 4.3, the elimination rules, together with the branching rules, were early introduced in [53] as constituting a refutation method for full **FDE** called RE_{fde}. So, the completeness of RE_{fde} trivially implies the completeness of the system presented in this Chapter. However, our intelim method can be used as a direct-proof method as well as a refutation method, and leads to more powerful approximations. A direct completeness proof can also be given based on the semantics, which implies the subformula property. In Lemma 4.5.9 we choose to prove a more general version of the subformula property by means of proof transformations.

Definitions 4.5.1.

- Let $X = \{\varphi_1, ..., \varphi_m\}$. Then $\mathcal{T}$ is an *intelim tree for X* if there is a finite sequence $(\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n)$ such that $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $(\varphi_1, ..., \varphi_m)$, $\mathcal{T}_n = \mathcal{T}$, and for each $i < n$, $\mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of an intelim rule to preceding S-formulae in the same branch, or by an application of PB or PB^*.

- A branch of an intelim tree is *closed* if it contains an S-formula φ and its conjugate $\bar{\varphi}$; otherwise, it is *open*.

- An intelim tree is said to be *closed* when all its branches are closed; otherwise, it is *open*.

- An *intelim proof of* φ *from* X is an intelim tree $\mathcal{T}$ for X such that φ occurs in all open branches of $\mathcal{T}$.

- An *intelim refutation of* X is a closed intelim tree $\mathcal{T}$ for X.

Note that every refutation of X is, simultaneously, a proof of φ from X, for every φ. This because there are no open branches and so the condition that φ occurs at the end of all open branches is vacuously satisfied. This is, of course, a kind of explosivity; but it regards *signed* formulae, and it is compatible with the non-explosivity regarding formulae in **FDE**. The reason of that compatibility is that a set consisting of S-formulae all of the form $\mathsf{T}\,A$ (i.e., formulae which the agent holds that are all **t** or **b**) cannot lead to explosion because there cannot be an intelim refutation of such a set. To begin with, starting from a set $\mathsf{T}\,\Gamma$, there is no way of obtaining S-formulae of the form $\mathsf{F}\,A$ or $\mathsf{T}^*\,A$ by applying only intelim rules. Starting from a set $\mathsf{T}\,\Gamma$, the only way of obtaining formulae of such forms is by applying PB or PB^* and, thus, adding virtual information. Nonetheless, a set $\mathsf{T}\,\Gamma$ cannot lead to explosion even if we add virtual information when unfolding the information contained in $\mathsf{T}\,\Gamma$. In fact, as the following result shows, for a set of S-formulae X to lead to explosion it must contain (in itself) S-formulae of the form $\mathsf{F}\,A$ or $\mathsf{T}^*\,A$, i.e., virtual information.

Proposition 4.5.2. *Any intelim tree for a set* $\mathsf{T}\,\Gamma$ *has at least a branch containing only S-formulae of the form* $\mathsf{T}\,A$ *or* F^*A.

Proof. We use the notation $\begin{array}{c}\mathcal{T}\\ \varphi\end{array}$ to denote either an empty intelim tree or a non-empty intelim tree such that φ is one of its terminal nodes. In turn, given an intelim tree $\mathcal{T}$ for X, we denote the number of assumptions (i.e., S-formulae in X) and outputs of applications of rules by $\delta(\mathcal{T})$. Now, let $\mathcal{T}$ be an arbitrary intelim tree for $\mathsf{T}\,\Gamma$. We proceed by induction on $\delta(\mathcal{T})$:

- Base case: Suppose that $\delta(\mathcal{T}) = 1$. Then both $\mathcal{T}$ and $\mathsf{T}\,\Gamma$ consist of a single S-formula $\mathsf{T}\,A$, and trivially the single branch of $\mathcal{T}$ contains only S-formulae of the form $\mathsf{T}\,A$ or $\mathsf{F}^*\,A$.

- Inductive hypothesis: Suppose that if $\delta(\mathcal{T}) = k$, $k \geq 1$, then $\mathcal{T}$ has at least a branch containing only S-formulae of the form $\mathsf{T}\,A$ or $\mathsf{F}^*\,A$.

- Induction step: Let $\delta(\mathcal{T}) = k+1$. Thus, if $\mathcal{T}$ consists only of assumptions, then the argument is analogous to that of the base case. Now, if $k+1$ refers to the output of a rule, this output was appended to the last node of either a branch of $\mathcal{T}_{n-1}$ containing S-formulae other than of the form $\mathsf{T}\, A$ or $\mathsf{F}^*\, A$ (if any), or a branch of $\mathcal{T}_{n-1}$ containing S-formulae only of the form $\mathsf{T}\, A$ or $\mathsf{F}^*\, A$ (there is at least one by the inductive hypothesis). In the former case, the proposition trivially holds because, regardless of how that branch could be extended, $\mathcal{T}$ still has a branch containing S-formulae only of the form $\mathsf{T}\, A$ or $\mathsf{F}^*\, A$. In the latter case, there are 16 cases depending on which of the possible rules has been applied. We consider only the cases in which the rule is an elimination of negation with major premise $\mathsf{F}^*\, \neg A$, and PB^*; the other cases being similar. Thus, $\mathcal{T}$ respectively is $\begin{array}{c}\mathcal{T}_{n-1}\\ \mathsf{T}\, A\end{array}$ or $\begin{array}{c}\mathcal{T}_{n-1}\\ \mathsf{F}^*\, A\end{array}$, where all the S-formulae in the same branch preceding the respective terminal node displayed are of the form $\mathsf{T}\, A$ or $\mathsf{F}^*\, A$. Therefore, given that by inductive hypothesis the proposition holds for $\mathcal{T}_{n-1}$, it also holds for $\mathcal{T} = \mathcal{T}_n$.

□

The above proposition implies that any intelim tree for a set $\mathsf{T}\,\Gamma$ is open and, thus, that there is no refutation of such a set. This fact regarding our proof-theoretic characterization of **FDE** corresponds to the fact regarding its 4-valued semantics according to which all the elements of any set of formulae can have a designated truth-value. More specifically, it corresponds to the fact that for any formula A there is a **4**-valuation v such that $v(A) = \mathbf{b}$; namely, v such that for all $p \in At$, $v(p) = \mathbf{b}$. Note that a set $\mathsf{T}\,\Gamma$ may well contain $\mathsf{T}\, A$ *and* $\mathsf{T}\,\neg A$, or either of them may well be obtained from that set by applying the rules; which precisely amounts to the formula A having the truth-value $\mathbf{b}$. However, that pair of S-formulae do not close a branch.

Now, as mentioned above, PB and PB^* may introduce formulae of arbitrary degree. However, as in the classical case, the set of formulae that can be used as PB-formulae or PB^*-formulae can be bounded in a variety of ways without loss of completeness. We call this set *virtual space* and define it as a function f of the set $\Gamma \cup \{A\}$, consisting of the premises Γ and of the conclusion A of the given inference. The strictest way of bounding the virtual space consists in allowing as PB-formulae only atomic formulae that occur in $\Gamma \cup \{A\}$. A more liberal option is allowing only subformulae of the formulae in $\Gamma \cup \{A\}$. Specifically, let $\mathcal{F}$ be the set of all functions f on the finite subsets of $F(\mathcal{L})$ such that: (i) for all Δ, $\mathsf{at}(\Delta) \subseteq f(\Delta)$; (ii) $f(\Delta)$ is closed under subformulae, i.e., $\mathsf{sub}(f(\Delta)) = f(\Delta)$; (iii) the *size* of $f(\Delta)$ is bounded above by a polynomial in the size of Δ, i.e., $|f(\Delta)| \leq p(|\Delta|)$ for some fixed polynomial p. (This last requirement will be essential in order to define *tractable* approximations bellow.) The choice of an specific function to yield suitable values of the virtual space for each particular deduction problem is the result of decisions that are conveniently made by the system designer, depending on the intended application.[10] In turn, the functions in $\mathcal{F}$ are partially ordered by the relation $\trianglelefteq$ such that $f_1 \trianglelefteq f_2$ iff, for every finite Δ, $f_1(\Delta) \subseteq f_2(\Delta)$.

Distinguished examples of functions in $\mathcal{F}$ are the identity function $f(\Delta) = \Delta$, sub and at. However, in general, $f(\Delta)$ may contain formulae that are not in sub. For instance, the operation f that maps Δ to the set of all formulae of bounded degree that can be built out of sub and at is also in $\mathcal{F}$. Thus, our intelim method allows for (possibly shorter) deductions that do not have the subformula property (SFP) simply by permitting applications of PB or PB^* to formulae that are not subformulae either of the premises or of the conclusion. However, even in this latter deductions the virtual space is still bounded.

The branching rules are not the unique rules of our intelim method that may bring about violations of the SFP. The introduction rules could in principle be indefinitely applied, leading to ever more complex formulae. Nonetheless, as we shall show below, the application of both kind of rules can be restricted so as to satisfy the SFP. More specifically, we shall show that every intelim proof of φ from X (intelim refutation of X) can be transformed into an intelim proof of φ from X (an intelim refutation of X) with the SFP.

4.5.1 Subformula property

As mentioned in Chapter 3, the subformula property is a key property of logical systems in that it allows us to search for proofs or refutations by *analytic methods*; i.e., by considering solely deduction steps involving formulae that are "contained" in the assumptions, or also in the conclusion in the case of proofs. This implies a drastic reduction of the search space which is crucial for the purpose of automated deduction. When it comes to propositional logics, this search space is finite for each putative inference, paving the way for decision procedures. Particularly, in our intelim method, the SFP guarantees that we can impose a bound on the applications of PB and PB^*, that could in principle be applied to arbitrary formulae, with no loss of deductive power. Similarly, it guarantees that we can impose a bound on the

[10] In the case of the approximations defined below, such decisions affect the deductive power of each given approximation, and so the "speed" at which the approximation process converges to the limiting logic at issue.

$\mathsf{T}^* \neg p^@$	$\mathsf{T}\, p^@$
$\mathsf{T}\, q^@$	$\mathsf{F}\, p^@$
$\mathsf{T}^* \neg r \vee \neg s^@$	$\mathsf{T}\, p \vee q$
$\mathsf{T}\, r^@$	$\mathsf{T}\, q$
$\mathsf{T}\, r \vee p$	
$\mathsf{F}\, p$	
$\mathsf{T}\, r$	
$\mathsf{F}^* \neg r$	
$\mathsf{T}^* \neg s$	

Figure 4.4: Redundant itelim sequences

sensible applications of introduction rules, which could in principle be indefinitely applied, yielding ever more complex formulae.

Definition 4.5.3. An intelim proof $\mathcal{T}$ of φ from X (an intelim refutation of X) has the *subformula property* (SFP) if, for every S-formula ψ occurring in $\mathcal{T}$, $\psi^u \in \mathsf{sub}(X^u \cup \{\varphi^u\})$ ($\psi^u \in \mathsf{sub}(X^u)$).

Now, consider the intelim sequences of Figure 4.4. The first one is a proof of $\mathsf{T}^* \neg s$ from $\{\mathsf{T}^* \neg p, \mathsf{T}\, q, \mathsf{T}^* \neg r \vee \neg s, \mathsf{T}\, r\}$. The second one is a proof of (an arbitrary) $\mathsf{T}\, q$ from $\{\mathsf{T}\, p, \mathsf{F}\, p\}$; i.e., an instance of the explosivity of our intelim method. Note that both proofs are *redundant*. In the first proof, the S-formula $\mathsf{T}\, r \vee p$ is first introduced (from premise $\mathsf{T}\, r$) and then eliminated (using the minor premise $\mathsf{F}\, p$) to re-obtain the S-formula $\mathsf{T}\, r$ which was already contained in the sequence; i.e., this proof contains circular reasoning. In the second proof, the S-formula $\mathsf{T}\, p \vee q$ is first introduced (from premise $\mathsf{T}\, p$) and then eliminated (using $\mathsf{F}\, p$ as minor premise); yet, the sequence was already closed before the application of the disjunction introduction and so, by Def. 4.5.1, the closed sequence $\mathsf{T}\, p, \mathsf{F}\, p$ was already a proof of $\mathsf{T}\, q$ from $\mathsf{T}\, p$ and $\mathsf{F}\, p$.

The same kind of redundancy is present whenever a formula is, simultaneously, the conclusion of an introduction and the major premise of an elimination.

Definition 4.5.4. An occurrence of an S-formula φ in an intelim tree $\mathcal{T}$ is a *detour* if φ is both the conclusion of an introduction and the major premise of an elimination.

Definition 4.5.5. An occurrence of an S-formula φ is *idle* in an intelim tree $\mathcal{T}$ if (i) it is not the terminal node of its branch, (ii) it is not used as premise of some application of an intelim rule, and (iii) it is not the conjugate of some S-formula occurring in the same branch.

Definitions 4.5.6. Given an intelim tree $\mathcal{T}$, a *path* in $\mathcal{T}$ is a finite sequence of nodes such that the first node is the root of $\mathcal{T}$ and each of the subsequent nodes is an immediate successor of the previous one. A path is *closed* if it contains both φ and $\bar{\varphi}$ for some φ.

Note that, according to the above definition, every branch is a maximal path.

Definition 4.5.7. Let $\mathcal{T}$ be an intelim proof of φ from X (an intelim refutation of X). $\mathcal{T}$ is *non-redundant* if it satisfies the following conditions:

1. it contains no idle occurrences of S-formulae;
2. none of its branches contains more than one occurrence of the same S-formula;
3. none of its branches properly includes a closed path.

Observe that if an intelim proof or refutation contains a detour, then either condition 2. or 3. above is violated. Thus:

Lemma 4.5.8. *If an intelim proof or refutation $\mathcal{T}$ is non-redundant, then it contains no detours.*

Proof. Suppose $\mathcal{T}$ contains a detour, i.e., an S-formula φ that is both the conclusion of an introduction and the major premise of an elimination. By inspection of the rules, either the conclusion of the elimination is equal to one of the premises of the introduction, or the minor premise of the elimination is the conjugate of the premise of the introduction and so the branch was already closed before the elimination. In either case, $\mathcal{T}$ is redundant. □

Now, turning an intelim proof or refutation $\mathcal{T}$ into a non-redundant one (with no increase in the size of the proof or refutation) is computationally easy, in that it only involves the following pruning steps:

1. check if there are closed paths and remove whatever follows after them;
2. remove any repetition of S-formulae in the same branch;
3. check if there are idle occurrences of S-formulae, and
4. for each idle occurrence of an S-formula φ:
    - if φ is the conclusion of an application of an intelim rule, just remove φ from $\mathcal{T}$;

- if φ is a virtual assumption introduced by an application of PB or PB^*, remove both φ and the whole subtree generated by its conjugate S-formula $\bar{\varphi}$ introduced in the same application of PB or PB^*; then attach the subtree below φ to the immediate predecessor of φ.

It is easy to verify that the result of this procedure is still an intelim proof of the same conclusion from the same premises, or an intelim refutation of the same assumptions.

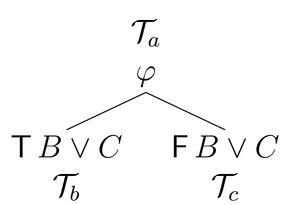
In turn, given a proof $\mathcal{T}$ of φ from X (a refutation of X), and any operation $f \in \mathcal{F}$, we say that an application of PB or PB^* in $\mathcal{T}$ is *f-analytic* if its PB-formula or, respectively, PB^*-formula is in $f(X^u \cup \{\varphi^u\})$ ($f(X^u)$); i.e., in the virtual space defined by the operation f. Recall that the latter is, by definition, closed under subformulae and polynomially bounded. When $f = \mathsf{sub}$, i.e., the virtual space consists exactly of the subformulae of $X^u \cup \{\varphi^u\}$ (X^u), we just say that the application of PB or PB^* is *analytic*. Thus, we can prove the following:

Lemma 4.5.9. *Given any $f \in \mathcal{F}$, every intelim proof $\mathcal{T}$ of φ from X (intelim refutation of X) can be transformed into an intelim proof $\mathcal{T}'$ φ from X (intelim refutation $\mathcal{T}'$ of X) such that every application of PB and PB^* in $\mathcal{T}$ is f-analytic.*

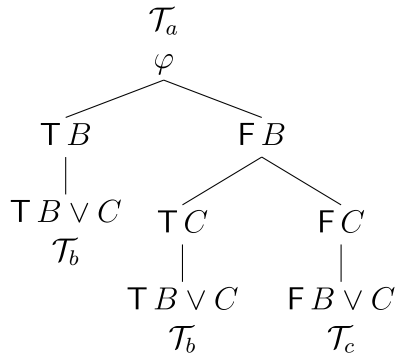
Proof. Again, we use the notation $\begin{array}{c}\mathcal{T}\\ \varphi\end{array}$ to denote either an empty intelim tree or a non-empty intelim tree such that φ is one of its terminal nodes. The proof is by lexicographic induction on $\langle\gamma(\mathcal{T}), \kappa(\mathcal{T})\rangle$, where $\gamma(\mathcal{T})$ denotes the maximum degree of a PB-formula or a PB^*-formula in $\mathcal{T}$ that is not f-analytic, and $\kappa(\mathcal{T})$ denotes the number of occurrences of such non-f-analytic PB-formulae or PB^*-formulae of maximal degree.

Let $\gamma(\mathcal{T}) = m > 0$ and let A be a PB-formula or a PB^*-formula of degree m. There are several cases depending on the logical form of A and on whether A is PB-formula or a PB^*-formula. We sketch only two cases: 1. one where $A = B \vee C$ and A is a PB-formula; 2. another where $A = B \wedge C$ and A is a PB^*-formula; the other cases being similar.

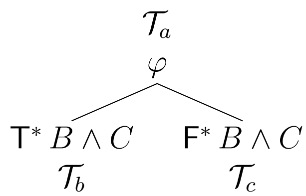
1. $\mathcal{T}$ has the following form:



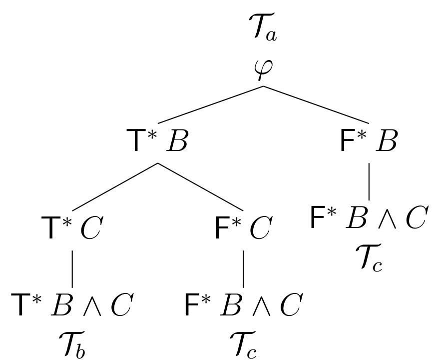
where $\mathcal{T}_b$ and $\mathcal{T}_c$ are intelim trees such that each of their open branches contains φ, or are both closed intelim trees in case $\mathcal{T}$ is a refutation of X. Let $\mathcal{T}'$ be the following intelim tree:



2. $\mathcal{T}$ has the following form:



where, again, $\mathcal{T}_b$ and $\mathcal{T}_c$ are intelim trees such that each of their open branches contains φ, or are both closed intelim trees. Let $\mathcal{T}'$ be the following intelim tree:



Clearly, in both cases, $\mathcal{T}'$ is an intelim proof of φ from X (an intelim refutation of X). Moreover, either $\gamma(\mathcal{T}') < \gamma(\mathcal{T})$, or $\gamma(\mathcal{T}') = \gamma(\mathcal{T})$ and $\kappa(\mathcal{T}') < \kappa(\mathcal{T})$. □

In fact, the transformations used in the proof of the above lemma show that every intelim tree can be turned into an equivalent one in which all the PB-formulae and PB^*-formulae are atomic. Thus, in principle, we could reformulate the notion of intelim tree in such a way that PB and PB^* are applied only to atomic formulae without loss of completeness. Nevertheless, if we demand that the applications of PB

and PB^* be restricted to atomic formulae, the property of being an intelim tree is no longer preserved under uniform substitutions of the atomic formulae occurring in the tree with arbitrary formulae.[11] On the other hand, if we require that the notion of intelim tree be restricted so as to permit only analytic applications of PB and PB^* (i.e., f-analytic applications with $f = \mathsf{sub}$), the property of being an intelim tree is indeed invariant under uniform substitutions.

The following Theorem states the SFP of our intelim method when $f = \mathsf{sub}$:[12]

Theorem 4.5.10 (Generalized SFP). *For every $f \in \mathcal{F}$, if $\mathcal{T}$ is an intelim proof of φ from X (an intelim refutation of X) such that (i) $\mathcal{T}$ is non-redundant, and (ii) every application of PB and PB^* in $\mathcal{T}$ is f-analytic, then for every S-formula ψ occurring in $\mathcal{T}$,*

$$\psi^u \in f(X^u \cup \{\varphi^u\}) \cup \mathsf{sub}(X^u \cup \{\varphi^u\})$$

if $\mathcal{T}$ is a proof of φ from X, or

$$\psi^u \in f(X^u) \cup \mathsf{sub}(X^u)$$

if $\mathcal{T}$ is a refutation of X.

Proof. Let $\mathcal{T}$ be a intelim proof of φ from X (refutation of X) satisfying (i) and (ii), and suppose that there are S-formulae ω in $\mathcal{T}$ such that $\omega^u \notin f(X^u \cup \{\varphi^u\}) \cup \mathsf{sub}(X^u \cup \{\varphi^u\})$ ($\omega^u \notin f(X^u) \cup \mathsf{sub}(X^u)$). Let us call such S-formulae *spurious*. Let ψ be a spurious formula such that ψ^u is of maximal degree in $\mathcal{T}$. Then ψ cannot result from the application of an elimination rule, otherwise $\mathcal{T}$ would contain an spurious formula whose unsigned part is of strictly greater degree; namely, the major premise of this elimination. Moreover, given that $\mathcal{T}$ contains only f-analytic applications of PB and PB^* according to (ii), no spurious S-formula can occur in it as a virtual assumption introduced by an application of PB or PB^*. Therefore ψ must be the conclusion of an introduction. Since, according to (i), $\mathcal{T}$ is non-redundant, it contains no idle occurrences of S-formulae and so either (a) $\psi = \bar{\theta}$ for some θ occurring in the same branch or (b) ψ is used as a premise of a rule application. However, both cases are impossible. Regarding (a), by the same arguments just used for ψ, θ (the conjugate of ψ) can only be the conclusion of an introduction. Then, it is not difficult to see, by inspection of the introduction rules, that case (a) implies that one of the

[11] Moreover, when we define the notion of *depth* of an intelim tree below, it be apparent that each application of the transformations used in the proof of the lemma increases the depth of the tree. So, it is convenient to use them only to the extent in which it is needed to remove applications of PB or PB^* which are not f-analytic.

[12] Note that whenever $\Delta \subseteq f(\Delta)$, then also $\mathsf{sub}(\Delta) \subseteq f(\Delta)$.

premises of the introduction of $\psi = \bar{\theta}$ must be the conjugate of one of the premises of the introduction of θ. So, one of the branches of $\mathcal{T}$ properly contains a closed path, against the assumption that $\mathcal{T}$ is non-redundant. As for (b), first note that ψ cannot be the minor premise of an elimination, otherwise there would be again an spurious formula whose unsigned part is of greater degree in $\mathcal{T}$; namely, the major premise of this elimination. Moreover, ψ cannot be used in $\mathcal{T}$ as major premise of an elimination, otherwise ψ would be a detour and, by Lem. 4.5.8, $\mathcal{T}$ would be redundant, against hypothesis (i). □

4.6 Depth-bounded approximation to FDE

Definition 4.6.1. The *depth* of an intelim tree $\mathcal{T}$ is the maximum number of virtual assumptions occurring in a branch of $\mathcal{T}$. An intelim tree $\mathcal{T}$ is a *k-depth intelim proof* of φ from X (a *k-depth intelim refutation* of X) if $\mathcal{T}$ is an intelim proof of φ from X (an intelim refutation of X) and $\mathcal{T}$ is of depth k.

Note that a 0-depth intelim tree is nothing but an intelim sequence. Examples of, respectively, two proofs of depth 1, a refutation of depth 1,and a proof of depth 2, all with the SFP, are given in Figure 4.5. Again, each assumption is marked with an '@'.

Definitions 4.6.2. For all X, φ,

- φ is *0-depth deducible* from X, $X \vdash_0 \varphi$, iff there is a 0-depth intelim proof of φ from X;
- X is *0-depth refutable*, $X \vdash_0$, iff there is a 0-depth intelim refutation of X.

Notation 4.6.3. We shall abuse of the same relation symbol '$\vdash_0$' to denote 0-depth deducibility and refutability.

Proposition 4.6.4. *$\langle \mathcal{L}, \vdash_0 \rangle$ is a (finitary) Tarskian propositional logic; i.e., $\vdash_0$ satisfies reflexivity, monotonicity, cut, and structurality.*

Proof. The proposition follows easily from the definitions involved, and so here we just outline the proof of monotonicity: Suppose that there is no 0-depth intelim proof of φ from $X \cup Y$. Then, there is no way of obtaining φ by applying intelim rules to the elements of X or the elements of Y. In particular, there is no way of obtaining φ by applying intelim rules to the elements of X alone and, so, there is no 0-depth intelim proof of φ from X. □

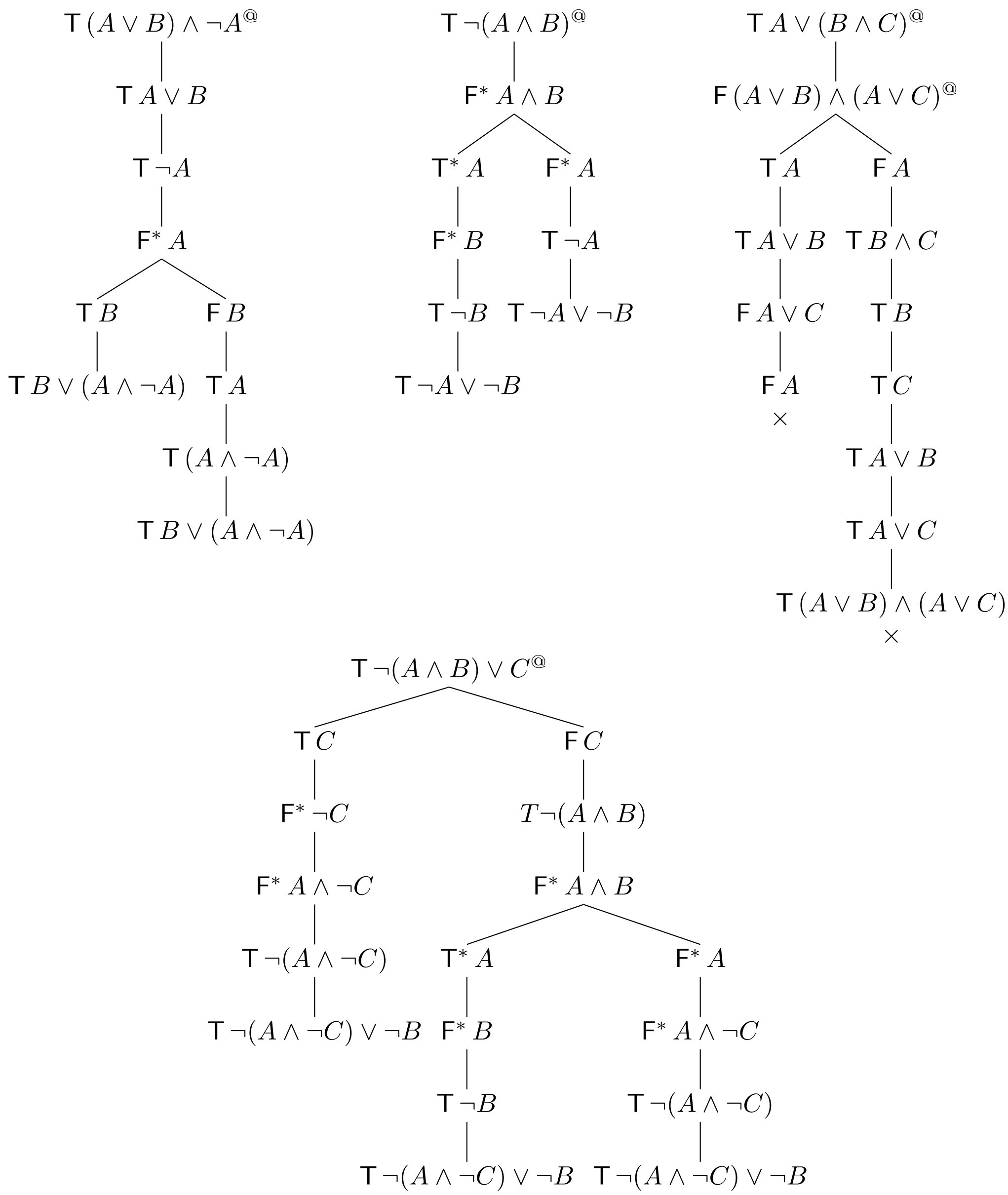


Figure 4.5: k-depth intelim proofs and refutations

Now, the notion of k-depth deducibility depends not only on the depth at which the use of virtual information is recursively allowed, but also on the virtual space discussed and defined above. So, finally:

Definitions 4.6.5. For all X, φ, and for all $f \in \mathcal{F}$,

- $X \vdash_0^f \varphi$ iff $X \vdash_0 \varphi$;
- for $k > 0$, $X \vdash_k^f \varphi$ iff $X \cup \{\psi\} \vdash_{k-1}^f \varphi$ and $X \cup \{\bar{\psi}\} \vdash_{k-1}^f \varphi$ for some $\psi^u \in f(X^u \cup \{\varphi^u\})$.

When $X \vdash_k^f \varphi$, we say that φ is *deducible at depth k from X over the f-bounded virtual space*. Note that the above definition covers also the case of k-depth refutability by assuming $X \vdash_k^f$ as equivalent to $X \vdash_k^f \varphi$ for all φ. Thus:

- $X \vdash_0^f$ iff $X \vdash_0$;
- for $k > 0$, $X \vdash_k^f$ iff $X \cup \{\psi\} \vdash_{k-1}^f$ and $X \cup \{\bar{\psi}\} \vdash_{k-1}^f$ for some $\psi^u \in f(X^u)$.

When $X \vdash_k^f$, we say that X *is refutable at depth k over the f-bounded virtual space.*

Notation 4.6.6. We shall abuse of the same relation symbol '$\vdash_k^f$' to denote k-depth deducibility and refutability over the f-bounded virtual space.

Observe that in the above definition the pair of S-formulae, ψ and $\bar{\psi}$, denote a pair of (conjugate) virtual assumptions introduced by respectively PB or PB^*. Thus, according to the definition, $X \vdash_k^f \varphi$ iff the conclusion φ is obtained at depth $k-1$ by introducing *either* $\mathsf{T}\, A$ and $\mathsf{F}\, A$, *or* $\mathsf{T}^*\, A$ and $\mathsf{F}^*\, A$, as virtual assumptions—for some A in the virtual space defined by f. More specifically, the conclusion φ obtains at depth k iff φ obtains at depth $k-1$ in either case of appending $\mathsf{T}\, A$ or appending $\mathsf{F}\, A$, *or* in either case of appending $\mathsf{T}^*\, A$ or appending $\mathsf{F}^*\, A$, in any branch of the corresponding intelim tree and with respect to some A in the virtual space defined by f. This corresponds to the fact that, in our intelim method, a formula φ may be obtained *at a certain depth* by introducing whichever $\mathsf{T}\, A$ or $\mathsf{F}\, A$ by an application of PB (probably together with the application of intelim rules), but not by introducing $\mathsf{T}^*\, A$ or $\mathsf{F}^*\, A$ by an application of PB^* (together with the application of intelim rules), and vice versa.

Now, it follows immediately from Def. 4.6.1 and 4.6.5 that:

Proposition 4.6.7. *For all X, φ and all $f \in \mathcal{F}$, $X \vdash_k^f \varphi$ ($X \vdash_k^f$) iff there is a k-depth intelim proof of φ from X (a k-depth intelim refutation of X) such that all its PB-formulae and PB^*-formulae are in $f(X^u \cup \{\varphi^u\})$ ($f(X^u)$).*

Proposition 4.6.8. *The k-depth deducibility relations $\vdash_k^f$ satisfy reflexivity, monotonicity, but not cut.*

Proof. The proposition follows easily from the involved definitions. To see why they are not transitive take, for example, $X := \{\mathsf{T}\, A \vee B, \mathsf{T}\, \neg A, \mathsf{T}\, \neg(C \wedge D)\}$, $\varphi := \mathsf{T}\, B \vee (A \wedge \neg A)$, and $\psi := \mathsf{T}\, (B \vee (A \wedge \neg A)) \wedge (\neg C \vee \neg D)$. Then it is easy to check that $X \vdash_1^{\mathsf{sub}} \varphi$ and $X \cup \{\varphi\} \vdash_1^{\mathsf{sub}} \psi$, but $X \nvdash_1^{\mathsf{sub}} \psi$. □

However, it is easy to verify that the relations $\vdash_k^f$ satisfy the following version of cut:

Depth-bounded cut: If $X \vdash_j^f \varphi$ and $X \cup \{\varphi\} \vdash_k^f \psi$, then $X \vdash_{j+k}^f \psi$.

Moreover, the relations $\vdash_k^f$ may not be structural in that structurality depends on the function f that defines the virtual space. For example, $\vdash_k^{\mathsf{sub}}$ is structural, while $\vdash_k^{\mathsf{at}}$ is not. In general, structurality can be imposed by restricting the operations in $\mathcal{F}$ to those such that, for all σ and all Δ, $\sigma(f(\Delta)) \subseteq f(\sigma(\Delta))$. This is not satisfied if $f = \mathsf{at}$, but it is satisfied if $f(\Delta) = \mathsf{sub}(\Delta)$, or $f(\Delta)$ is the set of all formulae of given bounded degree that can be built out of $\mathsf{sub}(\Delta)$. Further, since $\vdash_0$ is monotonic, its successors are ordered: $\vdash_j^f \subseteq \vdash_k^f$ whenever $j \leq k$. The transition from $\vdash_k^f$ to $\vdash_{k+1}^f$ corresponds to an increase in the depth at which the nested use of virtual information—restricted to formulae in the virtual space defined by f—is allowed. Note also that $\vdash_j^{f_1} \subseteq \vdash_k^{f_2}$ whenever $f_1 \trianglelefteq f_2$.

4.6.1 Tractability

We now show that the decision problem for the k-depth logics is tractable. Theorem 4.5.10 immediately suggests a decision procedure for k-depth deducibility: to establish whether φ is k-depth deducible from a finite set X we apply the intelim rules, together with PB and PB^* up to a number k of times, in all possible ways starting from X and restricting to applications which preserve the subformula property. If the resulting intelim tree is closed or φ occurs at the end of all its open branches, then φ is k-depth deducible from X, otherwise it is not. We shall first show the tractability of the 0-depth logic, and then the tractability of the k-depth logics, $k > 0$. Again, we denote by $|X|$ the total number of occurrences of symbols in X.

Theorem 4.6.9. *Whether or not $X \vdash_0 \varphi$ ($X \vdash_0$) can be decided in time $O(n^2)$, where $n = |X \cup \{\varphi\}|$ ($n = |X|$).*

Proof sketch. The proof can be adapted from [59]. We just sketch the decision procedure and give a hint about the upper bound.

We now describe a general procedure to generate the closure of a set Y of signed formulae under the intelim rules restricting our attention to a finite search space Δ that includes all the formulae in Y^u and is closed under subformulae. Start by constructing the subformula graph associated with Δ, i.e., the graph in which the nodes are the subformulae of Δ, while the edges represent the subformula relation. Observe that the number of distinct subformulae of a formula is always less than or equal to the number of occurrences of symbols in that formula. So the number of distinct subformulae of the formulae in Y is $O(n)$ where n is the number of occurrences of symbols in Y. Constructing this graph takes time $O(n^2)$. A *neighbour* of a node A is a node consisting of either (i) one of the immediate subformulae of A (if any), or (ii) one of the immediate superformulae of A (if any), or (iii) else one of the immediate subformulae of the immediate superformulae of A (if any). The number of neighbours of each node is $O(n)$.

Let us say that a node in the subformula graph is associated with a premise (a conclusion) of an intelim rule, if it consists of a formula that is the unsigned part of a premise or of the conclusion. Note that

The relation "A is a neighbour of B" is symmetric. (4.1)

The node associated with a premise of an intelim rule is a neighbour both of the node associated with the second premise (if any) and of the node associated with the conclusion. (4.2)

Nodes are labelled with a subset of the four signs as follows. Initially, all nodes are marked as "fulfilled". Whenever a new sign is added to the labelling set, the node turns "unfulfilled". At the beginning all the nodes consisting of the formulae in X^u are labelled in accordance with their signs in X (and therefore turn "unfulfilled") while all the others are labelled with the empty set. Fulfilling a node means that all the possible intelim rules involving this node and any of its neighbours are applied, which may lead to adding new signs to the labelling sets of the nodes in the neighbours, making them unfulfilled. This amounts to using the formula in the node to be fulfilled, prefixed with each of the signs in its labelling set, as premise of an intelim rule, possibly involving one of its neighbours as second premise. Yet-unfulfilled nodes are fulfilled in turn (the order is immaterial) and marked as such. Since there are $O(n)$ neighbours, fulfilling a node takes $O(n)$ steps.

A node is *inconsistent* if its label contains a pair of conjugate signs, otherwise it is *consistent*. Note that the labelling set of each consistent node may contain at most two signs. Note also that it may be necessary to fulfil a node more than once, when a *new* sign is added to its labelling set as result of an application of an intelim rule to one of its neighbours. However, no consistent node needs to be fulfilled more than twice (once for each sign in its labelling set). To see this, observe that if the procedure leads to adding a new sign to a node n' (e.g., the sign F to A) that may, in turn, be used together with a previously fulfilled neighbour n (e.g., $A \vee B$ signed with T) as premise of an intelim rule, then n' turns unfulfilled, and the rule in question will be applied anyway when fulfilling n'. For, n is a neighbour of n', by (4.1), and so is the node n'' consisting of the conclusion of the rule application (B), by (4.2), whose labelling set will be updated accordingly (adding T).

A *graph* is *inconsistent* if it contains an inconsistent node, otherwise it is *consistent*. In turn, a graph is 0-*depth saturated* if it is either inconsistent or it is consistent and each node is marked as fulfilled. A 0-depth saturated graph is obtained in $O(n^2)$ steps, since there are $O(n)$ nodes in the graph, each node is fulfilled at most twice and fulfilling a node takes $O(n)$ steps. Figure 4.6 shows the initialized graph for the set $X = \{\mathsf{T}\, C \vee (A \vee B), \mathsf{F}\, C, \mathsf{F}\, A \vee (B \wedge \neg C)\}$. The corresponding saturated graph, with a possible order of fulfillment of the nodes, is shown in Figure 4.7. The reader can verify that any alternative sequence leads to the same saturated graph. Figure 4.8 shows the initialized graph for the set $X = \{\mathsf{T}\, A \vee (B \wedge C), \mathsf{F}\, (A \vee B) \wedge (A \vee C), \mathsf{F}\, A\}$. A corresponding saturated graph, with a possible order of fulfillment of the nodes, is shown in Figure 4.9. Note that, for inconsistent graphs, not any alternative sequence leads to the same saturated graph. In general, all the signed formulae φ of the form $\mathsf{S}\, A$, where S is in the labelling set of A, that occur in a saturated graph are 0-depth deducible from X.

To decide whether $X \vdash_0 \varphi$ ($X \vdash_0$), consider the graph associated with $X^u \cup \varphi^u$ (X^u), initialize it by adding signs to the labelling sets in accordance with X, and then run the saturation procedure. When the graph is saturated $X \vdash_0 \varphi$ iff the sign of the signed formula φ belongs to the labelling set of φ^u or the graph is inconsistent. Note that an inconsistent graph detects a "metalevel" inconsistency that concerns an incoherent assignment of the imprecise values associated with the signs. Note also that a 0-depth saturated graph starting with nodes labelled with $\{\mathsf{T}\}$ is always consistent and may contain only the signs T and F^* in the labelling sets. □

Corollary 4.6.10. *Whether or not $X \vdash_k^{\mathsf{sub}} \varphi$ ($X \vdash_k^{\mathsf{sub}}$) can be decided in time $O(n^{k+2})$, where $n = |X \cup \{\varphi\}|$ ($n = |X|$).*

Hint. From Definition 4.6.5 and the observation that there are $O(n)$ distinct subformulae of $X^u \cup \{\varphi^u\}$ (X^u). □

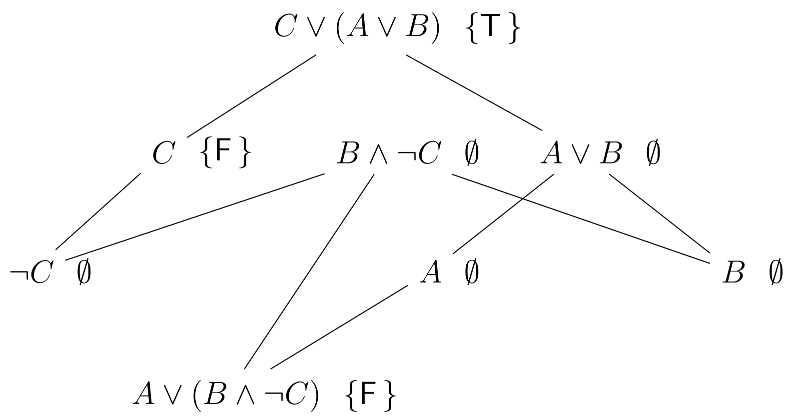


Figure 4.6: Initialized graph

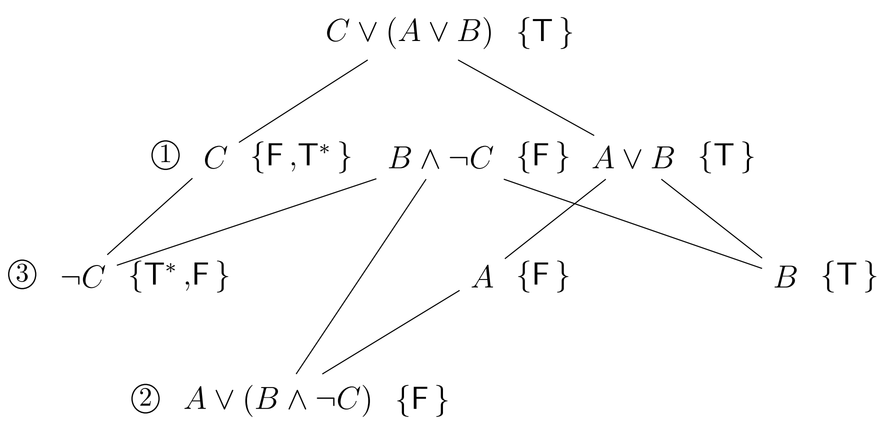


Figure 4.7: Saturated graph

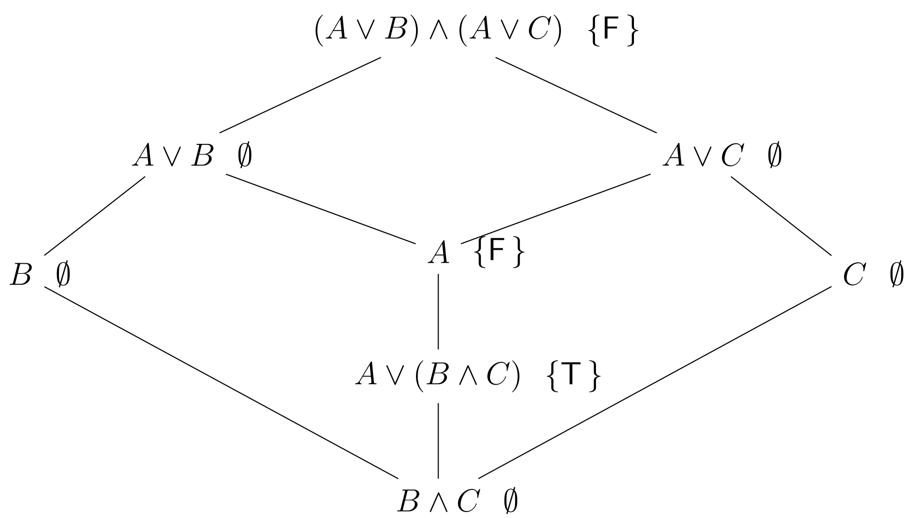


Figure 4.8: Initialized graph

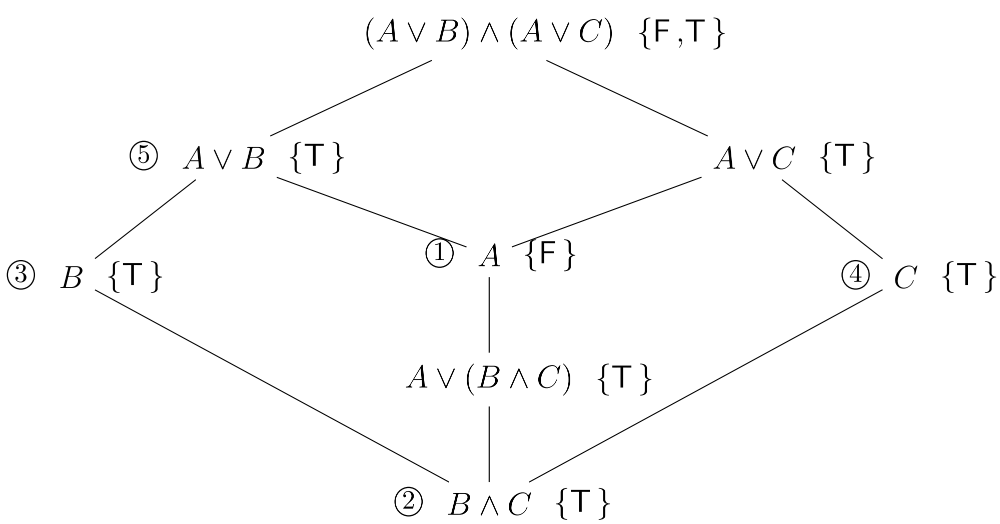


Figure 4.9: Saturated graph

The above Corollary refers to the basic case where the function that defines the virtual space is sub, but is not difficult to generalize it for any polynomially bounded virtual space [see 59, 58]. More precisely, much as in the classical case, when $f \trianglelefteq \mathsf{sub}$, the complexity of the decision problem is $O(n^{k+2})$. In general, the complexity is $O(p(n)^{k+2})$ where p is a polynomial depending on f (recall that, by definition, the virtual space is polynomially bounded).

4.7 5-valued non-deterministic semantics

The signs of our intelim method can be taken as *imprecise* truth-values that intuitively encode partial information about the standard truth-values in **4** [see 18]; namely, two-element sets of the standard truth-values:

$$\mathrm{t} = \{\mathbf{t}, \mathbf{b}\}, \mathrm{f} = \{\mathbf{f}, \mathbf{n}\}, \mathrm{t}^* = \{\mathbf{t}, \mathbf{n}\}, \mathrm{f}^* = \{\mathbf{f}, \mathbf{b}\}.$$

Note that $\mathrm{t} \cap \mathrm{t}^* = \mathbf{t}$ and $\mathrm{f} \cap \mathrm{f}^* = \mathbf{f}$. Let us denote the set of these imprecise truth-values, $\{\mathrm{t}, \mathrm{f}, \mathrm{t}^*, \mathrm{f}^*\}$, by *4*. Now, we can take the elements of *4* as primitive, and use Dunn-style relational semantics [70] to define analogous notions to those of set-up and **4**-valuation:

Definition 4.7.1. A *4-valuation* is a relation $\eta \subseteq At(\mathcal{L}) \times 4$ such that:

i. for no p, $\langle p, \mathrm{t}\rangle$ and $\langle p, \mathrm{f}\rangle$ are both in η;

ii. for no p, $\langle p, \mathrm{t}^*\rangle$ and $\langle p, \mathrm{f}^*\rangle$ are both in η.

Given a *4*-valuation, η, this is extended to a relation $\eta \subseteq F(\mathcal{L}) \times 4$ by recursive clauses:

$$\neg A\eta\mathrm{t} \text{ iff } A\eta\mathrm{f}^*;$$
$$\neg A\eta\mathrm{t}^* \text{ iff } A\eta\mathrm{f};$$

$$A \wedge B\eta\mathrm{t} \text{ iff } A\eta\mathrm{t} \text{ and } B\eta\mathrm{t};$$
$$A \wedge B\eta\mathrm{t}^* \text{ iff } A\eta\mathrm{t}^* \text{ and } B\eta\mathrm{t}^*;$$
$$A \wedge B\eta\mathrm{f} \text{ iff } A\eta\mathrm{f} \text{ or } B\eta\mathrm{f};$$
$$A \wedge B\eta\mathrm{f}^* \text{ iff } A\eta\mathrm{f}^* \text{ or } B\eta\mathrm{f}^*;$$

$$A \vee B\eta\mathrm{t} \text{ iff } A\eta\mathrm{t} \text{ or } B\eta\mathrm{t};$$
$$A \vee B\eta\mathrm{t}^* \text{ iff } A\eta\mathrm{t}^* \text{ or } B\eta\mathrm{t}^*;$$
$$A \vee B\eta\mathrm{f} \text{ iff } A\eta\mathrm{f} \text{ and } B\eta\mathrm{f};$$
$$A \vee B\eta\mathrm{f}^* \text{ iff } A\eta\mathrm{f}^* \text{ and } B\eta\mathrm{f}^*.$$

The above notion would yield yet another alternative semantics for *full* **FDE**. However, we introduce it here only as a first step towards devising a semantics for the depth-bounded approximations defined proof-theoretically above.

4.7.1 The 0-depth logic

Within our conceptual framework, the truth-value of formulae may be completely *undefined* when the agent's information of Ω is insufficient to even establish any of the imprecise truth-values. As mentioned above, there is no reason to assume that an agent is "told" about the values of atoms only. In most practical contexts, it may well be that the sources inform the agent that a certain disjunction is true without informing her which of the two disjuncts is the true one, or, analogously, that a certain conjunction is false without informing her which of the two conjuncts is the false one.

We shall denote $A\eta\bot$ whenever η is *undefined* for A. It is technically convenient to treat $\bot$ as a fifth truth-value. So, let us denote by *5* the set consisting of the elements of *4* together with $\bot$. Intuitively, $\bot$ may eventually turn into an imprecise or even a standard truth-value by the development of the agent's reasoning or querying process.[13] Thus, we take the five truth-values as partially ordered by two relations: (i) $\preceq_a$ such that $x \preceq_a y$ (read "x is less defined than, or equal to, y") iff $x = \bot$ or $x = y$ for $x, y \in \{\mathrm{t}, \mathrm{f}, \bot\}$; (ii) $\preceq_b$ such that $x \preceq_b y$ iff $x = \bot$ or $x = y$ for $x, y \in \{\mathrm{t}^*, \mathrm{f}^*, \bot\}$.

Definition 4.7.2. A *5 non-deterministic valuation* is a relation $\eta' : F(\mathcal{L}) \times 2^{\mathit{5}}$ such that:

For no formula A, and $S_1, S_2 \in 2^{\mathit{5}}$, it is the case that:

i. $A\eta' S_1$, $A\eta' S_2$ and $\{\mathrm{t}, \mathrm{f}\} \subseteq S_1 \cup S_2$;

ii. $A\eta' S_1$, $A\eta' S_2$ and $\{\mathrm{t}^*, \mathrm{f}^*\} \subseteq S_1 \cup S_2$.

Moreover:

$$\neg A\eta'\{\mathrm{f}^*\} \text{ iff } A\eta\mathrm{t};$$
$$\neg A\eta'\{\mathrm{t}^*\} \text{ iff } A\eta\mathrm{f};$$
$$\neg A\eta'\{\mathrm{f}\} \text{ iff } A\eta\mathrm{t}^*;$$

[13] Note that **n** and $\bot$ denote two different notions. While **n** intuitively means that the agent *knows* that she is told nothing about the truth-value of the (atomic) formula at issue, $\bot$ intuitively means that she *does not know yet* whether she is told something about the truth-value of the formula under consideration or not. That is, $\bot$ denotes (full) *ignorance* about defined truth-values.

$$\neg A\eta'\{\mathrm{t}\} \text{ iff } A\eta\mathrm{f}^*;$$
$$\neg A\eta'\{\bot\} \text{ iff } A\eta\bot;$$

$$A \wedge B\eta'\{\mathrm{f}\} \text{ iff } A\eta\mathrm{f} \text{ or } B\eta\mathrm{f};$$
$$A \wedge B\eta'\{\mathrm{f}^*\} \text{ iff } A\eta\mathrm{f}^* \text{ or } B\eta\mathrm{f}^*;$$
$$A \wedge B\eta'\{\mathrm{f} \cap \mathrm{f}^*\} \text{ iff } A\eta\mathrm{f} \text{ and } B\eta\mathrm{f}^*, \text{ or } A\eta\mathrm{f}^* \text{ and } B\eta\mathrm{f};$$
$$A \wedge B\eta'\{\mathrm{t}\} \text{ iff } A\eta\mathrm{t} \text{ and } B\eta\mathrm{t};$$
$$A \wedge B\eta'\{\mathrm{t}^*\} \text{ iff } A\eta\mathrm{t}^* \text{ and } B\eta\mathrm{t}^*;$$
$$A \wedge B\eta'\{\bot\} \text{ iff } A\eta\mathrm{t} \text{ and } B\eta\mathrm{t}^*, \text{ or } A\eta\mathrm{t}^* \text{ and } B\eta\mathrm{t};$$
$$A \wedge B\eta'\{\bot, \mathrm{f}^*\} \text{ iff } A\eta\mathrm{t} \text{ and } B\eta\bot, \text{ or } A\eta\bot \text{ and } B\eta\mathrm{t};$$
$$A \wedge B\eta'\{\bot, \mathrm{f}\} \text{ iff } A\eta\mathrm{t}^* \text{ and } B\eta\bot, \text{ or } A\eta\bot \text{ and } B\eta\mathrm{t}^*;$$
$$A \wedge B\eta'\{\mathrm{f}, \mathrm{f}^*, \bot\} \text{ iff } A\eta\bot \text{ and } B\eta\bot;$$

$$A \vee B\eta'\{\mathrm{t}\} \text{ iff } A\eta\mathrm{t} \text{ or } B\eta\mathrm{t};$$
$$A \vee B\eta'\{\mathrm{t}^*\} \text{ iff } A\eta\mathrm{t}^* \text{ or } B\eta\mathrm{t}^*;$$
$$A \vee B\eta'\{\mathrm{t} \cap \mathrm{t}^*\} \text{ iff } A\eta\mathrm{t} \text{ and } B\eta\mathrm{t}^*, \text{ or } A\eta\mathrm{t}^* \text{ and } B\eta\mathrm{t};$$
$$A \vee B\eta'\{\mathrm{f}\} \text{ iff } A\eta\mathrm{f} \text{ and } B\eta\mathrm{f};$$
$$A \vee B\eta'\{\mathrm{f}^*\} \text{ iff } A\eta\mathrm{f}^* \text{ and } B\eta\mathrm{f}^*;$$
$$A \vee B\eta'\{\bot\} \text{ iff } A\eta\mathrm{f} \text{ and } B\eta\mathrm{f}^*, \text{ or } A\eta\mathrm{f}^* \text{ and } B\eta\mathrm{f};$$
$$A \vee B\eta'\{\bot, \mathrm{t}^*\} \text{ iff } A\eta\mathrm{f} \text{ and } B\eta\bot, \text{ or } A\eta\bot \text{ and } B\eta\mathrm{f};$$
$$A \vee B\eta'\{\bot, \mathrm{t}\} \text{ iff } A\eta\mathrm{f}^* \text{ and } B\eta\bot, \text{ or } A\eta\bot \text{ and } B\eta\mathrm{f}^*;$$
$$A \vee B\eta'\{\mathrm{t}, \mathrm{t}^*, \bot\} \text{ iff } A\eta\bot \text{ and } B\eta\bot.$$

Now, this 5-valued relational semantics *à la* Dunn can be summarized by the 5-valued non-deterministic truth-tables (5N-tables, for short) in Table 4.10. Thus, we are in a position to introduce the following notion:

Definition 4.7.3. Let $\mathcal{M}_{dfde}$ be the Nmatrix for $\mathcal{L}$, where $\mathcal{V} = \mathit{5}$, $\mathcal{D} = \{\mathrm{t}\}$ and the functions in $\mathcal{O}$ are defined by the 5N-tables in Table 4.10.[14]

Therefore, using the 5N-tables, a 5 non-deterministic valuation can be defined as a function. Namely, a *5-valuation* for $\mathcal{L}$ is a function $v : F(\mathcal{L}) \longrightarrow \mathit{5}$. Then, we pick out from the set of all *5*-valuations those which agree with the intended meaning of the connectives via $\mathcal{M}_{dfde}$:

Definition 4.7.4. A *5N-valuation* is a *5*-valuation v such that for all $A, B \in F(\mathcal{L})$:

1. $v(\neg A) = \tilde{\neg}(v(A))$;
2. $v(A \circ B) \in \tilde{\circ}(v(A), v(B))$.

[14]Owing to the logical symmetry between **b** and **n**, we can alternatively take $\mathcal{D} = \{\mathrm{t}^*\}$.

$\widetilde{\vee}$	t	f	t*	f*	⊥
t	{t}	{t}	{**t**}	{t}	{t}
f	{t}	{f}	{t*}	{⊥}	{⊥, t*}
t*	{**t**}	{t*}	{t*}	{t*}	{t*}
f*	{t}	{⊥}	{t*}	{f*}	{⊥, t}
⊥	{t}	{⊥, t*}	{t*}	{⊥, t}	{t, t*, ⊥}

$\widetilde{\wedge}$	t	f	t*	f*	⊥
t	{t}	{f}	{⊥}	{f*}	{⊥, f*}
f	{f}	{f}	{f}	{**f**}	{f}
t*	{⊥}	{f}	{t*}	{f*}	{⊥, f}
f*	{f*}	{**f**}	{f*}	{f*}	{f*}
⊥	{⊥, f*}	{f}	{⊥, f}	{f*}	{f, f*, ⊥}

$\widetilde{\neg}$	
t	f*
f	t*
t*	f
f*	t
⊥	⊥

Table 4.10: 5N-tables

Where $\circ$ is $\vee$ or $\wedge$.

Remark 11. A 5N-valuation can be seen as describing an *information state* that is closed under the implicit information that depends only on the informational meaning of the connectives. This is information that the agent holds and with which she can operate, in the precise sense that she has a feasible procedure to decide, for every A, whether the information that A is t or f (analogously, t* or f*), or neither of them actually belongs to her information state.

Now, the following definitions are respectively analogous to Definitions 4.2.3 and 4.3.4:

Definition 4.7.5. Given a 5N-valuation v, we say that a formula A is:

- *at least true* under v iff $v(A) = \mathrm{t}$;
- *non-true* under v iff $v(A) = \mathrm{f}$;
- *non-false* under v iff $v(A) = \mathrm{t}^*$;
- *at least false* under v iff $v(A) = \mathrm{f}^*$.

Definition 4.7.6. A 5N-valuation v *realizes* a S-formula

- $\mathsf{T}\,A$ iff A is at least true under v;
- $\mathsf{F}\,A$ iff A is non-true under v;
- $\mathsf{T}^*\,A$ iff A is non-false under v;
- $\mathsf{F}^*\,A$ iff A is at least false under v.

A set X is said to be *5N-realizable* if there is a 5N-valuation v which realizes every element of X.

Definitions 4.7.7. For all X, φ,

- φ is a *0-depth consequence* of X, $X \vDash_0 \varphi$, iff for every 5N-valuation v, v realizes φ whenever v realizes all the elements of X;
- X is *0-depth inconsistent*, $X \vDash_0$, iff it is not 5N-realizable.

Analogously to the explosivity of the proof-theoretic characterization above, this explosivity regards inconsistent sets of *signed* formulae. Recall our explanation above of why this is compatible with the non-explosivity regarding (unsigned) formulae.

Example 1. $\{\mathsf{T}\,\neg(A \vee B), \mathsf{T}\,\neg C\} \vDash_0 \mathsf{T}\,\neg(A \vee C)$
By inspection of the 5N-tables, it is easy to check that for any 5N-valuation v s.t. v realizes both $\mathsf{T}\,\neg(A \vee B)$ and $\mathsf{T}\,\neg C$, then v also realizes $\mathsf{T}\,\neg(A \vee C)$.

Example 2. $\{\mathsf{T}\,(A \vee B) \wedge \neg A\} \nvDash_0 \mathsf{T}\,B \vee (A \wedge \neg A)$
Let v be a 5N-valuation s.t. $v(A) = \text{f}^*$, $v(B) = \bot$, $v(A \vee B) = v(\neg A) = v((A \vee B) \wedge \neg A) = \text{t}$, $v(A \wedge \neg A) = \text{f}^*$ and $v(B \vee (A \wedge \neg A)) = \bot$.

Example 3. $\{\mathsf{T}\,\neg(A \wedge B)\} \nvDash_0 \mathsf{T}\,\neg A \vee \neg B$
Let v be a 5N-valuation s.t. $v(A) = \text{t}$, $v(B) = v(\neg B) = \bot$, $v(A \wedge B) = v(\neg A) = \text{f}^*$, $v(\neg(A \wedge B)) = \text{t}$ and $v(\neg A \vee \neg B) = \bot$.

Example 4. $\{\mathsf{T}\,(A \vee B) \wedge \neg A, \mathsf{F}\,B \vee (A \wedge \neg A)\} \vDash_0$
By inspection of the 5N-tables, it is easy to check that there is no 5N-valuation v which realizes both $\mathsf{T}\,(A \vee B) \wedge \neg A$ and $\mathsf{F}\,B \vee (A \wedge \neg A)$.

Example 5. $\{\mathsf{T}\,A \vee (B \wedge C), \mathsf{F}\,(A \vee B) \wedge (A \vee C)\} \nvDash_0$
Let v be a 5N-valuation s.t. $v(A) = \text{f}^*$, $v(B) = v(A \vee B) = \text{t}^*$, $v(C) = v(A \vee C) = v(B \wedge C) = \bot$, $v(A \vee (B \wedge C)) = \text{t}$, and $v((A \vee B) \wedge (A \vee C)) = \text{f}$.

Let us now show the adequacy of our informational 5-valued non-deterministic semantics with respect to the relation $\vdash_0$.

Proposition 4.7.8. *For all X and φ,*

$$X \vDash_0 \varphi \text{ iff } X \vdash_0 \varphi.$$

Proof.
The soundness of the intelim rules can be immediately verified by inspection of the 5N-tables: every 5N-valuation which realizes the premise(s) of an intelim rule realizes also the conclusion of the rule. For example, if an agent holds the information that both A and B are t^*, then she also holds the information that $A \wedge B$ is t^*, since the 5N-table for $\wedge$ excludes the other imprecise values. Thereby, it follows by an elementary inductive argument that, if a 5N-valuation v realizes all the initial S-formulae of a 0-depth intelim tree $\mathcal{T}$ (i.e., an intelim sequence), then v realizes all the S-formulae occurring in $\mathcal{T}$. But, of course, no 5N-valuation can realize two conjugate S-formulae simultaneously. Thus, if $\mathcal{T}$ is a closed intelim tree, no 5N-valuation can realize all the initial S-formulae of $\mathcal{T}$. Therefore, on the one hand, if $\mathcal{T}$ is a 0-depth proof of φ from X, then for every 5N-valuation v, v realizes φ whenever v realizes all the elements of X. On the other hand, if $\mathcal{T}$ is a 0-depth refutation of X, then no 5N-valuation v realizes all the elements of X.

As for completeness, suppose that $X \nvdash_0 \varphi$. Then X is not 0-depth refutable; otherwise, by definition of 0-depth intelim proof, it should hold that $X \vdash_0 \varphi$, contrary to our hypothesis. Next, consider the set $Y = \{\psi | X \vdash_0 \psi\}$. Since X is not 0-depth refutable, for no A, $\mathsf{S}A$ and $\bar{\mathsf{S}}A$ are both in Y. Then, it is not difficult to verify that the function v defined as follows:

$$v(A) = \begin{cases} \text{t} & \text{if } \mathsf{T}\, A \in Y \\ \text{f} & \text{if } \mathsf{F}\, A \in Y \\ \text{t}^* & \text{if } \mathsf{T}^*\, A \in Y \\ \text{f}^* & \text{if } \mathsf{F}^*\, A \in Y \\ \bot & \text{otherwise} \end{cases}$$

is a 5N-valuation. Here we just outline a typical case. Suppose $v(A) = v(B) = \bot$. Then, $\mathsf{F}\, A \vee B \notin Y$. Otherwise, if $\mathsf{F}\, A \vee B \in Y$ then, by definition of Y and by the corresponding elimination rule for $\vee$, $\mathsf{F}\, A$ and $\mathsf{F}\, B$ should also be in Y. Hence, by definition of v, $v(A) = v(B) = \text{f}$, against our assumption. Thus, by the 5N-table for $\vee$, $v(A \vee B) \neq \text{f}$. Analogously, $\mathsf{F}^*\, A \vee B \notin Y$. Otherwise, if $\mathsf{F}^*\, A \vee B \in Y$ then, $\mathsf{F}^*\, A$ and $\mathsf{F}^*\, B$ should also be in Y. So, by definition of v, $v(A) = v(B) = \text{f}^*$, against our assumption. Then, by the 5N-table for $\vee$, $v(A \vee B) \neq \text{f}^*$. On the other hand, $\mathsf{T}\, A \vee B$ or $\mathsf{T}^*\, A \vee B$, may or may not belong to Y, and so $v(A \vee B) = \text{t}$, $v(A \vee B) = \text{t}^*$, or $v(A \vee B) = \bot$. Finally, observe that: (i) $\psi \in Y$ for all $\psi \in X$ and so, by definition of v, v realizes all $\psi \in X$; (ii) by the hypothesis that $X \nvdash_0 \varphi$, $\varphi \notin Y$ and so v does not realize φ. Therefore, $X \nvDash_0 \varphi$. □

Corollary 4.7.9. *For all X,*

$$X \vDash_0 \textit{ iff } X \vdash_0.$$

4.7.2 k-depth logics

Examples 2, and 3 above are valid inferences in **FDE** that are not so in the 0-depth approximation. Again, the latter is simply the logic of deductive reasoning restricted to the use of actual information. For those valid inferences that cannot be justified solely by the meaning of the connectives—i.e., by the 5N-tables—the incorporation of virtual information is required. This is information that is not even potentially contained in the current information state. Accordingly, the k-depth logics, $k \geq 0$, require the simulation of virtual extensions of the current information state. These extensions are formally defined through the following notion:

Definition 4.7.10. Let v, w be 5N-valuations. Then, w is a *5-refinement* of v, $v \sqsubseteq_5 w$, iff $v(A) \preceq_a w(A)$ or $v(A) \preceq_b w(A)$ for all A.

Now, the following definition mimics Definition 4.6.5:

Definition 4.7.11. For all X, φ, and for all $f \in \mathcal{F}$,

- $X \vDash_0^f \varphi$ iff $X \vDash_0 \varphi$;
- for $k > 0$, $X \vDash_k^f \varphi$ iff $X \cup \{\psi\} \vDash_{k-1}^f \varphi$ and $X \cup \{\bar{\psi}\} \vDash_{k-1}^f \varphi$ for some $\psi^u \in f(X^u \cup \{\varphi^u\})$.

When $X \vDash_k^f \varphi$ ($X \vDash_k^f$), we say that φ is a *k-depth consequence of X* (*X is k-depth inconsistent*) *over the f-bounded virtual space*.

As Definition 4.6.5, the above covers the case of k-depth inconsistency by assuming $X \vDash_k^f$ as equivalent to $X \vDash_k^f \varphi$ for all φ. Moreover, according with the above definition, $X \vDash_k^f \varphi$ iff by simulating *either* a pair of refinements (of the current information state) in which the truth-value of some A (in the virtual space defined by f) is respectively t or f, *or* a pair of refinements in in which the truth-value of some A is respectively t* or f*, the conclusion φ is realized by either of the members of the pair at depth $k-1$.[15] That use of a defined truth-value for A, which is not even potentially contained in the current information state, is what we call virtual information.

Example 6. $\{\mathsf{T}\,(A \vee B) \wedge \neg A\} \vDash_1^{\mathsf{sub}} \mathsf{T}\,B \vee (A \wedge \neg A)$
It is easy to check that $\{\mathsf{T}\,(A \vee B) \wedge \neg A\} \cup \{\mathsf{S}B\} \vDash_0 \mathsf{T}\,B \vee (A \wedge \neg A)$, and $\{\mathsf{T}\,(A \vee B) \wedge \neg A\} \cup \{\bar{\mathsf{S}}B\} \vDash_0 \mathsf{T}\,B \vee (A \wedge \neg A)$.

Example 7. $\{\mathsf{T}\,\neg(A \wedge B)\} \vDash_1^{\mathsf{sub}} \mathsf{T}\,\neg A \vee \neg B$
It is easy to check that $\{\mathsf{T}\,\neg(A \wedge B)\} \cup \{\mathsf{T}^*\,A\} \vDash_0 \mathsf{T}\,\neg A \vee \neg B$ and $\{\mathsf{T}\,\neg(A \wedge B)\} \cup \{\mathsf{F}^*\,A\} \vDash_0 \mathsf{T}\,\neg A \vee \neg B$.

Example 8. $\{\mathsf{T}\,A \vee (B \wedge C), \mathsf{F}\,(A \vee B) \wedge (A \vee C)\} \vDash_1^{\mathsf{sub}}$
It is to check that $\{\mathsf{T}\,A \vee (B \wedge C), \mathsf{F}\,(A \vee B) \wedge (A \vee C)\} \cup \{\mathsf{T}\,A\} \vDash_0$ and $\{\mathsf{T}\,A \vee (B \wedge C), \mathsf{F}\,(A \vee B) \wedge (A \vee C)\} \cup \{\mathsf{F}\,A\} \vDash_0$.

Now, the next proposition follows from the fact that RE_{fde} is sound and complete for full **FDE** [53]:

Proposition 4.7.12. *For all X, φ, and all $f \in \mathcal{F}$,*

$$X \vDash_k^f \varphi \text{ iff } X \vdash_k^f \varphi.$$

[15] Analogously to Def. 4.6.5, an S-formula φ may be realized *at certain depth* by one of those pairs of refinements but not by the other.

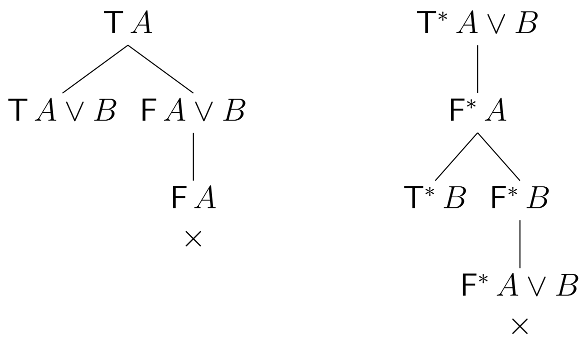


Figure 4.10: Simulating an introduction on the left and an elimination on the right

The above given that RE_{fde} is a subsystem of our intelim method for unbounded k; i.e., a subsystem of the system constituted by the intelim rules together with an arbitrary number of applications of PB and PB^*. Indeed, the elimination rules together with PB and PB^* can be used to simulate any of the introduction rules. Conversely, the introduction rules together with PB and PB^* can be used to simulate any of the elimination rules. It is easy to see how to make these simulations and, thus, here we just show two examples in Fig. 4.10, the other cases being similar. This clearly implies that the direct-proof system constituted by the introduction rules together with PB and PB^*—let us call it RI_{fde}—is also complete for full **FDE**. Nevertheless, there are two reasons for using both introduction and elimination rules: (i) it allows for more natural and shorter proofs, although not essentially shorter because the corresponding simulation is polynomial; (ii) it reduces the number of applications of PB and PB^* that, as stated above, is key to define the *depth* of an inference. In fact, regarding (i):[16]

Proposition 4.7.13. *RE_{fde} and RI_{fde} can linearly simulate each other. Moreover, the simulation preserves the subformula property.*

Proof. We denote the number of nodes of a tree $\mathcal{T}$ by $\lambda(\mathcal{T})$. Now, if $\mathcal{T}$ is an RE_{fde}-tree with assumptions in $X \cup \{\bar{\varphi}\}$ then replace each application of an RE_{fde}-rule with its RI_{fde}-simulation (the applications of PB and PB^* can be left unchanged since both are also RI_{fde}-rules). The result is an RI_{fde}-tree $\mathcal{T}'$ containing at most $\lambda(\mathcal{T}) + c \cdot \lambda(\mathcal{T})$ nodes, where c is the maximum number of additional nodes generated by an RI_{fde}-simulation of an RE_{fde}-rule; namely, 2. Thus, if $\mathcal{T}$ is a closed RE_{fde}-tree for $X \cup \{\bar{\varphi}\}$, then $\mathcal{T}'$ is a closed RI_{fde}-tree for $X \cup \{\bar{\varphi}\}$. Moreover, $\lambda(\mathcal{T}') \leq 3\lambda(\mathcal{T})$

[16]The notions of RI_{fde}-tree (-proof) should be clear.

and $\mathcal{T}'$ does not contain any formula which does not occur in $\mathcal{T}$. Hence, by applying PB or PB^* to φ^u, we obtain the required RI_{fde}-proof of φ from X.

In turn, if $\mathcal{T}$ is a RI_{fde}-tree with assumptions X, then replace each application of a RI_{fde}-rule with its RE_{fde}-simulation (again, the applications of PB and PB^* can be left unchanged since both are also RE_{fde}-rules). The result is a RE_{fde}-tree $\mathcal{T}'$ containing at most $\lambda(\mathcal{T}) + c \cdot \lambda(\mathcal{T})$ nodes, where c is the maximum number of additional nodes generated by an RE_{fde}-simulation of an RI_{fde}-rule; namely, 2. Thus, if $\mathcal{T}$ is a RI_{fde}-proof of φ from X, $\mathcal{T}'$ is a RE_{fde}-tree such that φ occurs in all its open branches. Moreover, $\lambda(\mathcal{T}') \leq 3\lambda(\mathcal{T})$ and $\mathcal{T}'$ does not contain any formulae which do not occur in $\mathcal{T}$. Simply adding $\bar{\varphi}$ to the assumptions will yield a closed RE_{fde}-tree for $X \cup \bar{\varphi}$. □

Analogously to the case of *KE* and *KI* in **CPL** (see Chapter 3 above), and as Propositions 4.7.12 and 4.7.13 suggest, RE_{fde} (RI_{fde}) may serve as the basis for defining depth-bounded approximations to **FDE**. The reason for using both introduction and elimination rules were stated above; however, as a refutation method, RE_{fde} may be still preferred for applications in automated reasoning.

4.8 A natural deduction variant

We present a variant of the proof system that we used to define the hierarchy of approximations to **FDE**; variant which is closer to Gentzen-Prawitz style natural deduction. We use again signed formulae (interpreted as before) and assume that $\mathcal{L}$ contains also the logical constant $\curlywedge$, denoting "the falsum" and intended as an absurd proposition. We think of falsum as a marker that a contradiction has been reached and display the rules concerning it in Tab. 4.13. Unlike the intelim rules of standard natural deduction, none of our intelim rules, shown in Tab. 4.11 and 4.12, is a discharge rule. Consequently, our intelim rules are not complete for full **FDE**, but only for the 0-depth logic of the hierarchy. To obtain a complete set of rules it is sufficient to add two discharge rules corresponding to PB and PB^*, which are displayed in Tab. 4.14.

If we allow unbounded applications of the discharge rules, then:

Definition 4.8.1. A *proof of φ depending on X* is a tree of occurrences of formulae constructed in accordance with the rules in Tab. 4.11 – 4.14, such that φ occurs at the root and X is the set of all undischarged assumptions that occur at the leaves.

Definition 4.8.2. We say that φ is *deducible from X*, $X \vdash \varphi$, if there is a proof of φ depending on $Y \subseteq X$.

$$\frac{\mathsf{F}\,A}{\mathsf{F}\,A \wedge B} \qquad \frac{\mathsf{F}\,B}{\mathsf{F}\,A \wedge B} \qquad \frac{\mathsf{F}^*\,A}{\mathsf{F}^*\,A \wedge B} \qquad \frac{\mathsf{F}^*\,B}{\mathsf{F}^*\,A \wedge B}$$

$$\frac{\mathsf{T}\,A}{\mathsf{T}\,A \vee B} \qquad \frac{\mathsf{T}\,B}{\mathsf{T}\,A \vee B} \qquad \frac{\mathsf{T}^*\,A}{\mathsf{T}^*\,A \vee B} \qquad \frac{\mathsf{T}^*\,B}{\mathsf{T}^*\,A \vee B}$$

$$\frac{\mathsf{T}\,A \quad \mathsf{T}\,B}{\mathsf{T}\,A \wedge B} \qquad \frac{\mathsf{F}\,A \quad \mathsf{F}\,B}{\mathsf{F}\,A \vee B} \qquad \frac{\mathsf{T}^*\,A \quad \mathsf{T}^*\,B}{\mathsf{T}^*\,A \wedge B} \qquad \frac{\mathsf{F}^*\,A \quad \mathsf{F}^*\,B}{\mathsf{F}^*\,A \vee B}$$

$$\frac{\mathsf{T}\,A}{\mathsf{F}^*\,\neg A} \qquad \frac{\mathsf{F}\,A}{\mathsf{T}^*\,\neg A} \qquad \frac{\mathsf{T}^*\,A}{\mathsf{F}\,\neg A} \qquad \frac{\mathsf{F}^*\,A}{\mathsf{T}\,\neg A}$$

Table 4.11: Introduction rules for the standard **FDE** connectives

$$\frac{\mathsf{F}\,A \wedge B \quad \mathsf{T}\,A}{\mathsf{F}\,B} \qquad \frac{\mathsf{F}\,A \wedge B \quad \mathsf{T}\,B}{\mathsf{F}\,A} \qquad \frac{\mathsf{F}^*\,A \wedge B \quad \mathsf{T}^*\,A}{\mathsf{F}^*\,B} \qquad \frac{\mathsf{F}^*\,A \wedge B \quad \mathsf{T}^*\,B}{\mathsf{F}^*\,A}$$

$$\frac{\mathsf{T}\,A \wedge B}{\mathsf{T}\,A} \qquad \frac{\mathsf{T}\,A \wedge B}{\mathsf{T}\,B} \qquad \frac{\mathsf{T}^*\,A \wedge B}{\mathsf{T}^*\,A} \qquad \frac{\mathsf{T}^*\,A \wedge B}{\mathsf{T}^*\,B}$$

$$\frac{\mathsf{T}\,A \vee B \quad \mathsf{F}\,A}{\mathsf{T}\,B} \qquad \frac{\mathsf{T}\,A \vee B \quad \mathsf{F}\,B}{\mathsf{T}\,A} \qquad \frac{\mathsf{T}^*\,A \vee B \quad \mathsf{F}^*\,A}{\mathsf{T}^*\,B} \qquad \frac{\mathsf{T}^*\,A \vee B \quad \mathsf{F}^*\,B}{\mathsf{T}^*\,A}$$

$$\frac{\mathsf{F}\,A \vee B}{\mathsf{F}\,A} \qquad \frac{\mathsf{F}\,A \vee B}{\mathsf{F}\,B} \qquad \frac{\mathsf{F}^*\,A \vee B}{\mathsf{F}^*\,A} \qquad \frac{\mathsf{F}^*\,A \vee B}{\mathsf{F}^*\,B}$$

$$\frac{\mathsf{T}\,\neg A}{\mathsf{F}^*\,A} \qquad \frac{\mathsf{F}\,\neg A}{\mathsf{T}^*\,A} \qquad \frac{\mathsf{T}^*\,\neg A}{\mathsf{F}\,A} \qquad \frac{\mathsf{F}^*\,\neg A}{\mathsf{T}\,A}$$

Table 4.12: Elimination rules for the standard **FDE** connectives

$$\frac{\mathsf{T}\,A \quad \mathsf{F}\,A}{\mathsf{T}\,\curlywedge} \qquad \frac{\mathsf{T}^*\,A \quad \mathsf{F}^*\,A}{\mathsf{T}^*\,\curlywedge} \qquad \frac{\mathsf{T}\,\curlywedge}{\mathsf{S}A} \qquad \frac{\mathsf{T}^*\,\curlywedge}{\mathsf{S}A}$$

Under any uniform substitution of S with T, F, T^*, or F^*.

Table 4.13: Falsum rules

$$\dfrac{\begin{matrix}[\mathsf{T}\,A] & [\mathsf{F}\,A]\\ \vdots & \vdots \\ \mathsf{S}B & \mathsf{S}B\end{matrix}}{\mathsf{S}B} \qquad \dfrac{\begin{matrix}[\mathsf{T}^*\,A] & [\mathsf{F}^*\,A]\\ \vdots & \vdots \\ \mathsf{S}B & \mathsf{S}B\end{matrix}}{\mathsf{S}B}$$

Under any uniform substitution of S with T , F , T^* , or F^* .

Table 4.14: Rules of generalized bivalence

The trees in Fig. 4.11 respectively show proofs of $\{\mathsf{T}\,\neg(A\vee B), \mathsf{T}\,\neg C\} \vdash \mathsf{T}\,\neg(A\vee C)$ and $\{\mathsf{T}\,\neg(A \wedge B)\} \vdash \mathsf{T}\,\neg A \vee \neg B$. (Where the numerals are used to keep track of the temporary assumptions that are discharged by the application of a rule. The numerals corresponding to the discharged assumptions are shown beside the inference line.)

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{T}\,\neg(A\vee B)}{\mathsf{F}^*\,\neg(A\vee B)}}{\mathsf{F}^*\,A}\quad \dfrac{\mathsf{T}\,\neg C}{\mathsf{F}^*\,C}}{\mathsf{F}^*\,A\vee C}}{\mathsf{T}\,\neg(A\vee C)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{T}\,\neg(A\wedge B)}{\mathsf{F}^*\,A\wedge B}\quad [\mathsf{T}^*\,A]^1}{\mathsf{F}^*\,B}}{\mathsf{T}\,\neg B}}{\mathsf{T}\,\neg A\vee\neg B}\qquad \dfrac{\dfrac{[\mathsf{F}^*\,A]^2}{\mathsf{T}\,\neg A}}{\mathsf{T}\,\neg A\vee\neg B}}{\mathsf{T}\,\neg A\vee\neg B}\ {}_{1,2}$$

Figure 4.11: A proof with no applications of discharge rules and another with one application

4.9 Depth-bounded approximations to LP and $\mathbf{K_3}$

4.9.1 Informational interpretation and the need for imprecise values

It is well-known that **FDE**, **LP**, and $\mathbf{K_3}$ are closely related to each other [see 81, 82, 127, 19]. As mentioned above, for a matrix to handle information that might be both inconsistent and partial, the availability of at least 4 different truth-values is required [see 19]. The matrix inducing **FDE** is an elegant example of such a matrix. Now, 3-valued matrices can be used to handle either inconsistency or partiality of information, one at a time. An example of a logic characterized by a 3-valued matrix handling inconsistency of information is the Logic of Paradox, **LP** (at this section $\mathcal{L} = \{\vee, \wedge, \neg, \rightarrow\}$):

$\tilde{\vee}$	*true*	*false*	*i*
true	*true*	*true*	*true*
false	*true*	*false*	*i*
i	*true*	*i*	*i*

$\tilde{\wedge}$	*true*	*false*	*i*
true	*true*	*false*	*i*
false	*false*	*false*	*false*
i	*i*	*false*	*i*

$\tilde{\neg}$	
true	*false*
false	*true*
i	*i*

$\tilde{\to}$	*true*	*false*	*i*
true	*true*	*false*	*i*
false	*true*	*true*	*true*
i	*true*	*i*	*i*

Table 4.15: **LP**/$\mathbf{K_3}$-tables

Definition 4.9.1. Let $\mathcal{M}_3^b$ be a matrix for $\mathcal{L}$ where $\mathcal{V} = \{true, false, i\}$, $\mathcal{D} = \{true, i\}$, and the functions in $\mathcal{O}$ are defined by the truth-tables in Tab. 4.15.

In the above definition, the truth-values *true* and *false* are the classical ones, and i stands for *both* (*true* and *false*). **LP** was introduced for rather philosophical purposes. Namely, although Asenjo introduced the logic itself first [14], Priest coined **LP** has a tool for handling some logical paradoxes (such as the Liar and Russell's paradoxes) involving sentences that, according to Priest's view, are simultaneously true and false [126]. Accordingly, Priest interprets the "inconsistent" third truth-value in an aletheic sense. However, **LP** can be plausibly interpreted along the lines of the standard informational semantics of **FDE**. Namely, *true* (*false*) can be interpreted as "there is a source assenting to p and *there is no* source dissenting to p" ("there is a source dissenting to p and *there is no* source assenting to p); whereas i can be interpreted as "there is a source assenting to p and there is a source dissenting to p".

In turn, an example of a logic characterized by a 3-valued matrix handling partiality of information is Strong Kleene Logic, $\mathbf{K_3}$:

Definition 4.9.2. Let $\mathcal{M}_3^n$ be a matrix for $\mathcal{L}$ where $\mathcal{V} = \{true, false, i\}$, $\mathcal{D} = \{true\}$, and the functions in $\mathcal{O}$ are defined by the truth-tables in Tab. 4.15.

Indeed, the only difference between $\mathcal{M}_3^b$ and $\mathcal{M}_3^n$ is their set of designated truth-values. Accordingly, although in $\mathcal{M}_3^n$ *true* and *false* are again the classical ones, i is interpreted differently; viz., it stands for *neither* (*true* nor *false*). $\mathbf{K_3}$ was introduced for purposes in computer science, or rather they would have been if computer science had existed by then. In $\mathbf{K_3}$, the "indeterminate" third truth-value was originally introduced to account for inferences involving sentences for which its truth (or falsity)

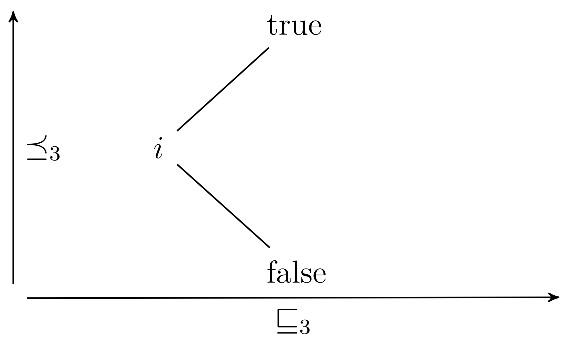


Figure 4.12: Orders over $\{\text{true}, \text{false}, i\}$

might not be decided by means of a function. More specifically, the third truth-value originally stands for "undecidable by the algorithms whether true or false". So, an informational interpretation of the truth-values is favoured. Correspondingly, the third truth-value behaves in a way compatible with any increase in information: If the value of some atomic formula p changed from indeterminate to either true or false, the value of any formula with p as a component must never change from true to false nor vice versa. Kleene referred to this as regularity; nowadays this is phrased in terms of monotony in an order. In fact, the three truth-values of $\mathbf{K_3}$ are ordered as in Fig. 4.12. Where $\sqsubseteq_3$ stands for an information order, and $\preceq_3$ stands for a truth order. (As expected, the corresponding truth order over the truth-values of **LP** is the same as $\preceq_3$; while, unlike $\sqsubseteq_3$, its information order places i to the right of both of *true* and *false*.) Thus, interpreting $\mathbf{K_3}$ along the lines of the standard informational semantics of **FDE** is not only plausible but natural: *true* (*false*) can be interpreted as "there is a source assenting to p" ("there is a source dissenting to p"); whereas i can be interpreted as "*there is no* source assenting to p and *there is no* source dissenting to p". (Warning: in $\mathbf{K_3}$ the possibility of contradictory information is discarded: once a source assents (dissents) to p, the possibility of there being a source dissenting (assenting) to p is discarded.)

Now, the valuations and consequence relations associated to the matrices of at issue are defined as for any many-valued logic. For instance:

Definition 4.9.3. A $\mathcal{M}_3^n$-*valuation* is a function $v : F(\mathcal{L}) \longrightarrow \{true, false, i\}$ such that for all A, B:

1. $v(\neg A) = \tilde{\neg}(v(A))$;
2. $v(A \circ B) = \tilde{\circ}(v(A), v(B))$.

Where $\circ$ is $\vee$, $\wedge$ or $\rightarrow$.

Definition 4.9.4. $\Gamma \vDash_{\mathcal{M}_3^n} A$ iff for every $\mathcal{M}_3^n$-valuation v, if $v(\Gamma) = \mathit{true}$, then $v(A) = \mathit{true}$.

The notion of $\mathcal{M}_3^b$-valuation and the corresponding relation $\vDash_{\mathcal{M}_3^b}$ are defined analogously.

Now, regarding their complexity, **LP** and $\mathbf{K_3}$ are both co-NP complete, and so also idealized models of how an agent *can* reason. That **LP** is co-NP complete can be shown analogously to Proposition 4.4.1 [see also 10];[17] whereas, that $\mathbf{K_3}$ is co-NP complete follows from Cook's result that **CPL** is co-NP complete [47] together with the following:

Proposition 4.9.5. *A is a classical tautology iff* $(p_1 \vee \neg p_1) \wedge ... \wedge (p_n \vee \neg p_n) \vDash_{\mathcal{M}_3^n} A$*, where* $p_1, ..., p_n$ *are the atoms occurring in A.*

Proof. By definition, A is a classical tautology iff for every classical valuation, $v : F(\mathcal{L}) \longrightarrow \{\mathit{true}, \mathit{false}\}$, $v(A) = \mathit{true}$. In turn, also by definition, this holds iff for every $\mathcal{M}_3^n$–valuation, $v : F(\mathcal{L}) \longrightarrow \{\mathit{true}, \mathit{false}, i\}$, if $v(p_i) \neq i$ for all p_i occurring in A, then $v(A) = \mathit{true}$. By the $\mathbf{K_3}$-tables for $\neg$ and $\vee$, this holds iff for every $\mathcal{M}_3^n$-valuation v, $v(A) = \mathit{true}$ whenever $v(p_i \vee \neg p_i) = \mathit{true}$ for all p_i occurring in A. In turn, by the $\mathbf{K_3}$-table for $\wedge$, this holds iff for every $\mathcal{M}_3^n$-valuation v, $v(A) = \mathit{true}$ whenever $v((p_1 \vee \neg p_1) \wedge ... \wedge (p_n \vee \neg p_n)) = \mathit{true}$ for all p_i occurring in A. Hence, the latter holds iff $(p_1 \vee \neg p_1) \wedge ... \wedge (p_n \vee \neg p_n) \vDash_{\mathcal{M}_3^n} A$. □

So, these results bring us to the need for tractable approximations. The basis for defining our approximations are sort of natural deduction systems based on observations analogous to those regarding **FDE**. Observe that, under the informational interpretation of **LP**, only the value i can be taken as *stable* without assuming complete information about the set of sources Ω. That is, given an epistemic state that evolves over time, the values *true* and *false* can be regarded as stable only if complete knowledge of Ω is assumed. Thus, these latter values are *information-transcendent* when interpreted as timeless, for they refer to an objective state of affairs concerning the domain of all sources. In a dual manner, under the informational interpretation of $\mathbf{K_3}$, the values *true* and *false* are stable without assuming full knowledge of Ω (since, recall, the possibility of contradictory information is discarded); whereas the value i is information-transcendent when interpreted as timeless. Much as in the case of **FDE**, this situation motivates the need for *stable imprecise values.* As before, we address this question by shifting to signed formulae, where the signs express such

[17] In fact, a formula A is a tautology in **CPL** iff A is a tautology in **LP** [126].

imprecise values associated with two distinct bipartitions of the corresponding set of standard values.

4.9.2 Intelim deduction in LP and $\mathbf{K_3}$

We use the same signed formulae used for **FDE** but, of course, interpret them differently. To state such a re-interpretation, we use x to refer to an agent and v to denote respectively a $\mathcal{M}_3^b$-valuation or a $\mathcal{M}_3^n$-valuation. For **LP**, we interpret: $\mathsf{T}\,A$ as "x holds that A is at least true" (expressing that $v(A) \in \{true, i\}$); $\mathsf{F}\,A$ as "x holds that A is false only" ($v(A) \in \{false\}$); $\mathsf{T}^*\,A$ as "x holds that A is true only" ($v(A) \in \{true\}$); $\mathsf{F}^*\,A$ as "x holds that A is at least false" ($v(A) \in \{false, i\}$). As for $\mathbf{K_3}$, we interpret: $\mathsf{T}\,A$ as "x holds that A is true" ($v(A) \in \{true\}$); $\mathsf{F}\,A$ as "x holds that A is non-true" ($v(A) \in \{false, i\}$); $\mathsf{T}^*\,A$ as "x holds that A is non-false" ($v(A) \in \{true, i\}$); $\mathsf{F}^*\,A$ as "x holds that A is false" ($v(A) \in \{false\}$). Crucially, according to the respective informational interpretation of the truth-values of **LP** and $\mathbf{K_3}$, whereas S-formulae of the form $\mathsf{T}\,A$ and $\mathsf{F}^*\,A$ involve only information that does not require complete knowledge of the sources, S-formulae of the form $\mathsf{T}^*\,A$ and $\mathsf{F}\,A$ involve information that does require such a complete knowledge.

Thereby, by making minor modifications to our proof system for **FDE**, we can obtain proof systems for **LP** and $\mathbf{K_3}$ which naturally lead to defining analogous hierarchies of tractable depth-bounded approximations to the latter logics. Namely, the systems for the latter logics are obtained by enriching the rules of the intelim method of the former logic with the intelim rules for implication displayed in Tab. 4.16, together with, respectively, the following rules:

$$\frac{\mathsf{T}^*\,A}{\mathsf{T}\,A} \qquad \frac{\mathsf{F}\,A}{\mathsf{F}^*\,A} \qquad\qquad\qquad \frac{\mathsf{T}\,A}{\mathsf{T}^*\,A} \qquad \frac{\mathsf{F}^*\,A}{\mathsf{F}\,A}$$

Additional structural rules for **LP**   Additional structural rules for $\mathbf{K_3}$

In turn, hierarchies of depth-bounded approximations can be defined in terms of the maximum number of nested applications of PB and PB^*, exactly as before. Even though, in the case of **LP** and $\mathbf{K_3}$, those branching rules are not the only structural rules, they are the only involving the introduction of virtual information. Moreover, it is straightforward to adapt the proofs of Theorem 4.6.9 and Corollary 4.6.10 to show the tractability of the approximations at issue. Further, by making minor modifications to our semantical framework for the hierarchy of approximations to **FDE**, it can be shown that the hierarchies for the 3-valued logics also admit of a 5-valued non-deterministic semantics.

$$\frac{\mathsf{F}^* A}{\mathsf{T}\, A \to B} \qquad \frac{\mathsf{T}\, B}{\mathsf{T}\, A \to B} \qquad \frac{\begin{array}{l}\mathsf{T}^* A\\ \mathsf{F}\, B\end{array}}{\mathsf{F}\, A \to B}$$

$$\frac{\mathsf{F}\, A}{\mathsf{T}^* A \to B} \qquad \frac{\mathsf{T}^* B}{\mathsf{T}^* A \to B} \qquad \frac{\begin{array}{l}\mathsf{T}\, A\\ \mathsf{F}^* B\end{array}}{\mathsf{F}^* A \to B}$$

$$\frac{\mathsf{F}\, A \to B}{\mathsf{T}^* A} \qquad \frac{\mathsf{F}\, A \to B}{\mathsf{F}\, B} \qquad \frac{\mathsf{F}^* A \to B}{\mathsf{T}\, A} \qquad \frac{\mathsf{F}^* A \to B}{\mathsf{F}^* B}$$

$$\frac{\begin{array}{l}\mathsf{T}\, A \to B\\ \mathsf{T}^* A\end{array}}{\mathsf{T}\, B} \qquad \frac{\begin{array}{l}\mathsf{T}^* A \to B\\ \mathsf{T}\, A\end{array}}{\mathsf{T}^* B} \qquad \frac{\begin{array}{l}\mathsf{T}\, A \to B\\ \mathsf{F}\, B\end{array}}{\mathsf{F}^* A} \qquad \frac{\begin{array}{l}\mathsf{T}^* A \to B\\ \mathsf{F}^* B\end{array}}{\mathsf{F}\, A}$$

Table 4.16: Intelim rules for the implication of **LP** and $\mathbf{K_3}$

Chapter 5

Towards tractable depth-bounded approximations to IPL

5.1 Introduction

In [96], Heyting provided a Hilbert-style system to codify patterns of reasoning used in Brouwer's intuitionistic mathematics. Heyting's formalization of the propositional fragment of those patterns is what has come to be called *Intuitionistic Propositional Logic*, **IPL**. While the basis of classical logic is ontological, the basis of intuitionistic logic is epistemic. The fundamental distinction between classical and intuitionistic logic lies on their underlying notion of *truth*. Classical truth is "external" in that it requires reference to a reality that exists independently of the agents' reasoning and perception. That is, an external reality which makes sentences determinately true or false quite independently of the agent's epistemic means. In contrast, intuitionistic truth is "internal" in that a sentence being true or false depends exclusively on the agent's epistemic means to determine whether the sentence is one or the other. More specifically—in tune with the intuitionistic tenet that mathematics and logic are mental constructive activities—the truth of a mathematical sentence can only be established by a *proof*, conceived as a certain kind of mental construction. The latter notion is not defined in formal terms as it is, for instance, in a proof system in Logic. Rather that notion is intended as an informal primitive one—just like truth in the case of classical logic. Thereby, intuitionisitc logic singles out valid inferences not in terms of preservation of mind-independent truth—as classical logic does—but preservation of mental constructibility.

Ever since Heyting's characterization of **IPL**, several and diverse equivalent formalizations of it have been given. We shall recall some of them below very succinctly, but of particular interest to us is that **IPL** admits of an informational semantics. Namely, there are two semantical characterizations of **IPL** based on informational terms which are closely related to each other: Kripke semantics [107] and Beth semantics [33]. In fact, the former has become the dominant semantics of **IPL**, presumably because it is particularly easy to work with and most resembles classical model theory.

Now, despite its informational interpretation, **IPL** is PSPACE-complete [140] and, thus, an idealized model of how an agent *can* reason. The present Chapter shows how the depth-bounded approach can be naturally extended to **IPL** and, accordingly, defines an infinite hierarchy of depth-bounded approximations to that logic, where each approximation is conjectured to be tractable. We identify the source of the intractability of **IPL** with the nested use of *virtual* or hypothetical information involved when evaluating the truth of formulae whose main connective is $\rightarrow$ or $\neg$. For instance, for an agent to recognize that $A \rightarrow B$ is proven at a state where A is unproven, she must transfer from the *actual* state to a *virtual* one where A is proven, and verify that in the latter state B is also proven; that is, she must reason *as if* her state were the latter one. Although all virtual information may be eventually discharged—to the effect that the conclusion depends only on information held by the agent—it is the case that the corresponding deduction steps could not be performed at all without using that virtual information. By contrast, the evaluation of the truth or falsity of other formulae involves only information that the agent holds, i.e., *actual* information. For instance, if an agent holds that $\neg A$ is unproven in the current *actual* information state, then she also holds that A can possibly be proven in some future state. That is, this inference involves only actual (modal) information.

Thereby, we provide a *KE*/*KI*-style proof system for **IPL** that neatly separates the use of actual and virtual information in deductions. However, unlike Gentzen-Prawitz style natural deduction, and following the key idea underlying the depth-bounded approach according to which the meaning of a connective is specified solely in terms of information that is held by the agent, in our system all the *operational* rules involve only actual information and there is a unique *structural* rule which introduces virtual information. We call this latter rule *PB* as it expresses a *generalized* Principle of Bivalence, intuitively saying: for any formula A and for any information state, either A is proven or unproven there. More precisely, our *KE*/*KI*-style system is such that: (i) it is formulated by means of signed formulae; (ii) it has linear introduction and elimination rules, which fix the meaning of the connectives; (iii) it has a sole branching rule which expresses a generalized rule of bivalence, is structural in that it does not involve any connective, and is essentially a cut rule;

(iv) it can be used as both a direct-proof and a refutation method; (v) obeys the subformula property. Since the examples introduced in Subsection 3.3.3 are hard for all tableau systems sharing the $\wedge/\vee$ rules with classical tableaux but easy for their *KE*-style counterparts, our *KE*/*KI*-style system at issue is interesting independently of the depth-bounded approach mainly because it has an exponential speed-up on its analogous tableau system. However, in this Chapter we focus on showing that it naturally leads to defining an infinite hierarchy of depth-bounded approximations to **IPL** in terms of the maximum number of nested applications that are allowed of the branching rule, approximations conjectured to be all tractable. Intuitively, the introduction and elimination rules govern the use of actual information, whereas the branching structural rule governs the manipulation of virtual information (i.e., hypothetical information about a formula being proven or unproven at an information state). As in the classical and many-valued cases addressed in the previous Chapters, the key intuition is that the more virtual information needs to be invoked via the branching rule, the harder the inference is for the agent. Thus, the nested applications of that rule provide a sensible measure of inferential depth, and so the levels of the corresponding hierarchy can be naturally related to the inferential power of agents.

We further pave the way for a non-deterministic semantics for the resulting hierarchy. Namely, as a first step towards a semantical characterization of the hierarchy, we provide an alternative 3-valued non-deterministic semantics for full **IPL** such that, unlike Kripke and Beth semantics, specifies the meaning of the connectives without appealing to any "structural" condition.

5.2 The BHK-interpretation

The hallmark of intuitionistic logic is to reject all non-constructive reasoning. More specifically, as mentioned above, the truth of a mathematical sentence can only be established by a proof. Accordingly, the meaning of a connective has to be given by establishing, for any sentence in which that connective is the main one, what counts as a proof of that sentence—being assumed that is already known what counts as a proof of any of the constituents. An intuitive interpretation of **IPL** in these terms was put forward by Brouwer and later developed by Heyting and Kolmogorov—and, thus, it is known as the *BHK-interpretation*. This consists of an explanation of the meaning of the connectives in terms of proofs, which can be roughly summarized as follows: (i) That a sentence A is true means that there is a proof of A or, more generally, some kind of evidence verifying A. (ii) Dually, that a sentence A is false means that there is a proof that A cannot possibly be proven or, more generally, that no evidence verifying A can possibly be given. So, A is false iff $\neg A$ is true, in the sense that a proof of $\neg A$ amounts to a proof that it is impossible to verify A, regardless how the corresponding agent's knowledge grows. Thus, assuming that a proof of A is available leads to some absurd sentence for which a proof will never be available—say, when dealing with arithmetical sentences, 0=1. In fact, in intuitionistic logic the constant $\curlywedge$ is often assumed to be contained in $\mathcal{L}$, denoting generically such an absurd sentence and, accordingly, $\neg A$ is often defined as $A \to \curlywedge$. (iii) The meaning of a connective $\diamond$ is fixed by establishing what kind of construction is to be counted as a proof of a complex sentence containing $\diamond$ its main connective. Besides, "π is a proof of p", where $p \in At(\mathcal{L})$, is taken as primitive (unexplained) notion. Thereby [144, p. 9]:

- π is a proof of $A \wedge B$ iff π is a pair (π_1, π_2) such that π_1 is a proof of A and π_2 is a proof of B;
- π is a proof of $A \vee B$ iff π is a proof of A or a proof of B (plus the stipulation that we want to regard the proof presented as evidence for $A \vee B$);
- π is a proof of $A \to B$ if π is a construction which transforms every proof π_1 of A into a proof $\pi(\pi_1)$ of B;
- $\curlywedge$ has no proof; a proof of $\neg A$ is a construction which transforms any hypothetical proof of A into a proof of an absurdity.

Thus, the BHK-interpretation of the meaning of the connectives proceeds in terms of the conditions under which a proof justifies the agent at issue in asserting an statement. Namely, the agent is entitled to assert a conjunction iff she is entitled to assert each of the conjuncts; the agent is entitled to assert a disjunction iff she is in a position to justifiably assert one or other disjunct—or, in a less strict way, she possesses an effective method whose application would put her in such a position; a conditional is assertible by the agent iff she possesses an effective method for turning any proof of the antecedent that might be constructed, into a proof of the consequent; a negation is assertible by the agent iff she possesses a way of turning any proof of the negated sentence into a proof of some absurdity.

Are the valid inferences of **IPL** exactly the principles justified by the BHK-interpretation? Clearly, there is no way to show that they are or are not, since the interpretation is not a formal semantics, but only an intuitive explanation of the meaning of the intuitionisitc connectives. Key notions appearing in it such as "proof", "construction", "transformation", and "effective method", are not defined, and can be interpreted in a variety of ways—and indeed they have been so. Nonetheless, already according to this informal explanation of the connectives, the principle of excluded middle $A \vee \neg A$ is not valid in general. Namely, there is no guarantee that for every sentence A there is either a proof of A or a proof that A cannot possibly be proven, i.e., a refutation of A. Excluded middle is valid only when A is decidable, i.e., there is an algorithm that always outputs a proof of A or a proof of $\neg A$.

5.3 IPL interpreted informationally

It is also possible to interpret intuitionistic logic in a formal semantics based on models of informational processes, where an agent—or a group of agents—progressively gains more information about a current information state, encoded in a propositional valuation. This kind of semantics for intuitionistic logic was first devised by Beth [33], and a later version is due to Kripke [107]. Intuitively, the corresponding models mimic abstract processes of mathematical investigation carried out by idealized agents with perfect memory. Such an investigation consists of constructing proofs of statements as well as mathematical objects, and the corresponding process takes place in time. At each moment the agent has acquired certain amount of information and, since this information is mathematical, once it is acquired, it "eternally" remains in the agent's memory. So, the agent gains information progressively and cumulatively; i.e., information increases monotone in time. Moreover, in general, when passing from one moment to the next, the agent has a number of possibilities to choose from to continue her investigation. Thus, the picture of the investigation process looks like a partially ordered set, even like a tree. What is more, each moment can be seen as an information state, and at each such an state there is a number of "accessible" next states. Therefore, truth at an informational state essentially depends on the future.

As shown below, rather than trying to formalize the BHK-interpretation, the semantics put forward by Beth and Kripke exploit a remarkable connection between intuitionistic logic and order theory. However, whether those semantics reflect any intuitive view of the meaning of the intuitionistic connectives and, particularly, the relationship they might bear to the BHK-interpretation, has been studied by, for example, Dummett [67]. In fact, in line with the BHK-interpretation, in those semantics the truth of a formula is intuitively equivalent to its *verifiability* at an information state [see 67, 34]. As explained below, although in both semantics formulae are evaluated at nodes in a partial order, the definition of the verifiability of a formula at a node is different.[1]

[1] Originally, Beth used finitely branching trees while Kripke used preorders, but nowadays it is

5.3.1 Kripke semantics

Kripke semantics [107] is more popular that Beth's presumably because, as we shall recall below, the former is somewhat easier to use than the latter.

An *intuitionistic Kripke model*, that we shall denote by $\mathfrak{M}$, is intended to represent an informational process with a set of information states, an "accessibility" relation representing (possible) time succession between those states, and a valuation that records which atomic formulas hold at each state. This valuation is then recursively extended on arbitrary formulae, defining a *forcing relation* $\Vdash_K$, whose intended interpretation is as follows: if at a particular information state a, the agent has enough information to prove a formula A, it is said that $\mathfrak{M}, a \Vdash_K A$; whereas, if she lacks such information, it is said that $\mathfrak{M}, a \nVdash_K A$. Correspondingly, $\mathfrak{M}, a \Vdash_K A$ is taken to mean that A *has been verified* to be true at a, and $\mathfrak{M}, a \nVdash_K A$ that A *has not been verified* to be true at a. The intended meaning of the latter is not that A has been proven false or refuted at a, but that A is not (yet) proven at a and may be established later. In turn, given an information state a, the "accessibility" relation is intended to represent the open possibilities for gaining more information. The intuitive meaning of "b is accessible from a" is that, as far as an agent knows, at state a, she may later gain enough information to advance to b. Further, in a finite Kripke model, each final state or leaf is a classical model because it forces every formula or its negation. Thus, Kripke models generalize classical models to intermediate states of information where classical laws may fail.

The following are formal definitions of the notions above as they are generally presented nowadays:

Definition 5.3.1. An *intuitionistic Kripke frame* is a pair $\mathfrak{F} = \langle S, R \rangle$, where S is a non-empty set and R is a partial order on S.

Elements of S are called information states and R is usually called "accessibility" relation. Thus, for all $a, b \in S$, aRb is read "b is accessible from a". Intuitively, aRb means that b is either equal to a or stands for a possible future development of a which can be envisaged from the viewpoint of the information contained in a. So, R tells us which possible future information states are "accessible" from each state.

We remind the reader that a binary relation R on a set S is called a partial order if the following three conditions are satisfied for all $a, b, c \in S$:

1. aRa (reflexivity)
2. If aRb and bRc, then aRc (transitivity)

customary to work with equivalent versions of both semantics using partial orders.

3. If aRb and bRa, then $a = b$ (antisymmetry)

Definition 5.3.2. An *intuitionistic Kripke valuation* is a function $\mathfrak{v} : At(\mathcal{L}) \longrightarrow \{T, F\}$.[2]

Intuitively, T stands for proven and F for unproven (i.e., there is no proof at the moment). Beware that A is unproven does not mean that A is refuted; i.e., it does not mean that $\neg A$ is proven. Accordingly, in terms of verifiability, the meaning of $\mathfrak{v}(p) = T$ is taken to be that p is verified (to be true), while $\mathfrak{v}(p) = F$ is taken to mean that p is not verified.

Definition 5.3.3. Let $\mathfrak{F}$ be an intuitionistic Kripke frame. An *intuitionistic Kripke model* is a pair $\mathfrak{M} = \langle \mathfrak{F}, \{\mathfrak{v}_a\}_{a \in S} \rangle$, where $\{\mathfrak{v}_a\}_{a \in S}$ is an intuitionistic Kripke valuation required to satisfy the "truth-persistence" condition: for all p and all $a, b \in S$, if $\mathfrak{v}_a(p) = T$ and aRb, then $\mathfrak{v}_b(p) = T$.

The latter condition tells us that if a p is proven at a given state, then it will remain proven forever. However, beware that, if p is unproven at a given state, it may (or may not) become proven at a "accessible" future state. On the other hand, naturally, if p is refuted (i.e., $\neg p$ is proven), then $\neg p$ will remain proven (and so p will remain refuted) forever. Accordingly, in terms of verficability, $\mathfrak{v}_a(p) = T$ means that p is verified at a; while $\mathfrak{v}_a(p) = F$ means that p is not verified at a. Further, for no $a \in S$, $\mathfrak{v}_a(\curlywedge) = T$—equivalently, for all $a \in S$, $\mathfrak{v}_a(\curlywedge) = F$.

Definition 5.3.4. Let $\mathfrak{M} = \langle \mathfrak{F}, \{\mathfrak{v}_a\}_{a \in S} \rangle$ be an intuitionistic Kripke model. Then, $\{\mathfrak{v}_a\}_{a \in S}$ is recursively extended on $\mathcal{L}$, defining a *forcing relation* $\Vdash_K$:

- $\mathfrak{M}, a \Vdash_K p$ iff $\mathfrak{v}_a(p) = T$;
- $\mathfrak{M}, a \Vdash_K A \vee B$ iff $\mathfrak{M}, a \Vdash_K A$ or $\mathfrak{M}, a \Vdash_K B$;
- $\mathfrak{M}, a \Vdash_K A \wedge B$ iff $\mathfrak{M}, a \Vdash_K A$ and $\mathfrak{M}, a \Vdash_K B$;
- $\mathfrak{M}, a \Vdash_K A \rightarrow B$ iff for all b such that aRb, if $\mathfrak{M}, b \Vdash_K A$, then $\mathfrak{M}, b \Vdash_K B$.
- $\mathfrak{M}, a \Vdash_K \neg A$ iff for all b such that aRb, $\mathfrak{M}, b \nVdash_K A$.

[2]In the literature, the definitions of this and the next notions are slightly different but equivalent to those presented here. As it will be apparent in Section 5.7, the reason we opted for using slightly different definitions is that they are better suited for comparing Kripke semantics with an alternative semantics we shall in that Section.

Note that the clause about $\neg A$ is compatible with defining of $\neg A$ as $A \to \curlywedge$ if we stipulate that for all $a \in S$, $\mathfrak{M}, a \nVdash_K \curlywedge$. Thereby, $\mathfrak{M}, a \Vdash_K \curlywedge$ iff for all b such that aRb, if $\mathfrak{M}, b \Vdash_K A$, then $\mathfrak{M}, b \Vdash_K \curlywedge$. But, since $\curlywedge$ is not forced by any information state and by contraposition, neither is A. In turn, intuitively, $\mathfrak{M}, a \Vdash_K A$ means that A *has been verified* at a, and $\mathfrak{M}, a \nVdash_K A$ that A *has not been verified* at a. Moreover, by an easy induction on the degree of an arbitrary formula A, it follows that the "truth-persistence" condition above extents to every formula. That is, if $\mathfrak{M}, a \Vdash_K A$ and aRb, then $\mathfrak{M}, b \Vdash_K A$. Thus, finally:

Definition 5.3.5. Let $\mathfrak{M} = \langle \mathfrak{F}, \{\mathfrak{v}_a\}_{a \in S} \rangle$ be an intuitionistic Kripke model. We say that:

- $\mathfrak{M}$ is a *model* of A iff $\mathfrak{M}, a \Vdash_K A$ for all $a \in S$.
- $\mathfrak{M}$ is a *model* of Γ iff $\mathfrak{M}$ is a model of every $B \in \Gamma$.
- A is a *logical consequence* of Γ, $\Gamma \vDash_K A$, iff every model $\mathfrak{M}$ of Γ is a model of A.

5.3.2 Beth semantics

Beth semantics [33], just as Kripke's, is intended to model informational process with a set of information states, a relation representing time succession between those states, and a valuation recording which atomic formulae hold at each state. Indeed, it is customary to present Beth semantics in terms of Kripke's, presumably because the latter is better known and used [see 107, 34]. As we shall recall below, the difference between the two semantics lies in the definition of their respective forcing relation for atoms and disjunction. To begin with, the notion of a frame is the same as in Kripke semantics:

Definition 5.3.6. An *intuitionistic Beth frame* is a pair $\mathfrak{G} = \langle S, R \rangle$, where S is a non-empty set and R is a partial order on S.

Of course, the intuitive interpretation of S and R is exactly as in a Kripke frame. However, as it will apparent below, owing to the forcing clauses for atoms and disjunction to be recalled, in Beth semantics we must make the extra assumption that there is necessarily an advance from each non-terminal information state to one of its "accessible" future states after a finite amount of time. Put differently, unlike Kripke semantics—according to which it is allowed to remain at a given information state indefinitely—in Beth's the advance from one information state to an "accessible" one within finite time is compulsory—unless the information state at issue is terminal and so it cannot be further developed. Regarding the notion of a valuation, there is no change from the analogous notion in Kripke semantics:

Definition 5.3.7. An *intuitionistic Beth valuation* is a function $\mathfrak{w} : At(\mathcal{L}) \longrightarrow \{T, F\}$.[3]

As above, intuitively, T stands for proven and F for unproven. Naturally, these latter notions are interpreted just as in Kripke semantics. Thus, the intuitive meaning in terms of verifiability of $\mathfrak{w}(p) = T$ and $\mathfrak{w}(p) = F$ is again, respectively, p is verified and p is not verified. Moreover, the notion of model is also defined as before:

Definition 5.3.8. Let $\mathfrak{G}$ be an intuitionistic Beth frame. An *intuitionistic Beth model* is a pair $\mathfrak{N} = \langle \mathfrak{G}, \{\mathfrak{w}_a\}_{a \in S} \rangle$, where $\{\mathfrak{w}_a\}_{a \in S}$ is an intuitionistic Beth valuation required to satisfy the "truth-persistence" condition: for all p and all $a, b \in S$, if $\mathfrak{w}_a(p) = T$ and aRb, then $\mathfrak{w}_b(p) = T$.[4]

Once more, as in Kripke semantics, the "truth-persistence" condition expresses that once an atom is proven, it will remain proven forever. As before, the intuitive meaning of a $\mathfrak{w}_a(p) = T$ ($\mathfrak{w}_a(p) = F$) is taken to be that p is (not) verified at a. Besides, for no $a \in S$, $\mathfrak{w}_a(\curlywedge) = T$.

Now, modulo the analogous definitions above, the forcing relation in Beth semantics is the same as in Kripke's for $\wedge$, $\rightarrow$ and $\neg$; yet it differs for atoms and $\vee$. The key concepts that differentiate the forcing relations at issue are those of *path* and *bar*.

Definitions 5.3.9. A *path in a poset* $\langle S, R \rangle$, is a maximal linearly ordered subset of S (linearly ordered by R, that is). Besides, if X is a path and $a \in X$, then we call X a *path through* a. In turn, a *bar for* a is a subset B of S such that each path through a intersects B.

Thereby:

Definition 5.3.10. Let $\mathfrak{N} = \langle \mathfrak{G}, \{\mathfrak{w}_a\}_{a \in S} \rangle$ be an intuitionistic Beth model. Then, $\{\mathfrak{v}_a\}_{a \in S}$ is recursively extended on $\mathcal{L}$, defining a *forcing relation* $\Vdash_B$:

[3] Again, the definitions of this and the next notions are slightly different but equivalent to those found in the literature. The reason for this is the same that in the previous footnote since, as customary, we are presenting Beth semantics in terms of Kripke's.

[4] A Beth model is often defined by imposing—besides the "truth-persistence" condition—what we shall call bellow the "barring" condition *directly* on (the valuation of) the model [see 107]. As pointed out in [e.g. 67, 34], another approach consists in imposing the "barring" condition not on the valuation, but on the basic clause for atoms of the forcing relation. We shall follow this second approach since it fits better our analysis.

- Barring: $\mathfrak{N}, a \Vdash_B p$ iff there is a bar B for a such that $B \subseteq \{b \in S \mid \mathfrak{w}_b(p) = T\}$;
- $\mathfrak{N}, a \Vdash_B A \vee B$ iff there is a bar B for a such that $B \subseteq \{b \in S \mid \mathfrak{N}, b \Vdash_B A$ or $\mathfrak{N}, b \Vdash_B B\}$;
- Same as in Kripke semantics for $\wedge$, $\rightarrow$ and $\neg$, *mutatis mutandis*.

Thereby, unlike Kripke semantics, in Beth's if at some informational state it can be envisaged that, within a finite time, an atom p will "inevitably" be verified, then p is already forced at that state. The underlying intuition is that if the agent knows that, for any possible course of investigation extending the current state a, p will be proven within finite time, then p is to be regarded as established at a. So, intuitively, $\mathfrak{N}, a \Vdash_B p$ means that, at a, it is *known that* p *will be verified*; while $\mathfrak{N}, a \nVdash_B p$ means that at a it is *known that* p *will not be verified*. In turn, if one disjuncts of a disjunction will, within a finite time, "inevitably" be forced, then the disjunction is already forced, even if neither of the disjuncts is already forced. Indeed, a constructivist view according to which a disjunction has been verified only if one of the disjuncts has been verified, does not imply that the agent *knows* that a disjunction will be verified only if she knows of one of the disjuncts that it will be verified. That is, unlike verification, knowledge is not assumed to distribute over disjunction [see 67, 34, 113]. Thus, given that Beth forcing is based on knowledge of what will be verified, the intuitive meaning of $\mathfrak{N}, a \Vdash_B A$ is that at a it is known that A will be verified; while the intuitive meaning of $\mathfrak{N}, a \nVdash_B A$ is that at a it is known that A will not be verified.

As above, it follows by an easy induction that both the "barring" condition and the corresponding "truth-persistence" condition extent to every formula. Finally, we obtain an analogous definition of intuitionistic logical consequence:

Definition 5.3.11. Let $\mathfrak{N} = \langle \mathfrak{G}, \{\mathfrak{w}_a\}_{a \in S} \rangle$ be an intuitionistic Beth model. We say that:

- $\mathfrak{N}$ is a *model* of A iff $\mathfrak{N}, a \Vdash_B A$ for all $a \in S$.
- $\mathfrak{N}$ is a *model* of Γ iff $\mathfrak{N}$ is a model of every $B \in \Gamma$.
- A is a *logical consequence* of Γ, $\Gamma \vDash_B A$, iff every model $\mathfrak{N}$ of Γ is a model of A.

5.3.3 Kripke vs. Beth semantics

On the one hand, owing to its more complex definition of forcing, Beth semantics is somewhat more difficult to use than Kripke's. To see how easier calculations are

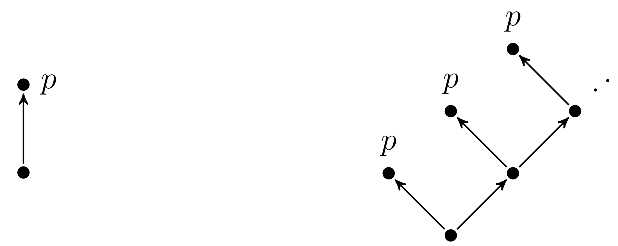


Figure 5.1: Simplest Kripke model refuting $p \vee \neg p$ on the left, and simplest Beth model refuting $p \vee \neg p$ on the right

in Kripke semantics relative to calculations in Beth's consider, for example, their simplest models refuting $p \vee \neg p$ in Fig. 5.1:

While the Kripke model consists only of two nodes, the Beth model—known as Beth comb—is infinite. In Beth semantics, every finite poset validates $p \vee \neg p$. The reason is that, if there is no infinite path, then every path through a contains an endpoint, and each endpoint satisfies p or $\neg p$; so, $\mathfrak{N}, b \Vdash_B p \vee \neg p$ for any b. Meanwhile, the Beth comb, as an infinite poset, does the job: Let $\{a \in S \mid \mathfrak{w}_a(p) = T\}$ be the set of all teeth of the comb. The spine of the comb is a path through the root that never intersects $\{a \in S \mid \mathfrak{w}_a(p) = T\}$. Thus, for every b in the spine, $\mathfrak{N}, b \nVdash_B p$; but also $\mathfrak{N}, b \nVdash_B \neg p$ since b can step to a tooth c such that $\mathfrak{N}, c \Vdash_B p$. Therefore, $p \vee \neg p$ does not hold at the root [see 34]. In fact, **IPL** does not have the finite model property w.r.t. Beth semantics.[5] What is more, every classically valid but intuitionistically invalid inference can only be shown to fail in an infinite Beth model.

On the other hand, while the intuitive verificationist interpretation of Kripke semantics accommodates only a very strict intuitionism, Beth's makes room for a more liberal intuitionism. While in Kripke semantics, that an atom is forced at a node amounts to it being already verified there ($\mathfrak{N}, a \Vdash_K p$ iff $\mathfrak{w}_a(p) = T$), in Beth's we can differentiate those notions. Namely, that at a, it is known that p will be verified ($\mathfrak{N}, a \Vdash_B p$) does not amount to p being verified at a ($\mathfrak{w}_a(p) = T$). To put it with Dummett:

> On this approach, we are distinguishing between the *verification* of an atomic statement in a given state of information, and its being *assertible*; the latter notion is represented by truth [being forced] at a node, and is defined, for all statements, in terms of the verification of atomic statements. The knowledge that a given atomic statement will be verified

[5]Recall that a logic is said to have the finite model property when each of its invalid inferences fails in some finite model.

> within a finite time does not itself constitute a verification of it, but is sufficient ground to entitle us to assert it [67, p. 139].

Put differently, if at some information state the agent knows that, for every foreseeable extension of such an state, p will be eventually verified, then she is entitled to assert p already at the original state. The same ideas underlie the different treatment of disjunction in Beth semantics w.r.t. Kripke's. The latter semantics is in line with an strict constructivist view according to which a disjunction is assertable on the basis of possession of a proof of at least one of the disjuncts. For, recall, if $\mathfrak{M}, a \Vdash_K A \vee B$ then $\mathfrak{M}, a \Vdash_K A$ or $\mathfrak{M}, a \Vdash_K B$. In contrast, although if $\mathfrak{N}, a \Vdash_B A$ or $\mathfrak{N}, a \Vdash_B B$ then $\mathfrak{N}, a \Vdash_B A \vee B$, the converse does not hold in general. Indeed, rephrasing what we said just after Def. 5.3.10, it does not follow from the aforementioned strict view that the agent *knows* that a disjunction will be verified only if she knows of one of the disjuncts that it will be verified. Once again, unlike verification, knowledge is not assumed to distribute over disjunction.[6] Again, intuitively, $\mathfrak{N}, a \Vdash_B A \vee B$ (if and) only if the agent knows that, in every foreseeable extension of her current information state, one of the disjuncts will be verified—though she may not know which. So, an agent's information state may entitle her to assert a disjunction without, as it currently stands, entitling her to assert either disjunct.

Observe that the clause for $\vee$ in Beth semantics is compatible with the idea that *verifying* a disjunction requires verifying a disjunct. To the same extent, such a clause is also compatible with the idea from the BHK-interpretation that *proving* a disjunction requires proving a disjunct. As expressed by Bezhanishvili and Holliday, "Beth semantics does not offer an alternative account of verification or proof, but rather an account of the validity of principles of propositional logic in terms of knowledge of what will inevitably be verified" [34, p. 431]. The BHK-interpretation is vague as to whether specifically a disjunction is assertable on the strict basis of having a proof of at least one of the disjuncts, or on the more liberal basis of having an effective method which would yield such a proof. Kripke semantics accords with such an strict basis, while Beth's is in line with the more liberal basis. An example discussed by Dummett [68] is enlightening to compare those basis: Consider the question of whether certain large number n is prime. A very strict constructivist might only accept that "n is prime or composite" is true if it has been already verified that either n is prime or that it is composite. In contrast, a more liberal constructivist might hold that "n is prime or composite" is true if it is known that it will eventually be verified that either n is prime or n is composite.

[6]Put differently, being based on knowledge of what will be verified, Beth semantics allows us to circumvent the *primeness* of disjunction—since knowledge is not prime [see 113].

5.4 Proof-theory of IPL revisited

In this section, we shall briefly recall some proof systems for **IPL**, and ideas thereof. Given that the formal definitions are to be found in the original references, our exposition of those systems is given in somewhat informal terms. Some of the systems to be recalled are closely related to our approach to defining tractable approximations to **IPL**. However, as shown by the hard examples introduced in 3.3.3, a crucial difference between the tableau methods to be recalled and the intelim method introduced below (as well as *KE*-style and *KI*-style systems) is that the latter has (have) an exponential speed-up on the former. As in the classical and many-valued cases previously addressed, roughly speaking, the reason of this is that while the tableau methods have operational branching rules that imply a good deal of redundant branchings in the corresponding tree, the intelim method has only a structural branching rule that reduces the amount of branching to a minimum by making all branches mutually exclusive. In the overall context of the Thesis, another important difference between the tableau methods and the intelim method—and, in fact, cut-based systems in general—is that since in the former cut is eliminable, no approximations can be defined by controlling the application of the cut rule. Further, regardless computational efficiency issues, the natural deduction system to be recalled below does not comply with the main idea underlying depth-bounded approximations, according to which the meaning of a logical operator is fixed only in terms of actual information. For in the natural deduction system, some of the (operational) "discharge rules" make essential use of virtual information.

5.4.1 Hilbert-style system

What has come to be called **IPL** was first characterized as an axiom system by Heyting [96], and was intended to codify patterns of reasoning used in (the propositional fragment of) Brouwer's intuitionistic mathematics. Heyting's characterization takes $\to$, $\wedge$, $\vee$ and $\neg$ as basic connectives, and is reproduced in Tab. 5.1.[7]

[7] Notice that all these axioms are actually axiom schemata. That is, one can substitute arbitrary formulae for A, B, C, obtaining instances of axioms.

Axioms:

1. $((A \to B) \wedge (B \to C)) \to (A \to C)$
2. $B \to (A \to B))$
3. $(A \wedge (A \to B)) \to B$
4. $A \to (A \wedge A)$
5. $(A \wedge B) \to (B \wedge A)$
6. $(A \to B) \to ((A \wedge C) \to (B \wedge C))$
7. $A \to (A \vee B)$
8. $(A \vee B) \to (B \vee A))$
9. $((A \to C) \wedge (B \to C)) \to ((A \vee B) \to C)$
10. $\neg A \to (A \to B)$
11. $((A \to B) \wedge (A \to \neg B)) \to \neg A$

Inference rule: $\dfrac{A \quad A \to B}{B}\ (MP)$

Table 5.1: Hilbert-style system for **IPL**

This is a Hilbert-style system, where:

Definitions 5.4.1. A *derivation* is a linearly ordered list of formulae, and each of them is either an instance of an axiom or is obtained from earlier formulae using *MP*. If there is a derivation ending with B, then B is called *derivable*, denoted by $\emptyset \vdash_{\mathbf{IPL}} B$. We also consider derivations from premises: we allow Γ to appear in derivations, along with axioms of **IPL**. If B is derivable using Γ, we write $\Gamma \vdash_{\mathbf{IPL}} B$.

Now, as in classical logic, we have:

Theorem 5.4.2 (Deduction Theorem). *Let Γ be a finite set of $\mathcal{L}$-formulae. Then $\Gamma, A \vdash_{\boldsymbol{IPL}} B$ iff $\Gamma \vdash_{\boldsymbol{IPL}} A \to B$.*

Proof. The *if* part is just an application of *MP*. The *only if* part follows by induction on the derivation of B from $\Gamma \cup \{A\}$ in **IPL**. □

If we add any of the following axioms to those in Tab. 5.1, we obtain the familiar **CPL**:

- $A \vee \neg A$
- $\neg\neg A \to A$
- $((A \to B) \to A) \to A$

5.4.2 Natural deduction

It is well known that both Kripke and Beth semantics involve a good deal of classical ("meta"-) reasoning, especially when it comes to the corresponding completeness proofs. In both Kripke's and Beth's models are not intended to be confined to intuitionistic methods. Indeed, Kripke semantics is the interpretation of **IPL** that most resembles classical model-theoretical semantics—and, as we explained above, Beth semantics coincides with Kripke's in several formal and conceptual aspects. Given this situation, purists prefer proof-theoretical formalizations of the BHK-interpretation. Among these formalizations, Gentzen's natural deduction system NJ [90] has been preferred since it embodies the intended meaning of the intuitionistic connectives—as expressed by the BHK-interpretation—far more accurately than other formalizations; in particular, more accurately than the existing Hilbert-style formalizations. The latter in the sense that, arguably, NJ reflects the specific constructive reasoning of the intuitionist best. In fact, Gentzen conceived his NJ as "a formal system that comes as close as possible to actual reasoning" [90, p. 68]; that is, as close as possible as the reasoning of intuitionistic mathematicians.

The dissemination and eventual recognition of Gentzen's insights is to a great part owing to Prawitz, who reintroduced natural deduction and considerably extended Gentzen's work [see 125]. A Gentzen-Prawitz style natural deduction system for **IPL**—which coincides with Gentzen's original NJ—is constituted by the rules in Tab. 5.2. The vertical dots appearing in some of the rules stand for a proof of the formula below the dots depending on assumptions that may include those enclosed in square brackets. The latter are "discharged" by the application of the rule at issue, in the sense that the conclusion no longer depends on them, but only on the still undischarged assumptions that occur in the leaves. Thereby:

Definitions 5.4.3. A *proof of A depending on* Γ is a tree of occurrences of formulae constructed according to the rules in Tab. 5.2 such that A occurs at the root and Γ

$$\frac{A}{A \vee B} \qquad \frac{B}{A \vee B} \qquad \frac{A \vee B \quad \begin{array}{c}[A]\\ \vdots \\ C\end{array} \quad \begin{array}{c}[B]\\ \vdots \\ C\end{array}}{C}$$

$$\frac{A \quad B}{A \wedge B} \qquad \frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B}$$

$$\frac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \to B} \qquad \frac{A \to B \quad A}{B} \qquad \frac{\begin{array}{c}[A]\\ \vdots \\ \curlywedge\end{array}}{\neg A}$$

$$\frac{\neg A \quad A}{\curlywedge} \qquad \frac{\curlywedge}{A}$$

Table 5.2: Gentzen-Prawitz style natural deduction rules for **IPL**

is the set of all the undischarged assumptions occurring at the leaves. In turn, A is deducible from Γ if there is a proof of A depending on some $\Delta \subseteq \Gamma$.[8]

Thus, in natural deduction, each logical operator is associated with suitable rules for introducing and eliminating it. Those rules are intended to represent the meaning of the connectives as faithfully as possible. In fact, Gentzen [90] famously suggested that such rules could be seen as definitions of the connectives themselves. What is more, Gentzen argued that the introduction rules alone are enough for that purpose and that the elimination rules are "no more, in the final analysis, than consequences of these definitions" [p. 80 90].[9]. So, purists about **IPL** find a stronghold in the idea that the introduction rules state themselves the meaning of the connectives. The introduction rules are taken as providing a proof-theoretic semantics, where the actual formal derivations in the system *are* the proofs or constructions that appear as primitive (unexplained) notions in the BHK-interpretation.

[8]For formal definitions see [125].

[9]Formally, the "soundness" of the elimination rules is given by the *inversion principle*

5.4.3 Beth tableaux

Beth [33] provided a Gentzen-sequent system for **IPL**. Fitting [79] modified such a system so as to turn it into a refutation tableau system and called it "Beth tableaux". Here we recall those tableaux as recast by Fitting [80] in tree form, branching downward. We use *signed formulae* (S-formulae, for short); namely, expressions of the form $\mathsf{T}\,A$ and $\mathsf{F}\,A$, where A is an (unsigned) formula. Their intended meaning is, respectively, "A is proven" and "A is (yet) unproven". Besides, we say that the conjugate of $\mathsf{T}\,A$ is $\mathsf{F}\,A$ and vice versa. Moreover, we use $\varphi, \psi, \theta, \ldots$, as variables ranging over S-formulae. The rules of Beth tableaux (as formulated by Fitting in [80]) are displayed in Tab. 5.3.

Definitions 5.4.4.

- Given an S-formulae φ, we say that $\mathcal{T}$ is a *tableau* for φ if there exists a finite sequence $(\mathcal{T}_1, \ldots, \mathcal{T}_n)$ such that $\mathcal{T}_1$ is a one-branch, one-node tree whose origin is φ, $\mathcal{T}_n = \mathcal{T}$, and for each $i < n$, $\mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of a rule to preceding S-formulae in the same branch.
- A branch of a tableau is *closed* if it contains both an S-formula and its conjugate.[10]
- A tableau is *closed* if all its branches are closed.
- A *proof* of A is a closed tableau for $\mathsf{F}\,A$.

Note that—with the exception of the double line in the rules for $\rightarrow$ and $\neg$ where the premise is F-signed—Beth tableaux rules are syntactically identical to the rules of the signed version of Smullyan's classical tableaux. Regarding the intuitive interpretation of the rules in Tab. 5.3, the non-branching rules are to be read: if the situation above the line is the case, then the situation(s) below the line is (are) possible, i.e., compatible with the current information state. Correspondingly, if the rule is branching, it says that if the situation above the line is the case, then one of the situations below the line must be possible. In turn, the double line in two of the rules stands for a "barrier" that can be crossed only by T-signed formulae.[11] The intuitive interpretation of these barriers is as follows: suppose that we think of a set of S-formulae on a branch as a (partial) description of an information state.

[10] In [80], closure is defined also with respect to T-signed *falsum* and F-signed *verum*; constants which are assumed to be contained in the language.

[11] In [80], Fitting originally stated the clause represented by those barriers as a rule by itself, and called it "Intuitionistic Branch Modification Rule".

$$
\frac{\mathsf{T}\,A \vee B}{\mathsf{T}\,A \mid \mathsf{T}\,B}
\qquad
\frac{\mathsf{F}\,A \vee B}{\begin{array}{c}\mathsf{F}\,A\\ \mathsf{F}\,B\end{array}}
\qquad
\frac{\mathsf{T}\,A \wedge B}{\begin{array}{c}\mathsf{T}\,A\\ \mathsf{T}\,B\end{array}}
\qquad
\frac{\mathsf{F}\,A \wedge B}{\mathsf{F}\,A \mid \mathsf{F}\,B}
$$

$$
\frac{\mathsf{T}\,A \to B}{\mathsf{F}\,A \mid \mathsf{T}\,B}
\qquad
\begin{array}{c}\mathsf{F}\,A \to B\\ \hline\hline \mathsf{T}\,A\\ \mathsf{F}\,B\end{array}
\qquad
\frac{\mathsf{T}\,\neg A}{\mathsf{F}\,A}
\qquad
\begin{array}{c}\mathsf{F}\,\neg A\\ \hline\hline \mathsf{T}\,A\end{array}
$$

Where the double line indicates that the application of corresponding rule requires all formulae of the form $\mathsf{F}\,A$ above such a line to be deleted.

Table 5.3: Beth tableaux rules

Thus, the rules having barriers imply a possible jump from a current information state to a future one where a previously unproven formula gets proven. However, in such a jump only already proven formulae can be carried to the future state, since previously unproven formulae might have been inadvertently proven. For example, consider the rule for $\to$ where the premise is F-signed: if $A \to B$ is unproven, it is possible to prove A without proving B; for if this were impossible, a proof of B would be "inherent" in a proof of A, but this would constitute a proof of $A \to B$. But there is a barrier in this rule because, in proving A, some additional previously unproven formula might have been inadvertently proven [see 79, 80]. Finally, a closed branch intuitively expresses that for any A and for any information state, it is impossible that A is proven and unproven there.

5.4.4 ***KE*-style Beth tableaux**

A *KE*-version of Beth tableaux is obtained by means of obvious adaptations. We use the same signed formulae, i.e., expressions of the form $\mathsf{T}\,A$ and $\mathsf{F}\,A$, interpreted exactly as in standard Beth tableaux. So, again, $\mathsf{T}\,A$ means "A is proven" and $\mathsf{F}\,A$ means "A is unproven", and we say that the conjugate of $\mathsf{T}\,A$ is $\mathsf{F}\,A$ and vice versa. Besides, we write $\mathsf{T}\,\Gamma$ to mean $\{\mathsf{T}\,B \mid B \in \Gamma\}$. Moreover, we use $\varphi, \psi, \theta, ...$, possibly with subscripts, as variables ranging over S-formulae; and $X, Y, Z, ...$, possibly with subscripts, as variables ranging over sets of S-formulae. Besides, we use $\bar{\varphi}$ to denote the conjugate of φ. Now, as expected, the rules of the *KE*-version of Beth tableaux (with the exception of a double line in the rules for $\to$ and $\neg$ where the premise is F-signed) are syntactically identical to the rules of (signed) *KE*. The rules of *KE*-style Beth tableaux are displayed in Tab. 5.4.

$$
\begin{array}{c}\mathsf{T}\,A \vee B \\ \mathsf{F}\,A \\ \hline \mathsf{T}\,B\end{array}
\qquad
\begin{array}{c}\mathsf{T}\,A \vee B \\ \mathsf{F}\,B \\ \hline \mathsf{T}\,A\end{array}
\qquad
\begin{array}{c}\mathsf{F}\,A \vee B \\ \hline \mathsf{F}\,A \\ \mathsf{F}\,B\end{array}
$$

$$
\begin{array}{c}\mathsf{F}\,A \wedge B \\ \mathsf{T}\,A \\ \hline \mathsf{F}\,B\end{array}
\qquad
\begin{array}{c}\mathsf{F}\,A \wedge B \\ \mathsf{T}\,B \\ \hline \mathsf{F}\,A\end{array}
\qquad
\begin{array}{c}\mathsf{T}\,A \wedge B \\ \hline \mathsf{T}\,A \\ \mathsf{T}\,B\end{array}
$$

$$
\begin{array}{c}\mathsf{T}\,A \to B \\ \mathsf{T}\,A \\ \hline \mathsf{T}\,B\end{array}
\qquad
\begin{array}{c}\mathsf{T}\,A \to B \\ \mathsf{F}\,B \\ \hline \mathsf{F}\,A\end{array}
\qquad
\begin{array}{c}\mathsf{F}\,A \to B \\ \hline\hline \mathsf{T}\,A \\ \mathsf{F}\,B\end{array}
$$

$$
\begin{array}{c}\mathsf{T}\,\neg A \\ \hline \mathsf{F}\,A\end{array}
\qquad
\begin{array}{c}\mathsf{F}\,\neg A \\ \hline\hline \mathsf{T}\,A\end{array}
\qquad
\begin{array}{c} \\ \hline \mathsf{T}\,A \mid \mathsf{F}\,A\end{array}
$$

Where the double line indicates that the application of corresponding rule requires all formulae of the form $\mathsf{F}\,A$ above such a line to be deleted.

Table 5.4: Rules of *KE*-style Beth tableaux

Definitions 5.4.5.

- Given a set of S-formulae $X = \{\varphi_1, ..., \varphi_m\}$, we say that $\mathcal{T}$ is a *KE-style tableau for X* if there exists a finite sequence $(\mathcal{T}_1, ..., \mathcal{T}_n)$ such that $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $(\varphi_1, ..., \varphi_m)$, $\mathcal{T}_n = \mathcal{T}$, and for each $i < n$, $\mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of an elimination rule to preceding S-formulae in the same branch, or by an application of *PB*.

- A branch of a *KE*-style tableau is *closed* if it contains both an S-formula and its conjugate.

- A *KE*-style tableau is *closed* if all its branches are closed.

- A *proof* of A from Γ is a closed *KE*-style tableau for $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$.

The intuitive interpretation of the rules in Tab. 5.4 is along the lines of that of standard Beth tableaux rules. That is, the non-branching rules in Tab. 5.4 are to be read: if the situation above the line is the case, then the situation(s) below the line is (are) compatible with the current information state. In turn, there is only one branching rule that we again call *PB* since it expresses a *generalized* Principle

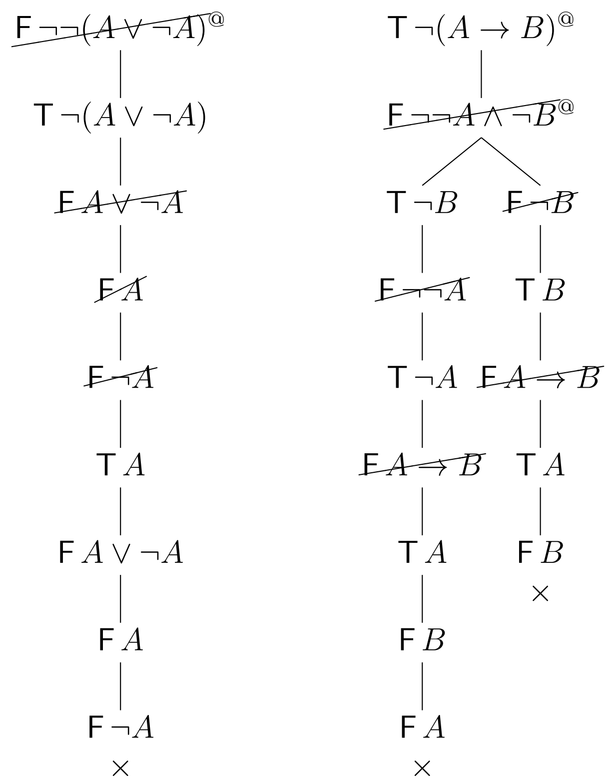


Figure 5.2: *KE*-style proofs

of Bivalence, and which intuitively says the following: for any formula A and for any information state, either A is proven or unproven there. As for the double line in two of the rules, its intuitive interpretation is exactly as in standard Beth tableaux. A couple of examples of *KE*-style proofs are displayed in Fig. 5.2. We mark the assumptions with an '@' and, for the sake of clarity, cross out formulae according to applications of rules with a barrier.

Soundness of our *KE*-version of Beth tableaux with respect to Kripke semantics is straightforward. Given a Kripke model $\mathfrak{M} = \langle\langle S, R\rangle, \{\mathfrak{v}_a\}_{a \in S}\rangle$, we say that an information state $a \in S$ *realizes* a S-formula $\mathsf{T}\,A$ iff $\mathfrak{M}, a \Vdash_K A$, and a S-formula $\mathsf{F}\,A$ iff $\mathfrak{M}, a \nVdash_K A$. We also say that a *realizes* a set of S-formulae X iff a realizes each $\varphi \in X$.

Proposition 5.4.6 (Soundness). *If there is a closed* KE*-style tableau* $\mathcal{T}$ *for* $\mathsf{T}\,\Gamma \cup \{\mathsf{F}\,A\}$*, then* $\Gamma \vDash_K A$*.*

Proof. The proof is by induction on the structure of $\mathcal{T}$. Let $\mathfrak{M} = \langle\langle S, R\rangle, \{\mathfrak{v}_a\}_{a\in S}\rangle$ be a Kripke model.

Basis: $\mathcal{T}$ consists only of the assumption $\mathsf{F}\,A$. Then suppose that $a \in S$ realizes $\mathsf{F}\,A$, then a trivially realizes $\mathsf{F}\,A$.

Inductive step: There are 12 cases according to the last rule applied in $\mathcal{T}$. We present two cases, the rest being similar.

- The last rule applied is a conditional elimination concluding $\mathsf{T}\,D$. So, the premises are $\mathsf{T}\,C \to D$ and $\mathsf{T}\,C$. Suppose that a realizes $\{\mathsf{T}\,C \to D, \mathsf{T}\,C\}$. Thus, $\mathfrak{M}, a \Vdash_K C \to D$ and $\mathfrak{M}, a \Vdash_K C$. By definition, the former means that for all b such that aRb, if $\mathfrak{M}, b \Vdash_K C$ then $\mathfrak{M}, b \Vdash_K D$. Given that aRa, we have that if $\mathfrak{M}, a \Vdash_K C$ then $\mathfrak{M}, a \Vdash_K D$. Since indeed $\mathfrak{M}, a \Vdash_K C$, then we have $\mathfrak{M}, a \Vdash_K D$. Thus, a realizes $\mathsf{T}\,D$.

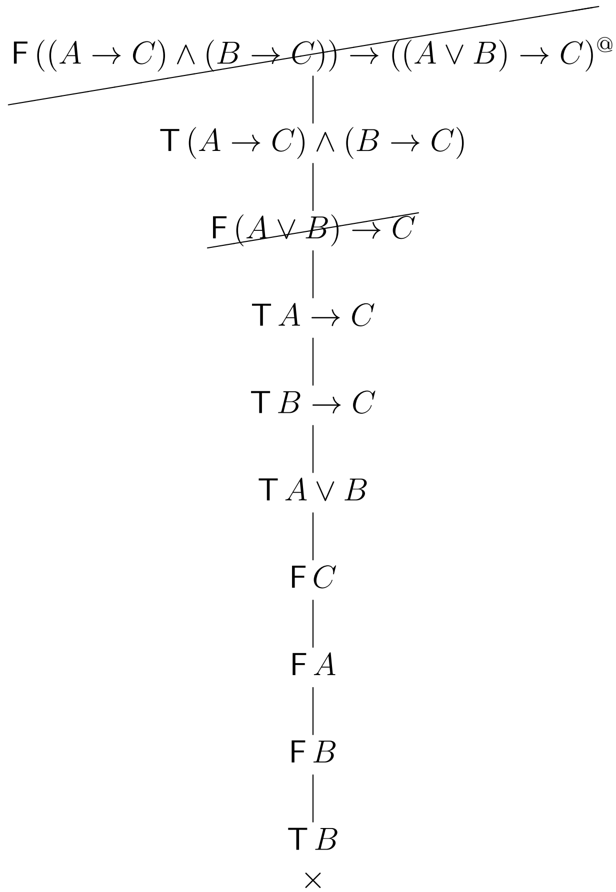
- The last rule applied is a negation elimination concluding $\mathsf{T}\,C$. So, the premise is $\mathsf{F}\,\neg C$. Suppose that a realizes $\mathsf{F}\,\neg C$. Thus, $\mathfrak{M}, a \nVdash_K \neg C$, and so by definition there is b, aRb, such that $\mathfrak{M}, b \Vdash_K C$. So b, aRb, realizes $\mathsf{T}\,C$.

Now, of course, no information state in a Kripke model can realize two conjugate S-formulae simultaneously. Then, if $\mathcal{T}$ is a closed *KE*-style tableau, no information state in a Kripke model can realize all the initial S-formulae of $\mathcal{T}$. Hence, if $\mathcal{T}$ is a closed *KE*-style tableau for $\mathsf{T}\,\Gamma \cup \{FA\}$, it follows that an information state in a Kripke model realizes $\mathsf{T}\,A$ whenever it realizes $\mathsf{T}\,\Gamma$, and so $\Gamma \vDash_K A$. □

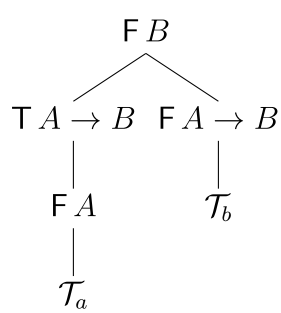
Now, the completeness of our *KE*-version of Beth tableaux can be shown in several ways. One is by proving that the set of theorems of such a system includes some standard set of axioms for **IPL** and is closed under *modus ponens*. Below we take this option with respect to the original set of axioms provided by Heyting.

Proposition 5.4.7 (Completeness). *If $\emptyset \vdash_{\boldsymbol{IPL}} A$, then there is a closed KE-style tableau for $\emptyset \cup \{\mathsf{F}\,A\}$.*

Proof. Checking that the 11 axioms given by Heyting are theorems of our *KE*-style Beth tableaux system is routine. As an example, we check one of those axioms:



Now, in order to show that our *KE*-style system is closed under *modus ponens*, suppose that there are closed *KE*-style Beth tableaux $\mathcal{T}_a$ and $\mathcal{T}_b$ respectively for $\{\mathsf{F}\, A\}$ and $\{\mathsf{F}\, A \rightarrow B\}$. Then, the following tableau:

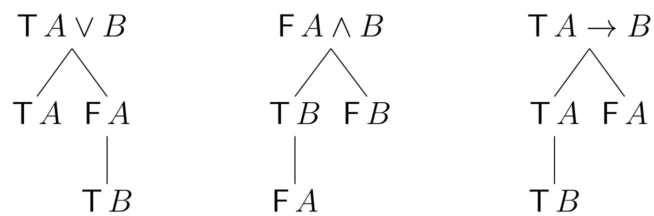


is closed for $\{\mathsf{F}\, B\}$. □

The above proof provides a simulation of a standard axiomatic system by our *KE*-style system. In fact, a direct simulation of standard Beth tableaux by our *KE*-style system provides another easy proof of completeness of the latter:

Proposition 5.4.8 (Completeness). *If there is a closed Beth tableau for* $\mathsf{F}\,A$*, then there is a closed KE-style tableau for* $\emptyset \cup \{\mathsf{F}\,A\}$.

Proof. First, observe that standard Beth tableaux and its *KE*-version share the rules $\mathsf{F}\vee$, $\mathsf{T}\wedge$, $\mathsf{F}\rightarrow$, $\mathsf{T}\neg$ and $\mathsf{F}\neg$. Then, it is easy to see that the rules $\mathsf{F}\rightarrow$ and $\mathsf{F}\neg$ formulated in terms of boxes with barriers are a purely notational variant of the original rules. Now, the rest of the rules of standard Beth tableaux can be simulated in our *KE*-version as follows:



Thus, if $\mathcal{T}$ is a closed Beth tableau for $\mathsf{F}\,A$, then replace each application of a (non-shared) Beth tableaux rule with its simulation in the *KE*-version. The result is a closed *KE*-style tableau $\mathcal{T}'$ for $\emptyset \cup \{\mathsf{F}\,A\}$. □

5.5 Intelim deduction in IPL

In order to define our depth-bounded approximations to **IPL** we shall define a *KE*/*KI*-style system that, as in previous Chapters, we call *intelim method*. So, we shall enrich our *KE*-style system above with suitable introduction rules for the connectives. We have two reasons for using both introduction and elimination rules: (i) it allows for more natural proofs; (ii) it reduces the number of applications of *PB* that, as explained below, is key to our measure of the *depth* of an inference.

Now, unlike the elimination rules of our *KE*-style system, some introduction rules for the intuitionistic connectives inherently involve temporary assumptions. In the *KE*-style system above, the unique rule involving hypothetical information about a formula is *PB*, information that is ultimately "discharged" when all the branches of the corresponding tree are closed. By contrast, the elimination rules do not involve, *per se*, any temporary assumption. For instance, if an agent actually possesses the information that $\neg A$ is unproven in the current information state, then she also actually possesses the information that A can possibly be proven in some future state. Again, information that an agent actually posses or holds is what we call *actual* information. So, the elimination rules of our *KE*-style system involve all only actual (in some cases, modal) information.

On the other hand, the intuitionistic *truth* of formulae whose main connective is $\rightarrow$ or $\neg$ cannot be determined without *assuming* that there is an essentially richer information state where certain formulae hold. For instance, for an agent to recognize that a conditional $A \rightarrow B$ is proven at a state where A is unproven, she must in principle transfer from the *actual* state to a *virtual* one where A is proven and any other formula remains the same; i.e., she must reason *as if* her state were the latter one. If in such a virtual state the agent verifies B as well, she can conclude that $A \rightarrow B$ must be proven at the actual state. In fact, this situation underlies the intractability of **IPL**: when the evaluation of the truth of a formula requires weaving in and out of a complex recursive pattern of virtual information states, the complexity of such an evaluation may soon get out of control. This is formally shown by the result that the decision problem for the pure implication fragment of **IPL** is P-SPACE complete [140]. In order to assent to a conditional $A \rightarrow B$, the agent essentially needs to go beyond the information that she holds, using hypothetical information; i.e., simulating situations in which she possesses information that she does not actually possess. Even though all hypothetical information may be eventually discharged—to the effect that the conclusion depends only on information actually possessed by the agent—it is the case that the corresponding deduction steps could not be performed at all without using that hypothetical information. Again, hypothetical information that an agent does not hold, but she temporarily assumes as if she held it is precisely what we call *virtual* information.

Conveniently, we can represent the scope of temporary assumptions with boxes, as in Jaśkowski-style natural deduction [102]. These boxes serve to demarcate the scope of *subordinate proofs* or, simply, subproofs where temporary assumptions are used. More specifically, boxes indicate that the enclosed formulae are considered as being derived only under corresponding temporary assumptions. Then, those assumptions are "discharged"—in the sense that the corresponding conclusion no longer depends on them but only on other still undischarged assumptions, if any—when the conditions stipulated by the corresponding rule are met. This discharge is symbolized by closing off a box and writing immediately outside it the corresponding conclusion. So, boxes can be nested in that inside them we continue applying rules. Besides, new boxes can be open after old ones have been closed; yet boxes cannot overlap. The scoping rule for boxes is then that each S-formula occurring in a box can be used, as premise of a rule application, in every box contained in it and cannot be used in any other box. Indeed, once a box is closed, no S-formula inside it can be used outside the box.

Thus, we can represent the introduction rules for the truth of formulae whose main connective is $\to$ and $\neg$ as follows:

$$\begin{array}{c} \boxed{\begin{array}{c} \mathsf{T}\,A \\ \vdots \\ \mathsf{T}\,B \end{array}} \\ \mathsf{T}\,A \to B \end{array} \qquad\qquad \begin{array}{c} \boxed{\begin{array}{c} \mathsf{T}\,A \\ \vdots \\ \times \end{array}} \\ \mathsf{T}\,\neg A \end{array}$$

Where: (i) the vertical dots stand for a derivation of the symbol below them depending on assumptions that may include that above the dots; (ii) the second rule requires the addition of another rule intuitively expressing a *generalized* Principle of Non-contradiction and that, thus, we shall call *PNC*:

$$\frac{\begin{array}{c} \mathsf{T}\,A \\ \mathsf{F}\,A \end{array}}{\times}$$

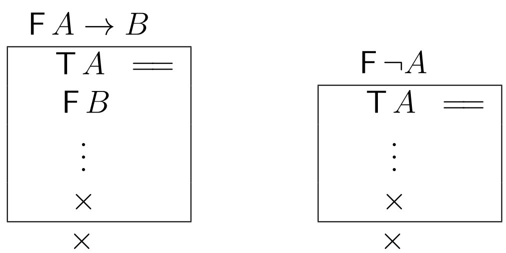
This latter rule, as *PB*, is structural in that it does not involve any particular logical operator and intuitively says: for any A and any information state, it is impossible that A is proven and unproven there. In turn, accordingly, we can reformulate *PB* using boxes as follows:

$$\begin{array}{c} \boxed{\begin{array}{c} \mathsf{T}\,A \\ \vdots \\ \varphi/\times \end{array}} \boxed{\begin{array}{c} \mathsf{F}\,A \\ \vdots \\ \varphi/\times \end{array}} \\ \varphi/\times \end{array}$$

Where: (i) the vertical dots stand for a derivation of the symbol below them depending on assumptions that may include that above the dots; (ii) $\varphi/\times$ means that the above derivation delivers either φ or $\times$; in turn, $\times$ is obtained outside the two parallel boxes if both symbols immediately above it are $\times$, while φ is obtained outside the parallel boxes when either one of the symbols immediately above it is φ and the other is $\times$, or both symbols immediately above it are φ.

These three boxed rules involve virtual (modal) information. Each of them involves a (pair of) temporary assumption(s) about an arbitrary formula being proven or unproven at a virtual information state. In turn, each of those assumptions appears at the top of the corresponding box, above the vertical dots. The derivation represented by the vertical dots *depends* on the respective temporary assumption. Then, when the symbol below the vertical dots is somehow derived, probably by means of the respective temporary assumption, the latter is "discharged" in the conclusion of the rule outside the box. Intuitively, although such a deduction takes place inside a virtual information state, once the information contained in the actual state combined with the corresponding assumption associated with the virtual one leads to the symbol at the bottom of the box, the latter can be transferred to the actual state. Thus, the intuitive interpretation of these introduction rules for $\rightarrow$ and $\neg$ is the same as their Gentzen-Prawitz style natural deduction counterparts. As for the reformulated rule *PB*, its intuitive interpretation is exactly as before: for any A and for any information state, either A is proven or unproven there. However, this "natural deduction variant" of *PB* allows us for the introduction of pieces of information that might not be nested (see the first example of Figure 5.4 below).

Now, the boxed introduction rules above can be taken as *derived rules* in a *KE*/*KI*-style system by simulating them using *PNC*, the boxed formulation of *PB*, and the following boxed formulation of the elimination rules with barriers:



Where: (i) '$=$' indicates that the application of corresponding rule requires that only T-signed formulae above that double line can be used inside the box; (ii) the vertical dots stand for a derivation of the conjugate of a T-signed formula occurring above the box or a pair of conjugate S-formulae. These boxes with barriers comply with the scoping rule of single boxes and a further condition: only T-signed formulae above the barrier can be used inside the corresponding box (yet, once the box with barrier is closed, F-signed formulae above the corresponding barrier can be used again). The intuition underlying these boxes with barriers is that they stand for a possible jump from the current information state to a future one, and thus only T-signed formulae can enter the box. Therefore, we must beware that, unlike single boxes, boxes with barriers *do not* demarcate the scope of temporary assumptions since, as explained above, the associated elimination rules involve only actual information. The S-formula(e) appearing at the top of the boxes with barriers, above the vertical dots, is (are) the *conclusion*(s) of the premise. Accordingly, unlike

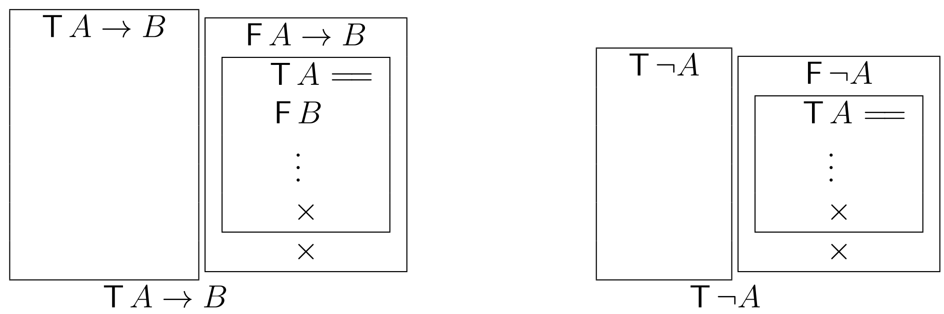


Figure 5.3: Simulated boxed introduction rules

the boxed introduction rules and PB, the boxed version of elimination rules allow us to obtain outside the corresponding box only $\times$. This is because these latter rules imply a jump from the current information state to a future one (jump which is represented by the barrier at the top of the box) and, naturally, a conclusion obtained in a future state cannot be transferred to the current one. However, if inside a box with a barrier an inconsistency is reached, that indeed would imply that the current information state is inconsistent and, so, $\times$ can be obtained outside the box. That said, the simulation that brought us to the boxed formulation of elimination rules is displayed in Figure 5.3.

Now, following the key idea underlying the depth-bounded approach according to which the meaning of a connective is specified solely in terms of information that is actually possessed by an agent, we shall take the boxed introduction rules as derived rules. This is motivated by technical as well as conceptual reasons. On the technical side, as suggested at the beginning of this section, when virtual information is essentially used in defining the meaning of a connective—say, by introduction rules as in proof-theoretic semantics—its unbounded use must be tolerated if this meaning is to be fully exploited in drawing inferences. But then the corresponding logic, as in the case of **IPL** at issue, may turn out to be (likely) intractable. On the conceptual side, if an agent understands the informational meaning of a sentence A, then she should be able to tell, in practice and not only in principle, whether or not she actually holds the information that A is proven, or the information that A is unproven or neither of them.[12] So, as in the classical and many-valued cases previously addressed, we seek for a system which has operational rules involving only actual information—so that they fix the meaning of the connectives solely in terms of information that the agent holds—and a purely structural principle that governs

[12]Recall that in the context of **CPL**, this intuition was called "Strong Manifestability".

the manipulation of virtual information—in terms of which the nested use of virtual information can be bounded. Therefore, the intelim method we shall use to define depth-bounded approximations is constituted by the rules in Table 5.5.

In these rules, we intuitively re-interpret $\mathsf{T}\,A$ as "the agent holds the information that A is proven", and $\mathsf{F}\,A$ as "the agent holds the information that A is unproven". We shall refer to the introduction and elimination rules of Table 5.5 taken together as *intelim* rules. So, the meaning of the connectives is specified solely in terms of actual information by the intelim rules. Observe that, corresponding to the fact that **IPL** is $\vee/\wedge$-classical, the intelim rules for those connectives are dual of each other. Besides, expressing a hallmark of **IPL**, a sentence and its negation are *not* treated in a symmetric way. Now, in the elimination rules, we shall refer to the premise containing the connective that is to be eliminated as *major* and to the other premise as *minor*. In turn, *PB* is the only rule introducing virtual information. Each application of *PB* stands for the introduction of virtual information about an arbitrary formula A; formula that we shall call the *PB-formula*. From our informational viewpoint, the main conceptual advantage of this proof-theoretical characterization consists in that it clearly separates the rules that fix the meaning of the connectives in terms of the information that an agent holds (the intelim rules) from the structural rule that introduces virtual information (*PB*). Now, the applications of the rules in Table 5.5 generate partially ordered sets (posets), and the resulting intelim method can be used as both a direct-proof method and refutation method.

Definitions 5.5.1.

- Let $X = \{\varphi_1, ..., \varphi_m\}$. Then $\mathcal{P}$ is an *intelim tableau for* X if there exists a finite sequence $(\mathcal{P}_1, ..., \mathcal{P}_n)$ such that $\mathcal{P}_1$ is a poset consisting of the sequence $(\varphi_1, ..., \varphi_m)$, $\mathcal{P}_n = \mathcal{P}$, and for each $i < n$, $\mathcal{P}_{i+1}$ results from $\mathcal{P}_i$ by an application of an intelim rule or *PNC* to preceding S-formulae in the same path, or by an application of *PB*.

- Given an intelim tableau $\mathcal{P}$ for X, $\mathcal{P}'$ is an intelim subtableau of $\mathcal{P}$ if $\mathcal{P}' = \mathcal{P}_j$ for some $j \leq n$.

- An intelim tableau for X is *completed* if no box occurring in it is open.

- An intelim tableau for X is *closed* if it is completed and $\times$ occurs in its last line; it is *open* if it is completed and $\times$ does not occur in its last line.

- An *intelim refutation* of X is a closed intelim tableau for X.

$$\begin{array}{c} \mathsf{T}\,A \lor B \\ \mathsf{F}\,A \\ \hline \mathsf{T}\,B \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \lor B \\ \mathsf{F}\,B \\ \hline \mathsf{T}\,A \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \lor B \\ \hline \mathsf{F}\,A \\ \mathsf{F}\,B \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \land B \\ \mathsf{T}\,A \\ \hline \mathsf{F}\,B \end{array}$$

$$\begin{array}{c} \mathsf{F}\,A \land B \\ \mathsf{T}\,B \\ \hline \mathsf{F}\,A \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \land B \\ \hline \mathsf{T}\,A \\ \mathsf{T}\,B \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \to B \\ \mathsf{T}\,A \\ \hline \mathsf{T}\,B \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \to B \\ \mathsf{F}\,B \\ \hline \mathsf{F}\,A \end{array}$$

$$\begin{array}{c} \mathsf{F}\,A \to B \\ \boxed{\begin{array}{l} \mathsf{T}\,A \;=\!= \\ \mathsf{F}\,B \\ \;\vdots \\ \times \end{array}} \\ \times \end{array} \qquad \begin{array}{c} \mathsf{T}\,\neg A \\ \hline \mathsf{F}\,A \end{array} \qquad \begin{array}{c} \mathsf{F}\,\neg A \\ \boxed{\begin{array}{l} \mathsf{T}\,A \;=\!= \\ \;\vdots \\ \times \end{array}} \\ \times \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \\ \hline \mathsf{T}\,A \lor B \end{array}$$

$$\begin{array}{c} \mathsf{T}\,B \\ \hline \mathsf{T}\,A \lor B \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \\ \mathsf{F}\,B \\ \hline \mathsf{F}\,A \lor B \end{array} \qquad \begin{array}{c} \mathsf{F}\,A \\ \hline \mathsf{F}\,A \land B \end{array} \qquad \begin{array}{c} \mathsf{F}\,B \\ \hline \mathsf{F}\,A \land B \end{array}$$

$$\begin{array}{c} \mathsf{T}\,A \\ \mathsf{T}\,B \\ \hline \mathsf{T}\,A \land B \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \\ \mathsf{F}\,B \\ \hline \mathsf{F}\,A \to B \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \\ \hline \mathsf{F}\,\neg A \end{array} \qquad \begin{array}{c} \mathsf{T}\,A \\ \mathsf{F}\,A \\ \hline \times \end{array}$$

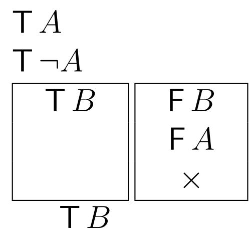
$$\begin{array}{c} \boxed{\begin{array}{c} \mathsf{T}\,A \\ \vdots \\ \varphi/\times \end{array}} \boxed{\begin{array}{c} \mathsf{F}\,A \\ \vdots \\ \varphi/\times \end{array}} \\ \varphi/\times \end{array}$$

Where: (i) '$=\!=$' indicates that the application of corresponding rule requires that only T-signed formulae above that double line can be used inside the box; (ii) the vertical dots stand for a derivation of the symbol below them depending on assumptions that may include those above the dots; (iii) $\varphi/\times$ means that the above derivation delivers either φ or $\times$; in turn, $\times$ is obtained outside the two parallel boxes if both symbols immediately above it are $\times$, while φ is obtained outside the parallel boxes when either one of the symbols immediately above it is φ and the other is $\times$, or both symbols immediately above it are φ.

Table 5.5: Rules of the intelim method

- An *intelim proof* of φ from X is an open intelim tableau for X such that φ occurs in its last line.

Note that the above definition accounts for the explosivity of **IPL** as follows:



A couple of examples of intelim proofs are displayed in Figure 5.4, where we mark each *initial* assumption with an '@'. The soundness of the rules in Table 5.5 with respect to Kripke semantics is again straightforward. For example, take the case where the last rule applied is a negation elimination and $\times$ occurs in the last line of the corresponding tableau. So, the premise is $\mathsf{F}\,\neg A$, occurring in the last line of the subtableau with undischarged assumptions (if any) Y. Now, suppose that and information state a in a Kripke model $\mathfrak{M}$ realizes $Y \cup \{\mathsf{F}\,\neg A\}$. Thus, $\mathfrak{M}, a \nVdash_K \neg A$, and so by definition there is b, aRb, such that $\mathfrak{M}, b \Vdash_K A$. Then suppose further that the subtableau with undischarged assumptions $Y \cup \{\mathsf{T}\,A\} \cup Z$ is closed. Thus, b cannot realize $Y \cup \{\mathsf{T}\,A\} \cup Z$ and so a cannot realize $Y \cup \{\mathsf{F}\,\neg A\} \cup Z$. Therefore, the subtableau with undischarged assumptions $Y \cup Z$ is also closed. In turn, the completeness of our intelim method trivially follows from the completeness of its *KE*-style subsystem (the boxed version of the elimination rules and *PB* is a mere notational variant of the unboxed version).

Remark 12. In Figure 5.3 we showed that boxed introduction rules can be simulated via *PB*, boxed elimination rules and *PNC* of our intelim method. It is not difficult to see that the (unboxed) introduction rules of the method can be simulated via *PB*, elimination rules, and *PNC*. Thus, the *KE*-style subsystem of the intelim method can simulate the whole of it. Observe that this implies that the intelim method enjoys the subformula property. Namely, given Proposition 5.4.8 (the *KE*-style subsystem can simulate Beth tableaux), the *KE*-style subsystem enjoys the subformula property and, in turn, since such a subsystem can simulate the whole intelim method, the latter also enjoys the property. In Subsection 5.5.1 below, we pave the way for a constructive proof of the subformula property of the intelim method via normalization, but we are still working on it.

Now, we identify the basic (*0-depth*) logic of our hierarchy of approximations with the inferences that an agent can draw without using virtual information; i.e., 1 T $A \vee B^{@}$

2 T $A \to (C \vee D)^{@}$

3 T $B \to (C \vee D)^{@}$

4 T $C \to E^{@}$

5 T $D \to F^{@}$

<table>
<tr><td>6 T A
7 T $C \vee D$</td><td>F A
T B
T $C \vee D$</td></tr>
</table>

9 T $C \vee D$

<table>
<tr><td>10 T C
11 T E
12 T $E \vee F$</td><td>F C
T D
T F
T $E \vee F$</td></tr>
</table>

14 T $E \vee F$

1 T $A \vee \neg A^{@}$

<table>
<tr><td>2 T $((A \to B) \to A) \to A$</td><td>F $((A \to B) \to A) \to A$
<table><tr><td>T $(A \to B) \to A$ $=\!=$
F A
T $\neg A$
F $A \to B$
<table><tr><td>T A $=\!=$
F B
F A
$\times$</td></tr></table>$\times$</td></tr></table>$\times$</td></tr>
</table>

13 T $((A \to B) \to A) \to A$

Figure 5.4: Intelim proofs

without making hypothetical assumptions that go beyond the information that she holds. We shall show that a natural proof-theoretic characterization of this basic logic is obtained by means of the set of intelim rules and *PNC*. Note that such a characterization bears some resemblance with natural deduction, but does not have discharge rules, since no hypothetical reasoning is involved. In turn, given that the intelim rules have all a linear format, their application generates *intelim sequences*. Namely, finite sequences $(\varphi_1, ..., \varphi_n)$ of S-formulae such that, for every $i = 0, ..., n$, either φ_i is an *initial* assumption or it is the conclusion of the application of an intelim rule to preceding S-formulae. In Figure 5.5 we show a pair of examples of these sequences.

The intelim rules and *PNC* are not complete for full **IPL** but only for the 0-depth logic of our hierarchy of approximations. Completeness for the k-depth logics, $k > 0$ (and in the limit for full **IPL**) is obtained by allowing the nested introduction of virtual information via *PB*. As explained above, each application of this latter rule stands for the introduction of virtual information about a formula being proven or unproven at some virtual information state. We shall refer to the temporary assumptions that are introduced by each application of *PB* as *virtual assumptions*. In turn, we shall call the (unsigned) formula A to which *PB* is applied as the *PB-formula*. Further, when applications of *PB* are allowed, deductions are represented by posets. *PB* is essentially a cut rule which may introduce formulae of arbitrary degree. However, we shall show later on, in Theorem 12, that we can restrict the applications of *PB* to subformulae of the *initial* assumptions or the conclusion of the given inference; i.e., restrict its application so as to satisfy the subformula property. As in the classical and many-valued cases addressed in the previous Chapters, intuitively, the more virtual information needs to be introduced via *PB*, the harder the inference is for the agent, both from the computational and the cognitive viewpoint. Thereby, the nested applications of *PB* provide a sensible measure of inferential *depth*. This naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations in terms of the maximum number of nested applications of *PB* that are allowed.

Now, although *PB* may introduce formulae of arbitrary degree, the set of formulae to which it is applied can be bounded in a variety of ways without loss of completeness. As in the previous Chapters, we call this set *virtual space* and define it as a function f of the set $\Gamma \cup \{A\}$, consisting of the premises Γ and of the conclusion A of the given inference. As before, the strictest way of bounding the virtual space consists in allowing as *PB*-formulae only atomic formulae that occur in $\Gamma \cup \{A\}$. A more liberal option is allowing only subformulae of the formulae in $\Gamma \cup \{A\}$. Specifically, let $\mathcal{F}$ be the set of all functions f on the finite subsets of

$\mathsf{T}\,(A \vee B) \rightarrow \neg C^{@}$
$\mathsf{T}\,A^{@}$
$\mathsf{T}\,(A \wedge D) \rightarrow C^{@}$
$\mathsf{T}\,A \vee B$
$\mathsf{T}\,\neg C$
$\mathsf{F}\,C$
$\mathsf{F}\,A \wedge D$
$\mathsf{F}\,D$

1 $\mathsf{T}\,A \rightarrow (B \rightarrow C)^{@}$
2 $\mathsf{F}\,B \rightarrow (A \rightarrow C)^{@}$
3 $\mathsf{T}\,B$ $=\!=$
4 $\mathsf{F}\,A \rightarrow C$
5 $\mathsf{T}\,A$ $=\!=$
6 $\mathsf{F}\,C$
7 $\mathsf{T}\,B \rightarrow C$
8 $\mathsf{T}\,C$
9 $\times$
10 $\times$
11 $\times$

Figure 5.5: Intelim sequences

$F(\mathcal{L})$ such that: (i) for all Δ, $\mathsf{at}(\Delta) \subseteq f(\Delta)$; (ii) $f(\Delta)$ is closed under subformulae, i.e., $\mathsf{sub}(f(\Delta)) = f(\Delta)$; (iii) the *size* of $f(\Delta)$ is bounded above by a polynomial in the size of Δ, i.e., $|f(\Delta)| \leq p(|\Delta|)$ for some fixed polynomial p. (This last requirement will be essential in order to define *tractable* approximations bellow.) Again, the choice of an specific function to yield suitable values of the virtual space for each particular deduction problem is the result of decisions that are conveniently made by the system designer, depending on the intended application.[13] In turn, the functions in $\mathcal{F}$ are partially ordered by the relation $\trianglelefteq$ such that $f_1 \trianglelefteq f_2$ iff, for every finite Δ, $f_1(\Delta) \subseteq f_2(\Delta)$. Once again, distinguished examples of functions in $\mathcal{F}$ are the identity function $f(\Delta) = \Delta$, sub and at. However, in general, $f(\Delta)$ may contain formulae that are not in sub. Thereby, our intelim method allows for (possibly shorter) deductions that do not have the subformula property simply by permitting applications of the boxed introduction rules or *PB* to formulae that are not subformulae either of the premises or of the conclusion. However, even in this latter deductions the virtual space is still bounded.

Now, *PB* is not the unique rule of our intelim method that may bring about violations of the subformula property. The introduction rules of the method could in principle be applied *ad infinitum*, leading to ever more complex formulae. Nevertheless, as we shall suggest below, the application of both *PB* and the introduction rules seem likely to be restricted so as to satisfy the subformula property. Namely, we shall suggest that every intelim proof of φ from X (intelim refutation of X) can be transformed into an intelim proof of φ from X (an intelim refutation of X) with the subformula property.

5.5.1 Towards normalization

Disclaimer: The content of this Subsection is based on [63] but is not fully settled, let alone verified. Particularly, the transformations for *PB*-canonicity in Table 5.6 seem problematic, and Table 5.11 is incomplete since the corresponding transformations for $\rightarrow$ and $\neg$ are missing. One of the main results of this Subsections would be a constructive proof of the subformula property of the intelim method. However, that the method enjoys such a property already follows from Proposition 5.4.8 and the fact that the *KE*-style subsystem of the method can simulate the whole of it, as noted in Remark 12.

As mentioned in the previous Chapters, the subformula property allows us to

[13] As in the cases addressed in the previous Chapters, in the approximations defined below, such decisions affect the deductive power of each given approximation, and so the “speed” at which the approximation process converges to full **IPL**.

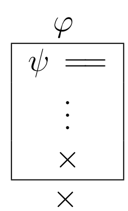
search for proofs or refutations by analytic methods; i.e., by considering solely deduction steps involving formulae that are "contained" in the assumptions, or also in the conclusion in the case of proofs. This implies a drastic reduction of the search space which is crucial for the purpose of automated deduction. At the propositional level, this search space is finite for each putative inference, paving the way for decision procedures. Particularly, in our intelim method, the subformula property guarantees that we can impose a bound on the applications of *PB*, which could in principle be applied to arbitrary formulae, with no loss of deductive power. Similarly, it guarantees that we can impose a bound on the sensible applications of introduction rules, which could in principle be indefinitely applied, leading to ever more complex formulae.

Now, we say that the *unsigned part* of an S-formula is the unsigned formula that results from it by removing its sign. Given an S-formula φ, we denote by φ^u the unsigned part of φ and by X^u the set $\{\varphi^u \,|\, \varphi \in X\}$. Moreover, henceforth with "(sub)tableau" we mean "*completed* intelim (sub)tableau", and with $\mathcal{P}$ (possibly with subscript) denote always a *completed* intelim (sub)tableau.

Definition 5.5.2. An intelim proof $\mathcal{P}$ of φ from X (an intelim refutation of X) has the *subformula property* if, for every S-formula ψ occurring in $\mathcal{P}$, $\psi^u \in \mathsf{sub}(X^u \cup \{\varphi^u\})$ ($\psi^u \in \mathsf{sub}(X^u)$).

Definition 5.5.3. An application of *PB* is *canonical* in an intelim tableau $\mathcal{P}$ if there is no application of an intelim rule or *PNC* in $\mathcal{P}$ below *PB*'s conclusion. An intelim tableau $\mathcal{P}$ is *PB-canonical* if all applications of *PB* in it are canonical.

Any intelim proof (refutation) can be turned into an *PB*-canonical one by applying the transformations in Table 5.6.[14] These transformations can be respectively applied whenever: φ/θ is an instance of a one-premise, one-conclusion, intelim rule; φ, $\bar{\varphi}/\times$ is an instance of *PNC*; φ, ψ/θ is an instance of a two-premise intelim rule; φ/ψ, θ is an instance of a one-premise, two-conclusion, intelim rule; and



[14]In what follows, our transformations are to be interpreted along these lines: replace locally a subtableau of the form shown on the left of "⇝" with a subtableau of the form shown on its right. Moreover, beware that the subscripts on $\mathcal{P}$ in our transformations are used differently that in Definitions 5.5.1.

is an instance of a boxed elimination rule. The repeated application of these transformations results in pushing downwards all the applications of *PB* so that, eventually, the conclusion of an application of *PB* is never used as a premise of an intelim rule or *PNC*, and must be identical to the conclusion of the whole proof or refutation.

Given a subtableau $\mathcal{P}'$ of $\mathcal{P}$, let $d_{PB}(\mathcal{P}')$ be equal to zero if $\mathcal{P}'$ ends with an application of *PB*, and equal to the total number of applications of *PB* in $\mathcal{P}'$ if $\mathcal{P}'$ ends with an application of an intelim rule or *PNC*. We introduce the parameter $d_1(\mathcal{P}) = \langle m, n\rangle$ where m is the maximum value of d_{PB} for a subtableau of $\mathcal{P}$ and n is the number of subtableaux for which the value of d_{PB} is maximum. Now, consider the lexicographic order on d_1 as usually defined: $\langle m, n\rangle < \langle m', n'\rangle$ iff $m < m'$ or $m = m'$ and $n < n'$. It can be shown that, for every tableau $\mathcal{P}$, there is a finite sequence of applications of transformation in Table 5.6 that progressively decreases the value of $d_1(\mathcal{P})$ until it yields a tableau $\mathcal{P}^*$ for which $d_1(\mathcal{P}^*) = \langle 0, 0\rangle$, i.e., a *PB*-canonical tableau.

Conjecture 1. *Every intelim proof of φ from X (intelim refutation of X) can be transformed into a PB-canonical intelim proof of φ from X (intelim refutation of X), by means of any sufficiently long sequence of applications of transformations in Table 5.6.*

Definition 5.5.4. Let the *depth* of a *PB*-canonical intelim proof of φ from X (intelim refutation of X) $\mathcal{P}$, denoted by $\mathsf{depth}(\mathcal{P})$, be defined as follows:

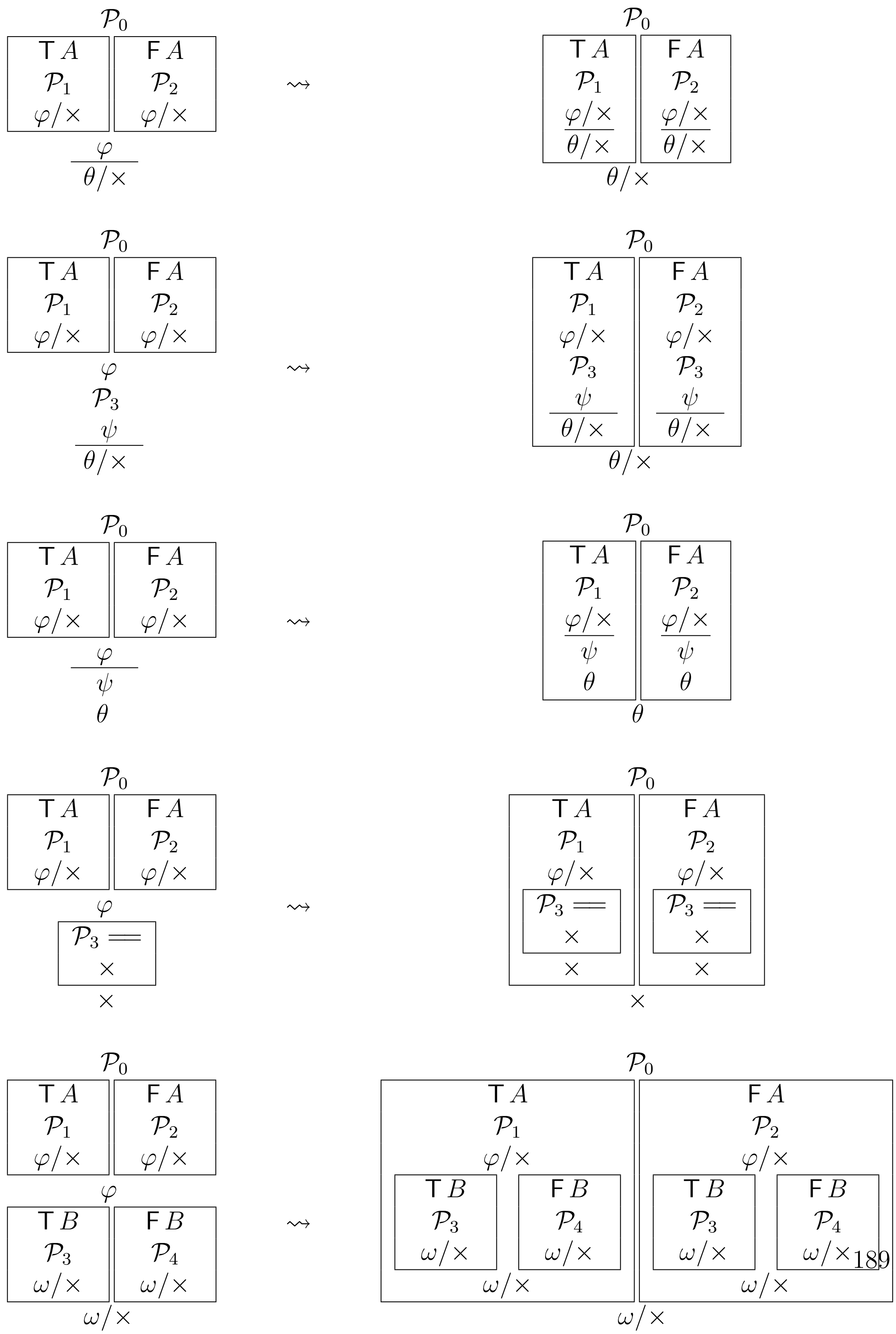
- If $\mathcal{P}$ contains no application of *PB*, then $\mathsf{depth}(\mathcal{P}) = 0$;
- if $\mathcal{P}$ has the form

$$\begin{array}{c} \mathcal{P}_0 \\ \begin{array}{|c|c|} \hline \mathsf{T}\,A & \mathsf{F}\,A \\ \mathcal{P}_1 & \mathcal{P}_2 \\ \varphi/\times & \varphi/\times \\ \hline \end{array} \\ \varphi/\times \end{array}$$

  then $\mathsf{depth}(\mathcal{P}) = \max\,(\mathsf{depth}(\mathcal{P}_1), \mathsf{depth}(\mathcal{P}_2)) + 1$.

Definition 5.5.5. A *0-depth component* of a *PB*-canonical intelim proof of φ from X (*PB*-canonical intelim refutation of X) $\mathcal{P}$ is any maximal 0-depth subtableau of $\mathcal{P}$, i.e., one that is not a proper subtableau of any 0-depth subtableau of $\mathcal{P}$.

Note that in a *PB*-canonical intelim proof (refutation) $\mathcal{P}$, the conclusion of every 0-depth component is the conclusion of $\mathcal{P}$ itself or $\times$. Each 0-depth component $\mathcal{P}_i$



Where: (i) *inside* the boxes on the right of $\rightsquigarrow$, if $\times$ is the case in $\varphi/\times$, nothing follows $\times$; (ii) in the last transformation, φ is used in the derivation of ω or $\times$.

Table 5.6: Transformations for *PB*-canonicity

$$
\begin{array}{c}
\mathcal{P}_0 \\
\begin{array}{|c|c|}
\hline
\mathsf{T}\,A & \mathsf{F}\,A \\
\mathcal{P}_1 & \mathcal{P}_2 \\
\varphi/\times & \varphi/\times \\
\hline
\end{array} \\
\varphi/\times
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{c}
\mathcal{P}_0 \\
\mathcal{P}_i \\
\varphi/\times
\end{array}
$$

Where $i = 1$ if $\mathsf{T}\,A$ is vacuously discharged in $\mathcal{P}_1$ or occurs as an undischarged assumption in $\mathcal{P}_2$; $i = 2$ if $\mathsf{F}\,A$ is vacuously discharged in $\mathcal{P}_2$ or occurs as an undischarged assumption in $\mathcal{P}_1$.

Table 5.7: Transformation for *PB*-non-redundancy

is a proof of φ from (a refutation of) $X_i \cup Y_i$ such that: (i) $X_i \subseteq X$; (ii) Y_i, with $|Y_i| \leq k$, is the set of virtual assumptions introduced in $\mathcal{P}_i$ that are subsequently discharged in $\mathcal{P}$ via applications of *PB*; (iii) $\mathcal{P}_i$ contains only applications of intelim rules and *PNC*. The node below the last node of each 0-depth component $\mathcal{P}_i$ is either an occurrence of the conclusion of $\mathcal{P}$ itself or $\times$, which results from applications of *PB* discharging the virtual assumptions in Y_i.

Definition 5.5.6. Given an intelim proof (refutation) $\mathcal{P}$, we say that an application of *PB* in $\mathcal{P}$ is *redundant* if one of the following conditions hold:

- at least one of its virtual assumptions is vacuously discharged, i.e., at least one of the corresponding virtual assumptions is not used;
- its conclusion still depends on one of the virtual assumptions.

Any intelim proof (refutation) can be turned into one that contains no redundant applications of *PB* by applying the transformation in Table 5.7. Note that the result of removing all redundant applications of *PB* from an intelim proof of φ from $X \cup Y$ (an intelim refutation of $X \cup Y$) is an intelim proof of φ from $Z \subseteq X \cup Y$ (an intelim refutation of $Z \subseteq X \cup Y$), where Y, with $|Y| \leq k$, is the set of virtual assumptions introduced in $\mathcal{P}$ that are subsequently discharged in $\mathcal{P}$ via applications of *PB*. Now, let $d_2(\mathcal{P})$ denote the number of redundant applications of *PB* in $\mathcal{P}$. Observe that each application of the transformation in Table 5.7 yields a tableau $\mathcal{P}'$ such that $d_2(\mathcal{P}') < d_2(\mathcal{P})$.

Conjecture 2. *Every intelim proof of φ from $X \cup Y$ (intelim refutation of $X \cup Y$) can be turned into an intelim proof of φ from $Z \subseteq X \cup Y$ (an intelim refutation of $Z \subseteq X \cup Y$) that contains no redundant applications of PB, by means of any* *sufficiently long sequence of applications of the transformation in Table 5.7 and with no increase in the size or depth of the proof (refutation).*

Remark 13. We emphasize that every application of the transformation in Table 5.7 which decreases $d_2(\mathcal{P})$ does not introduce any new application of *PB* and so cannot increase $d_1(\mathcal{P})$.

Definition 5.5.7. A *detour* in an intelim proof (refutation) $\mathcal{P}$ is an occurrence of an S-formula as conclusion of an introduction and, simultaneously, as major premise of an elimination.

The transformations in Tables 5.8 and 5.9 show how detours can be removed from an intelim proof (refutation); to save space we use the variable i ranging over $\{1, 2\}$. Observe that the transformations in Tables 5.8 and 5.9: (i) do not increase the size nor depth of the proof (refutation); (ii) in some cases, their application may introduce new detours, but these new detours are always of lower degree (i.e., have lower number of occurrences of connectives) than the one that is removed by the transformation. Thus, let $d_3(\mathcal{P})$ be the sum of the degrees of all detours occurring in $\mathcal{P}$ and equal to 0 when there are no detours. Hence, each application of the transformations in Tables 5.8 and 5.9 decreases the value of $d_3(\mathcal{P})$ until it eventually drops to 0, and so yielding a proof that does not have detours.

Conjecture 3. *Any intelim proof of φ from X (intelim refutation of X) can be transformed into a intelim proof of φ from X (intelim refutation of X) that contains no detours, by any sufficiently long sequence of applications of transformations in Tables 5.8 and 5.9 and with no increase in the size or depth of the proof (refutation).*

Remark 14. We emphasize that the transformations in Tables 5.8 and 5.9 do not introduce any new application of *PB*, and thus cannot increase either $d_1(\mathcal{P})$ or $d_2(\mathcal{P})$.

Remark 15. The notion of *detour* can be generalized by considering any sequence $(\varphi_1, ..., \varphi_n)$ of occurrences of the *same* S-formula such that: (i) φ_1 is the conclusion of an introduction; (ii) φ_n is the major premise of an elimination; (iii) for all i such that $1 < i \leq n$, φ_i is an immediate successor of φ_{i-1} resulting from an application of *PB*. We call such a sequence a *detour of level n*. An example of a detour of level 2 is shown in Figure 5.6. However, those higher level detours cannot occur in *PB*-canonical proofs (refutations), for φ_n would be at the same time the conclusion of an application of *PB* and a premise of an intelim rule or *PNC*. Then, the removal of higher level detours is a side-effect of transforming proofs (refutations) into *PB*-canonical ones. Another kind of “indirect” detour is displayed in Figure 5.7 and is related to the possibility of simulating introductions via *PB*, eliminations and *PNC*.

$$\begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A_i \\ \hline \mathsf{T}\,A_1 \vee A_2 \\ \mathcal{P}_2 \\ \mathsf{F}\,A_i \\ \hline \mathsf{T}\,A_{(3-i)} \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A_i \\ \mathcal{P}_2 \\ \mathsf{F}\,A_i \\ \times \end{array}$$

$$\begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A_i \\ \hline \mathsf{T}\,A_1 \vee A_2 \\ \mathcal{P}_2 \\ \mathsf{F}\,A_{(3-i)} \\ \hline \mathsf{T}\,A_i \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A_i \end{array}$$

$$\begin{array}{c} \mathcal{P}_1 \\ \mathsf{F}\,A_1 \\ \mathcal{P}_2 \\ \mathsf{F}\,A_2 \\ \hline \mathsf{F}\,A_1 \vee A_2 \\ \hline \mathsf{F}\,A_1 \\ \mathsf{F}\,A_2 \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \mathcal{P}_1 \\ \mathsf{F}\,A_1 \\ \mathcal{P}_2 \\ \mathsf{F}\,A_2 \end{array}$$

$$\begin{array}{c} \mathcal{P}_1 \\ \mathsf{F}\,A_i \\ \hline \mathsf{F}\,A_1 \wedge A_2 \\ \mathcal{P}_2 \\ \mathsf{T}\,A_i \\ \hline \mathsf{F}\,A_{(3-i)} \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \mathcal{P}_1 \\ \mathsf{F}\,A_i \\ \mathcal{P}_2 \\ \mathsf{T}\,A_i \\ \times \end{array}$$

Table 5.8: Transformations for removing detours 1

$$\begin{array}{c} \mathcal{P}_1 \\ \mathsf{F}\,A_i \\ \hline \mathsf{F}\,A_1 \wedge A_2 \\ \mathcal{P}_2 \\ \mathsf{T}\,A_{(3-i)} \\ \hline \mathsf{F}\,A_i \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \mathcal{P}_1 \\ \mathsf{F}\,A_i \end{array}$$

$$\begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A_1 \\ \mathcal{P}_2 \\ \mathsf{T}\,A_2 \\ \hline \mathsf{T}\,A_1 \wedge A_2 \\ \hline \mathsf{T}\,A_1 \\ \mathsf{T}\,A_2 \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A_1 \\ \mathcal{P}_2 \\ \mathsf{T}\,A_2 \end{array}$$

$$\begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A \\ \hline \mathsf{F}\,\neg A \\ \boxed{\begin{array}{c} \mathsf{T}\,A == \\ \mathcal{P}_2 \\ \times \end{array}} \\ \times \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A \\ \mathcal{P}_2 \\ \times \end{array}$$

$$\begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A_1 \\ \mathcal{P}_2 \\ \mathsf{F}\,A_2 \\ \hline \mathsf{F}\,A_1 \rightarrow A_2 \\ \boxed{\begin{array}{c} \mathsf{T}\,A_1 == \\ \mathsf{F}\,A_2 \\ \mathcal{P}_3 \\ \times \end{array}} \\ \times \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \mathcal{P}_1 \\ \mathsf{T}\,A_1 \\ \mathcal{P}_2 \\ \mathsf{F}\,A_2 \\ \mathcal{P}_3 \\ \times \end{array}$$

Table 5.9: Transformations for removing detours 2

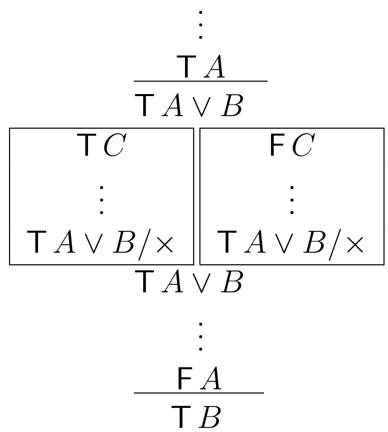


Figure 5.6: A detour of level 2

Again, the removal of this kind of indirect detours is a byproduct of transforming proofs (refutations) into *PB*-canonical ones.

Definition 5.5.8. Given an intelim proof (refutation) $\mathcal{P}$, an application of *PNC* is *canonical* in $\mathcal{P}$ if it is not the case that its premises are both conclusions of introductions. An intelim proof (refutation) is *PNC-canonical* if it contains no non-canonical applications of *PNC*.

Now, non-canonical applications of *PNC* can be removed by means of the transformations in Table 5.10 (for i ranging over $\{1, 2\}$). By the *degree of an application of PNC* with premises $\mathsf{T}\,A$ and $\mathsf{F}\,A$, we mean the degree of A. The removal of a non-canonical application of *PNC* may introduce a new non-canonical application, but the degree of the latter is always lower. Thus, let $d_4(\mathcal{P})$ be the sum of the degrees of the non-canonical applications of *PNC* in $\mathcal{P}$ and equal to 0 when all the applications of *PNC* are canonical. Each application of the transformations in Table 5.10 decreases $d_4(\mathcal{P})$ until its value drops to 0.

Conjecture 4. *Any intelim proof (refutation) can be transformed into a PNC-canonical one by means of any sufficiently long sequence of applications of transformations in Table 5.10 and with no increase in the size or depth of the proof (refutation).*

Remark 16. Observe that applications of transformations in Table 5.10 do not introduce new applications of *PB* nor new detours, and so cannot increase any of the parameters $d_1(\mathcal{P})$–$d_3(\mathcal{P})$.

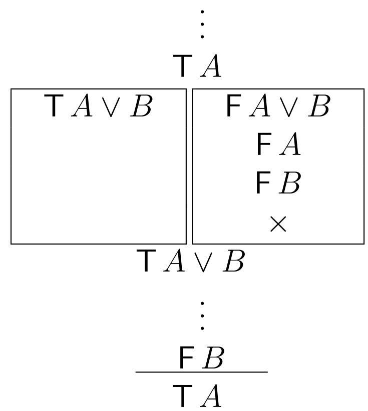


Figure 5.7: An indirect detour

$$
\begin{array}{c}
\mathcal{P}_0 \\
\underline{\mathsf{T}\,A_i} \\
\mathsf{T}\,A_1 \vee A_2 \\
\mathcal{P}_1 \\
\mathsf{F}\,A_1 \\
\mathcal{P}_2 \\
\underline{\mathsf{F}\,A_2} \\
\underline{\mathsf{F}\,A_1 \vee A_2} \\
\times
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{c}
\mathcal{P}_0 \\
\mathsf{T}\,A_i \\
\mathcal{P}_i \\
\underline{\mathsf{F}\,A_i} \\
\times
\end{array}
$$

$$
\begin{array}{c}
\mathcal{P}_0 \\
\underline{\mathsf{F}\,A_i} \\
\mathsf{F}\,A_1 \wedge A_2 \\
\mathcal{P}_1 \\
\mathsf{T}\,A_1 \\
\mathcal{P}_2 \\
\underline{\mathsf{T}\,A_2} \\
\underline{\mathsf{T}\,A_1 \wedge A_2} \\
\times
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{c}
\mathcal{P}_0 \\
\mathsf{F}\,A_i \\
\mathcal{P}_i \\
\underline{\mathsf{T}\,A_i} \\
\times
\end{array}
$$

Table 5.10: Transformations for PNC-canonicity

Definition 5.5.9. An intelim proof (refutation) $\mathcal{P}$ is *quasi-normal* if the following three conditions are satisfied:

- $\mathcal{P}$ is *PB*-canonical and *PNC*-canonical;
- $\mathcal{P}$ contains no redundant applications of *PB*;
- $\mathcal{P}$ contains no detours.

Now, let

$$d(\mathcal{P}) = \langle d_1(\mathcal{P}), d_2(\mathcal{P}), d_3(\mathcal{P}), d_4(\mathcal{P})\rangle$$

and consider the usual lexicographic order on $d(\mathcal{P})$ for every intelim proof (refutation) $\mathcal{P}$. It can be shown that a transformation that decreases $d_i(\mathcal{P})$ for some $i < 4$, may increase $d_j(\mathcal{P})$ for some $j > i$. However, as observed in Remarks 13-16, transformations that decrease $d_i(\mathcal{P})$ for $i > 1$, never increase d_j for any $j < i$. So, each of the transformations in Tables 5.6-5.10 decreases $d(\mathcal{P})$. Therefore, the repeated application of these transformations, regardless of their order, eventually yields a proof $\mathcal{P}'$ such that $d(\mathcal{P}') = \langle\langle 0, 0\rangle, 0, 0, 0\rangle$, which is thus quasi-normal.

Conjecture 5. *Any intelim proof (refutation) can be turned into a quasi-normal intelim proof (refutation), by means of any sufficiently long sequence of applications of the transformations in Tables 5.6-5.10.*

Further, observe that the transformations in Tables 5.7-5.10 that decrease $d_2(\mathcal{P})$ – $d_4(\mathcal{P})$ never increase the size of the proof (refutation) and do not introduce any new application of *PB*. Thus:

Conjecture 6. *Any PB-canonical intelim proof (refutation) can be turned into a quasi-normal intelim proof (refutation), by means of any sufficiently long sequence of applications of the transformations in Tables 5.7-5.10, with no increase in the size or depth of the proof (refutation).*

Remark 17. If $\mathcal{P}$ is a quasi-normal intelim proof (refutation), then every subtableau of $\mathcal{P}$ is also quasi-normal.

Quasi-normal proofs and refutations avoid trivially redundant applications of *PB*, intelim rules and *PNC*. Besides, in those proofs and refutations, the applications of *PB* are pushed down at the end of the corresponding tableau, so that their conclusion is always the conclusion of the whole proof or refutation. So, the role of applications of *PB* consists in gradually discharging the virtual assumptions made in what we called the 0-depth components. In quasi-normal proofs and refutations, *PB* is the only rule that may bring about violations of the subformula property. We now continue with the notion of *normal* proof (refutation), which is simply a quasi-normal proof (refutation) where *PB* is applied only to subformulae either of the *initial* assumptions or of the conclusion. Thus, normal proofs and refutations enjoy the subformula property.

Remark 18. Observe that:

a) The relation "A is a proper subformula of B" is transitive;

b) the unsigned part of the minor premise of an elimination is always a proper subformula of the unsigned part of its major premise;

c) the unsigned part of the conclusion of an elimination is always a proper subformula of the unsigned part of its major premise;

d) the unsigned part of a premise of an introduction is always a proper subformula of the unsigned part of its conclusion.

Definitions 5.5.10. Given an intelim proof of φ from X (an intelim refutation of X) $\mathcal{P}$, we say that an application of *PB* in $\mathcal{P}$ is *analytic* if its *PB*-formula is in $\mathsf{sub}(X^u \cup \{\varphi^u\})$ $(\mathsf{sub}(X^u))$. An application of *PB* in $\mathcal{P}$ is *atomic* if its *PB*-formula is atomic, i.e., the virtual assumptions discharged by it have the form $\mathsf{T}\,p$ and $\mathsf{F}\,p$ for some atomic p.

Definition 5.5.11. An intelim proof (refutation) $\mathcal{P}$ is *normal* if it is quasi-normal and every application of *PB* in $\mathcal{P}$ is analytic.

Note that a quasi-normal 0-depth intelim proof (refutation), i.e., one that contains no applications of *PB*, is normal by definition.

Remark 19. If $\mathcal{P}$ is a normal intelim proof (refutation), every subtableau of $\mathcal{P}$ is also normal.

Conjecture 7. *If $\mathcal{P}$ is a normal 0-depth proof of φ from X (normal 0-depth refutation of X), and ψ is a S-formula occurring in $\mathcal{P}$, then either $\psi^u \in X^u \cup \{\varphi^u\}$ (X^u) or ψ^u is a proper subformula of some formula in $X^u \cup \{\varphi^u\}$ (X^u), i.e., $\psi^u \in \mathsf{sub}(X^u \cup \{\varphi^u\})$ $(\psi^u \in \mathsf{sub}(X^u))$.*

Proof sketch. Suppose that $\mathcal{P}$ is a normal 0-depth intelim proof of φ from X (normal 0-depth refutation of X). Let Y be the set of all S-formulae θ occurring in $\mathcal{P}$ such that (i) $\theta^u \notin X^u \cup \{\varphi^u\}$ $(\theta^u \notin X^u)$, and (ii) θ^u is not a proper subformula of any formula in $X^u \cup \{\varphi^u\}$ (in X^u). Let us assume that $Y \neq \emptyset$ and take a formula $\omega \in Y$ such that ω^u is of maximum degree in Y^u. Given that $\omega^u \notin X^u$, ω occurs in $\mathcal{P}$ as conclusion of an application of an intelim rule.

Now, ω cannot be a conclusion of an elimination. To see this, note that the unsigned part of the major premise of this elimination cannot be in $X^u \cup \{\varphi^u\}$ (X^u), otherwise, by Remark 18-c), ω^u would be a proper subformula of a formula in $X^u \cup \{\varphi^u\}$ (X^u) and thus ω would not belong to Y. Moreover, the unsigned part of this major premise cannot be a proper subformula of a formula in $X^u \cup \{\varphi^u\}$ (X^u), for in this case, by Remark 18-a) and Remark 18-c), ω^u would also be a proper subformula of some formula in $X^u \cup \{\varphi^u\}$ (X^u) and so ω would not belong to Y. Therefore, the major premise of the elimination should be an S-formula in Y such that its unsigned part is of greater degree than ω^u, against the assumption that ω^u is of maximum degree in Y^u.

Hence, ω can only be the conclusion of an introduction. Given that $\omega \neq \varphi$, ω must be used in $\mathcal{P}$ as premise of some intelim rule or of *PNC*. Now, ω cannot be used as major premise of an elimination rule, otherwise ω would be a detour, against the assumption that $\mathcal{P}$ is normal and so contains no detours. Besides, ω cannot be used as premise of *PNC*, because in this case $\bar{\omega}$ could also be only the conclusion of an introduction, for the same reasons as ω; but this is impossible because $\mathcal{P}$ is normal and so *PNC*-canonical (it is not the case that the premises of an application of *PNC* are both conclusions of an introduction). Moreover, ω cannot be used as minor premise of an elimination, otherwise, by Remark 18-b), ω^u would be a proper subformula of the unsigned part of the major premise. So, either the unsigned part of this major premise belongs to $X^u \cup \{\varphi^u\}$ (X^u) and then, by Remark 18-a), ω^u would be a proper subformula of some formula in $X^u \cup \{\varphi^u\}$ (X^u), in which case ω would not belong to Y; or the major premise of this elimination would be an S-formula in Y whose unsigned part is of greater degree than ω^u, against the assumption that ω^u is of maximum degree in Y^u.

Thus, ω must be used as premise of an introduction. But this is impossible since, by Remark 18-d), ω^u would be a proper subformula of the unsigned part of the conclusion of this introduction. So, either (i) the unsigned part of this conclusion belongs to $X^u \cup \{\varphi^u\}$ (X^u) and, by Remark Remark 18-a), ω^u would be a proper subformula of some formula in $X^u \cup \{\varphi^u\}$ (X^u), in which case ω would not belong to Y, or (ii) the conclusion of this introduction would be a S-formula in Y whose unsigned part is of greater degree than ω^u, against the assumption that ω^u is of maximum degree in Y^u. Therefore, Y must be empty. □

It would follow immediately from the above Conjecture that:

Conjecture 8 (SFP of 0-depth proofs and refutations). *Every normal 0-depth intelim proof (every normal 0-depth intelim refutation) has the subformula property.*

Remark 20. Note that if $\mathcal{P}$ is a quasi-normal proof of φ from X (a quasi-normal refutation of X), whose 0-depth components are $\mathcal{P}_1, ..., \mathcal{P}_n$, every 0-depth component $\mathcal{P}_i$ is a normal proof of φ from $X_i \cup Z_i$ (a normal refutation of $X_i \cup Z_i$), where $X_i \subseteq X$ and Z_i are virtual assumptions that are subsequently discharged in $\mathcal{P}$ by applications of *PB*.

Conjecture 9 (SFP of normal proofs and refutations). *If $\mathcal{P}$ is a normal intelim proof of φ from X (normal intelim refutation of X), then $\mathcal{P}$ has the subformula property.*

Proof sketch. Let $\mathcal{P}_1, ..., \mathcal{P}_n$ be the 0-depth components of $\mathcal{P}$. By Remark 20, every 0-depth component of $\mathcal{P}$ is normal. Recall that in a normal proof (refutation), every S-formula occurring in $\mathcal{P}$ occurs also in some of its 0-depth components, since all the conclusions of applications of *PB* are equal to the conclusion of all 0-depth components. Then, for every S-formula ψ occurring in $\mathcal{P}$, there is a 0-depth component $\mathcal{P}_i$ of $\mathcal{P}$ such that, by Conjecture 7, either $\psi^u \in X_i^u \cup Z_i^u \cup \{\varphi^u\}$ ($\psi^u \in X_i^u \cup Z_i^u$), or ψ^u is a proper subformula of some formula in $X_i^u \cup Z_i^u \cup \{\varphi^u\}$ ($\psi^u \in X_i^u \cup Z_i^u$), where X_i are the assumptions of $\mathcal{P}_i$ that are left undischarged in $\mathcal{P}$ and Z_i are the virtual assumptions subsequently discharged in $\mathcal{P}$. If $\mathcal{P}$ is normal, for every S-formula $\theta \in Z_i$, θ^u is a subformula of a formula in $X_i^u \cup \{\varphi^u\}$ (X_i^u). So, either (i) $\psi^u \in X_i^u \cup Z_i^u \cup \{\varphi^u\}$ ($\psi^u \in X_i^u \cup Z_i^u$) and so ψ^u is a subformula of a formula in $X_i^u \cup \{\varphi^u\}$ (X_i^u), or (ii) ψ^u is a proper subformula of some formula in $X_i^u \cup Z_i^u \cup \{\varphi^u\}$) ($X_i^u \cup Z_i^u$). Since for every $\theta \in Z_i$, θ^u is a subformula of some formula in $X_i^u \cup \{\varphi^u\}$, it is not difficult to verify, that if ψ^u is a proper subformula of ω^u and ω^u is a subformula of χ^u, then ψ^u is a subformula of χ^u. Therefore, ψ^u must be a subformula of some formula in $X_i^u \cup \{\varphi^u\}$ (X_i^u). □

Conjecture 10. *If $\mathcal{P}$ is a quasi-normal intelim proof of φ from X (quasi-normal intelim refutation of X) and all the non-atomic applications of PB in $\mathcal{P}$ are analytic, then all the atomic applications of PB in $\mathcal{P}$ are also analytic, i.e., $\mathcal{P}$ is normal.*

Proof sketch. Let $\mathcal{P}$ is a quasi-normal intelim proof of φ from X (quasi-normal intelim refutation of X) such that all the non-atomic applications of *PB* in $\mathcal{P}$ are analytic. We show that $\mathcal{P}$ cannot contain any non-analytic atomic application of *PB* and, so, $\mathcal{P}$ is normal. To this end we prove, by induction on k, that every k-depth subtableau of $\mathcal{P}$ is normal.

By Remark 20, every 0-depth subtableau of $\mathcal{P}$ is normal. For $k > 0$, suppose that every subtableau of $\mathcal{P}$ of depth $k-1$ is normal. We show that under this supposition every k-depth subtableau is also normal. Since $k > 0$, any k-depth subtableau either ends with a non-atomic application of *PB* (which is by hypothesis analytic, so that the k- depth subtableau is normal) or has the following form, for some atomic p:

$$\begin{array}{c} \mathcal{P}_0 \\ \begin{array}{|c|c|} \hline \mathsf{T}\,p & \mathsf{F}\,p \\ \mathcal{P}_1 & \mathcal{P}_2 \\ \varphi/\times & \varphi/\times \\ \hline \end{array} \\ \varphi/\times \end{array}$$

Suppose, ex absurdo, that this atomic application of *PB* is non-analytic, i.e. that p does not occur in $X^u \cup \{\varphi^u\}$. By inductive hypothesis, we know that for some $X_1, X_2 \subseteq X$:

- $\mathcal{P}_1$ is a normal proof of φ from $X_1 \cup Y_1 \cup \{\mathsf{T}\,p\}$ (normal intelim refutation of $X_1 \cup Y_1 \cup \{\mathsf{T}\,p\}$),
- $\mathcal{P}_2$ is a normal proof of φ from $X_2 \cup Y_2 \cup \{\mathsf{F}\,p\}$ (normal intelim refutation of $X_2 \cup Y_2 \cup \{\mathsf{F}\,p\}$),

where Y_1 and Y_2 are the sets of virtual assumptions that are still undischarged in $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively.

Since $\mathcal{P}$ is quasi normal, it contains no redundant applications of *PB*, and so neither $\mathsf{T}\,p$ nor $\mathsf{F}\,p$ are vacuously discharged. Thus, $\mathsf{T}\,p$ must be used as premise of some application of an intelim rule or *PNC* in $\mathcal{P}_1$, and $\mathsf{F}\,p$ as premise of some application of an intelim rule or *PNC* in $\mathcal{P}_2$. Given the logical form of p, $\mathsf{T}\,p$ cannot be used in $\mathcal{P}_1$ as major premise of an elimination. In turn, if $\mathsf{T}\,p$ is used as minor premise of an elimination, p would occur in the unsigned part of the major premise and the latter would not be a subformula of some formula in $X_1^u \cup Y_1^u \cup \{p\} \cup \{\varphi^u\}$; this because every formula in Y_1^u is either an atomic formula, or the negation of an atomic formula, or a subformula of some formula in $X_1^u \cup \{\varphi^u\}$. But this is impossible, since, by inductive hypothesis, $\mathcal{P}_1$ is normal and, by Conjecture 9, it has the subformula property.

Moreover, $\mathsf{T}\,p$ cannot be used as premise of an introduction, for the unsigned part of the conclusion of this introduction would not be a subformula of any of the formulae in $X_1^u \cup Y_1^u \cup \{p\} \cup \{\varphi^u\}$, which would again contradict the hypothesis that $\mathcal{P}_1$ is normal and has the subformula property. So, $\mathsf{T}\,p$ must be used in $\mathcal{P}_1$ as premise of an application of *PNC*. But, in such a case, the other premise $\mathsf{F}\,p$ cannot result from the application of an elimination, otherwise $\mathcal{P}_1$ would not have the subformula property, against the hypothesis that it is normal. Thus, $\mathsf{F}\,p$ should belong to Y_1 and

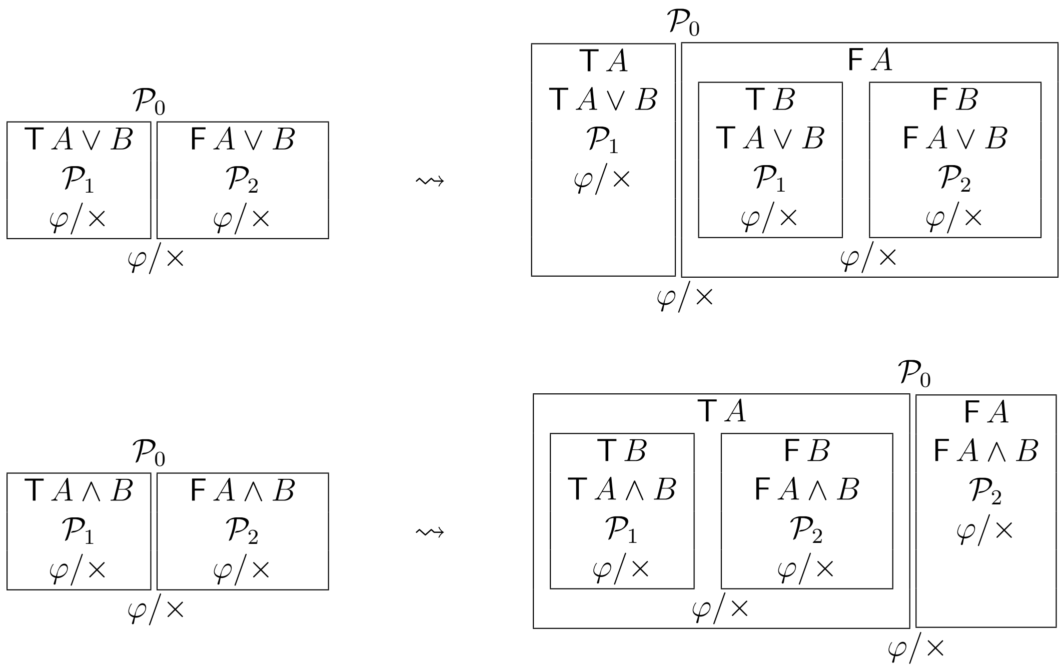


Table 5.11: Transformations for normality (the transformations for $\to$ and $\neg$ are missing)

the corresponding application of PB would be redundant, against the hypothesis that $\mathcal{P}$ is quasi normal, which implies that it contains no redundant applications of PB. Hence, it is impossible that p is an atomic formula that does not occur in $X^u \cup \{\varphi^u\}$. Therefore, all applications of PB in $\mathcal{P}$ are analytic and $\mathcal{P}$ is normal. $\square$

Any intelim proof (refutation) can be turned into a normal one by applying the transformations in Table 5.11. In turn, for every intelim tableau $\mathcal{P}$, let $g(\mathcal{P})$ be defined as follows:

$$g(\mathcal{P}) = \begin{cases} \sharp(A) & \text{if } \mathcal{P} \text{ ends with a non-analytic application of } PB \text{ and } A \text{ is the} \\ & PB\text{-formula of this application} \\ 0 & \text{otherwise} \end{cases}$$

where $\sharp(A)$ denotes the degree of A.

Observe that, in general, the transformations in Table 5.11 increase the size of the proof (refutation). Besides, they may introduce new detours; for example in the first transformation in the Table, it may be the case that $\mathsf{T}\, A \vee B$ or $\mathsf{F}\, A \vee B$ or both are respectively used in $\mathcal{P}_1$ or $\mathcal{P}_2$ as major premises of eliminations. Moreover, the transformations may also introduce new non-canonical applications of *PNC*.

Now, let $d_0(\mathcal{P}) = \langle m, n \rangle$, where m is the maximum value taken by g for a subtableau of $\mathcal{P}$, and n is the number of subtableaux for which the value of g is maximum. Consider again the usual lexicographic order on d_0. If the transformations in Table 5.11 are applied to arbitrary subtableaux, the index d_0 may not always decrease. However, it never increases, and it can be shown that eventually it reaches the minimum value $\langle 0, 0 \rangle$ in a finite number of steps. Thus, there is a finite sequence of applications of transformations in Table 5.11, independently to which subtableaux they are applied, that yields a tableau in which all the applications of *PB* are either analytic or atomic.

Conjecture 11. *Any intelim proof of φ from $X \cup Y$ (intelim refutation of $X \cup Y$) can be transformed into an intelim proof of φ from $Z \subseteq X \cup Y$ (intelim refutation of $Z \subseteq X \cup Y$) where all the applications of PB are either analytic or atomic by means of any sufficiently long sequence of applications of transformations in Table 5.11.*

Now, let

$$d(\mathcal{P}) = \langle d_0(\mathcal{P}), d_1(\mathcal{P}), d_2(\mathcal{P}), d_3(\mathcal{P}), d_4(\mathcal{P}) \rangle$$

and consider the usual lexicographic order on $d(\mathcal{P})$ for every intelim proof (refutation) $\mathcal{P}$. By inspection of the transformations in Tables 5.6-5.11, it can be verified that each transformation that decreases $d_i(\mathcal{P})$ for any $i < 4$, may increase $d_j(\mathcal{P})$ for some $j > i$. However, no transformation that decreases $d_i(\mathcal{P})$ for $i > 0$, never increases d_j for any $j < i$. So, each of the transformations in Tables 5.6-5.11 decreases $d(\mathcal{P})$. Therefore, the repeated application of these transformations, regardless of their order, eventually yields a proof $\mathcal{P}'$ such that $d(\mathcal{P}') = \langle \langle 0, 0 \rangle, \langle 0, 0 \rangle, 0, 0, 0 \rangle$. Note that such a proof (refutation) is quasi-normal, for the value of $d_1(\mathcal{P}')$ is equal to $\langle 0, 0 \rangle$ and the values of $d_2(\mathcal{P}')$-$d_4(\mathcal{P}')$ are all equal to 0. Moreover, all the applications of *PB* in $\mathcal{P}'$ are either analytic or atomic. Then, by Conjecture 10, the proof (refutation) is normal.

Conjecture 12. *Any intelim proof from φ of $X \cup Y$ (intelim refutation of $X \cup Y$) can be transformed into a normal intelim proof of φ from $Z \subseteq X \cup Y$ (intelim refutation of $Z \subseteq X \cup Y$) by means of any sufficiently long sequence of applications of transformations in Tables 5.6-5.11.*

5.6 Depth-bounded approximations to IPL

Definitions 5.6.1. The *depth* of an intelim tableau $\mathcal{P}$ is the maximum number of nested boxes occurring in $\mathcal{P}$ which are associated with applications of *PB*. An intelim tableau $\mathcal{P}$ is a *k-depth intelim proof of φ from X* (a *k-depth intelim refutation of X*) if $\mathcal{P}$ is an intelim proof of φ from X (an intelim refutation of X) and $\mathcal{P}$ is of depth k.[15]

In Figure 5.4, both examples are 1-depth proofs. In Figure 5.5, both examples are intelim sequences and, so, they respectively are a 0-depth proof and a 0-depth refutation. Now, we are in a position to introduce the following definitions:

Definitions 5.6.2. For all X, φ,

- φ is *0-depth deducible* from X, $X \vdash_0 \varphi$, iff there is a 0-depth intelim proof of φ from X;
- X is *0-depth refutable*, $X \vdash_0$, iff there is a 0-depth intelim refutation of X.

Notation 5.6.3. We shall abuse of the same relation symbol '$\vdash_0$' to denote 0-depth deducibility and refutability.

Proposition 5.6.4. *$\langle \mathcal{L}, \vdash_0 \rangle$ is a (finitary) Tarskian propositional logic; i.e., $\vdash_0$ satisfies reflexivity, monotonicity, cut, and structurality.*

Proof. The proposition follows easily from the definitions involved. For example, to see that $\vdash_0$ satisfies cut, suppose that there is a 0-depth intelim proof of φ from X and that there is a 0-depth intelim proof of ψ from $X \cup \{\varphi\}$. Then, clearly, there is a 0-depth intelim proof of ψ from X. □

Furthermore, note that $\vdash_0$ has no tautologies; i.e., for no A, it holds that $\emptyset \vdash_0 A$. This is in tune with the informational tenet of the depth bounded-approach according to which there is no way of extracting information from the empty set of assumptions without introducing virtual information. Accordingly, tautologies make their appearance only at depths $k > 0$, when the use of virtual information is allowed, and the set of provable tautologies increases with k. Interestingly, according to our definitions, $\vdash_0$ is *not* explosive. In a sort of duality with tautologyhood, explosivity appears only at depths $k > 0$. Relatedly, k-depth refutability, $k \geq 0$, is stricter than intuitionisitc refutability in that a set X may well be k-depth non-refutable but

[15] Note that the transformations in Tables 5.6 and 5.11 may increase the depth of a tableau $\mathcal{P}$, so that it is convenient to use them only to the extent they are needed to normalize $\mathcal{P}$.

intuitionistically refutable. More importantly, as stated in the following Subsection, we conjecture that 0-depth refutability can be feasibly detected.

Now, the notion of k-depth deducibility depends not only on the depth at which the use of virtual information is recursively allowed, but also on the virtual space discussed and defined above. Thus, finally:

Definitions 5.6.5. For all X, ψ, all $f \in \mathcal{F}$,

- $X \vdash_0^f \psi$ iff $X \vdash_0 \psi$;
- for $k > 0$, $X \vdash_k^f \psi$ iff there is a tableau for X in which there is a finite sequence $(\varphi_1, ..., \varphi_n)$ such that $\varphi_n = \psi$ and, for every φ_i, $1 \leq i \leq n$,

  $\{\varphi_1, ..., \varphi_{i-1}\} \cup \{\mathsf{T}\, B\} \vdash_{k-1}^f \varphi_i$ and $\{\varphi_1, ..., \varphi_{i-1}\} \cup \{\mathsf{F}\, B\} \vdash_{k-1}^f \varphi_i$ for some $B \in f(X^u \cup \{\psi^u\})$.

When $X \vdash_k^f \psi$, we say that ψ is *deducible at depth k from X over the f-bounded virtual space.*

In turn, we shall denote the case of k-depth refutability by $X \vdash_k^f$, assuming it as equivalent to $X \vdash_k^f \psi$ for all ψ, and defined as follows:

Definitions 5.6.6. For all X, all $f \in \mathcal{F}$,

- $X \vdash_0^f$ iff $X \vdash_0$;
- for $k > 0$, $X \vdash_k^f$ iff there is a tableau for X in which there is a finite sequence $(\varphi_1, ..., \varphi_n)$ such that for every φ_i, $1 \leq i \leq n$, $X \vdash_{k-1}^f \varphi_i$ and

  $\{\varphi_1, ..., \varphi_n\} \cup \{\mathsf{T}\, B\} \vdash_{k-1}^f$ and $\{\varphi_1, ..., \varphi_n\} \cup \{\mathsf{F}\, B\} \vdash_{k-1}^f$ for some $B \in f(X^u)$.

When $X \vdash_k^f$, we say that X is *refutable at depth k over the f-bounded virtual space.*

Notation 5.6.7. We shall abuse of the same relation symbol '$\vdash_k^f$' to denote k-depth deducibility and refutability over the f-bounded virtual space.

Now, it follows immediately from Def. 5.6.1, 5.6.5, and 5.6.6 that:

Proposition 5.6.8. *For all X, ψ and all $f \in \mathcal{F}$, $X \vdash_k^f \psi$ ($X \vdash_k^f$) iff there is a k-depth intelim proof of ψ from X (a k-depth intelim refutation of X) such that all its PB-formulae are in $f(X^u \cup \{\psi^u\})$ ($f(X^u)$).*

Now, it is easy to verify that the relations $\vdash_k^f$ satisfy reflexivity and monotonicity, but they may or may not satisfy cut depending on the function f that defines the virtual space. A sufficient condition obtains when $f(\Delta)$ is the set of all formulae of given bounded degree that can be built out of $\mathsf{at}(\Delta)$. Nonetheless, the relations $\vdash_k^f$ always satisfy the following version of cut:

Bounded cut: If $X \vdash_k^f \varphi$ and $X \cup \{\varphi\} \vdash_k^f \psi$, then $X \vdash_k^f \psi$.

When $f = \mathsf{sub}$ we call the latter *analytic cut*. Moreover, the relations $\vdash_k^f$ may not be structural in that structurality depends also on the function f that defines the virtual space. For example, $\vdash_k^{\mathsf{sub}}$ is structural, while $\vdash_k^{\mathsf{at}}$ is not. As in the classical and many-valued cases, in general structurality can be imposed by restricting the operations in $\mathcal{F}$ to those such that, for all σ and all Δ, $\sigma(f(\Delta)) \subseteq f(\sigma(\Delta))$. This is not satisfied if $f = \mathsf{at}$, but it is satisfied if $f(\Delta) = \mathsf{sub}(\Delta)$, or $f(\Delta)$ is the set of all formulae of given bounded degree that can be built out of $\mathsf{sub}(\Delta)$. Further, since $\vdash_0$ is monotonic, its successors are ordered: $\vdash_j^f \subseteq \vdash_k^f$ whenever $j \leq k$. The transition from $\vdash_k^f$ to $\vdash_{k+1}^f$ corresponds to an increase in the depth at which the nested use of virtual information (restricted to formulae in the virtual space defined by f) is allowed. Note also that $\vdash_j^{f_1} \subseteq \vdash_k^{f_2}$ whenever $f_1 \trianglelefteq f_2$.

5.6.1 Tractability

We conjecture that the decision problem for the k-depth logics is tractable. Analogously to the cases addressed in previous Chapters, that our intelim method enjoys the subformula property (see Remark 12) immediately suggests a decision procedure for k-depth deducibility: to establish whether φ is k-depth deducible from a finite set X we apply the intelim rules, together with PB up to a number k of times, in all possible ways starting from X and restricting to applications which preserve the subformula property. If the resulting intelim tableau is closed or φ occurs in its last line, then φ is k-depth deducible from X, otherwise it is not.

Conjecture 13. *Whether or not $X \vdash_k^f \varphi$ ($X \vdash_k^f$), $k \geq 0$, can be decided in polynomial time.*

5.7 3-valued non-deterministic semantics for full IPL

In Kripke and Beth semantics, the meanings of $\rightarrow$ and $\neg$ depend on "structural" constraints in the sense that they essentially appeal to an "accessibility" relation.[16] Following the key idea of the depth-bounded approach according to which the meaning of a connective is specified solely in terms of information that the agent holds, in our proof-theoretic characterization of the hierarchy of depth-bounded approximations, the unique *structural* rule which introduces virtual information (PB) has no role in specifying the meaning of the connectives. In such a characterization, the intelim rules fix the meaning of the connectives without appealing to any structural condition.[17] Accordingly, a natural semantical characterization of the hierarchy should be one under which the meanings of all the connectives are independent of structural constraints. As a first step towards that characterization, in this section we shall introduce an alternative semantics for full **IPL** where the meaning of all the connectives is completely specified by a non-deterministic matrix (Nmatrix) and the notion of *model* so defined is what must satisfy some structural constraints.[18]

The primary notions of our alternative semantics are intuitionistic *truth*, *falsity* and *indeterminacy*. That is, *holding the information* that a formula is, respectively, proven, refuted or undecided. In turn, we use the truth-values 1, 0 and u to respectively denote those notions. Based on the idea that the meaning of a connective is specified solely in terms of the information that is held by an agent, our semantics is intended to model *actual* information states, as opposed to *virtual* or *potential* ones; the latter being those considered when evaluating the truth of some complex formulae in both Kripke and Beth semantics. Put differently, our semantics is aimed to model the *readiness* of agents to answer questions on the basis of information they hold. Accordingly, that the value of a formula A is 1 means that the agent holds that there is a proof of A; that the value of A is 0 means that the agent holds that there is a refutation of A; and that the value of A is u means that the agent holds that there is no proof nor refutation of A—the three cases, under the current *actual* information state, as we explain and define below.

[16] In the case of Beth semantics also the meaning of '$\vee$' depends on structural constraints.

[17] Recall that, in our intelim method, the introduction rules for the truth of formulae whose main connective is $\rightarrow$ and $\neg$ are derived rules via PB.

[18] Avron [e.g., 16] has already explored the use of non-deterministic Kripke frames for **IPL**. However, he did so by embedding Nmatrices in intuitionistic Kripke models. Instead, here we opt to offer a structural non-deterministic semantics for that logic which does not rely on the full Kripke-style apparatus.

$\tilde{\vee}$	1	0	u
1	$\{1\}$	$\{1\}$	$\{1\}$
0	$\{1\}$	$\{0\}$	$\{u\}$
u	$\{1\}$	$\{u\}$	$\{u\}$

$\tilde{\wedge}$	1	0	u
1	$\{1\}$	$\{0\}$	$\{u\}$
0	$\{0\}$	$\{0\}$	$\{0\}$
u	$\{u\}$	$\{0\}$	$\{0, u\}$

$\tilde{\rightarrow}$	1	0	u
1	$\{1\}$	$\{0\}$	$\{u\}$
0	$\{1\}$	$\{1\}$	$\{1\}$
u	$\{1\}$	$\{0, u\}$	$\{1, u\}$

$\tilde{\neg}$	
1	$\{0\}$
0	$\{1\}$
u	$\{0, u\}$

Table 5.12: $3\mathrm{N}_I$-tables

The truth-values 1, 0 and u are all *defined* and partially ordered by the relation $\preceq_3$. As in our informational approach to the classical case, this relation is intended as an information order and such that $x \preceq_3 y$ (read "x is less defined than, or equal to, y") iff $x = u$ or $x = y$ for $x, y \in \{1, 0, u\}$. Thus, a 3_I*-valuation* v for $\mathcal{L}$ is a function $v : F(\mathcal{L}) \longrightarrow \{1, 0, u\}$. Now, we pick out from the set of all 3_I-valuations those which agree with the intended meaning of the connectives. We do this through the following Nmatrix, which conservatively extends the standard matrix of **CPL**:

Definition 5.7.1. Let $\mathcal{M}_I$ be the Nmatrix for $\mathcal{L}$ where $\mathcal{V} = \{1, 0, u\}$, $\mathcal{D} = \{1\}$ and the functions in $\mathcal{O}$ are defined by the $3\mathrm{N}_I$-tables in Table 5.12.

Remark 21. We shall show below that **IPL** is sound and complete with respect to the semantics induced by the $3\mathrm{N}_I$-tables together with structural constraints imposed on all corresponding refinements. To begin with, we explain the method to obtain the $3\mathrm{N}_I$-tables, which is completely analogous to the method applied to obtain the 3N-tables in the classical case (Table 3.7 and Remark 5): we start by pointing out that, in this case, u plays the role of a sort of "undefined" truth-value.[19] Accordingly, the part of each $3\mathrm{N}_I$-table which involves solely 1 and 0 plays the role of the corresponding "defined kernel" of the table. Each of those "defined kernels" is respectively identical with a standard truth-table of a classical connective—where the meaning of the truth-values is different, of course. As it is well-known, those "defined kernels" agree with the intuitionistic meaning of the connectives in that they can be mimicked by,

[19] As explained below, under our conceptual framework, u is not a (genuine) undefined truth-value. Actually, as expected, the undefined truth-value for **IPL** would appear just until the corresponding 0-depth logic is introduced, and would denote (full) *ignorance* about defined truth-values.

for example, Gentzen-Prawitz style natural deduction [see, 143].[20] Regarding the part of each $3N_I$-table involving u, we follow exactly the same method applied for getting the 3N-tables, which was outlined in Remark 5: the entries involving u are established by checking its compatibility with the "defined kernel". Now, as before, a truth-value is *compatible* with the corresponding "defined kernel" if, on the basis of the very "defined kernel", such a truth-value does not imply that one particular argument of those which are u had a defined truth-value "instead", i.e., 1 or 0. Let us take as an example the $3N_I$-table for $\rightarrow$: if the antecedent is 1 and the consequent is u, then the conditional can only be u; it cannot be 1 nor 0 because that would respectively imply that the consequent was already 1 or 0. Now, when the antecedent is 0, the conditional can only be 1 regardless of the truth-value of the consequent. Analogously, if the consequent is 1, the conditional can only be 1 regardless of the truth-value of the antecedent. In turn, if the antecedent is u and the consequent is 0, the unique excluded truth-value for the conditional is 1 since it would imply that the antecedent was already 0. By contrast, the conditional may well be either u or 0, depending on whether or not the agent holds the additional information that antecedent and consequent cannot be both 0, i.e., in case one is the negation of the other. Similarly, when both antecedent and consequent are u, the only excluded truth-value for the conditional is 0 since it would imply that the consequent was already 0. However, the conditional may well be either u or 1, depending on whether or not the agent holds the additional information that antecedent and consequent can be either both 1 or 0, i.e., in case one is not the negation of the other.

The entries involving u in the rest $3N_I$-tables can be explained in similar terms. Furthermore, *a fortiori*, all those entries can be analogously be explained in terms of Gentzen-Prawitz style natural deduction. To give an example, the first case of the $3N_I$-table for $\rightarrow$ explained in the previous paragraph, can be analogously explained as in Figure 5.8. (Where the "falsum" logical constant $\curlywedge$ is assumed to be part of $\mathcal{L}$ and intended as an absurd proposition. Besides, the numerals are used to keep track of the temporary assumptions that are discharged by the application of a rule. The numerals corresponding to the discharged assumptions are shown beside the inference line.)

Definition 5.7.2. A *$3N_I$-valuation* is a 3_I-valuation v satisfying the following conditions for all $A, B \in F(\mathcal{L})$:

1. $v(\neg A) \in \tilde{\neg}(v(A))$;

[20] This simulation requires replacing each occurrence of 1 by the formula to which it is assigned, and each occurrence of 0 by the negation of the formula to which it is assigned.

$$\frac{A \quad [A \to B]^1}{B} \qquad\qquad \cfrac{[\neg(A \to B)]^1 \quad \cfrac{[B]^2}{A \to B}}{\cfrac{\curlywedge}{\neg B}\ {}^2}$$

Figure 5.8: Explaining a non-deterministic truth-table table entry

2. $v(A \circ B) \in \tilde{\circ}(v(A), v(B))$.

Where $\circ$ is $\vee$, $\wedge$ or $\to$.

We take $3\mathrm{N}_I$-valuations as partially ordered by the following relation:

Definition 5.7.3. Let v, w be $3\mathrm{N}_I$-valuations. Then, w is a *3-refinement* of v, $v \sqsubseteq_3 w$, iff $v(A) \preceq_3 w(A)$ for all A.

Definition 5.7.4. Given a $3\mathrm{N}_I$-valuation v, let $\hat{v}$ be a partial function defined as follows:

$$\hat{v}(\neg A) = \begin{cases} x & \text{if } \tilde{\neg}(v(A)) = \{x\} \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\hat{v}(A \circ B) = \begin{cases} x & \text{if } \tilde{\circ}(v(A), v(B)) = \{x\} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Where $\circ$ is $\vee$, $\wedge$ or $\to$. We call $\hat{v}$ the *deterministic restriction* of v.

Definition 5.7.5. Let $\mathfrak{F} = \langle S, R\rangle$ be an intuitionistic Kripke frame. A *$3N_I$-model* is a pair $M_3 = \langle \mathfrak{F}, \{v_a\}_{a \in S}\rangle$, where for all $a, b \in S$, aRb iff $v_a \sqsubseteq_3 v_b$, and which satisfies the following *structural constraints*:

If $v_a(A) = u$, then

i. there is v_b such that $v_a \sqsubseteq_3 v_b$ and $\hat{v}_b(A) = 1$;

ii. there is v_c such that $v_a \sqsubseteq_3 v_c$ and $\hat{v}_c(A) \neq 1$.

Remark 22. A $3\mathrm{N}_I$-valuation can be seen as describing an *actual information state* that is closed under the implicit information that depends only on the meaning of the connectives. This is information that an agent *holds* and with which she can operate. Accordingly, v_a denotes a $3\mathrm{N}_I$-valuation relative to an actual information state which, in turn, is an element of a partially ordered structure representing an informational process where an agent progressively gains more information—much as in Kripke and Beth semantics. Nonetheless, a crucial difference with the latter is that, unlike a Kripke valuation $\mathfrak{v}_a$ and a Beth valuation $\mathfrak{w}_a$, a 3N$_I$-valuation v_a records which atomic and *non-atomic* formulae hold at state a. In turn, that the truth-value of a formula A is u under the 3N$_I$-valuation at issue (i.e., at the current actual information state) means that an agent holds the information that currently there is no proof nor refutation of A. However, the possibility is open that the agent acquires *new* information, which is not even potentially contained in the current information state, according to which there is a proof or a refutation of A. Otherwise, if that possibility was closed, then the agent would already hold the information that there was, respectively, a refutation or a proof of A at the current state. Thus, u can—and, in fact, eventually will—become 1 or 0, yet not by the agent's deployment of the information she holds in the current state but only by the acquisition, and possibly deployment, of new information. In turn, intuitively, new information comes from reliable external sources—say, from another agent or theory—and so it is used as actual information in the corresponding future information states (i.e., refinements of the current state) by the agent who acquired it. To illustrate these ideas, recall Euclid's Fifth postulate which, after two thousand years!, was showed to be independent from—in our terminology "undecided according to"—the remaining four postulates. As it is well-known, counterexamples to the Fifth postulate emerged outside the theory based on the other four postulates, thanks to the information exchange between agents.

Now, regarding the structural constraints and according to the meaning of u, constraint i intuitively says that if an agent can never envisage a future information state where 1 is the truth-value of A, then 0 is A's truth-value already. As for constraint ii, we are still working on an intuitive explanation for it. For now, we restrict ourselves to say that ii conveys the idea that every valuation eventually becomes deterministic, and that it is conceptually close to the notion of *bar* in Beth semantics.

Definition 5.7.6. For every M_3, and for all Γ and A,

- M_3 is a *model* of A iff $v_a(A) = 1$ for all $a \in S$;
- M_3 is a *model* of Γ iff it is a model of every $B \in \Gamma$;
- A is a *logical consequence* of Γ, $\Gamma \vDash_{3N_I} A$, iff every model M_3 of Γ is a model of A.

Remark 23. Our 3N$_I$-semantics differs from Kripke and Beth semantics in two crucial intertwined aspects that render the former suitable for applying the depth-bounded approach to **IPL**, unlike the latter: (i) In 3N_I-semantics, the meaning of each connective is specified solely by an Nmatrix. (ii) The structural constraints of 3N_I-semantics do not involve at all the meaning of the connectives. On the other hand, a rather expected concurrence between our semantics and Kripke and Beth semantics is that the notion of *model* requires to be defined considering a set of information states. However, while in Kripke and Beth semantics the information states are partially ordered by an "accessibility" relation, in our semantics those states are so ordered by the usual refinement relation. Now, as explained above, refinements from u to 1, 0 or u are related to the introduction of *new* information, which is not even potentially contained in the information that the agent holds, but that is intuitively provided by reliable external sources and so used as actual information once it is acquired. Thus, those refinements are *not* related with the introduction of *virtual* information; i.e., information that the agent does not hold but temporarily assumes as if she held it. Put differently, those refinements are an integral part of the evaluation of formulae in **IPL** because—via the structural constraints—they are required for defining the notion of model. So, there is no virtual information involved in those refinements: if an agent is at a state in which the value of a formula A is u, then in that very state she holds also the information that (i) there is a refinement (of that state) under which the value of A is deterministically 1; and (ii) there is a refinement under which the value of A is deterministically different from 1. That is, whenever an agent holds the information that the truth-value of A is u, she also holds the information that the structural constraints are satisfied.

Before proving the adequacy of our 3N_I-semantics, let us consider some examples.

Notation 5.7.7. We write $A_1, ..., A_n \vDash_{3N_I} B$ instead of $\{A_1, ..., A_n\} \vDash_{3N_I} B$.

Example 9. $\nvDash_{3N_I} A \vee \neg A$
Let M_3 be s.t. there is $a \in S$ s.t. $v_a(A) = v_a(\neg A) = u$. Then, $v_a(A \vee \neg A) = u$. Besides the structural constraints (s.c.) are satisfied: For i, let v_b be s.t. $v_a \sqsubseteq_3 v_b$ and $v_b(A) = 1$, and so $\hat{v}_b(A \vee \neg A) = 1$. For ii, let v_c be s.t. $v_a \sqsubseteq_3 v_c$, $v_c(A) = u$ and $v_c(\neg A) = 0$, and so $\hat{v}_c(A \vee \neg A) = u$.

Example 10. $\neg(A \wedge B) \nvDash_{3N_I} \neg A \vee \neg B$
Let M_3 be s.t. there is $a \in S$ s.t. $v_a(A) = v_a(B) = v_a(\neg A) = v_a(\neg B) = u$ and $v_a(A \wedge B) = 0$. Then $v_a(\neg(A \wedge B)) = 1$ and $v_a(\neg A \vee \neg B) = u$. Besides s.c. are satisfied: For i, let v_b be s.t. $v_a \sqsubseteq_3 v_b$ and $v_b(A) = 0$; then $v_b(\neg A) = 1$ and so $\hat{v}_b(\neg A \vee \neg B) = 1$. For ii, let v_c be s.t. $v_a \sqsubseteq_3 v_c$ and $v_c(\neg A) = v_c(\neg B) = u$; then $\hat{v}_c(\neg A \vee \neg B) = u$.

Example 11. $\vDash_{3N_I} \neg\neg(A \vee \neg A)$
Suppose that there is M_3 s.t. there is $a \in S$ s.t. either a) $v_a(\neg\neg(A \vee \neg A)) = 0$ or b) $v_a(\neg\neg(A \vee \neg A)) = u$. In either case, $v_a(\neg(A \vee \neg A)) = u$ since, if $v_a(\neg(A \vee \neg A)) = 1$, then $v_a(A \vee \neg A) = 0$, which is impossible. Thus, s.c. i is not satisfied: For i to be satisfied there should be v_b s.t. $v_a \sqsubseteq_3 v_b$ and $\hat{v}_b(\neg(A \vee \neg A)) = 1$. That is, v_b should be such that $v_b(\neg(A \vee \neg A)) = 1$ and so $v_b(A \vee \neg A) = 0$, which is impossible.

Example 12. $\neg A \vee \neg B \vDash_{3N_I} \neg(A \wedge B)$
Suppose that there is M_3 s.t. there is $a \in S$ s.t. $v_a(\neg A \vee \neg B) = 1$ but either a) $v_a(\neg(A \wedge B)) = 0$ or b) $v_a(\neg(A \wedge B)) = u$. If a), either a$_1$) $v_a(A \wedge B) = 1$ or a$_2$) $v_a(A \wedge B) = u$. In turn, if a$_1$), $v_a(A) = v_a(B) = 1$; but then $v_a(\neg A \vee \neg B) = 0$. If a$_2$), $v_a(A) = u$ or $v_a(B) = u$; but then either $v_a(\neg A \vee \neg B) = u$ or $v_a(\neg A \vee \neg B) = 0$. Now, if b), $v_a(A \wedge B) = u$, which amounts to a$_2$).

Example 13. $A \vee B, A \rightarrow C, B \rightarrow C \vDash_{3N_I} C$
Suppose that there is M_3 s.t. there is $a \in S$ s.t. $v_a(A \vee B) = v_a(A \rightarrow C) = v_a(B \rightarrow C) = 1$ but either a) $v_a(C) = 0$ or b) $v_a(C) = u$. If a), $v_a(A) = v_a(B) = 0$, but then $v_a(A \vee B) = 0$. If b), either $v_a(A) = 0$ or $v_a(A) = u$ and either $v_a(B) = 0$ or $v_a(B) = u$, but then either $v_a(A \vee B) = 0$ or $v_a(A \vee B) = u$.

Example 14. $\vDash_{3N_I} ((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) \rightarrow C$
Suppose that there is M_3 s.t. there is $a \in S$ s.t. either a) $v_a(((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) \rightarrow C) = 0$ or b) $v_a(((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) \rightarrow C) = u$. If a), either a$_1$) $v_a((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) = 1$ and $v_a(C) = 0$, or a$_2$) $v_a((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) = u$ and $v_a(C) = 0$. Now, if a$_1$), $v_a(((A \rightarrow C) \rightarrow A) \rightarrow A) = 0$ which, in turn, implies that either a$_{1.1}$) $v_a((A \rightarrow C) \rightarrow A) = 1$ and $v_a(A) = 0$, or a$_{1.2}$) $v_a((A \rightarrow C) \rightarrow A) = u$ and $v_a(A) = 0$. In turn, if a$_{1.1}$), $v_a(A \rightarrow C) = 0$, which is impossible. Now, if a$_{1.2}$), i is not satisfied: For i to be satisfied there should be v_b s.t. $v_a \sqsubseteq_3 v_b$ and $\hat{v}_b((A \rightarrow C) \rightarrow A) = 1$. That is, v_b s.t. $v_b((A \rightarrow C) \rightarrow A) = 1$ and $v_b(A) = 0$, which amounts to a$_{1.1}$). Now, if a$_2$), i is not satisfied: for i to be satisfied there should be v_b s.t. $v_a \sqsubseteq_3 v_b$ and $\hat{v}_b((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) = 1$. That is, v_b s.t. $v_b((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) = 1$ and $v_b(C) = 0$, which amounts to a$_1$). In turn, if b), ii is not satisfied: for ii to be satisfied there should be v_c s.t. $v_a \sqsubseteq_3 v_c$ and $\hat{v}_c(((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) \rightarrow C) \neq 1$. That is, v_c s.t. either b$_1$) $v_c((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) = 1$ and $v_c(C) = 0$, or b$_2$) $v_c((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) = 1$ and $v_c(C) = u$. b$_1$) amounts to a$_1$) and if b$_2$), either b$_{2.1}$) $v_c(((A \rightarrow C) \rightarrow A) \rightarrow A) = 0$ or b$_{2.2}$) $v_c(((A \rightarrow C) \rightarrow A) \rightarrow A) = u$. Analogously to a$_1$, b$_{2.1}$) implies that either b$_{2.1.1}$) $v_c((A \rightarrow C) \rightarrow A) = 1$ and $v_c(A) = 0$, or b$_{2.1.2}$) $v_c((A \rightarrow C) \rightarrow A) = u$ and $v_c(A) = 0$. Now, if b$_{2.1.1}$), $v_c(A \rightarrow C) = 0$, which is impossible. Similarly, if b$_{2.1.2}$), $v_c(A \rightarrow C) = 0$, which

is also impossible. In turn, if $b_{2.2}$), ii is not satisfied: for ii to be satisfied there should be v_d s.t. $v_c \sqsubseteq_3 v_d$ and $\hat{v}_d(((A \to C) \to A) \to A) \neq 1$. That is v_d s.t. either $b_{2.2.1}$) $v_d((A \to C) \to A) = 1$ and $v_d(A) = 0$, or $b_{2.2.2}$) $v_d((A \to C) \to A) = 1$ and $v_d(A) = u$. $b_{2.2.1}$) amounts to $b_{2.1.1}$). If $b_{2.2.2}$), either $v_d(A \to C) = 0$ or $v_d(A \to C) = u$. If the first, $v_d(C) = 0$, which is impossible (cf. b_2).) If the second, ii is not satisfied: For ii to be satisfied there should be v_e s.t. $v_d \sqsubseteq_3 v_e$ and v_e s.t. $\hat{v}_e(A \to C) \neq 1$. That is, v_e s.t. either $v_e(A) = 1$ and $v_e(C) = 0$, or $v_e(A) = 1$ and $v_e(C) = u$, which are both impossible (cf. $b_{2.2}$).)

We now show that $3N_I$-semantics can simulate the Hilbert-style presentation of **IPL**:

Proposition 5.7.8. *Every intuitionistic tautology is valid in $3N_I$-semantics.*

Proof. It suffices to show that (I) every axiom of **IPL** is valid in $3N_I$-semantics and (II) *MP* preserves validity.

(I) We must check all the 11 axioms. It is time-consuming but routine. As an example, we check one:

- $\vDash_{3N_I} ((A \to B) \land (B \to C)) \to (A \to C)$
  Suppose that there is M_3 s.t. there is $a \in S$ s.t. either a) $v_a(((A \to B) \land (B \to C)) \to (A \to C)) = 0$ or b) $v_a(((A \to B) \land (B \to C)) \to (A \to C)) = u$. If a), either a_1) $v_a((A \to B) \land (B \to C)) = 1$ and $v_a(A \to C) = 0$, or a_2) $v_a((A \to B) \land (B \to C)) = u$ and $v_a(A \to C) = 0$. In turn, if a_1), $v_a(A \to B) = v_a(B \to C) = 1$ and, either $v_a(A) = 1$ and $v_a(C) = 0$, or $v_a(A) = u$ and $v_a(C) = 0$; which are both impossible. Now, if a_2), i is not satisfied: For i to be satisfied there should be v_b s.t. $v_a \sqsubseteq_3 v_b$ and $\hat{v}_b((A \to B) \land (B \to C)) = 1$. That is, v_b should be s.t. $v_b((A \to B) \land (B \to C)) = 1$ and $v_b(A \to C) = 0$, which amounts to a_1). In turn, if b), ii is not satisfied: For ii to be satisfied there should be v_c s.t. $v_a \sqsubseteq_3 v_c$ and $\hat{v}_c(((A \to B) \land (B \to C)) \to (A \to C)) \neq 1$. That is, v_c should be s.t. either b_1) $v_c((A \to B) \land (B \to C)) = 1$ and $v_c(A \to C) = 0$, or b_2) $v_c((A \to B) \land (B \to C)) = 1$ and $v_c(A \to C) = u$. b_1) amounts to a_1), and if b_2) then ii is not satisfied: For ii to be satisfied there should be v_d s.t. $v_c \sqsubseteq_3 v_d$ and $\hat{v}_d(A \to C) \neq 1$. That is, v_d should be s.t. either $v_d(A) = 1$ and $v_d(C) = 0$, or $v_d(A) = 1$ and $v_d(C) = u$; which are both impossible.

The rest of the axioms are checked similarly.

(II) From the $3N_I$-table for $\to$ it follows immediately that for every $3N_I$-valuation v, if $v(A) = 1$ and $v(A \to B) = 1$, then $v(B) = 1$. □

In fact, $3N_I$-semantics is equivalent to Kripke semantics in the following sense:

Theorem 5.7.9 (Semantics equivalence).

1. *For every intuitionistic Kripke model $\mathfrak{M}$, there is a $3N_I$-model M_3 such that for every A, $\mathfrak{M}$ is a model of A iff M_3 is a model of A.*

2. *For every $3N_I$-model M_3, there is a Kripke model $\mathfrak{M}$ such that for every A, M_3 is a model of A iff $\mathfrak{M}$ is a model of A.*

Proof.
1. Given an $\mathfrak{M} = \langle \mathfrak{F}, \{\mathfrak{v}_a\}_{a \in S} \rangle$, we generate out of it an $M_3 = \langle \mathfrak{F}, \{v_a\}_{a \in S} \rangle$ by defining each $v_a \in \{v_a\}_{a \in S}$ as follows: For all A, for all $p \in \mathsf{at}(\{A\})$, let

$$v_a(p) = \begin{cases} 1 & \text{if } \mathfrak{v}_a(p) = T \\ u & \text{if } \mathfrak{v}_a(p) = F, \text{ and there is } b \text{ s.t. } aRb \text{ and } \mathfrak{v}_b(p) = T \\ 0 & \text{otherwise} \end{cases}$$

Recall that the structural constraints, i and ii, on M_3 must be satisfied. Next, we show, by induction on the degree of A, that the given $\mathfrak{M}$ is a model of A iff the generated M_3 is a model of A:

Base case: Suppose that $A := p$. Thus, $\mathfrak{M}, a \Vdash_K p$ for all $a \in S$ iff $\mathfrak{v}_a(p) = T$ for all $a \in S$. Now, according to the generation of M_3, the latter holds iff $v_a(p) = 1$ for all $a \in S$.

Inductive hypothesis: Suppose that if A has degree n, then $\mathfrak{M}, a \Vdash_K A$ for all $a \in S$ iff $v_a(A) = 1$ for all $a \in S$.

Inductive step: Let A have degree $n + 1$, then we have four cases.

($\vee$) Suppose that $A := B \vee C$. Thus, $\mathfrak{M}, a \Vdash_K B \vee C$ for all $a \in S$ iff $\mathfrak{M}, a \Vdash_K B$ or $\mathfrak{M}, a \Vdash_K C$ for all $a \in S$. By inductive hypothesis, the latter holds iff $v_a(B) = 1$ or $v_a(C) = 1$ for all $a \in S$, iff $v_a(B \vee C) = 1$ for all $a \in S$.

($\wedge$) Suppose that $A := B \wedge C$. Thus, $\mathfrak{M}, a \Vdash_K B \wedge C$ for all $a \in S$ iff $\mathfrak{M}, a \Vdash_K B$ and $\mathfrak{M}, a \Vdash_K C$ for all $a \in S$. By inductive hypothesis, the latter holds iff $v_a(B) = 1$ and $v_a(C) = 1$ for all $a \in S$, iff $v_a(B \wedge C) = 1$ for all $a \in S$.

($\to$) Suppose that $A := B \to C$. Thus, there is $a \in S$ s.t. $v_a(B \to C) \neq 1$ iff there is $a \in S$ s.t. either a) $v_a(B \to C) = 0$ or b) $v_a(B \to C) = u$. Now, a) holds iff there is $a \in S$ s.t. either a$_1$) $v_a(B) = 1$ and $v_a(C) = 0$, or a$_2$) $v_a(B) = u$ and $v_a(C) = 0$. By inductive hypothesis, a$_1$) holds iff there is $a \in S$ s.t. $\mathfrak{M}, a \Vdash_K B$ but $\mathfrak{M}, a \nVdash_K C$, iff there is $a \in S$ s.t. $\mathfrak{M}, a \nVdash_K B \to C$. Now, a$_2$) holds iff there are $a, b \in S$ s.t. $v_a \sqsubseteq_3 v_b$, $v_a(B) = u$, $v_a(C) = 0$, $\hat{v}_b(B) = 1$, and $v_b(C) = 0$. So, by inductive hypothesis, a$_2$) holds iff there are $a, b \in S$ s.t. aRb, $\mathfrak{M}, b \Vdash_K B$

but $\mathfrak{M}, b \nVdash_K C$, iff there is $a \in S$ s.t. $\mathfrak{M}, a \nVdash_K B \to C$. In turn, b) holds iff there are $a, b \in S$ s.t. $v_a \sqsubseteq_3 v_b$, $v_a(B \to C) = u$, and either b$_1$) $\hat{v}_b(B \to C) = 0$ or b$_2$) $\hat{v}_b(B \to C) = u$. Thus, b$_1$) is analogous to a$_1$); whereas b$_2$) holds iff $\hat{v}_b(B) = 1$ and $\hat{v}_b(C) = u$, and so b$_2$) is analogous to a$_2$).

($\neg$) Suppose that $A := \neg B$. Thus, there is $a \in S$ s.t. $v_a(\neg B) \neq 1$ iff there is $a \in S$ s.t. either *a*) $v_a(\neg B) = 0$ or *b*) $v_a(\neg B) = u$. Now, *a*) holds iff there is $a \in S$ s.t. either a_1) $v_a(B) = 1$ or a_2) $v_a(B) = u$. By inductive hypothesis, a_1) holds iff there is $a \in S$ s.t. $\mathfrak{M}, a \Vdash_K B$, iff there is $a \in S$ s.t. $\mathfrak{M}, a \nVdash_K \neg B$. Now, a_2) holds iff there are $a, b \in S$ s.t. $v_a \sqsubseteq_3 v_b$, $v_a(B) = u$ and $\hat{v}_b(B) = 1$. So, by inductive hypothesis, a_2) holds iff there are $a, b \in S$ s.t. aRb and $\mathfrak{M}, b \Vdash_K B$, iff there is $a \in S$ s.t. $\mathfrak{M}, a \nVdash_K \neg B$. In turn, *b*) holds iff there is $a \in S$ s.t. $v_a(B) = u$ and so *b*) is analogous to a_2).

2. Conversely, given an $M_3 = \langle \mathfrak{F}, \{v_a\}_{a \in S} \rangle$, we generate out of it an $\mathfrak{M} = \langle \mathfrak{F}, \{\mathfrak{v}_a\}_{a \in S} \rangle$ by defining each $\mathfrak{v}_a \in \{\mathfrak{v}_a\}_{a \in S}$ as follows: For all A, for all $p \in \mathsf{at}(\{A\})$, let

$$\mathfrak{v}_a(p) = \begin{cases} T & \text{if } v_a(p) = 1 \\ F & \text{if } v_a(p) = u \text{ or } v_a(p) = 0 \end{cases}$$

Next, we show, by induction on the degree of A, that the given M_3 is a model of A iff the generated $\mathfrak{M}$ is a model of A:

Base case: Suppose that $A := p$. Thus, according to the generation of $\mathfrak{M}$, $v_a(p) = 1$ for all $a \in S$ iff $\mathfrak{v}_a(p) = T$ for all $a \in S$, iff $\mathfrak{M}, a \Vdash_K p$ for all $a \in S$.

Inductive hypothesis: Suppose that if A has degree n, then $v_a(A) = 1$ for all $a \in S$ iff $\mathfrak{M}, a \Vdash_K A$ for all $a \in S$.

Inductive step: Let A have degree $n + 1$, then we have four cases.

($\vee$) Suppose that $A := B \vee C$. Thus, $v(B \vee C) = 1$ for all $a \in S$ iff $v(B) = 1$ or $v(C) = 1$ for all $a \in S$. By inductive hypothesis, the latter holds iff $\mathfrak{M}, a \Vdash_K B$ or $\mathfrak{M}, a \Vdash_K C$ for all $a \in S$, iff $\mathfrak{M}, a \Vdash_K B \vee C$ for all $a \in S$.

($\wedge$) Suppose that $A := B \wedge C$. Thus, $v(B \wedge C) = 1$ for all $a \in S$ iff $v(B) = 1$ and $v(C) = 1$ for all $a \in S$. By inductive hypothesis, the latter holds iff $\mathfrak{M}, a \Vdash_K B$ and $\mathfrak{M}, a \Vdash_K C$ for all $a \in S$, iff $\mathfrak{M}, a \Vdash_K B \wedge C$ for all $a \in S$.

($\to$) Suppose that $A := B \to C$. Thus, there is $a \in S$ s.t. $\mathfrak{M}, a \nVdash_K B \to C$ iff there are $a, b \in S$ s.t. aRb, $\mathfrak{M}, b \Vdash_K B$ but $\mathfrak{M}, b \nVdash_K C$. By inductive hypothesis, the latter holds iff there is $a \in S$ s.t. $v_a(B) = 1$ but either $v_a(C) = 0$ or $v_a(C) = u$, iff there is $a \in S$ s.t. either $v_a(B \to C) = 0$ or $v_a(B \to C) = u$, iff there is $a \in S$ s.t. $v_a(B \to C) \neq 1$.

(¬) Suppose that $A := \neg B$. Thus, there is $a \in S$ s.t. $\mathfrak{M}, a \nVdash_K \neg B$ iff there are a, b s.t. aRb and $\mathfrak{M}, b \Vdash_K B$. By inductive hypothesis, the latter holds iff there is $a \in S$ s.t. $v_a(B) = 1$, iff there is $a \in S$ s.t. $v_a(\neg B) = 0$, iff there is $a \in S$ s.t. $v_a(\neg B) \neq 1$.

□

Corollary 5.7.10 (Completeness). *If* $\Gamma \vDash_{3N_I} A$*, then* $\Gamma \vDash_K A$*.*

Proof. Suppose that every model M_3 of Γ is a model of A, but that there is a model $\mathfrak{M}$ of Γ that is not a model of A. Given 1. in the previous proposition, from the latter assumption it follows that we can generate, out of $\mathfrak{M}$, an M_3 such that M_3 is a model of Γ but is not a model of A; which contradicts the first assumption. □

Corollary 5.7.11 (Soundness). *If* $\Gamma \vDash_K A$*, then* $\Gamma \vDash_{3N_I} A$*.*

Proof. Analogous to the previous one. □

5.7.1 Coda

Our $3N_I$-semantics is interesting in its own right. However, we conceived it as a basis to extend the notions of depth-bounded consequence and inconsistency to **IPL**. We consider that $3N_I$-semantics paves the way for developing a non-deterministic semantics suitable for the hierarchy of approximations defined in proof-theoretic terms above.

As explained above, the truth-values 1, 0 and u are all *defined* and respectively denote the primary notions of intuitionistic truth, falsity and indeterminacy. Now, when a formula A takes neither of those truth-values, we could say that the truth-value of A is *unknown*. Accordingly, a *partial valuation* v for $\mathcal{L}$ would be a partial function $v : F(\mathcal{L}) \longrightarrow \{1, 0, u\}$, and we would denote by $v(A) = \bot$ whenever v is undefined for A. As in the cases addressed in previous Chapters, it would be then technically convenient to treat $\bot$ as a fourth truth-value, and so interpret it as denoting a fourth primary notion: (full) *ignorance*. Here it is important to note that u and $\bot$ would denote two different notions. Again, that the truth-value of a formula A is u means that an agent *holds* the information that, under the current actual information state, there is no proof nor refutation of A. In contrast, that the truth-value of a formula A is $\bot$ would mean that an agent *does not hold* any information determining whether or not there is a proof or a refutation of A. Intuitively, $\bot$ would represent situations where, at the current actual information state, an agent has not deployed the information she holds, either because she has not started the deployment process yet or the process is still running. Accordingly, $\bot$ would eventually become 1, 0 or u, but becoming so could involve not only actual information (which either the agent holds or receives from external sources), but also virtual information (that she temporarily assumes as she held it). Thus, intuitively, unlike 3-refinements (from u to 1, 0 or u) which correspond to the structural constraints of our 3N_I-semantics and involve solely actual information, the corresponding refinements from $\bot$ to the defined values may involve the introduction of virtual information.

Therefore, the four truth-values would be partially ordered by the relation $\preceq_4$, defined as the minimum partial order over $\{1, 0, u, \bot\}$ such that $\bot$ is the least element, and $u \preceq_4 1$, $u \preceq_4 0$. For the sake of clarity, the graph corresponding to this partial order is depicted in Figure 5.9.[21] Thereby, a 4_I*-valuation* v for $\mathcal{L}$ could be defined as a function $v : F(\mathcal{L}) \longrightarrow \{1, 0, u, \bot\}$. In turn, as we did in previous cases, we could pick out from the set of all 4_I-valuations those that agree with the intended meaning of the connectives by means of an Nmatrix, which would conservatively extend $\mathcal{M}_I$. Currently we are still working on devising such an Nmatrix. In the case of **IPL**, obtaining suitable semantics for our depth-bounded approximations is not so straightforward since we deal with the additional difficulty that the corresponding Nmatrix is embedded into a partially ordered structure.

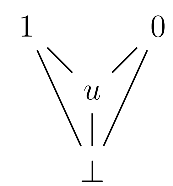


Figure 5.9: Information order over $\{1, 0, u, \bot\}$

[21]Moreover, the graph allow us to represent an important intuition: $\bot$ can directly turn into 1 or 0, without necessarily turning into u first.

Part III

A digression into complexity

Chapter 6

Computational complexity revisited

6.1 Introduction

Rational decisions on the relative computational merits of different formalizations of a logic are of paramount importance for the mechanization of deduction. Is there a rational and relatively stable basis for such decisions? In Chapters 2, 3 and 5, we have considered the traditional approach to the relative complexity of proof systems in terms of the p-simulation relation introduced by Cook and Rechow [49]. In this Chapter, we propose a way of enhancing such an approach and produce results which are more relevant to the problem of mechanical proof.

In [35, pp. 133-134], Bibel wrote:

> [...] evaluation of the performance of theorem provers is quite a complicated thing, so that it would be helpful to have a clearer view of what we mean by "quantitatively better" or by "improved methods". In principle, the performance may be measured either by experience with running systems or by mathematical analysis. In the former case, one would compare the relative performance of implementations of different methods on a number of samples. Since [...] there is relatively little experience in the current state of the art of building and testing theorem provers, any such experimental comparisons at present should be taken with much caution. [...] Under these circumstances it is not surprising that the present techniques of mathematical analysis, the other possibility of measuring performance, are rather limited as well. For such an analysis, what we would need is a realistic mathematical model of the binary relation which

> captures the natural and practically relevant meaning of the term "better than" with respect to proof procedures.

Almost forty years later the situation has not changed much. The unreliability of experimental "evidence" is a general methodological problem: its results are meaningless if they are not supported by theoretical analysis. On the theoretical side, the methods which can be found in the literature are the worst-case and average-case analysis of particular algorithms, as well as the analysis of the relative efficiency of (non-deterministic) proof systems in terms of the p-simulation relation. None of these treatments seems, on its own, to provide an adequate basis for positive judgements on the relative complexity of proof systems. All proof procedures are very likely (if P$\neq$ NP) to have a superpolynomial worst-case complexity. On the other hand, are such worst-case results sufficient to describe the computational behaviour of an algorithm? In Karp's words:

> The traditional worst-case analysis—the dominant strain in complexity theory—corresponds to a scenario in which the instances of a problem to be solved are constructed by an infinitely intelligent adversary who knows the structure of the algorithm and chooses inputs that will embarrass it to the maximal extent [104, p. 106].

Karp suggests an alternative approach in which "the inputs are assumed to come from a user who simply draws his instances from some reasonable probability distribution, attempting neither to foil nor to help the algorithm" [104, p. 106]. This approach too involves methodological difficulties. As Karp admits, a result obtained in this way "is meaningful only if the assumed probability distribution of problem instances bears some resemblance to the population of instances that arise in real life, or if the probabilistic analysis is robust enough to be valid for a wide range of probability distributions" [104, p. 106]. To put it with Rabin:

> We may postulate a certain distribution such as all instances being equally likely, but in a practical situation the source of instances of the problem to be solved may be biased in an entirely different way. The distribution may be shifting with time and will often not be known to us. In the extreme case, most instances which actually come up are precisely those for which the algorithm behaves worst [130, p.632].

The situation is even more complicated than it appears to be, if we consider that what we often intend to compare are non-deterministic *proof systems* rather than deterministic *proof procedures.* Roughly, a proof system is a collection of inference rules with only partial or no control on their application, so that in some cases we choose which rule to apply next among several possibilities. As mentioned above, rational decisions on the relative computational merits of proof systems are extremely important when selecting a system as the underlying formalization of future algorithmic developments. From this point of view, a comparison between deterministic algorithms based on different formal systems—namely, on two different sets of allowed inference rules—is not a crucial test unless the algorithms in question can be proven to be *optimal.* Otherwise it is always possible to produce a better version of one of the two algorithms which overthrows the previous judgement. In this way negative empirical evidence can always be blamed on the particular "current version" of the algorithm and diverted from the formalization itself. On the other hand, a mathematical model of the notion "is better than" when referred to proof systems, is meaningful only if its results are both *relevant* and *stable.* It may seem, then, that such a mathematical model cannot be obtained. However, there is a natural limit on the possible algorithms that can be developed on the basis of a given formalization, and this limit could be different for different systems. So, we need criteria for measuring the relative *potential* of different formalizations from the point of view of mechanical proof.

6.2 How hard is it to find an easy proof?

The standard way of measuring the resources r (time or space) required by a non-deterministic algorithm M for a given input I is in terms of the best case for the input I—if we are interested in the time complexity of our algorithm, this is represented by the shortest path in the computation tree which leads to success. Then the complexity of M itself is measured as a function of the input size, by taking the maximum value of r over all inputs of a given size—alternatively one can take the average value over a given probability distribution. Accordingly, Cook and Rechow [49, 50] addressed the problem of comparing different proof systems in terms of their *shortest* proofs. This measure of complexity is certainly adequate to establish negative results with respect to a given proof system, like in Haken's proof that resolution has exponential size shortest proofs for the pigeon-hole principle [95]. The p-simulation relation, introduced and studied by Cook and Rechow, serves the purpose of propagating such negative results to other proof systems: if S_2 p-simulates S_1 and S_2 is not polynomially bounded, than neither is S_1. On the other hand, if S_2 does not p-simulate S_1 there is still a chance that S_1 is polynomially bounded.

However, the p-simulation relation is misleading if it is taken as a basis for positive judgements about the superiority of a proof system over another with respect to the *problem of mechanical proof.* If we assumed that "S_2 is better than S_1" is modelled by

S_2 p-simulates S_1 but not viceversa

we would soon run into highly counterintuitive judgements. For instance, the existence of polynomial-size proofs of the pigeon-hole principle in Frege (i.e. Hilbert-style axiomatic) systems, proven by Buss in [40], together with Haken's result on the intractability of this problem for resolution [95], imply that

resolution cannot p-simulate Frege systems.

Since it is well-known [49] that

Frege systems can p-simulate resolution

we should conclude that

Frege systems are essentially more efficient than resolution.

Nonetheless, this clashes with our intuition that an unrestricted Frege system, as it stands, with no control structure, is no use for the purpose of mechanical proof. In fact, this kind of result does not take into account the difficulty of *finding* the short Frege proofs of the pigeon hole principle required to prove that resolution cannot p-simulate Frege systems. In fact these proofs are so "hard" to find that Cook and Rechow had conjectured that they did not exist [50]. In general, if we show that S_2 p-simulates S_1 in Cook and Rechow's sense, we know that every algorithm for S_1 can be turned into an algorithm for S_2 which can perform the same inferences within (essentially) the same resource bounds. However, proving that the converse does not hold, namely that S_1 cannot p-simulate S_2 would not be sufficient to show that S_2 is "more efficient" than S_1 in any intuitive sense of this expression: it may very well be that the short S_2-proofs of classes of formulae which are hard for S_1 require a good deal of ingenuity to be found, as is the case for the short Frege proofs of the pigeon-hole principle.

This difficulty of the present theoretical framework in expressing *positive* results has led many researchers in the area of automated deduction to dismiss the work on the computational complexity of proof systems as irrelevant. We are usually interested in knowing not only that a system *admits* short proofs in cases in which another does not, but also *how hard it is to find* such short proofs. We need some extra information that cannot be obtained from the standard way of evaluating the relative efficiency of non-deterministic proof algorithms via p-simulation.

6.3 From p-simulation to p-emulation

A natural way of measuring the "difficulty" of finding the solution of a problem within a given formal system is in terms of the amount of information required to obtain it. This amount of information is, in turn, inversely related to the probability of finding the required solution "by chance", using the rules of the formal system completely "at random". So, it seems natural to measure the relative "difficulty" of finding short proofs within two proof systems in terms of the relative frequency with which such short proofs are found when we apply the rules of the system "blindly". This approach suits very well the non-deterministic nature of a proof system. In general, proof systems are characterized by a set of rules such that the next rule to be applied is not uniquely determined by the current state of the proof. The proof generated by the algorithm on input I can be seen as the outcome of a *random process* whose sample space contains all possible sequences of rule-applications.[1]. Since such sequences depend on random choices, we shall call them *choice sequences*.

The resource requirement of the system for input I is treated as a *random variable* whose value depends on these choice sequences. This approach embraces algorithms with lower and lower "degree of freedom" including the usual deterministic ones as a limiting case. By a *proof algorithm* based on a proof system S we shall intend the proof system S plus some control structure on the applications of its rules, including as a limiting case, the original proof system with no control structure. The successive configurations will be *proof-states*. A *determination* M^* of a proof system M is a proof system with a smaller degree of freedom; i.e., for each given input, its sample space is properly contained in the sample space of M. A determination of a proof system is obtained as a result of a stricter *control* on the applications of the derivation rules. A *deterministic* proof algorithm is just the limit of a sequence of determinations of a proof system, obtained when the sample space for each given input is a singleton; that is, the control structure determines uniquely the next configuration at each given

[1]This sample space is often referred to in the literature as *the search space* associated with the proof system. This terminology can be misleading, since it suggests that such a space should in some sense be "explored" and, therefore, it is always convenient to reduce it to smaller proportions. However, in many proof systems, finding a proof (or a short proof) does not depend at all on exploring the space of all possible sequences of rule applications, since either all sequences lead to a proof or no sequence does. In this case, there is no guarantee that restricting the "search space" will improve the performance. It may well be that most of the short proofs which are possible in the unrestricted system are left out by the restriction. So if we applied the rules at random in the original system we might have a better chance of obtaining short proofs than we have in the "refinement" with a smaller "search space". This is perfectly clear if we look at the space of all possible proofs as at the sample space of a random process: a bigger sample space does not mean a smaller probability of obtaining short proofs.

step. Following this approach, we shall define a series of notions which yield a more complete profile of the complexity of a proof system. On the basis of these notions, we shall then define a preorder relation, called *p-emulation*, which is more adequate than p-simulation to capture the intuitive meaning of "more efficient" when referred to non-deterministic algorithms. In fact, we shall show that "S_2 p-emulates S_1 but not viceversa" is a good rendering of the intuitive notion of "S_2 is a refinement of S_1", so allowing for relevant positive results about the relative efficiency of logical systems. We shall also show how such results can be made *stable* by means of a stronger relation that we call *monotonic p-refinement*. These relations lead to a different classification of proof systems than Cook and Rechow's p-simulation, which agrees with our intuitions about efficiency in the clear-cut cases, and provides a rational basis for comparing the computational merits of different formalizations.

6.3.1 Random algorithms and choice sequences

Random algorithms have been employed successfully by Karp and Rabin for different purposes. Karp focused on algorithms which allow errors, whereas Rabin was more interested in algorithms which always solve the problem in exact terms. In particular, Rabin was interested in studying random algorithms with a polynomially bounded *expected computation time* for problems whose membership of P is still open. As far as exact algorithms are concerned, this is of course possible only if the problem under consideration is in NP. However, we shall show in the sequel that even for problems, like the tautology problem for **CPL** (TAUT), whose membership of NP is unknown (and unlikely), or indeed for every problem at all, probability considerations can be fruitfully used to evaluate the *relative* efficiency of non-deterministic algorithms and to measure the relative "difficulty" of finding short proofs when such proofs are possible. The analysis in this chapter will be rather informal; although everything said here can be (tediously) translated into rigorous definitions. There is a trade-off between precision and readability. We have decided to privilege readability in a way that still captures the essential features of the problem we are considering, without letting "overprecision" hide the main points.

Let Π be a computational problem, i.e., a collection of computational tasks each of which is called an *instance* of Π. For every instance I, let $|I|$ be the *size* of I, measured in any reasonable way.[2] A *non-deterministic algorithm M* is, informally, characterized by a set of transition rules $\{\vdash_i\}$ for passing from a configuration to another, a (possibly empty) control component which partially determines the set of allowed transition steps for each given configuration, and by a halting condition.

[2]For a definition of "reasonable" in this context, see [89].

Non-determinism here means that the control component may, in some cases, allow several alternatives for the step to enter next; that is, for some configuration C there may be $d > 1$ possible C' such that $C \vdash_i C'$ is allowed for some i, and even the local transition rules $\vdash_i$ may be non-functional. So, the possible computations generated by M on input I can be arranged as a tree that we shall call *the computation tree of M on input I*.

Thereby, the efficiency of a non-deterministic algorithm can be measured (at least) in two different ways. The first way is the traditional one, in which the *minimal* resource requirements of the algorithm for each given input are taken into account. The second way consists in regarding a non-deterministic algorithm as a *random* algorithm. We look at its computation tree on a given input as at a description of a *random process*, where the branches represent the possible outcomes. Let us suppose that the immediate successors of each node in the computation tree are enumerated. Then, computations are identified by sequences of integers that we call *choice sequences*. If we make the simplifying assumption that all the steps (including the control component) have the same unitary time-cost, the length of a choice sequence corresponds to the time-cost of the computation that it encodes. If we are interested in other resources, like space, we can consider a function that maps each complete choice sequence to the integer representing the amount of resources consumed by the particular computation that it encodes. In the sequel we shall consider only the length of computations; i.e., their time-requirements, but clearly whatever we say can be referred, *mutatis mutandis*, to other resources.

We shall denote the length of choice sequence a by $|a|$. This length is then treated as a *random variable* taking values in the positive integers. We can imagine that at each configuration C with d possible successors, the algorithm uses some random process to choose which of the d successors of C to enter next.[3] In this approach, randomness is not in the occurrence of the instances I, but is introduced into the algorithm itself, in the choice of a particular sequence. For each given input I there is a well-defined *sample space* $\mathbf{\Omega}_I$ containing all the possible outcomes. The successive configurations of M on input I can be described by a partial function $M(I, a)$, where I is the usual input and a is a string of integers, which is undefined when a is a string not corresponding to any path in the computation tree.

A non-deterministic algorithm computes a partial function $\phi_M(I, a)$. If $M(I, a)$ is not a halting configuration, $\phi_M(I, a)$ is undefined and the choice sequence is said

[3]This does not mean, of course, that the algorithm needs to enumerate all the possible successor configurations before choosing one, just as we do not need to be aware of all the possibilities before we take some action. The issue here is that the probability of choosing one alternative or the other is determined by the number of alternatives, not by our (the algorithm's) knowledge of them.

to be *partial*. Otherwise, the choice sequence is said to be *total*. The sample space $\mathbf{\Omega}_I$ contains all and only the choice sequences generated by the algorithm on input I which are either total or infinite (just total, if the algorithm always terminates). If M is an *exact* algorithm, $\phi_M(I,a) = \phi_M(I,a')$ for all total a, a' in $\mathbf{\Omega}_I$. In the sequel we shall deal only with exact algorithms. A *determination* of a non-deterministic algorithm M is a non-deterministic algorithm M^* such that for each input I, its sample space $\mathbf{\Omega}_I^*$ is properly included in the sample space $\mathbf{\Omega}_I$ generated by M on the same input. A determination results from a more restrictive control component on the application of the transitions rules. A *deterministic algorithm* is a non-deterministic algorithm, such that for every input I the sample space $\mathbf{\Omega}_I$ is a singleton.

6.3.2 Notation

In the rest of this chapter we shall consistently use the following notation

Notation	Definition
$\lvert a \rvert$	length of the choice sequence a
$M_j(I,a)$	Configuration associated with input I and choice sequence a in algorithm M_j
$\mathbf{\Omega}_{j,I}$	sample space generated by algorithm M_j on input I
$pr(A)$	probability of the event A
$\rho_{j,I}(r)$	$pr\{a : \lvert a \rvert = r\}$
$P_{j,I}(r)$	$\sum_{s \leq r} \rho_{j,I}(s)$
2^A	set of all subsets of set A

6.3.3 Simulations

Let us write $a \sqsubseteq b$ if the choice sequence a is a prefix of the choice sequence b. We also denote by $\downarrow \mathbf{\Omega}_{j,I}$ the set of all b such that $b \sqsubseteq a$ for some $a \in \mathbf{\Omega}_{j,I}$.

Definition 6.3.1. Let M_1 and M_2 be non-deterministic algorithms computing the functions ϕ_1 and ϕ_2 respectively. A *simulation of* M_1 *in* M_2 is a computable function $f : \downarrow \mathbf{\Omega}_{1,I} \longrightarrow 2^{\downarrow \mathbf{\Omega}_{2,I}}$ mapping choice sequences of M_1 to *sets* of choice sequences of M_2 such that:

1. $(\forall I), (\forall a \in \mathbf{\Omega}_{1,I}), (\forall b \in f(x)), \phi_1(I,a) = \phi_2(I,b)$.

2. If $(\forall a, b \in \mathbf{\Omega}_{1,I}), a \sqsubseteq b$ then $(\forall c \in f(a)), (\exists d \in f(b)),\ c \sqsubseteq d$ and $(\forall d \in f(b)), (\exists c \in f(a)),\ c \sqsubseteq d$.

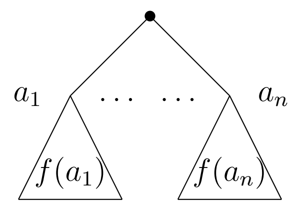


Figure 6.1: Simulation process.

Condition 1 means that the function ϕ_2 computed by M_2 is a refinement of the function ϕ_1 computed by M_1. Conditions 2 means that the function f is a *step-by-step* simulation of one algorithm into the other. In general, we allow for more than one M_2-computation for each given M_1-computation. Let $\simeq_f$ be the equivalence relation defined as follows:

$$a \simeq_f b \text{ iff } f(a) = f(b). \tag{6.1}$$

Let $\|a\|_f$ denote the equivalence class of a under $\simeq_f$. Clearly,

$$\|a\| = \|b\| \text{ iff } f(a) = f(b).$$

Let $M_{f(1)}$ denote the determination of M_2 consisting in the simulation of M_1 by f; i.e., at each step if the random process generated by M_1 has outcome a, $M_{f(1)}$ "reads" the outcome, maps it to the set (possibly a singleton) $f(a)$ of M_2-computations and enters another random process to generate an outcome $b \in f(a)$. A typical simulation step is illustrated in Figure 6.1, where $a_1, ..., a_n$ are the outcomes of M_1 and $f(a_1), ..., f(a_n)$ the outcomes of the simulation algorithm $M_{f(1)}$.

Clearly,

$$pr(a) \leq pr(f(a)) \text{ and } pr(\|a\|_f) = pr(f(a)).$$

Suppose, moreover, that the function f has the additional property that

$$\text{If } a \not\simeq_f b \text{ then } f(a) \cap f(b) = \emptyset. \tag{6.2}$$

Then, for every I, the partition induced by $\simeq_f$ in $\mathbf{\Omega}_{1,I}$ induces a corresponding partition in the image of $\mathbf{\Omega}_{1,I}$ under f, and the partitions are one-one. If (6.2) is satisfied, we say that the simulation function is *discriminating*.

Then, the following holds:

Proposition 6.3.2. *Let A be an event (a set of outcomes) in the sample space $\mathbf{\Omega}_{1,I}$ of M_1 on input I. Let us denote by $f(A)$ the set $\bigcup_{a\in A} f(a)$. Then, if f is a discriminating simulation function,*

$$pr(A) \leq pr(f(A)) = \sum_{a\in A} pr(f(a)).$$

We shall make use of this property in the sequel.

In this context, the *p-simulation* relation can be redefined as follows:

Definition 6.3.3. M_2 *p-simulates* M_1 iff there are a simulation function $f :\downarrow \mathbf{\Omega}_{1,I} \mapsto 2^{\downarrow \mathbf{\Omega}_{2,I}}$ computable in polynomial time and a fixed polynomial p such that, for every input I, every $a \in \mathbf{\Omega}_{1,I}$ and every $b \in f(a)$, $|b| \leq p(|a|)$.

This definition is slightly different from the Cook and Rechow definition in that it expresses the requirement that the simulation proceeds *step by step*. It is also different from another notion of p-simulation that frequently occurs in the literature—for instance in [40]—according to which M_2 p-simulates M_1 iff there is a fixed polynomial p such that, for every input I and every computation a generated by M_1 on input I, *there exists* an M_2-computation b on the same input with $\phi_2(I,b) = \phi_1(I,a)$ and $|b| \leq p(|a|)$. We shall refer to this weaker notion of p-simulation as to *weak p-simulation*.

6.3.4 Complexity functions

Throughout this Chapter we make the simplifying assumptions that the random process which leads from one configuration to the next has unitary cost. If we think of a non-deterministic algorithm as an algorithm which makes "guesses", this means that guesses which lead to an illegal step are not charged for. Although this assumption is hardly realistic, it is reasonable to assume that the process of "applying an arbitrary transformation rule to the present configuration", which involves (a) selecting a rule and (b) checking its applicability to the present configuration, is bounded above by some polynomial function of the size of the current configuration (since there is always a finite number of rules).[4] Therefore, our simplifying assumption is theoretically appropriate. In practice the cost of the process of "applying *a* rule", if the

[4]The fact that there is always a finite number of transformation rules should not be confused with the fact that there are finitely many *applications* of rules. For instance, if the rules of our non-deterministic algorithm are the rules of an axiomatic system of logic with a fixed number of axiom schemes, the number of possible next configurations is infinite.

rules are reasonable from the algorithmic point of view, can be considered roughly the same for all non-deterministic algorithms.[5]

Given a particular input I and a non-deterministic algorithm M_j, we can consider:

1. its worst-case requirement, i.e., $\max_{a \in \mathbf{\Omega}_{j,I}} |a|$;
2. its expected requirement, i.e., the expected value of $|a|$ for $a \in \mathbf{\Omega}_{j,I}$;
3. its best-case requirement, i.e., $\min_{a \in \mathbf{\Omega}_{j,I}} |a|$.

The latter corresponds to the time requirement of a non-deterministic algorithm as it is traditionally defined. Notice that here the expressions "worst-case", "expected" and "best-case", do not refer to the *global* resource requirements of the algorithm as a function of the complexity of the input, but to the *local* resource requirements for each particular input I as a function of the choice sequence. Each of these local functions gives rise to a different global time (space) complexity function depending on whether we are interested, respectively, in its maximal value, its expected value, or its minimal value over all the inputs of a given size. So, if we are interested in the worst-case complexity over all the inputs of a given size, we have the following functions, where $|I|$ is the size of input I, and we write $pr(a)$ for $pr(\{a\})$, the probability of the event consisting only of the choice-sequence a:

1. *$\mathcal{WW}$-complexity*
$$\mathcal{WW}(n) = \max\{\max_{a \in \mathbf{\Omega}_{j,I}} |a| : |I| = n\}$$
2. *$\mathcal{WE}$-complexity*
$$\mathcal{WE}(n) = \max\{\sum_{a \in \mathbf{\Omega}_{j,I}} pr(a) \cdot |a| : |I| = n\}$$
3. *$\mathcal{WB}$-complexity*
$$\mathcal{WB}(n) = \max\{\min_{a \in \mathbf{\Omega}_{j,I}} |a| : |I| = n\}$$

One can define similar functions $\mathcal{EW}, \mathcal{EE}, \mathcal{EB}, \mathcal{BW}$, etc., by taking the expected or the minimal value of the different local functions over all inputs of size n. The complexity of a non-deterministic algorithm, as traditionally defined, corresponds to

[5] An example of an "unreasonable" rule would be the following: given a set S of formulae of propositional logic, extend S with A if A is a logical consequence of S. On the other hand, the rules of logical systems are usually reasonable.

its $\mathcal{WB}$-complexity. There is no special reason—except reasons related to the development of the NP-completeness theory—that this measure should play a privileged role in evaluating the performance of non-deterministic algorithms. In fact both the $\mathcal{WW}$-complexity and the $\mathcal{WE}$-complexity appear to play an equally important role. The former yields an upper bound on the resources required by *any* deterministic version of the non-deterministic algorithm under consideration. The latter yields an estimate of the worst-case *expected* performance of the non-deterministic algorithm (seen as a random algorithm) over all inputs of given size. This measures allows us to ask interesting questions about "heuristics". A heuristic can be described as a determination of a non-deterministic algorithm. So we can ask ourselves: is the $\mathcal{WE}$-complexity of this determination lower than the corresponding complexity of the original algorithm?[6] If not, the original algorithm augmented with a module for making random choices could be a better bet.

The complexity functions defined above obviously satisfy the following condition, for all arguments n:

$$\mathcal{WB}(n) \leq \mathcal{WE}(n) \leq \mathcal{WW}(n) \tag{6.3}$$

So, as far as this type of measures are concerned, the strongest result we can prove about the relative efficiency of two non-deterministic algorithms, say M_1 and M_2 is that the $\mathcal{WB}$-complexity of one algorithm ($\mathcal{WB}_1(n)$) is not bounded above by any polynomial in the $\mathcal{WW}$-complexity of the other($\mathcal{WW}_2(n)$). Namely,

$$\neg\exists p, \forall n, \mathcal{WB}_1(n) \leq p(\mathcal{WW}_2(n)) \tag{6.4}$$

However, these measures *are not sufficient* for our purposes. A result like the above does not provide a complete description of the relative complexity of two non-deterministic algorithms. It might detect only a *local* improvement which is compatible with a worse *global* behaviour of the algorithm.

6.3.5 Distribution functions

We want to capture the notion of a proof system's global behaviour being "better" than another. As argued in the introduction to this section, "S_2 p-simulates S_1 but not viceversa" cannot serve this purpose. Notice that:

If S_2 p-simulates S_1 then there is a polynomial p such that $\mathcal{WB}_2(n) \leq p(\mathcal{WB}_1(n))$,

[6] Of course such a question makes sense also in the case in which the determination is deterministic, so that for instance its $\mathcal{WE}$-complexity coincides with its $\mathcal{WW}$-complexity.

where $\mathcal{WB}_i$ indicates the $\mathcal{WB}$-complexity of proof system S_i. However the p-simulation relation says nothing about the $\mathcal{WW}$ and the $\mathcal{WE}$-complexity of the two proof systems. So we look for an alternative preorder of proof systems which yields a more complete profile of the relative efficiency of two non-deterministic algorithms and reflects more faithfully our intuitions.

Let us go back to the non-deterministic algorithm M_j seen as generating a *random process* where the length of a computation is treated as a random variable. Let us denote by $\rho_{j,I}(r)$ the probability that M_j terminates exactly in r steps on input I that is

$$\rho_{j,I}(r) = pr_{j,I}\{a : |a| = r\}.$$

Clearly, for each M_j and each fixed I:

$$\rho_{j,I}(r) \geq 0, \qquad \sum_r \rho_{j,I}(r) = 1$$

so that $\rho_{j,I}(r)$ is the *(probability) mass function* of the discrete random variable r. Hence,

$$P_{j,I}(r) \stackrel{\text{def}}{=} \sum_{s \leq r} \rho_{j,I}(s)$$

is *the distribution function* of the random variable r. Notice that for all I, r_1, r_2:

$$\text{If } r_1 \leq r_2 \text{ then } P_{j,I}(r_1) \leq P_{j,I}(r_2). \tag{6.5}$$

The distribution function yields the probability of terminating on input I *within* resources r if a "blind", purely random, strategy is adopted in the choice of the next step when several are allowed. As argued above, this appears to be a rather natural way or measuring the "difficulty" of terminating within a given resource bound for a non-deterministic algorithm (i.e., a set of rules). If we are given a problem and a set of rules which does not uniquely determine a particular solution, we can measure the "difficulty" of finding that particular solution as the inverse of its probability in the random process generated by the rules. So the difficulty of finding a solution within bounded resources will be the inverse of the probability of the event consisting of all the possible solutions which can be obtained within the given resources. So, given a non-deterministic algorithm M_j, we define

$$H_j(I, r) \stackrel{\text{def}}{=} \frac{1}{P_{j,I}(r)} \tag{6.6}$$

and take this as a measure of the difficulty of finding a solution within resources r.

Note that for $r \to \infty$, the above measure yields an estimate of the "absolute" difficulty of finding a solution. This is still meaningful even when the problem is not mechanically solvable. So, anyone who is not interested in resource bounds, or believes that an exponential resource bound is acceptable—so that from her point of view the tautology problem is "easy"—can still apply this criterion meaningfully.

Thereby, we are now able to express the fact that, for instance, finding a short proof of the pigeon-hole principle may be hard in unrestricted Frege systems: the probability of generating such a proof simply applying the rules (axioms plus proof-rules) at random, without any control structure, may be vanishing. If we define a strategy—in our terminology a "determination"—which helps us generating these short proofs, it may very well be that the same strategy would not work for simple examples which are easily solved by resolution or other proof systems which are strictly "less powerful" from the point of view of the p-simulation relation.

Now that we are able to express the difficulty of finding solutions within bounded resources, we can express the *relative* difficulty of this problem for different non-deterministic algorithms.

6.3.6 *p*-emulation

Given two non-deterministic algorithms M_1 and M_2 it is interesting to compare the distribution functions $P_{1,I}(r)$ and $P_{2,I}(r)$ of the random variables associated with them. Suppose we can show that, given algorithms M_1 and M_2, there is a polynomial p such that for *every* input I, the distribution of the random variable r for M_2 is less than or equal to the distribution of $p(r)$ for M_1, in symbols:

$$\exists p, \forall I, \forall r, P_{1,I}(r) \leq P_{2,I}(p(r)). \tag{6.7}$$

What does (6.7) mean? It means that for every input I and every resource bound r, the non-deterministic algorithm M_2 (seen as a random algorithm) terminates within the resource bound $p(r)$ *at least as often* as M_1. This can be taken as a formal way of expressing the fact that *finding* a solution within the resource bound r is (essentially) *no harder* for M_2 than it is for M_1.[7] In terms of the H-measure defined above:

$$\exists p \forall I \forall r H_{2,I}(p(r)) \leq H_{1,I}(r).$$

Definition 6.3.4. We say that M_2 *p-emulates* M_1 if (6.7) holds.

[7] Of course, for all practical purposes, the relation is meaningful only when p is a polynomial of low degree.

The p-emulation relation is clearly a preorder so that its symmetric closure is an equivalence relation. Non-deterministic algorithms which p-emulate each other can be considered equivalent from the computational point of view. Notice that unrestricted Frege systems or the unrestricted sequent calculus with cut cannot p-emulate any of the "mechanical" proof systems, like the cut-free sequent calculus, tableaux or resolution, because of the unlimited number of "misleading" alternatives which are available at each step, so that the probability of stumbling upon the required short proof is vanishing: these systems do not have an "algorithmic nature".[8]. For the same reason, any of the algorithmic proof system can p-emulate Frege systems. In this way the p-emulation relation agrees with our intuitions about the relative efficiency of different formalizations from the point of view of algorithmic proof.

Note that M_2 p-simulates M_1 does not imply that M_2 p-emulates M_1, but only that there is a determination of M_2, say M_2^*, which p-emulates M_1; namely, that determination of M_2 which consists of the simulation of M_1. So if we prove that the converse does *not* hold, namely that M_1 does not p-simulate M_2 we have shown that no determination of M_1 can simulate M_2 but we have shown nothing about M_2^*: it may very well be that M_1 p-simulates M_2^* and the short M_2-computations which cannot be simulated in M_1 lie outside the scope of M_2^*.

Suppose, instead, that we can show that M_2 *p-emulates* M_1, but M_1 cannot p-emulate M_2. That is (6.7) holds and, moreover,:

$$\neg\exists p, \forall I, \forall r, P_{2,I}(r) \leq P_{1,I}(p(r)). \tag{6.8}$$

Then (6.7) and (6.8) taken together allow us to say that M_2 provides a *uniform* and *essential* improvement on M_1—especially if the polynomial p in (6.7) has the form cr for some small constant c. Note that in the limiting case of deterministic algorithms the relation is one of dominance: for all inputs, the time required by M_2 is bounded above by a fixed polynomial of the time required by M_1 on the same input, but not viceversa. This corresponds to the intuitive meaning of "M_2 is uniformly and essentially more efficient than M_1" in the case of deterministic algorithms. Our definition can be seen as a generalization of this intuitive idea to the case of non-deterministic algorithms.

Proposition 6.3.5. *If M_2 p-emulates M_1, then M_2 weakly p-simulates M_1.*

[8] The fact that the sample space is infinite is not, *per se*, the reason of this bad algorithmic behaviour. An algorithm with an infinite sample space in which most of the outcomes are "good" ones is perfectly conceivable. Again, it is not the *size* of the sample space which determines the good or bad computational behaviour of the algorithm, but the frequency of "good" choice sequences.

Proof. If M_2 p-emulates M_1, then for every input I and every integer r, $P_{2,I}(p(r)) \leq P_{1,I}(r)$. If there exists an M_1-computation on input I which terminates within r steps, then $P_{1,I}(r) > 0$ and therefore $P_{2,I}(p(r)) > 0$; that is, there exists an M_2-computation which terminates on input I within $p(r)$ steps. □

That the p-emulation relation is much more informative than the p-simulation relation is also shown by the following propositions.

Proposition 6.3.6. *If M_2 p-emulates M_1, then there is a polynomial p such that for all n*

1. $\mathcal{WB}_2(n) \leq p(\mathcal{WB}_1(n))$
2. $\mathcal{WE}_2(n) \leq p(\mathcal{WE}_1(n))$
3. $\mathcal{WW}_2(n) \leq p(\mathcal{WW}_1(n))$

Proof. Observe that 1 follows from the previous Proposition. To show 3 reason as follows. The negation of 3 is equivalent to

$$\forall p, \exists n, p(\mathcal{WW}_1(n)) < \mathcal{WW}_2(n).$$

Therefore, let N be a value of n satisfying the above condition and let $\mathcal{WW}_2(N) = r$. By definition of $\mathcal{WW}(n)$, for all inputs I with $|I| = N$, and all choice sequences $a \in \mathbf{\Omega}_{1,I}$, $|a| \leq r$. But there is an I, with $|I| = N$, such that $b > p(r)$ for some choice sequence $b \in \mathbf{\Omega}_{2,I}$. Therefore for such an I, $P_{1,I}(r) = 1$, but $P_{2,I}(p(r)) < 1$. Hence

$$\forall p, \exists I, \exists r, P_{1,I}(p(r)) < P_{2,I}(r),$$

that is M_2 cannot p-emulate M_1, against the hypothesis. Now, to show 2, consider that if M_2 p-emulates M_1, then there is a fixed polynomial function of r, say r^c, such that for all r and all inputs I:

$$P_{1,I}(r) \leq P_{2,I}(r^c),$$

and therefore

$$P_{1,I}(\sqrt[c]{r}) \leq P_{2,I}(r).$$

So,

$$1 - P_{2,I}(r) \leq 1 - P_{1,I}(\sqrt[c]{r}),$$

and

$$\begin{aligned}
\sum_{a \in \mathbf{\Omega}_{2,I}} pr_{2,I}(a) \cdot |a| &= \sum_{r=0}^{\infty} \rho_{2,I}(r) \cdot r \\
&= \sum_{r=0}^{\infty} 1 - P_{2,I}(r) \\
&\leq \sum_{r=0}^{\infty} 1 - P_{1,I}(\sqrt[c]{r}) \\
&\leq \sum_{r=0}^{\infty} \sum_{s > \sqrt[c]{r}} \rho_{1,I}(s) \\
&\leq \sum_{s=0}^{\infty} \sum_{r=0}^{s^c - 1} \rho_{1,I}(s) \\
&\leq \sum_{s=0}^{\infty} s^c \cdot \rho_{1,I}(s).
\end{aligned}$$

Since

$$\sum_s s \cdot \rho_{1,I}(s) \leq \sum_s s^c \cdot \rho_{1,I}(s) \leq (\sum_s s \cdot \rho_{1,I}(s))^c,$$

then for every input I the expected resource requirement in the sample space $\mathbf{\Omega}_{2,I}$ is bounded by n^c where n is the expected resource requirement in the sample space $\mathbf{\Omega}_{1,I}$. □

Proposition 6.3.7. *If M_2 p-emulates M_1, then there is a polynomial p such that for all n*

1. $\mathcal{EW}_2(n) \leq p(\mathcal{EW}_1(n))$
2. $\mathcal{EB}_2(n) \leq p(\mathcal{EB}_1(n))$
3. $\mathcal{EE}_2(n) \leq p(\mathcal{EE}_1(n))$

Proposition 6.3.8. *If M_2 p-emulates M_1, then there is a polynomial p such that for all n*

1. $\mathcal{BW}_2(n) \leq p(\mathcal{BW}_1(n))$
2. $\mathcal{BB}_2(n) \leq p(\mathcal{BB}_1(n))$
3. $\mathcal{BE}_2(n) \leq p(\mathcal{BE}_1(n))$

The proof of these propositions is similar to the proof of Proposition 6.3.6.

6.3.7 *p*-refinement

Propositions 6.3.6–6.3.8 above indicate that the p-emulation relation provides a much more complete picture of the relative efficiency of two non-deterministic algorithms than the p-simulation relation. As argued above, "M_2 p-emulates M_1 but not viceversa" is a good mathematical modelling of the informal notion of "M_2 is uniformly and essentially more efficient than M_1" when referred to non-deterministic algorithms. We can assert something even stronger. Suppose that M_2 p-emulates M_1 as before, but that M_1 cannot even weakly p-simulate M_2. This means that there are problems Π such that for every $I \in \Pi$ there is a *finite* probability that M_2 solves I within, say, $g(|I|)$ steps for some fixed function g, but for every polynomial p, the probability that M_2 solves I within $p(g(|I|))$ steps is null for almost all I. This leads to the following definitions:

Definition 6.3.9. A non-deterministic algorithm M_2 is *a p-refinement* of another non-deterministic algorithm M_1 if M_2 p-emulates M_1 but not viceversa. We also say that M_2 is a *strong* p-refinement of M_1 if M_2 p-emulates M_1 but M_1 cannot weakly p-simulate M_2.

However, this is not the whole story. We would like our efficiency estimates to be "stable". So, we would like to be able to show that the improvement of the distribution function of M_2 over the distribution function of M_1 is preserved under *every possible* determination of the latter algorithm. So our rational preference for M_2 does not risk to be overturned by further "heuristic" developments of M_1. This is a form of monotonicity. More precisely:

Definition 6.3.10. A non-deterministic algorithm M_2 is a *(strong) monotonic p-refinement* of a non-deterministic algorithm M_1 iff for every given determination M_1^* of M_1 there is a determination M_2^* of M_2 such that M_2^* is a (strong) p-refinement of M_1^*.

If M_2 is a monotonic p-refinement of M_1 according to this definition, then the improvement of M_2 on M_1 is stable and *cannot be reversed* by any further determination of the weaker algorithm. We can visualize this property by immagining an enumeration of all possible determinations of M_1:[9]

$$M_1, M_1^*, M_1^{**}, M_1^{***} \dots$$

[9] The reader with an interest in the Philosophy of Science may notice the analogy between this way of representing the p-refinement relation and Lakatos' notion of a research programme that *supersedes* another research programme in his Methodology of Scientific Research Programmes [108]. In this context, the proof systems are *rival research programmes* and the sequence of M_2-determinations is *a progressive shift*.

If M_2 is a (strong) p-refinement of M_1, there is a corresponding sequence of determinations of M_2

$$M_2, M_2^*, M_2^{**}, M_2^{***} \dots$$

such that

$$M_2^{\overbrace{*,\dots,*}^{n}} \text{ is a (strong) } p\text{-refinement of } M_1^{\overbrace{*,\dots,*}^{n}}$$

for all $n \geq 0$. Monotonic p-refinement is, like p-refinement, a strict partial order of proof systems and can be employed to make appraisals of the relative efficiency of proof systems that are more *relevant* to the problem of mechanical proof than the appraisals based on the p-simulation relation, and are also more *stable* than the appraisals based on specific deterministic algorithms.

The problem arises of how to show that an algorithm is a monotonic p-refinement of another. The following lemma provides a sufficient condition for *strong* p-refinement.

Lemma 6.3.11. *Let M_1 be a non-deterministic algorithm and let $\mathbf{\Omega}_{1,I}$ denote its sample space on input I. The non-deterministic algorithm M_2, with sample space $\mathbf{\Omega}_{2,I}$, is a strong monotonic p-refinement of M_1 if there is a simulation function $f :\downarrow \mathbf{\Omega}_{1,I} \mapsto 2^{\downarrow \mathbf{\Omega}_{2,I}}$ (see above Sec. 6.3.3), computable in polynomial time, such that the following conditions are satisfied:*

1. *The function f is a p-simulation, i.e. there is a polynomial p such that*

$$(\forall I)(\forall a \in \mathbf{\Omega}_{1,I})(\forall b \in f(a)) \ |b| \leq p(|a|).$$

2. *There is a problem Π, such that for all polynomials p*

$$(\exists I \in \Pi)(\forall a \in \mathbf{\Omega}_{1,I})(\exists b \in f(a)) \ |a| > p(|b|).$$

3. *The function f is a discriminating simulation (see above, Sec. 6.3.3), i.e.,*

$$\textit{If } a \not\approx_f b \textit{ then } f(a) \cap f(b) = \emptyset.$$

Proof. If conditions 1–3 are fulfilled, then, for every determination M_1^* of M_1, there is a determination M_2^* of M_2 such that M_2^* p-emulates M_1^* but M_1^* cannot weakly p-simulate M_2^*. For, let M_1^* be an arbitrary determination of M_1. Let $M_{f(1)}^*$ be the determination of M_2 consisting in the simulation of M_1^* via f (see above page 229), so that

$$\mathbf{\Omega}_{f(1)^*,I} = \bigcup_{a \in \mathbf{\Omega}_{1^*,I}} f(a).$$

Clearly,

$$pr(a) \leq pr(\|a\|_f),$$

where $\|a\|_f$ is the equivalence class of a under $\simeq_f$ defined as in (6.1). Since

$$pr(a) \leq pr(f(a)),$$

It follows from Proposition 6.3.2 that, for all r:

$$pr\{a : |a| \leq r\} \leq pr\{f(a) : |a| \leq r\}.$$

Since, again by Proposition 6.3.2,

$$pr\{f(a) : |a| \leq r\} = pr \bigcup_{|a| \leq r} f(a),$$

then, by condition 1,

$$pr\{a : |a| \leq r\} \leq pr\{b : |b| \leq p(r)\},$$

for the fixed polynomial p. So

$$\sum_{s \leq r} \rho_{1^*,I}(s) \leq \sum_{s \leq p(r)} \rho_{f(1)^*,I}(s),$$

therefore for all r

$$P_{1^*,I}(r) \leq P_{f(1)^*,I}(p(r)).$$

This shows that $M^*_{f(1)}$ p-emulates M^*_1. Now, it follows from condition 2 that there is problem Π such that for all polynomials p there is an $I \in \Pi$ such that for all choice sequences a in the sample space $\mathbf{\Omega}_{1^*,I} \subset \mathbf{\Omega}_{1,I}$, there exists a choice sequence b in the set $f(a)$ such that $p(|b|) < |a|$. This means that $M^*_{f(1)}$ cannot weakly p-simulate M^*_1. Therefore, for every determination M_{1^*} of M_1 there is a determination M_{2^*} of M_2, namely $M^*_{f(1)}$, such that M_{2^*} p-emulates M_{1^*} but M^*_1 cannot weakly p-simulate M^*_2 (M_{2^*} is a strong p-refinement of M_{1^*}), that is M_2 is a strong monotonic p-refinement of M_1. □

6.4 A case study: *KE* vs. Smullyan's tableaux

In this section we show that *KE* is a strong monotonic p-refinement of Smullyan's (binary) tableaux [138]. We first define a p-simulation of such a tableau method in *KE*; then we show that all conditions in Lemma 6.3.11 are satisfied. In this context, each choice sequence in $\downarrow \mathbf{\Omega}_{TB,I}$, the sample space of the tableau method, identifies a tableau (a completed tableau) which is mapped by the simulation function to a finite set of *KE*-trees (completed *KE*-trees).

With the notation

$$\begin{array}{c}\mathcal{T}\\ x\end{array}$$

we denote a tree such that φ is among its leaves.

We consider a tableau as a tree with a labelling function l mapping the nodes of $\mathcal{T}$ to formulae. Below we define simultaneously the relation $\mathcal{T}'$ *is a KE-simulation of* $\mathcal{T}$, and a *partial* function v which for every pair $<\mathcal{T},\mathcal{T}'>$ in the relation, maps the labelled nodes of $\mathcal{T}$ to labelled nodes of $\mathcal{T}'$ (v is one-one). So, the simulation function f will simply be defined as

$$f(\mathcal{T}) = \{\mathcal{T}' :\ \mathcal{T}' \text{ is a } KE\text{-simulation of } \mathcal{T}\}$$

Definition 6.4.1. Let $\mathcal{T}$ be a tableau. The relation $\mathcal{T}'$ *is a KE-simulation of* $\mathcal{T}$ and, for every such $\mathcal{T}'$, the partial function v from nodes of $\mathcal{T}$ to nodes of $\mathcal{T}'$ are defined by induction on the number of nodes of $\mathcal{T}$:

A. For every one-node tableau $\mathcal{T}$, with x as its only labelled node, the *KE*-tree $\mathcal{T}'$ with y as its only labelled node and $l(x) = l(y)$ is a *KE*-simulation of $\mathcal{T}$ and $v(x) = y$.

B. Let $\mathcal{T}$ be a tableau of the form

$$\begin{array}{c}\mathcal{T}_0\\ x\\ y_1(\alpha_1)\\ y_2(\alpha_2)\end{array}$$

for some α such that $l(u) = \alpha$ and u is on the path to x, and let $\mathcal{T}_0'$ be a *KE*-simulation of $\mathcal{T}_0$. Then,

1. if $v(x)$ is defined, the following is a *KE*-simulation of $\mathcal{T}$

$$\begin{array}{c}\mathcal{T}_0'\\ v(x)\\ z_1(\alpha_1)\\ z_2(\alpha_2)\end{array}$$

   and $v(y_i) = z_i$, $i = 1, 2$;

2. if $v(x)$ is undefined, then $\mathcal{T}_0'$ is a *KE*-simulation of $\mathcal{T}$.

C. Let $\mathcal{T}$ be a tableau of the form

$$\begin{array}{c} \mathcal{T}_0 \\ x \\ \hline y_1(\beta_1) \mid y_2(\beta_2) \end{array}$$

for some β such that $l(u) = \beta$ and u is on the path to x. Let $\mathcal{T}_0'$ be a *KE*-simulation of $\mathcal{T}_0$. Then,

1. if $v(x)$ is defined and u, with $l(u) = \beta_i'$, is on the path to $v(x)$ in $\mathcal{T}_0'$, then the following is a *KE*-simulation of $\mathcal{T}$

$$\begin{array}{c} \mathcal{T}_0' \\ v(x) \\ z(\beta_j) \end{array}$$

   where j is equal to 1 if $i = 2$ and to 2 if $i = 1$; moreover $v(y_j) = z$ and $v(y_i)$ is undefined;

2. if $v(x)$ is defined and u, with $l(u) = \beta_i$ $(i = 1, 2)$, is on the path to $v(x)$ in $\mathcal{T}_0'$, then $\mathcal{T}_0'$ is a *KE*-simulation of $\mathcal{T}$;

3. if $v(x)$ is defined and, for all $i(= 1, 2)$ and all u on the path to $v(x)$, $l(u) \neq \beta_i'$ and $l(u) \neq \beta_i$, then both the following trees are *KE*-simulations of $\mathcal{T}$:

$$\begin{array}{c} \mathcal{T}_0' \\ v(x) \\ \hline \begin{array}{c|c} z_1(\beta_1) & s(\beta_1') \\ & z_2(\beta_2) \end{array} \end{array} \qquad \begin{array}{c} \mathcal{T}_0' \\ v(x) \\ \hline \begin{array}{c|c} s(\beta_2') & z_2(\beta_2) \\ z_1(\beta_1) & \end{array} \end{array}$$

   and $v(y_i) = z_i$, $i = 1, 2$;

4. if $v(x)$ is undefined, then $\mathcal{T}_0'$ is a *KE*-simulation of $\mathcal{T}$.

This concludes the definition.

Remark 24. The simulation procedure can be easily adapted to the case in which disjunctions with more than two components are allowed in the language. So the procedure can deal with trees of clauses.

Let f be the function mapping every tableau $\mathcal{T}$ to the (finite) set of *KE*-simulations of $\mathcal{T}$. It is easy to show that:

Proposition 6.4.2. *If $\mathcal{T}$ is a closed tableau, then every $\mathcal{T}'$ in $f(\mathcal{T})$ is also closed. Moreover, by construction, every $\mathcal{T}' \in f(\mathcal{T})$ enjoys the subformula property.*

Note that it follows immediately from the above proposition that *KE* is complete and enjoys the subformula property.

Theorem 6.4.3. *KE is a strong monotonic p-refinement of Smullyan's tableaux.*

Proof. Let f be the simulation function defined above and let $\sharp(\mathcal{T})$ be the number of branches in the tree $\mathcal{T}$. We can consider the "crude" complexity measure $\sharp(\mathcal{T})$ as adequate in this context, since each branch is assumed to contain only subformulae of the assumption nodes (and only one occurrence of each). Hence any more precise measure is bounded above by a polynomial in our crude measure. It is easy to see that for all $\mathcal{T}$ and for all $\mathcal{T}' \in f(\mathcal{T})$:

$$\sharp(\mathcal{T}') \leq \sharp(\mathcal{T}). \tag{6.9}$$

This implies that condition 1 in Lemma 6.3.11 is satisfied.

Let H_k^c be the class of "truly fat" expressions recalled in Subsection 3.3.2. It is not difficult to verify that if $\mathcal{T}$ is a closed tableau for H_k^c, and f is the simulation function defined above, then for *every KE*-tree $\mathcal{T}'$ in $f(\mathcal{T})$:

$$\sharp(\mathcal{T}') = 2^{k-1}. \tag{6.10}$$

Given that the *minimal* closed tableau for H_k^c must have a factorial number of branches (Theorem 3.3.19), condition 2 in Lemma 6.3.11 is also satisfied.

Finally, by inspection of the definition of the simulation f, it is easy to verify that condition 3 of Lemma 6.3.11 is satisfied. □

Remark 25. In fact, (6.10) and Theorem 3.3.19 taken together are much *stronger* than what we need to satisfy condition 2: for all polynomials p and for all tableaux $\mathcal{T}$ for H_k^c, not only *there is* a *KE*-tree $\mathcal{T}'$ in $f(\mathcal{T})$ such that $\sharp(\mathcal{T}') < p(\sharp(\mathcal{T}))$, but the latter holds *for all* $\mathcal{T}'$ in $f(\mathcal{T})$. To satisfy condition 2, Cook and Rechow's examples H_n, recalled in Subsection 3.3.2, would be sufficient. As observed in that Subsection, while in the case of the examples H_k^c the size of the resulting refutation is not affected by the choice of the atom to which the branching rule *PB* is applied, in the case of the H_n examples wrong choices can increase the size of the refutation trees up to an exponential factor. However, as we recalled also there, D'Agostino and

Mondadori [65] suggested a simple heuristic principle that is sufficient to overcome the problem as far as input in clausal form is concerned. The study of more general criteria for input in non-clausal form is a topic we consider for further investigation. In the context of this Chapter, however, it is worth remarking that *even without any additional heuristics* there is a very low probability that a randomized *KE*-procedure generates a tree whose number of branches is as big as that of the shortest tableau for this examples. Such a *KE*-tree is obtained only when the atom with the least number of occurrences in the branch is *always* chosen. Since the probability of choosing any particular atom among n atoms is $\frac{1}{n}$ and there are $2^n - 1$ atoms in H_n, it is easy to see that the probability of generating a refutation tree whose number of branches is as big as the number of branches in the smallest tableau is $\frac{1}{(2^n-1)!}$. All the other possible *KE*-trees have a strictly smaller number of branches.

The simulation function f can be seen as an *optimization* technique: given a tableau-based *deterministic* algorithm M_{TB} we can turn it into a *KE*-based algorithm M_{KE} which is uniformly and essentially more efficient. In terms of p-simulation, our results show that no deterministic M_{TB} can p-simulate its *KE*-optimization. The reader can verify how tableaux "shrink", when they are processed by the simulation function: in most cases the *KE*-tree obtained via the simulation closes (or becomes completed) much more quickly than the simulated tableaux. Since the simulation is a step-by-step one, our results can be read in the following more pictorial way. Let M_{TB} be *any* deterministic algorithm based on the tableau method. The simulation function maps every instruction of M_{TB} to another instruction which expands a *KE*-tree. For the case 3 of the definition, we can imagine that one of the two alternative simulations is chosen at random. The algorithm M_{KE} consisting of the composition of M_{TB} and the simulation function (plus the random mechanism) is a uniform and essential improvement of the original algorithm M_{TB}; i.e., M_{KE} always terminates *within* approximately the same number of steps as M_{TB}, but often terminates before and sometimes (very) long before. For input in clausal form, M_{KE} has in fact a much better worst-case complexity than M_{TB}: $O(2^k)$ vs. $O(k!)$.

Now, let KE_f be the determination of *KE* obtained by simulating the tableau method, where f is the simulation function. Then, it is not difficult to see that KE_f linearly emulates (and so linearly simulates, as shown in [65]) the main refinements of Smullyan's tableaux; namely, merging and "lemma generation". So, a certain restricted use of the rule *PB* is sufficient to simulate and emulate both refinements. On the other hand, clearly, more liberal uses of *PB* yield strictly more powerful versions of *KE* which may well be more powerful from the complexity viewpoint and still obey the subformula property. As pointed out in [65], an interesting question is whether more liberal uses of the rule would lift *KE* to a better complexity class in terms of the p-simulation relation, and we add here that the same question is interesting in terms of the p-emulation relation. As mentioned in [65], an answer to this question is presently available only for the totally unrestricted version of *PB*, which yields a version of *KE* equivalent to the sequent calculus with cut and to unrestricted Frege systems. However, obviously, the complete lack of control resulting from such an unrestricted use would render the corresponding version of *KE* useless in practice; that is, the unrestricted version would not p-emulate the restricted versions. Hence, such a version does not provide any improvement from the point of view of automated deduction. Nonetheless, it is still an open question whether between the totally unrestricted version and the severely restricted one employed to obtain the simulation results recalled in Section 3.3.1, there is room for a more liberal, yet "controlled", version which is strictly more powerful in terms of both the p-simulation and the p-emulation relations.

Chapter 7

Conclusions and future work

We reassessed and extended the depth-bounded approach to **CPL** introduced by D'Agostino and co-authors. Part of our reassessment of the proof-theoretic basis of the approach—constituted by a *KE*/*KI* system that we called the *intelim method*—led us to prove new lower bounds on analytic tableaux. Namely, we introduced a class of examples in the pure disjunction-conjunction fragment of the language, and proved a factorial lower bound on that class of examples for all tableau methods sharing the $\wedge$ and $\vee$ rules with classical tableaux. On the other hand, our new examples are easy for *KE*-style (and so *KE*/*KI*-style) variants of those tableau systems. Moreover, we stated factorial lower bound for the strongest possible version of clausal tableaux on the class of "truly fat" expressions used in [54] to state an analogous factorial lower bound for simple clausal tableaux (by contrast, the "truly fat" expressions are easy for truth-tables and *KE*). The latter settled a problem left open in the literature [115, 116, 9]. Another result of our reassessment is that, following a suggestion in [65], we explored a hierarchy of depth-bounded approximations to **CPL** based only on *KE*. Although arguably less natural than the analogous hierarchy based on the intelim method, the hierarchy based on *KE* might be preferred for potential uses in automated reasoning.

We also showed how the depth-bounded approach can be naturally extended to useful non-classical logics such as **FDE**, **LP**, $\mathbf{K_3}$ and **IPL**. In each case, the basis for the extension was the introduction of a corresponding *KE*/*KI*-style system. Each of these systems: (i) is formulated in terms of signed formulae, where the signs have an intuitive informational interpretation; (ii) has linear introduction and elimination (intelim) rules, which fix the meaning of the connectives; (iii) has branching structural rule(s) expressing a generalized rule of bivalence; (iv) can be used as both a direct-proof and a refutation method; (v) obeys the subformula property. Given our new lower bounds on analytic tableaux, these *KE*/*KI*-style systems are interesting independently of the depth-bounded approach since they have an exponential speed-up on their tableau systems counterparts.

Then, we focused on showing that each of our *KE*/*KI*-style systems naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations to the corresponding logic, in terms of the maximum number of nested applications that are allowed of the branching rule(s).[1] The latter is (are) essentially cut rule(s) which intuitively govern(s) the manipulation of virtual information, as opposed to the operational rules that intuitively govern the use of actual information. As in the classical case, the key intuition is that the more virtual information needs to be invoked via the branching rule(s), the harder the inference is for the agent, from both the cognitive and computational viewpoints. Thus, the nested application of those rules provides a sensible measure of inferential depth, and so the levels of the corresponding hierarchy can be naturally related to the inferential power of agents. Furthermore, we showed that, in the case of the many-valued logics, each hierarchy admits of a 5-valued non-deterministic semantics. As for the case of **IPL**, we paved the way for a non-deterministic semantics suitable for the corresponding hierarchy by providing an alternative 3-valued non-deterministic semantics for full **IPL** which specifies the meaning of the connectives without appealing to any "structural" condition.

Finally, we presented what we consider is a methodological enhancement for comparing the relative complexity of proof systems. Namely, we proposed a refinement of the p-simulation relation; refinement that is adequate to establish positive results about the superiority of a proof system over another with respect to proof-search. We tested our refinement with a case study; viz., we showed the superiority of *KE* over Smullyan's (binary) tableaux.

We consider that our new lower bounds on analytic tableaux, our hierarchies of tractable depth-bounded approximations to classical and non-classical logics, as well as our refinement of the p-simulation relation, can be of interest for researchers in Computer Science, Artificial Intelligence, Philosophy and Cognitive Science, to name some. The Thesis contribution is added to recent and ongoing work showing that the depth-bounded approach is flexible enough to cover a variety of reasoning phenomena in a natural and fruitful way, accounting for the computational and cognitive cost of those phenomena as performed by situated resource-bounded agents. In this sense, the approach seems to comply with the virtues that we suggested (in Chapter 1) one may expect from an informational approach to Logic.

[1]While in the case of the many-valued logics tractability of the approximations is proven, in the case of **IPL** it is (for now) only conjectured.

Some future directions of research suggest themselves and others have been pointed out by the external reviewers. The trio of many-valued logics (closely related to each other) addressed in Chapter 4 provides a case study for extending the depth-bounded framework to a variety of finite-valued logics, in the spirit of [43, 94, 42]. In fact, [42] already paves the way for carrying out such an extension since it introduces a general method for extracting *analytic cut-based* tableaux for any finite-valued logic. Given this general method, defining hierarchies of depth-bounded approximations to any finite-valued logic suggests itself as a natural next step in our research. Further, in the same [42] a generalized notion of analyticity—different to the f-analyticity of our framework—is proposed. When extending the depth-bounded approach to finite-valued logics we shall test whether the generalized notion in [42] is better suited than f-analyticity for such a extension.[2].

Moreover, the hierarchy of approximations to **IPL** that we defined in Chapter 5 paves the way for defining similar hierarchies for a wide variety of logics characterized by Kripke-style semantics. Even further, it seems plausible to extend the depth-bounded approach to relevant and linear logics, and even to non-monotonic reasoning and paradigmatic phenomena of logical dynamics. The latter two extensions would imply: (i) a switch from hard (knowledge-like) information—which is the one present in all the logics hitherto covered by the approach—to soft (belief-like) information [see 5, 113, 152]; (ii) considering the single agent case as the limit of the multi-agent case, and so not considering the latter as a mere extension of the former. Intuitively, the way agents access or fail to access certain information is tied to the distributed nature of information. In the real world, agents cannot access all information at once and, thus, using information in inference often involves communication and, more generally, information flow between agents. Anyway, the adaptation of the approach to multi-agent settings seems natural: the focus would be on distributed systems, possibly assigning different inferential powers to different agents [see 46].

Thereby, we envisage extensions of the depth-bounded framework covering an ample variety of non-classical logics. What is more, hierarchies of depth-bounded approximations to these logics based only on the "*KE*-fragment" of the corresponding *KE*/*KI*-style systems seem also worth of further research. As in the classical case, these hierarchies may be preferred for potential uses in automated reasoning.

Last but not least, among other useful and interesting remarks, João Marcos has pointed out that the "imprecise truth-values" in our approach to many-valued logics—associated to the signs of our respective proof systems—seem to be related to the designated/anti-designated "cognitive attitudes" used in [38, 91] when advocat-

[2] Another generalized notion of analitycity which seems worth exploring when further extending the depth-bounded framework is the one introduced in [91]

ing a two-dimensional approach to consequence. However, the latter approach allows one to dispense with signs in the formulation of the corresponding proof systems, and allows also for a perspective on logical pluralism which is different to that outlined in our Chapter 1. The key difference between those perspectives is that while ours crucially relies on pragmatic/extra-logical criteria (viz., on the purpose of the agents' modelling), the perspective arising from the two-dimensional approach vindicates "a variety of *logical pluralism* in which logics of different kinds may be said to 'cohabitate' the same generalized logical structure" [38, p. 258]. So, this point raised by Marcos seems worth exploring, for choosing between our approach in terms of signs or the two-dimensional approach when implementing the depth-bounded framework would clearly yield different results, both technically and conceptually. We shall deal with this point in future work.

List of Figures

List of Tables

Bibliography

[1] A. Aliseda. *Abductive Reasoning: Logical Investigations into Discovery and Explanation*, volume 330 of *Synthese Library*. Springer, 2006.

[2] A. Aliseda. *La lógica como Herramienta de la Razón. Razonamiento Ampliativo en la Creatividad, la Cognición y la Inferencia.* College Publications. Milton Keynes, Reino Unido, 2014.

[3] P. Allo. Information and logical discrimination. In *Conference on Computability in Europe*, pages 17–28. Springer, 2012.

[4] P. Allo. The logic of information. In *The Routledge Handbook of Philosophy of Information*, pages 75–92. Routledge, 2016.

[5] P. Allo. Hard and soft logical information. *Journal of Logic and Computation*, 27(8):2505–2524, 2017.

[6] P. Allo and E. Mares. Informational semantics as a third alternative? *Erkenntnis*, 77(2):167–185, 2012.

[7] A. Anderson and N. Belnap. Tautological entailments. *Philosophical studies*, 13(1-2):9–24, 1962.

[8] A. Anderson and N. Belnap. *Entailment: The Logic of Relevance and Neccessity, Vol. I.* Princeton University Press, 1975.

[9] N. Arai, T. Pitassi, and A. Urquhart. The complexity of analytic tableaux. *The Journal of Symbolic Logic*, 71(3):777–790, 2006.

[10] O. Arieli and M. Denecker. Reducing preferential paraconsistent reasoning to classical entailment. *Journal of Logic and Computation*, 13(4):557–580, 2003.

[11] S. Artemov and R. Kuznets. Logical omniscience via proof complexity. In *International Workshop on Computer Science Logic*, pages 135–149. Springer, 2006.

[12] S. Artemov and R. Kuznets. Logical omniscience as a computational complexity problem. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 14–23, 2009.

[13] S. Artemov and R. Kuznets. Logical omniscience as infeasibility. *Annals of pure and applied logic*, 165(1):6–25, 2014.

[14] F. Asenjo. A calculus of antinomies. *Notre Dame Journal of Formal Logic*, 7(1):103–105, 1966.

[15] A. Avron. Tableaux with four signs as a unified framework. In *Automated Reasoning with Analytic Tableaux and Related Methods, Int. Conf., TABLEAUX 2003*, volume 2796 of *Lecture Notes in Computer Science*, pages 4–16. Springer, 2003.

[16] A. Avron. A non-deterministic view on non-classical negations. *Studia Logica*, 80(2-3):159–194, 2005.

[17] A. Avron. Non-deterministic semantics for families of paraconsistent logics. *Handbook of Paraconsistency*, 9:285–320, 2007.

[18] A. Avron. Multi-valued semantics: Why and how. *Studia Logica*, 92(2):163–182, 2009.

[19] A. Avron, O. Arieli, and A. Zamansky. *Theory of effective propositional paraconsistent logics.* College Publications, 2018.

[20] A. Avron and I. Lev. Canonical Propositional Gentzen-Type Systems. In *Proceedings of the 1st International Joint Conference on Automated Reasoning, IJCAR*, volume 2083 of *Lecture Notes in Artificial Inteligence*, pages 529–544. Springer, 2001.

[21] A. Avron and I. Lev. Non-deterministic multiple-valued structures. *Journal of Logic and Computation*, 15(3):241–261, 2005.

[22] A. Avron and A. Zamansky. Non-deterministic semantics for logical systems. In *Handbook of philosophical logic*, volume 16, pages 227–304. Springer, 2011.

[23] A. Ayer. *Language, truth and logic*. London: Gollancz, 1936.

[24] P. Baldi, F. D'Asaro, and G. Primiero. Introducing k-lingo: a k-depth bounded version of ASP system Clingo. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 18, pages 661–665, 2021.

[25] P. Baldi and H. Hosni. Depth-bounded belief functions. *International Journal of Approximate Reasoning*, 123:26–40, 2020.

[26] P. Baldi and H. Hosni. Logical approximations of qualitative probability. In *ISIPTA*, pages 12–21. PMLR, 2021.

[27] P. Baldi and H. Hosni. A logic-based tractable approximation of probability. *Journal of Logic and Computation*, in press.

[28] A. Barceló-Aspeitia. Words and images in argumentation. *Argumentation*, 26(3):355–368, 2012.

[29] J. Barwise and J. Perry. Situations and attitudes. *The Journal of Philosophy*, 78(11):668–691, 1981.

[30] N. Belnap. How a computer should think. In *Contemporary Aspects of Philosophy*, pages 30–55. Oriel Press, 1977.

[31] N. Belnap. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logics*, pages 7–37. Reidel Publishing Company, 1977.

[32] N. Belnap. *A Formalization of Entailment*. PhD thesis, Yale University, Pisa, 1959.

[33] E. Beth. Semantic construction of intuitionistic logic. *Koninklijke Nederlandse Akad. von Wettenscappen*, 19(11):357–388, 1956.

[34] G. Bezhanishvili and W. Holliday. A semantic hierarchy for intuitionistic logic. *Indagationes Mathematicae*, 30(3):403–469, 2019.

[35] M. Bibel. *Automated Theorem Proving*. Friedr. Vieweg Sohn, Braunschweig, Wiesbaden, second edition, 1987.

[36] C. Blasio. Up the hill: on the notion of information in logics based on the four-valued bilattice. *LSFA 2015 GeTFun 3.0 FILOMENA 2 LFIs^15 TRS Reasoning School*, page 99, 2015.

[37] C. Blasio. Revisitando a lógica de Dunn-Belnap. *Manuscrito*, 40:99–126, 2017.

[38] C. Blasio, J. Marcos, and H. Wansing. An inferentially many-valued two-dimensional notion of entailment. *Bulletin of the Section of Logic*, 46(3/4):233–262, 2017.

[39] A. Bloesch. A tableau style proof system for two paraconsistent logics. *Notre Dame Journal of Formal Logic*, 34(2):295–301, 1993.

[40] S. Buss. Polynomial size proofs of the pigeon-hole principle. *The Journal of Symbolic Logic*, 52:916–927, 1987.

[41] M. Cadoli and M. Schaerf. Approximate entailment. In *Congress of the Italian Association for Artificial Intelligence*, pages 68–77. Springer, 1991.

[42] C. Caleiro, J. Marcos, and M. Volpe. Bivalent semantics, generalized compositionality and analytic classic-like tableaux for finite-valued logics. *Theoretical Computer Science*, 603:84–110, 2015.

[43] W. Carnielli. Systematization of finite many-valued logics through the method of tableaux. *The Journal of Symbolic Logic*, 52(2):473–493, 1987.

[44] C. Cherniak. *Minimal rationality*. Mit Press, 1986.

[45] A. Church. A note on the entscheidungsproblem. *The Journal of Symbolic Logic*, 1(1):40–41, 1936.

[46] G. Cignarale and G. Primiero. A multi-agent depth bounded Boolean logic. In *International Conference on Software Engineering and Formal Methods*, pages 176–191. Springer, 2020.

[47] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[48] S. Cook. The P versus NP problem. *The millennium prize problems*, pages 87–104, 2006.

[49] S. Cook and R. Reckhow. On the lengths of proofs in the propositional calculus, preliminary version. In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing*, pages 135–148, 1974.

[50] S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

[51] J. Crawford and D. Etherington. A non-deterministic semantics for tractable inference. In *AAAI/IAAI*, pages 286–291, 1998.

[52] M. D'Agostino. Tractable depth-bounded logics and the problem of logical omniscience. In *Probability, Uncertainty and Rationality*, pages 245–275. CRM series. Springer.

[53] M. D'Agostino. *Investigations into the Complexity of some Propositional Calculi*. Oxford University. Computing Laboratory. Programming Research Group, 1990.

[54] M. D'Agostino. Are tableaux an improvement on truth-tables? *Journal of Logic, Language and Information*, 1(3):235–252, 1992.

[55] M. D'Agostino. Classical natural deduction. In *We Will Show Them! (1)*, pages 429–468. College Publications, 2005.

[56] M. D'Agostino. Depth-bounded logic for realistic agents. *L&PS–Logic & Philosophy of Science*, 11(1):3–57, 2013.

[57] M. D'Agostino. Analytic inference and the informational meaning of the logical operators. *Logique et Analyse*, 57(227):407–437, 2014.

[58] M. D'Agostino. An informational view of classical logic. *Theoretical Computer Science*, 606:79–97, 2015.

[59] M. D'Agostino, M. Finger, and D. Gabbay. Semantics and proof-theory of depth bounded Boolean logics. *Theoretical Computer Science*, 480:43–68, 2013.

[60] M. D'Agostino, T. Flaminio, and H. Hosni. Rational beliefs real agents can have–a logical point of view. In *Imperfect Decision Makers: Admitting Real-World Rationality*, pages 97–109, 2017.

[61] M. D'Agostino and L. Floridi. The enduring scandal of deduction. *Synthese*, 167(2):271–315, 2009.

[62] M. D'Agostino, D. Gabbay, and K. Broda. Tableau methods for substructural logics. In *Handbook of Tableau Methods*, pages 397–467. Springer, 1999.

[63] M. D'Agostino, D. Gabbay, and S. Modgil. Normality, non-contamination and logical depth in classical natural deduction. *Studia Logica*, 108(2):291–357, 2020.

[64] M. D'Agostino, C. Larese, and S. Modgil. Towards depth-bounded natural deduction for classical first-order logic. *Journal of Applied Logics*, 2631(2):423, 2021.

[65] M. D'Agostino and M. Mondadori. The taming of the cut. Classical refutations with analytic cut. *Journal of Logic and Computation*, 4(3):285–319, 1994.

[66] M. Dalal. Anytime families of tractable propositional reasoners. In *Annals of Mathematics and Artificial Intelligence 22*, pages 297–318, 1998.

[67] M. Dummett. Elements of intuitionism, reprinted (1985) edn., series, 1977.

[68] M. Dummett. Truth from the constructive standpoint. *Theoria*, 64(2-3):122–138, 1998.

[69] J. Dunn. Information in computer science. In *Handbook of the Philosophy of Information*, pages 581–608. Elsevier.

[70] J. Dunn. Intuitive semantics for first-degree entailments and 'coupled trees'. *Philosophical studies*, 29(3):149–168, 1976.

[71] J. Dunn. The concept of information and the development of modern logic. In *Zwischen traditioneller und moderner Logik: Nichtklassiche Ansätze*, pages 423–447. Mentis Verlag GmbH, 2001.

[72] J. Dunn. Two, three, four, infinity: The path to the four-valued logic and beyond. In *New Essays on Belnap-Dunn Logic*, pages 77–97. Springer, 2019.

[73] L. Estrada-González. The Bochum plan and the foundations of contra-classical logics. *CLE e-Prints*, 19(1):00–00, 2020.

[74] R. Fagin and J. Halpern. Belief, awareness, and limited reasoning. *Artificial intelligence*, 34(1):39–76, 1987.

[75] M. Finger. Polynomial approximations of full propositional logic via limited bivalence. In *European Workshop on Logics in Artificial Intelligence*, pages 526–538. Springer, 2004.

[76] M. Finger and D. Gabbay. Cut and pay. *Journal of Logic, Language and Information*, 15(3):195–218, 2006.

[77] M. Finger and R. Wasserman. Approximate and limited reasoning: Semantics, proof theory, expressivity and control. *Journal of Logic and Computation*, 14(2):179–204, 2004.

[78] M. Finger and R. Wassermann. The universe of propositional approximations. *Theoretical Computer Science*, 355(2):153–166, 2006.

[79] M. Fitting. *Intuitionistic Logic Model Theory and Forcing.* North-Holland Publishing Co., Amsterdam, 1969.

[80] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 34. Synthese Library, D. Reidel Publishing, Dordrecht, 1983.

[81] M. Fitting. Bilattices and the semantics of logic programming. *The Journal of Logic Programming*, 11(2):91–116, 1991.

[82] M. Fitting. Kleene's three valued logics and their children. *Fundamenta informaticae*, 20(1, 2, 3):113–131, 1994.

[83] M. Fitting, Primary Course Notes: First Degree Entailment, Continued. The Graduate Center, City University of New York, April, 2018.

[84] L. Floridi. The logic of being informed. *Logique et analyse*, 49(196):433–460, 2006.

[85] D. Gabbay. What is a logical system? In *What is a Logical System?*, pages 179–216. Oxford University Press, 1994.

[86] D. Gabbay. What is a logical system? An evolutionary view: 1964–2014. In *Computational Logic. Handbook of the History of Logic*, volume 9, pages 41–132. North-Holland, 2014.

[87] D. Gabbay and J. Woods. The new logic. *Logic Journal of the IGPL*, 9(2):141–174, 2001.

[88] D. Gabbay and J. Woods. Fallacies as cognitive virtues. In *Games: unifying logic, language, and philosophy*, pages 57–98. Springer, 2009.

[89] M. Garey and D. Johnson. *Computers and Intractability. A Guide to the theory of NP-Completeness.* W.H. Freeman & Co., San Francisco, 1979.

[90] G. Gentzen. Untersuchungen über das logische schließen. *Mathematische zeitschrift*, 39(1):176–210, 1935. English translation in Szabo, M., (ed.), *The Collected Papers of Gerhard Gentzen*. North-Holland, Amsterdam, 1969.

[91] V. Greati, S. Marcelino, and J. Marcos. Proof search on bilateralist judgments over non-deterministic semantics. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 129–146. Springer, 2021.

[92] S. Haack. Analyticity and logical truth in The Roots of Reference. *Theoria*, 43(2):129–143, 1977.

[93] R. Hähnle. Towards an efficient tableau proof procedure for multiple-valued logics. In *International Workshop on Computer Science Logic*, pages 248–260. Springer, 1990.

[94] R. Hähnle. Tableaux for many-valued logics. In *Handbook of Tableau Methods*, pages 529–580. Springer, 1999.

[95] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[96] A. Heyting. Die formalen regeln der intuitionistischen logik. *Sitzungsbericht PreuBische Akademie der Wissenschaften Berlin, physikalisch-mathematische Klasse II*, pages 42–56, 1930.

[97] J. Hintikka. Language-games for quantifiers. In Nicholas R., editor, *American Philosophical Quarterly Monograph Series 2: Studies in Logical Theory*, pages 46–72. Blackwell, 1968.

[98] J. Hintikka. Surface information and depth information. In J. Hintikka and P. Suppes, editors, *Information and inference*, pages 263–297. Springer, 1970.

[99] J. Hintikka. *Logic, language games and information. Kantian themes in the philosophy of logic.* Clarendon Press, Oxford, 1973.

[100] L. Humberstone. Contra-classical logics. *Australasian Journal of Philosophy*, 78(4):438–474, 2000.

[101] L. Humberstone. Logical discrimination. In *Logica Universalis*, pages 225–246. Birkhäuser Verlag, 2nd edition, 2007.

[102] S. Jaśkowski. On the rules of suppositions in formal logic. *Studia Logica*, 1:5–32, 1934.

[103] R. Jeffrey. *Formal Logic: Its Scope and Limits*. McGraw-Hill, New York, 1967.

[104] R. Karp. Combinatorics, complexity, and randomness. *Commun. ACM*, 29(2):98–109, 1986.

[105] S. Kleene. *Introduction to Metamathematics*. Van Nostrand, Princeton, NJ, 1950.

[106] B. Kooi and A. Tamminga. Completeness via correspondence for extensions of the logic of paradox. *The Review of Symbolic Logic*, 5(4):720–730, 2012.

[107] S. Kripke. Semantical analysis of intuitionistic logic i. In *Studies in Logic and the Foundations of Mathematics*, volume 40, pages 92–130. Elsevier, 1965.

[108] I. Lakatos. *The methodology of scientific research programmes*. Cambridge university press, 1978.

[109] G. Lakemeyer and H. Levesque. A first-order logic of limited belief based on possible worlds. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, pages 624–635, 2020.

[110] C. Larese. *The Principle of Analyticity of Logic*. PhD thesis, Scuola Normale Superiore, Pisa, 2019.

[111] A. Laureano-Cruces and A. Barceló-Aspeitia. Formal verification of multi-agent systems behaviour emerging from cognitive task analysis. *Journal of Experimental & Theoretical Artificial Intelligence*, 15(4):407–431, 2003.

[112] H. Levesque. A logic of implicit and explicit belief. In *AAAI*, pages 198–202, 1984.

[113] M. Martinez and S. Sequoiah-Grayson. Logic and information. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2019 edition, 2019.

[114] F. Massacci. A proof theory for tractable approximations of propositional reasoning. In *Congress of the Italian Association for Artificial Intelligence*, pages 219–230. Springer, 1997.

[115] F. Massacci. Cook and Reckhow are wrong: Subexponential tableau proofs for their family of formulae. In *13th European Conference on Artificial Intelligence*, pages 408–409, 1998.

[116] F. Massacci. The proof complexity of analytic and clausal tableaux. *Theoretical Computer Science*, 243(1-2):477–487, 2000.

[117] M. Mondadori. Classical analytical deduction. *Annali dell'Università di Ferrara.* Sez. III, Discussion Paper 1, pages 1–23, 1988.

[118] M. Mondadori. Classical analytical deduction, part II. *Annali dell'Università di Ferrara.* Sez. III, Discussion Paper 5, 1988.

[119] M. Mondadori. On the notion of a classical proof. *Temi e prospettive della logica e della filosofia della scienza contemporanee*, 1:211–224, 1988.

[120] M. Mondadori. Efficient inverse tableaux. *Logic Journal of the IGPL*, 3(6):939–953, 1995.

[121] R. Morado and L. Savion. Rationality, logic and heurisitcs. In *IC-AI*, pages 791–797, 2002.

[122] N. Murray and E. Rosenthal. On the computational intractability of analytic tableau methods. *Logic Journal of the IGPL*, 2(2):205–228, 1994.

[123] R. Parikh. Sentences, belief and logical omniscience, or what does deduction tell us? *The Review of Symbolic Logic*, 1(4):459–476, 2008.

[124] K. Popper. *The logic of scientific discovery.* Routledge, London and New York; original published in German in 1934; first English translation in 1959, 1934/1959 [2002].

[125] D. Prawitz. *Natural Deduction: A Proof-Theoretical Study.* Almqvist & Wiksell, Uppsala, 1965.

[126] G. Priest. The logic of paradox. *Journal of Philosophical logic*, 8(1):219–241, 1979.

[127] G. Priest. *An introduction to non-classical logic.* Cambridge University Press, 2001.

[128] G. Priest. Paraconsistent logic. In *Handbook of philosophical logic*, volume 6, pages 287–393. Springer, 2002.

[129] W. Quine. *The Roots of Reference*. Open Court, LaSalle, 1973.

[130] M. Rabin. Complexity of computations. *Commun. ACM*, 20(9):625–633, 1977.

[131] R. Reckhow. *On the lengths of proofs in the propositional calculus*. PhD thesis, University of Toronto, 1976.

[132] T. Roy. Natural derivations for Priest, An Introduction to Non-Classical Logic. *The Australasian Journal of Logic*, 5:47–192, 2006.

[133] L. Savion and R. Morado. The role of logical inference in heuristic rationality. In *The Proceedings of the Twenty-First World Congress of Philosophy*, volume 5, pages 13–18, 2007.

[134] M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249–310, 1995.

[135] M. Sheeran and G. Stålmarck. A tutorial on stålmarck's proof procedure for propositional logic. *Formal Methods in System Design*, 16(1):23–58, 2000.

[136] Y. Shramko and H. Wansing. Some useful 16-valued logics: How a computer network should think. *Journal of Philosophical Logic*, 34(2):121–153, 2005.

[137] G. Sillari. Models of awareness. In *Logic and the Foundations of Games and Decisions*, pages 209–240. University of Amsterdam, 2008.

[138] R. Smullyan. *First-order logic*. Springer, Berlin, 1968.

[139] A. Solaki. *Logical Models for Bounded Reasoners*. PhD thesis, Institute for Logic, Language and Computation, Amsterdam, 2021.

[140] R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9(1):67–72, 1979.

[141] A. Tamminga. Correspondence analysis for strong three-valued logic. *Logical Investigations*, 20:255–268, 2014.

[142] A. Tamminga and K. Tanaka. A natural deduction system for first degree entailment. *Notre Dame Journal of Formal Logic*, 40(2):258–272, 1999.

[143] N. Tennant. *Natural Logic*. Edinburgh University Press, 1990.

[144] A. Troelstra and D. van Dalen. *Constructivism in Mathematics*, volume I. North-Holland, Amsterdam, 1988.

[145] A. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.

[146] A. Urquhart. Semantics for relevant logics. *The Journal of Symbolic Logic*, 37(1):159–169, 1972.

[147] A. Urquhart. The undecidability of entailment and relevant implication. *The Journal of Symbolic Logic*, 49(4):1059–1073, 1984.

[148] A. Urquhart. The complexity of decision procedures in relevance logic. In *Truth or consequences*, pages 61–76. Springer, Dordrecht, 1990.

[149] A. Urquhart. The complexity of propositional proofs. *The Bulletin of Symbolic Logic*, 1(4):425–467, 1995.

[150] J. van Benthem. Exploring logical dynamics. *Sudies in Logic, Language and Information, CSLI Publications*, 1996.

[151] J. van Benthem. Logical dynamics meets logical pluralism? *The Australasian Journal of Logic*, 6, 2008.

[152] J. van Benthem and M. Martinez. The stories of logic and information. In *Philosophy of Information*, volume 8, pages 225–288. Elsevier, 2008.

[153] E. Voishvillo. Philosophical-methodological aspects of relevant logic. *Bulletin of the Section of Logic*, 18(4):146–152, 1989.