

# A Simple Logic of Concepts

Thomas F. Icard and Lawrence S. Moss

August 12, 2022

## Abstract

In Pietroski (2018) a simple representation language called **SMPL** is introduced, construed as a hypothesis about core conceptual structure. The present work is a study of this system from a logical perspective. In addition to establishing a completeness result and a complexity characterization for reasoning in the system, we also pinpoint its expressive limits, in particular showing that the fourth corner in the square of opposition (“**Some\_not**”) eludes expression. We then study a seemingly small extension, called **SMPL**<sup>+</sup>, which allows for a minimal predicate-binding operator. Perhaps surprisingly, the resulting system is shown to encode precisely the concepts expressible in first-order logic. However, unlike the latter class, the class of **SMPL**<sup>+</sup> expressions admits a simple procedural (context-free) characterization. Our contribution brings together research strands in logic—including natural logic, modal logic, description logic, and hybrid logic—with recent advances in semantics and philosophy of language.

## 1 Introduction and Motivation

The idea that thoughts and concepts are logically structured is a guiding hypothesis for much of modern cognitive science (Fodor, 1975; Carey, 2009; Piantadosi, 2021). A fundamental question is what, precisely, the relevant structure is. The answer to this question will likely depend in part on how we answer various related questions, e.g., what kinds of operations are defined over such structure. An influential approach to these question starts from a “worst case” position. For example, we might take thoughts to have the structure of expressions in a higher-order modal logic (see, e.g., Montague 1974; Muskens 2007). Similarly, as a first pass, we may take the space of possible mental operations to encompass anything that can be computed by a Turing machine (e.g., Zylberberg et al. 2011).

What this worst case approach gains in generality it arguably lacks in precision. That is, while these frameworks may pinpoint absolute limits of thought, they do not seem to carve cognition at its natural joints. The challenge is often posed as one of explanation (Partee, 1975; Keenan and Stavi, 1985; Lidz, 2018). Why are certain possibilities widely or even universally attested, while others are systematically absent? Human thought—and human language as well—appears to be more finely sculpted than the rather blunt worst-case approaches would be able to reveal. Or at any rate, even if the wider range of possibilities is ultimately available, we may still want to recognize certain structures as psychologically more basic or prominent.

A famous early example of this style of argument is from Chomsky (1957), who suggested that whatever generates linguistic output in humans does not seem to involve arbitrary reordering of lexical items. A concomitant technical insight was that a representational system could be quite powerful—even infinitely so, under common idealizations—while still highly constrained. For instance, a *context-free grammar* could generate many of the patterns we see in natural languages, while disallowing arbitrary reshuffling and indeed preserving a type of constituency structure (see also Chomsky 1959). If our goal is to zero in on linguistic mechanisms, context-free grammars may well be a reasonable starting point, even if we believe they are still too general (Pullum, 1983) or not quite general enough (Joshi et al., 1991).

A similar style of investigation has been pursued in connection with logical representation and inference, under the label of *natural logic* (van Benthem, 1987b; Sánchez-Valencia, 1991; Keenan, 2004; Pratt-Hartmann, 2004; Moss, 2008). The goal in this line of research is to isolate constrained logical fragments that enjoy prima facie correspondence with “surface-level” natural language inference patterns. Some such

patterns, such as those involving monotonicity, have been invoked in both computational work and in empirical studies on human inference (see Icard and Moss 2014 for an overview). A different, but related tradition has studied specific features of semantic representation that appear to be universal, and thus seem to demand some kind of cognitive—e.g., learning-theoretic or communication-theoretic—explanation (Barwise and Cooper, 1981; Steinert-Threlkeld and Szymanik, 2019), which might in turn shed light on the structure of linguistic thought. Beyond the domain of language, and in a less formal vein, developmental psychologists have offered detailed accounts of what conceptual repertoires are present in pre-linguistic infants and how those structures are transformed through learning and maturation (e.g., Carey 2009).

Spanning different domains and from different perspectives, all of these strands aim at uncovering the specific forms that human thought and conceptual structure can take, out of the vast space of *prime facie* possibilities. In the present work we pursue one intriguing recent hypothesis that is informed by all of this work, as well as much other work in cognitive science. In a recent book, Pietroski (2018) offers a view of core conceptual structure as “massively monadic and minimally relational,” introducing the idea through a simple system called SMPL (see especially Chapters 6 and 7 in Pietroski 2018).<sup>1</sup> Ultimately, the proposal is that we understand natural language *meaning* to involve instructions for retrieving and combining concepts in a representation language like SMPL. The bold conjecture is that SMPL is up to this task, that it can capture at least a substantial class of the concepts that can be expressed in natural languages.

Part of the motivation for SMPL is the classical observation that there is something special about *conjunction* and *existential quantification*. It has long been observed that conjunctive concepts are psychologically simple, and generally easy to learn (Shepard et al., 1961). Meanwhile, the idea of existentially quantifying over unspecified arguments (or “variables”) is a common theme across analyses of natural language phenomena, appearing prominently in event semantics (Davidson, 1966; Parsons, 1990; Pietroski, 2005), dynamic semantics (Kamp, 1981; Heim, 1982), and elsewhere.<sup>2</sup> SMPL augments this conjunctive-existential base with a type of *polarization* operator, inspired by Tarski’s (1944) seminal treatment of satisfaction in first-order logic. Polarizing a concept turns it into a predicate that applies to everything (“true”) or nothing (“false”), recovering a class of concepts that effectively behave like propositions.

Our study of SMPL—and some of its extensions—is logical. The purpose is not to offer further evidence that SMPL is empirically adequate, but rather to understand it as a formal system and to clarify its properties from a logical point of view. In that direction, we first formulate SMPL precisely as a (modal) logical system and establish some basic facts about its expressive power. A notable observation is that through assertions of concept equality, SMPL is able to express the standard quantifiers *Some*, *All*, and *No*; however, the unattested basic quantifier “*Some\_not*” eludes expression. We also provide a complete axiomatization of the system, isolating its fundamental reasoning principles, and show that reasoning in the system is no more difficult than in propositional logic (both possessing an NP-complete satisfiability problem).

We next turn to a seemingly modest extension of SMPL, also introduced in Pietroski (2018) for capturing a wider array of phenomena than SMPL could accommodate. The extension, which we call SMPL<sup>+</sup>, allows a minimal type of predicate binding, which we observe is closely related to operators explored independently in hybrid logic (Blackburn and Seligman, 1995; Goranko, 1996). Perhaps surprisingly, we show that SMPL<sup>+</sup> is expressively equivalent to standard first-order logic: SMPL<sup>+</sup> encodes precisely the concepts representable by formulas of first-order logic with one free variable. However, while the latter is known to be a syntactically complex set—specifically, this set of formulas is not context-free (van Benthem, 1987a; Marsh and Partee, 1987)—by construction, SMPL<sup>+</sup> is generated in a context-free manner. This may be important if we want to think of conceptual repertoires not just extensionally, but in generative or procedural terms.

Our contribution in this paper can be viewed from two different perspectives. On the one hand, we offer SMPL and SMPL<sup>+</sup> as interesting systems from a logical point of view. While they are motivated by many of the same considerations as various natural logics—syllogistic systems, monotonicity calculi, etc.—they have a rather different character from the systems studied in this tradition. On the other hand, the work here helps clarify the nature of an important recent proposal in philosophy of language, revealing extensive

<sup>1</sup>The acronym comes from Simple Monadic Predicate Logic.

<sup>2</sup>The idea has also featured in recent computational work; see, e.g., Liang et al. (2013); Buch et al. (2021). So called conjunctive queries serve as a “basic building block” in much of database theory as well (Abiteboul et al., 1995, p. 39).

connections with existing work in philosophical and computational logic. More generally, we hope that the present work may help to build further bridges between prominent research strands in logic (natural logic, modal logic, hybrid logic, etc.) and more cognitively inspired work in the semantics of natural language.

In what follows we assume only minimal familiarity with (classical) first-order logic; see e.g., Barwise (1977) for a review. All other logical tools and ideas are introduced in a self-contained way. The details for the main results of the paper appear in technical appendices.

## 2 SMPL

Our first contribution in this section is to present SMPL in terms of a precisely defined syntax and semantics. The system can be understood as a language for forming complex monadic (or “unary”) predicates (or “concepts”) from some set of “atomic” monadic predicates  $\mathcal{A} = \{A, B, \dots\}$  and a set  $\mathcal{R} = \{R, Q, \dots\}$  of dyadic (or binary-relational) predicates. However, relations appear in a highly restricted way, with their second argument existentially quantified.

**Definition 1.** We define the formal language  $\mathcal{L}$  of concepts using the following (context-free) grammar:

$$\varphi ::= A \mid (\varphi \wedge \varphi) \mid \exists R.\varphi \mid \Downarrow\varphi$$

where  $A \in \mathcal{A}$  and  $R \in \mathcal{R}$ .

For example, a typical concept in  $\mathcal{L}$  is  $(A \wedge \exists R.B)$ , which intuitively picks out those  $A$ s that are related by  $R$  to some  $B$ . (The precise semantics is given below in Definition 2.)

**Remark 1.** An alternative formulation of the language, highlighting a connection to multimodal logic with relational modalities  $\langle R \rangle$  and an existential modality  $E$  (see Hemaspaandra 1996), would be:

$$\varphi ::= A \mid (\varphi \wedge \varphi) \mid \langle R \rangle \varphi \mid \neg E\varphi$$

Such a formulation shows that  $\mathcal{L}$  is evidently a small fragment of a larger polymodal logic with  $E$  (or  $U$ , the universal modality). Our notation adopts the syntax of *description logic* for  $\exists R.\varphi$ , making the existential nature of the operator apparent. Similarly minimal systems have been explored in the literature on description logic (see, e.g., Kurtonina and de Rijke 1999).

The expressions in  $\mathcal{L}$  are to be understood not as denoting sets, but rather something like mental *classifiers* (Pietroski, 2018, Chapter 2), perhaps more akin to Aristotelian predication (see, e.g., the discussion in Malink 2013 of the modal syllogistic). Nonetheless, just as in the study of Aristotle’s logic, it can be helpful to investigate the properties of the system with respect to a standard set-theoretic interpretation (viz. the set of objects that fall under the concept). In that direction:

**Definition 2.** A model is a pair  $\mathcal{M} = (M, \llbracket \ ])$  consisting of a non-empty set  $M$  and a valuation function  $\llbracket \ ]$  such that  $\llbracket A \rrbracket \subseteq M$  for each  $A \in \mathcal{A}$  and  $\llbracket R \rrbracket \subseteq M \times M$  for  $R \in \mathcal{R}$ . We extend this to the rest of the language of monadic predicates as follows:

$$\begin{aligned} \llbracket \varphi \wedge \psi \rrbracket &= \llbracket \varphi \rrbracket \cap \llbracket \psi \rrbracket \\ \llbracket \exists R.\varphi \rrbracket &= \{a \in M \mid \text{there is } b \in \llbracket \varphi \rrbracket \text{ such that } \langle a, b \rangle \in \llbracket R \rrbracket\} \\ \llbracket \Downarrow\varphi \rrbracket &= \{a \in M \mid \llbracket \varphi \rrbracket = \emptyset\}. \end{aligned}$$

In other words,  $\Downarrow\varphi$  is true of everything if  $\varphi$  is empty, and true of nothing if  $\varphi$  is non-empty. Pietroski (2018)—inspired by Tarski’s (1944) treatment of satisfaction and work on polarity phenomena in logic (e.g., Sánchez-Valencia 1991)—introduces this operator as a “polarizer” in the sense that expressions of the form  $\Downarrow\varphi$  behave like propositional logical formulas (see Section 2.1 below).

Although the language  $\mathcal{L}$  is quite spare, it already suffices to define a number of other useful concepts. For instance, using  $\Downarrow\varphi$  we can define a new operator  $\Uparrow\varphi$ , which intuitively asserts the existence of something

satisfying  $\varphi$ :  $\llbracket \uparrow\varphi \rrbracket = \{a \in M \mid \llbracket \varphi \rrbracket \neq \emptyset\}$ . Likewise, we can define the trivial concept  $\perp$  that applies to nothing by  $A \wedge \Downarrow A$ ,<sup>3</sup> while the other trivial concept  $\top$ , defined by  $\Downarrow \perp$ , applies to anything whatsoever.

**Example 1.** Part of the motivation for SMPL comes from the structure of verb phrases, and in particular the analysis of verb phrases through event semantics (see, e.g., Pietroski 2005). To give a sense of the natural language phenomena straightforwardly encoded in a formalism like this, consider the following examples inspired by Pietroski’s (2018) discussion (see especially the discussion in Chapter 7). A sentence like

(1) Scarlet stabbed Mustard

can be analyzed in terms of a predicate of events. We might assume that **Stab** is a predicate that holds of stabbing events, that **Before** is the precedence relation on events, and that **Speech-Time** is a contextually determined “reference time.” Then consider the concept described in (2):

(2)  $\exists \text{Before.Speech-Time} \wedge \exists \text{External.Scarlet} \wedge \text{Stab} \wedge \exists \text{Internal.Mustard}$ .

This holds of any event that happened before the time of utterance and whose “external” role is played by Scarlet, whose “internal” role is played by Mustard, and which is a stabbing.<sup>4</sup> Note also that the proper names, Scarlet and Mustard, are being treated here as unary predicates (cf. Quine 1948).

The meaning of a declarative sentence like (1) could then be obtained by (“upward”) polarizing an expression like the one in (2):

(3)  $\uparrow(\exists \text{Before.Speech-Time} \wedge \exists \text{External.Scarlet} \wedge \text{Stab} \wedge \exists \text{Internal.Mustard})$ .

In a similar vein, we could analyze the meaning of a negative claim like

(4) Scarlet didn’t stab Mustard

by polarizing “downward”:

(5)  $\Downarrow(\exists \text{Before.Speech-Time} \wedge \exists \text{External.Scarlet} \wedge \text{Stab} \wedge \exists \text{Internal.Mustard})$ .

This will be false—that is, the concept that applies to nothing—just in case there is such an event, true otherwise. SMPL is able to encode more complex examples as well. For instance, a sentence like:

(5) Scarlet didn’t stab anyone who lives in Tudor Manor

might be rendered as:

(6)  $\Downarrow(\exists \text{Before.Speech-Time} \wedge \exists \text{External.Scarlet} \wedge \text{Stab} \wedge \exists \text{Internal}.\exists \text{LivesIn.TudorManor})$ ,

involving a further monadic predicate **TudorManor** and a further dyadic predicate **LivesIn**. See Pietroski (2018) for many more examples and constructions.

## 2.1 Embedding Propositional Logic

We noted above that the polarization operator  $\Downarrow$  (together with the definable  $\uparrow$  operator) facilitates encoding of propositional reasoning. We now proceed to make this observation more precise.

Consider the language  $\mathcal{L}^{prop}$  of propositional logic, construing the unary predicates  $\mathcal{A}$  here as propositional atoms. That is,  $\mathcal{L}^{prop}$  is generated by the grammar:

$$\alpha ::= A \mid (\alpha \wedge \alpha) \mid \neg \alpha .$$

<sup>3</sup>Formally speaking, this looks like a strong kind of “Moore sentence,”  $A \wedge \Box \neg A$  (cf. Holliday and Icard 2010), here meaning something like “being an  $A$  such that nothing is an  $A$ .”

<sup>4</sup>Alternatively, the subexpression  $\text{Stab} \wedge \exists \text{Internal.Mustard}$  could be replaced by  $\exists \text{StabOf.Mustard}$ , where **StabOf** is a dyadic relation between stabbing events and those stabbed. Thanks to Paul Pietroski for mentioning this.

The semantics of  $\mathcal{L}^{prop}$  is given as usual by a Boolean valuation  $v : \mathcal{A} \rightarrow \{0, 1\}$ , canonically extended to a valuation  $\bar{v} : \mathcal{L}^{prop} \rightarrow \{0, 1\}$ . We say  $\alpha$  is *satisfiable* if there is a valuation  $v$  such that  $\bar{v}(\alpha) = 1$ .

Our aim is to show that SMPL can emulate propositional reasoning in  $\mathcal{L}^{prop}$ . To do this we can restrict attention to one-point models of  $\mathcal{L}$ . In particular, a propositional valuation  $v$  determines a one-point model  $\mathcal{M}_v = (\{\ast\}, \llbracket \cdot \rrbracket_v)$  whereby  $\llbracket R \rrbracket_v = \emptyset$  for all  $R \in \mathcal{R}$ , and for all  $A \in \mathcal{A}$ :

$$\llbracket A \rrbracket_v = \begin{cases} \{\ast\} & \text{if } v(A) = 1 \\ \emptyset & \text{if } v(A) = 0 \end{cases}$$

We want a translation function  $\text{tr}$  from  $\alpha \in \mathcal{L}^{prop}$  into  $\mathcal{L}$  guaranteeing the following:

$$v(\alpha) = 1 \quad \text{iff} \quad \llbracket \text{tr}(\alpha) \rrbracket = \{\ast\} \quad (1)$$

That is, the propositional formula  $\alpha$  is assigned 1 if and only if its translation denotes the non-empty set  $\{\ast\}$ . In fact, such a translation is given recursively by:  $\text{tr}(A) = A$ ,  $\text{tr}(\alpha \wedge \beta) = \text{tr}(\alpha) \wedge \text{tr}(\beta)$ , and  $\text{tr}(\neg\alpha) = \Downarrow \text{tr}(\alpha)$ . It is easy to show by induction on the complexity of  $\alpha$  that (1) holds under this choice of  $\text{tr}$ .

In the appendix we use this observation to show that reasoning about concepts in SMPL, e.g., deciding whether there is a model with  $\varphi$  interpreted as a non-empty set, is at least as difficult as the satisfiability problem for propositional logic. The latter problem is well-known to be NP-hard (see, e.g., Papadimitriou 1994), which gives us a first result about SMPL (the proof in the appendix primarily uses (1) above):<sup>5</sup>

**Theorem 2.** The problem of determining whether two SMPL concepts differ in extension in some model is NP-hard, that is, at least as difficult as deciding satisfiability of a formula in propositional logic. Likewise, determine whether two SMPL concepts are equivalent is as difficult as determining propositional validity.

## 2.2 Concept Definability

Despite the fact that SMPL can encode propositional logical reasoning, there are a variety of concepts that cannot be expressed in SMPL. A first observation is that there is no general way to define concept negation in SMPL. That is, given an atomic predicate  $A$  there is no concept  $\varphi$  such that, in every model  $(M, \llbracket \cdot \rrbracket)$  we have  $\llbracket \varphi \rrbracket = M \setminus \llbracket A \rrbracket$ , i.e.,  $\varphi$  is interpreted as the complement of  $A$ . The proof of this is instructive:

**Proposition 3.** Concept negation cannot be defined in SMPL.

*Proof.* Our aim is to show there is no formula  $\varphi$  whose meaning always complements  $A$ . Consider the model  $\mathcal{M}_1 = (\{d, e\}, \llbracket \cdot \rrbracket_1)$ , with  $\llbracket A \rrbracket_1 = \{d\}$  and any other predicates empty. We claim for every  $\varphi$  that  $\llbracket \varphi \rrbracket_1$  is equal either to  $\emptyset$ ,  $\{d\}$ , or  $\{d, e\}$  (see Fig. 1). This is by a simple induction on the structure of formulas in  $\mathcal{L}$ . Consequently there is no formula whose extension is  $\{e\}$ , which is exactly  $\{d, e\} \setminus \llbracket A \rrbracket_1$ .  $\dashv$

Notably, adding negation to  $\mathcal{L}$  results in a well-known modal logical system, multimodal  $\mathbf{K}$  with an existential modality. This logic is very expressive, and while decidable, is known to be highly complex.<sup>6</sup>

A similar argument shows that concept disjunction cannot be defined directly either:

**Proposition 4.** Concept disjunction cannot be defined in SMPL.

*Proof.* We show there is no concept  $\varphi$  whose meaning is always given by the union of the meanings of  $A$  and  $B$ , i.e.,  $\llbracket \varphi \rrbracket = \llbracket A \rrbracket \cup \llbracket B \rrbracket$ . Consider the model  $\mathcal{M}_3 = (\{i, j, k\}, \llbracket \cdot \rrbracket_3)$  with  $\llbracket A \rrbracket_3 = \{i\}$ ,  $\llbracket B \rrbracket_3 = \{j\}$ , and any other predicates empty. We observe by induction on that for any  $\varphi$  we have  $\llbracket \varphi \rrbracket_3$  equal to  $\emptyset$ ,  $\{i\}$ ,  $\{j\}$ , or  $\{i, j, k\}$ . So in particular there is no concept whose extension is exactly  $\{i, j\} = \llbracket A \rrbracket_3 \cup \llbracket B \rrbracket_3$ .  $\dashv$

A similar argument shows that we cannot define a concept  $\forall R.\varphi$  with meaning  $\llbracket \forall R.\varphi \rrbracket = \{a \in M \mid \text{for all } b \text{ s.t. } \langle a, b \rangle \in \llbracket R \rrbracket : b \in \llbracket \varphi \rrbracket\}$ , codifying a phrase like “likes only cookies.” Though we do not explore exact characterizations of expressive power here, it is worth mentioning that existing tools from the literature on modal and description logics, such as Kurtonina and de Rijke’s (1999) “one-way bisimulations,” may point the way to a general theory. Instead, we turn briefly to connections with syllogistic logics.

<sup>5</sup>It has been claimed that NP-completeness is actually the right target for analyses of linguistic phenomena; see, e.g., Ristad (1993) and more recently, Szymanik (2016).

<sup>6</sup>Specifically, the satisfiability problem is complete for the class of exponential time problems (Hemaspaandra, 1996).



Figure 1: Models  $\mathcal{M}_1$  and  $\mathcal{M}_2$  used in Propositions 3 and 5. Definable (non-empty) subsets are circled.

### 2.3 SMPL and Syllogistics

Up to this point, we have worked with SMPL as a language of concepts. In logical terms, we have a first-order model whose elements are the concepts, and with operations given in Definition 2. As in any model, we write  $\mathcal{M} \models \varphi = \psi$  to mean that  $\llbracket \varphi \rrbracket = \llbracket \psi \rrbracket$ . It is natural to think of this type of concept equality as a basic judgment in SMPL. Using equality we can also express one way of encoding the (binary) universal quantifier, whereby **All  $\varphi$  are  $\psi$**  is expressed by  $\varphi \wedge \psi = \varphi$ —semantically this says  $\llbracket \varphi \rrbracket \cap \llbracket \psi \rrbracket = \llbracket \varphi \rrbracket$ , i.e.,  $\llbracket \varphi \rrbracket \subseteq \llbracket \psi \rrbracket$ .<sup>7</sup> Henceforth we will abbreviate this statement by  $\varphi \leq \psi$ .

In a similar way we can also express the thought **Some  $\varphi$  are  $\psi$**  by  $\uparrow(\varphi \wedge \psi) = \top$ , while **No  $\varphi$  are  $\psi$**  is expressed by  $\varphi \wedge \psi = \perp$ . This already shows that SMPL is at least as expressive as basic syllogistic logic with monadic variables and quantifiers **Some**, **All**, and **No** (see Moss 2008 for such a system). In fact, SMPL allows transitive verbs and relative clauses in both  $\varphi$  and  $\psi$ —recall Example 1 above—so it even extends syllogistic logic in the direction of richer syllogistic systems (e.g., Pratt-Hartmann and Moss 2009).

It is natural to ask which other judgments SMPL can express. For instance, Moss (2008) considers extending the language of **Some**, **All**, and **No** with Boolean connectives. In such a language it is then easy to define further operators such as **Some.not**, so that **Some  $A$  are not  $B$**  is expressed as  $\neg \text{All}(A, B)$ . Another way of expressing **Some.not** is to add concept negation  $\bar{\varphi}$ , leading to **Some( $A, \bar{B}$ )** (see, e.g., Pratt-Hartmann and Moss 2009). The fact that concept negation cannot be defined (Proposition 3) casts some doubt on whether something analogous is possible here. In fact, we can show directly that **Some.not** cannot be expressed. If we could express **Some.not( $A, B$ )**, then there would have to be some  $\varphi$  and  $\psi$ —presumably containing  $A$  and  $B$  as subexpressions—such that the equality statement  $\varphi = \psi$  holds in exactly the models where **Some.not( $A, B$ )** is true, i.e., exactly when  $\llbracket A \rrbracket \not\subseteq \llbracket B \rrbracket$ .

**Proposition 5.** **Some.not** cannot be expressed. That is, there are no expressions  $\varphi$  and  $\psi$  involving subexpressions  $A$  and  $B$  such that, for every model  $M$ ,  $\llbracket \varphi \rrbracket_M = \llbracket \psi \rrbracket_M$  iff  $\llbracket A \rrbracket_M \not\subseteq \llbracket B \rrbracket_M$ .

*Proof.* Consider two models:  $\mathcal{M}_1$  as in Proposition 3 in which we set  $\llbracket A \rrbracket_1 = \{d, e\}$  and  $\llbracket B \rrbracket_1 = \{d\}$ ; in the second model we have  $\mathcal{M}_2 = (\{i\}, \llbracket \cdot \rrbracket_2)$  with  $\llbracket A \rrbracket_2 = \llbracket B \rrbracket_2 = \{i\}$ , and all else empty (see again Figure 1). Note that  $\llbracket A \rrbracket_1 \not\subseteq \llbracket B \rrbracket_1$ , while  $\llbracket A \rrbracket_2 \subseteq \llbracket B \rrbracket_2$ . We will show that they nonetheless agree on all equalities.

Clearly every subset of  $\{i\}$  is the extension of some predicate in  $\mathcal{M}_2$ . The definable subsets of  $M_1$  are  $\emptyset, \{d\}, \{d, e\}$  (recall Proposition 3). However, we also have, for any  $\varphi$ , that  $\llbracket \varphi \rrbracket_1 = \emptyset$  iff  $\llbracket \varphi \rrbracket_2 = \emptyset$ . This is by induction on  $\varphi$ . We now claim, for all  $\varphi$  and  $\psi$ , that if  $\llbracket \varphi \rrbracket_1 = \llbracket \psi \rrbracket_1$  then  $\llbracket \varphi \rrbracket_2 = \llbracket \psi \rrbracket_2$ . If  $\llbracket \varphi \rrbracket_1 = \emptyset$ , this is immediate by the previous observation. If  $\llbracket \varphi \rrbracket_1 \neq \emptyset$ , then  $\llbracket \psi \rrbracket_1 \neq \emptyset$ , and again by the previous observation both  $\llbracket \varphi \rrbracket_1 \neq \emptyset$  and  $\llbracket \psi \rrbracket_2 \neq \emptyset$ . But this means  $\llbracket \varphi \rrbracket_2 = \{i\} = \llbracket \psi \rrbracket_2$ .

Suppose finally for a contradiction that there were  $\varphi$  and  $\psi$  such that, for every  $\mathcal{M}$ :  $\llbracket \varphi \rrbracket = \llbracket \psi \rrbracket$  iff  $\llbracket A \rrbracket \not\subseteq \llbracket B \rrbracket$ . Then since  $\llbracket A \rrbracket_1 \not\subseteq \llbracket B \rrbracket_1$  and thus  $\llbracket \varphi \rrbracket_1 = \llbracket \psi \rrbracket_1$ , we would have by the previous observation that  $\llbracket \varphi \rrbracket_2 = \llbracket \psi \rrbracket_2$ , which again would mean that  $\llbracket A \rrbracket_2 \not\subseteq \llbracket B \rrbracket_2$ , quod non.  $\dashv$

The question of why no known natural language lexicalizes an expression like **Some.not** is a well known and longstanding puzzle. For instance, 19th century metaphysician William Hamilton claimed that the

<sup>7</sup>In other words, **All  $\varphi$  are  $\psi$**  is rendered as something like, “The  $\varphi$ ’s are the same as the  $\psi$ ’s that are  $\psi$ .” See Knowlton et al. (2021) for relevant evidence on how universal quantifiers may in fact be represented.

$\frac{}{\Downarrow\Downarrow\Downarrow\varphi = \Downarrow\varphi}$	$\frac{}{\varphi \leq \top \wedge \Uparrow\varphi}$	$\frac{}{\varphi \wedge \Downarrow\varphi = \psi \wedge \Downarrow\psi}$	$\frac{}{\exists R.\varphi \leq \Uparrow\varphi}$
	$\frac{\psi \leq \varphi}{\Downarrow\varphi \leq \Downarrow\psi}$	$\frac{\varphi \leq \psi}{\exists R.\varphi \leq \exists R.\psi}$	

Figure 2: The rules of SMPL. We also include the usual rules of equational logic and of meet-semi-lattices, in addition to a rule of proof by cases. See Definition 3 for details.

fourth Aristotelian<sup>8</sup> form in the square of opposition

was only laid down from a love of symmetry, in order to make out the opposition of all the corners in the square of opposition. . . In reality and in thought, every quantity is necessarily either all, or none, or some. (Hamilton, 1860, p. 261)

Perhaps the most prominent explanation for why there is no such determiner comes from Horn (1989). He argues that there is no need to lexicalize such a meaning given the other syllogistic quantifiers and some general pragmatic principles; see, e.g., Hoeksema (1999) for criticism. Proposition 5 suggests a different possible explanation. Assuming SMPL captures some very basic conceptual repertoire, perhaps the reason we do not see lexicalization of such determiners relates to the fact that such thoughts cannot even be formulated without going beyond SMPL. Pursuing this idea would take us beyond the scope of the present contribution, but we leave this as an intriguing possibility for further exploration.<sup>9</sup>

### 3 A Complete Axiomatization

The foregoing discussion provides some motivation for taking SMPL seriously as a hypothesis about the basic logic of concepts; see Pietroski (2018) for much more. In this section we show that reasoning in the system can be fully axiomatized. The importance of axiomatization in semantics and theory of meaning has been stressed in previous work (see Moss 2010; Holliday and Icard 2018, *inter alia*). In the present case the interest is twofold. First, we would like a perspicuous presentation of the basic inference steps licensed by the interpretation given in Section 2. Second, proving the completeness of our set of axioms will show that the lower-bound on complexity of SMPL established in Theorem 2 can be matched by the very same upper-bound. That is, we will be able to conclude that reasoning in SMPL is exactly as difficult as reasoning in propositional logic, thereby substantiating the intuitive claim that this system is indeed simple, despite its ability to encode a reasonably wide class of complex concepts.

Our proof calculus is a natural deduction system which incorporates equational reasoning, taking as basic statements of the form  $\varphi = \psi$ . In order to formulate the system we adopt a useful abbreviation. As a counterpart to  $\varphi = \perp$ , let us write  $\varphi \neq \perp$  for  $\top \leq \Uparrow\varphi$ . Then a model  $\mathcal{M}$  satisfies  $\varphi \neq \perp$  iff  $\llbracket \varphi \rrbracket \neq \emptyset$ . The reason that we use this terminology comes from the following observation. Fix a model  $\mathcal{M}$ . For all  $\varphi$ , either  $\llbracket \varphi \rrbracket = \emptyset$ , in which case  $\llbracket \Downarrow\varphi \rrbracket = M$ , or else  $\llbracket \varphi \rrbracket \neq \emptyset$ , in which case  $\llbracket \Uparrow\varphi \rrbracket = M$ . So, our model satisfies  $\varphi = \perp$ , or it satisfies  $\varphi \neq \perp$ . The proof system is defined as follows:

**Definition 3** (Proof System). The proof rules include the usual rules of equational logic and meet-semi-lattices (see, e.g., Grätzer 2008), as well as the rules in Figure 2. We also allow applications of a rule by cases. In words: to prove  $\varphi = \psi$  from  $\Gamma$ , one may (i) make a temporary assumption of  $\chi = \perp$ , add this to  $\Gamma$ , and then derive  $\varphi = \psi$ ; and (ii) make a temporary assumption of  $\chi \neq \perp$ , add this to  $\Gamma$ , and then derive  $\varphi = \psi$ . In symbols: if  $\Gamma \cup \{\chi = \perp\} \vdash \varphi = \psi$  and  $\Gamma \cup \{\chi \neq \perp\} \vdash \varphi = \psi$ , then  $\Gamma \vdash \varphi = \psi$ . Since our proof system incorporates proofs-with-hypotheses, it is fundamentally a natural deduction proof system. For such systems, it is standard to define *proof trees with hypotheses* and hence the notion  $\Gamma \vdash \varphi = \psi$ .

<sup>8</sup>Even Aristotle remarked in the *Posterior Analytics* that there are really only three basic forms. See Horn (1989).

<sup>9</sup>Cf. also Sbardolini (2021) for work in a similar vein, targeting the absence of connectives like NAND in natural languages.

We read  $\Gamma \vdash \varphi = \psi$  as “ $\varphi = \psi$  follows from  $\Gamma$  by a formal proof in our system.” Correspondingly, we write  $\Gamma \models \varphi = \psi$  to mean that  $\varphi = \psi$  semantically follows from the assumptions in  $\Gamma$ : every model satisfying each equality statement in  $\Gamma$  also satisfies  $\varphi = \psi$ . The statement of soundness is that whenever something is provable,  $\Gamma \vdash \varphi = \psi$ , this equality really follows, i.e.,  $\Gamma \models \varphi = \psi$ . Meanwhile, a result of completeness would tell us that the rules in Definition 3—and in particular those in Figure 2—summarize everything there is to know about reasoning in SMPL. That is, when  $\varphi = \psi$  really follows from  $\Gamma$ , there is a proof using these rules that witnesses this fact. In fact, we have both results:

**Theorem 6.**  $\Gamma \models \varphi = \psi$  iff  $\Gamma \vdash \varphi = \psi$ .

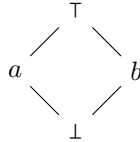
The full proof of this theorem is given in the appendix. As an example of a derivation in this system, consider a simple inference in a modest extension of syllogistic logic (a slightly elaborated version of the classical syllogism *Celarent*): from “No student lives in a house” and “Every administrator lives in a big house” it follows that “No student is an administrator”:

$$\begin{array}{c}
 \text{Every administrator lives in a big house} \quad \frac{\overline{H \wedge B \leq H}}{\exists L.(H \wedge B) \leq \exists L.H} \\
 \frac{A \leq \exists L.(H \wedge B)}{A \leq \exists L.H} \\
 \text{No student lives in a house} \quad \frac{(S \wedge A) \leq (S \wedge \exists L.H)}{(S \wedge A) = \perp} \\
 \frac{(S \wedge \exists L.H) = \perp}{(S \wedge A) = \perp}
 \end{array}$$

As a corollary of Theorem 6 we obtain the following complexity upper-bound on reasoning in SMPL (again, see the appendix for the argument):

**Corollary 7.** The problem of determining whether  $\Gamma \not\vdash \alpha = \beta$  is decidable in nondeterministic polynomial (NP) time, i.e., is no harder than determining satisfiability of propositional logic formulas.

**Remark 8.** In Section 2 we defined the models of SMPL to be based on sets. As a purely mathematical point, the SMPL as a logical system possesses other models that are not based on sets. Here is one example. In a set model, the interpretation of every monadic predicate  $\varphi$  has the property that either  $\llbracket \varphi \rrbracket = \llbracket \perp \rrbracket$ , or else  $\llbracket \uparrow \varphi \rrbracket = \llbracket \top \rrbracket$ . We actually depend on this feature in our soundness arguments (see appendix). However, this property is not expressible in SMPL itself, and for that matter it is not implied by the logical rules. To see this, consider the following model:



We interpret the symbols  $\wedge$ ,  $\Downarrow$  and  $\exists R$  in the following ways. For  $\wedge$ , we use the greatest lower bound in the picture. (For example,  $a \wedge b = \perp$ .) For  $\Downarrow$ , we say that  $\Downarrow a = b$ ,  $\Downarrow b = a$ ,  $\Downarrow \top = \perp$ , and  $\Downarrow \perp = \top$ . For each  $R \in \mathcal{R}$ , we take  $\llbracket R \rrbracket = \emptyset$ . Then it is not hard to check that all of the rules of the logic are sound for this interpretation. (The key points of the verification are that  $\Downarrow$  is order-reversing, and  $\Downarrow \Downarrow$  is the identity function.) And of course  $a$  has the property that both  $a \neq \perp$  and also  $\uparrow a = \Downarrow \downarrow a = a \neq \top$ .

Whether such alternative interpretations have any implications for our understanding of language or cognition—e.g., viz. inferentialist views of meaning (Brand 2000, etc.)—we leave as an open question.

## 4 From SMPL to SMPL<sup>+</sup>

The system SMPL capitalizes on the evident fundamentality of conjunction and existential quantification in conceptual structure. But it goes without saying that various thoughts which can be entertained cannot be formulated in SMPL. The challenge is to augment the system just enough to capture combinatorially complex thoughts that are succinctly expressed in natural languages, without collapsing into familiar “worst case” analyses with no meaningful constraints at all.



**Example 2.** As one example of such a construction, Pietroski considers an intriguing pattern pointed out by Chierchia (1984); see also Pietroski (2019). In a phrase like *Romeo loves Juliet*, English allows abstracting out *Romeo* to define a new type of predicate, namely the relative clause *who loves Juliet*. Such a meaning can be encoded in SMPL by  $\exists \text{Loves}.\text{Juliet}$ . But English also allows us to abstract out *Juliet* with the relative clause *who Romeo loves*. This we cannot yet symbolize. Standard tools like the lambda calculus (as in the higher-order system from Montague 1974, for example) are tailor-made for such abstractions. The problem is that they allow for far too many unwanted possibilities. For instance, there is no way in English (or in other languages) to abstract out the verb to give expressions like *which Romeo Juliet*, meaning something like, “a relation that Romeo holds to Juliet.” We would like a system that allows the first two abstractions but disallows the third, and more generally comes closest to carving natural languages at their natural joints.

In this section we explore a concrete proposal extending SMPL, obtained by adding a very limited type of abstraction (cf. Pietroski 2018, Chapter 7).<sup>10</sup> Given a monadic predicate  $A$  we allow expressions of the form  $\zeta A.\varphi$ . We stipulate that the interpretation  $\llbracket \zeta A.\varphi \rrbracket$  is the set of those objects  $a$  such that  $\llbracket \varphi \rrbracket$  is non-empty when  $A$  is reinterpreted as  $\{a\}$ —see Section 4.1 below for details.

**Definition 4.** Our language  $\mathcal{L}^+$  is then defined by the following context-free grammar (again given a set  $\mathcal{A}$  of monadic predicates and  $\mathcal{R}$  of dyadic predicates):

$$\varphi ::= A \mid \varphi \wedge \varphi \mid \exists R.\varphi \mid \Downarrow \varphi \mid \zeta A.\varphi$$

As before, we use  $\Uparrow$  as an abbreviation for  $\Downarrow\Downarrow$ .<sup>11</sup>

One immediate observation is that using this new operator, in combination with the rest of  $\mathcal{L}$ , we can now define negation (construed as set complement):  $\neg\varphi$  is expressed by  $\zeta A.\Downarrow(A \wedge \varphi)$ . Note the linguistically relevant fact that negation here is not invoked as a primitive operation on monadic concepts (cf. Horn 1989). We are also able to perform the subject extraction mentioned above, and much else:

**Example 3.** The phrase *who Romeo loves* is simply  $\zeta A.(\text{Romeo} \wedge \exists \text{Loves}.A)$ . Likewise, *who Romeo loves and Desdemona despises* is  $\zeta A.(\text{Romeo} \wedge \exists \text{Loves}.A \wedge \Uparrow(\text{Desdemona} \wedge \exists \text{Despises}.A))$ . We can also express more complex concepts like *self-respecting* simply by  $\zeta A.(A \wedge \exists \text{Respect}.A)$ , so that, e.g., *self-respecting journalist* is rendered as  $\zeta A.(A \wedge \exists \text{Respect}.A) \wedge \text{Journalist}$ .

**Remark 9.** Operators like  $\zeta$  have been studied in an area of modal logic known as hybrid logic—see especially Blackburn and Seligman 1995; Goranko 1996—although the base system is usually taken to be a more expressive (multi)modal logic with negation, disjunction, etc., rather than a weak fragment like  $\mathcal{L}$ . For instance, the operator  $\zeta$  is closely related to a binding operator that Blackburn and Seligman (1995) (confusingly, given the notation adopted here from Pietroski 2018) call  $\Downarrow$ . It is in fact the strongest of the so-called binding operators they consider; a weaker operator that they call  $\downarrow$  would have (in our notation)  $\llbracket \downarrow_A \varphi \rrbracket$  correspond to the set of those objects  $a$  such that  $\llbracket \varphi \rrbracket$  is non-empty when  $A$  is mapped to  $a$ , for some way of reinterpreting all of the predicate variables *other than*  $A$ .

Although the semantics do not line up perfectly—specifically, the binding operators in hybrid logic all involve possible changes to other variables—we will draw on some of this work in the next section. Most notably, Goranko (1996) studies a hybrid logic that is shown to be equivalent to first-order logic. While the details of our argument differ in various respects, the broad idea is in the same spirit.

## 4.1 Semantics of SMPL<sup>+</sup>

To give a precise interpretation of  $\mathcal{L}^+$ , we again define a model to be a pair  $\mathcal{M} = (M, \llbracket \cdot \rrbracket)$ . But we will now also speak about  $\llbracket \varphi \rrbracket_g$  where  $g$  is a partial function from  $\mathcal{A}$  to  $M$ . That is,  $g$  will be a set of assignments

<sup>10</sup>Strictly speaking, the TARSKI operator in Pietroski (2018) was intended to be restricted to polarized monadic predicates.

<sup>11</sup>Were further succinctness desired, we could alternatively wrap  $\Downarrow$  and  $\zeta A$  into one abstraction operator  $\mu A$ , where the interpretation of  $\mu A.\varphi$  would give those objects  $a$  such that  $\varphi$  is *empty* when  $A$  is reinterpreted as  $\{a\}$ . This system, including just this operator,  $\exists R$  and conjunction, would be as expressive as  $\mathcal{L}^+$ .

$\{A_1 \mapsto a_1, \dots, A_n \mapsto a_n\}$ , with the property that each  $A \in \mathcal{A}$  appears at most once in this set. We say  $A \in \text{dom}(g)$  if  $g$  includes an assignment for  $A$ . Given  $g$ , we define  $g[A \mapsto a]$  as that function which adds the assignment  $A \mapsto a$ , thereby replacing whatever assignment to  $A \in \text{dom}(g)$  (if any) was present in  $g$ . We think of  $g[A \mapsto a]$  as overriding the assignment given by  $[[\ ]]$ , always mapping  $A$  to the point  $a$ . This is made precise in the base step of the recursive definition below:

$$\begin{aligned} [[A]]_g &= \begin{cases} \{g(A)\} & \text{if } A \in \text{dom}(g) \\ [[A]] & \text{if } A \notin \text{dom}(g) \end{cases} \\ [[\varphi \wedge \psi]]_g &= [[\varphi]]_g \cap [[\psi]]_g \\ [[\exists R.\varphi]]_g &= \{a \in M \mid \exists b : \langle a, b \rangle \in [[R]] \text{ and } y \in [[\varphi]]_g\} \\ [[\Downarrow\varphi]]_g &= \{a \in M \mid [[\varphi]]_g = \emptyset\} \\ [[\zeta A.\varphi]]_g &= \{a \in M \mid [[\varphi]]_{g[A \mapsto a]} \neq \emptyset\} \end{aligned}$$

Thus, in particular,  $[[\varphi]]_\emptyset = [[\varphi]]$  for  $\varphi$  in  $\mathcal{L}$  and  $\emptyset$  the empty function. This is the system  $\text{SMPL}^+$ .

## 4.2 Equivalence with First-Order Logic

We saw above in Example 3 that  $\text{SMPL}^+$  is quite a bit more expressive than  $\text{SMPL}$ . The natural question from a logical point of view is how much more expressive.

Remarkably,  $\text{SMPL}^+$  is equivalent to first-order logic, in the sense that these two systems define exactly the same complex concepts. Throughout this section let us assume a set  $\mathcal{A} = \{A, B, \dots\}$  of monadic predicate symbols and  $\mathcal{R} = \{R, Q, \dots\}$ , a set of dyadic predicate symbols. While  $\mathcal{L}^+$  is defined as above, these predicate symbols also define a signature for a first-order language  $\mathcal{L}^{FO}$ , which includes individual variables  $\mathcal{V} = \{x_0, x_1, x_2, \dots\}$  as well (and no constant or function symbols). A model  $\mathcal{M} = (M, [[\ ]])$  for  $\mathcal{L}^+$  can thereby also be considered a model for  $\mathcal{L}^{FO}$ , while a variable assignment is as usual a function  $f : \mathcal{V} \rightarrow M$ .

In order to compare  $\text{SMPL}^+$  and first-order logic, let us assume that  $\mathcal{L}^+$  comes with an augmented set of monadic predicates  $\mathcal{X} = \{X_i\}_{x_i \in \mathcal{V}}$ , one for each first order variable from  $\mathcal{L}^{FO}$ , in addition to the usual predicates of  $\mathcal{A}$ . Moreover let us assume that we only ever  $\zeta$ -bind using these symbols. That is, we allow  $\zeta X_i.\varphi$  but not  $\zeta A.\varphi$  for  $A \in \mathcal{A}$ . Then we can define the ‘‘free occurrences’’ of  $X_i$  in a formula  $\varphi \in \mathcal{L}^+$  analogously to how these are defined in first-order logic (see, e.g., Barwise 1977).

Given a first-order variable assignment  $f$  and a formula  $\varphi$ , we will write  $g_f$  to denote a corresponding assignment to elements of  $\mathcal{X}$ : we include an assignment  $X_i \mapsto f(x_i)$  for every  $X_i$  that occurs free in  $\varphi$ . Thus,  $g_f$  always assumes a contextually given set of variables, namely those free in a given formula.

We embed  $\mathcal{L}^+$  into  $\mathcal{L}^{FO}$  using an extension of the ‘‘standard translation’’ of modal logic into  $\mathcal{L}^{FO}$  (van Benthem, 1985). The translation  $\tau_x(\varphi)$  of a formula in  $\mathcal{L}^+$  is relative to a choice of variable  $x$ . Our goal will be to show that, for any first-order variable assignment  $f : \mathcal{V} \rightarrow M$ , and any formula  $\varphi \in \mathcal{L}^+$ , we have:

**Theorem 10.**  $\mathcal{M}, f \models \tau_x(\varphi)$  iff  $f(x) \in [[\varphi]]_{g_f}$ .

In other words, the objects  $a$  in the denotation of  $\varphi$  are exactly those that make  $\tau_x(\varphi)$  true when  $x$  is mapped to  $a$ . We define the family of translations  $\tau_x : \mathcal{L}^+ \rightarrow \mathcal{L}^{FO}$  simultaneously for all  $x \in \mathcal{V}$ :

$$\begin{aligned} \tau_x(A) &= A(x) \\ \tau_x(Y) &= x = y \\ \tau_x(\varphi \wedge \psi) &= \tau_x(\varphi) \wedge \tau_x(\psi) \\ \tau_x(\exists R.\varphi) &= \exists z (R(x, z) \wedge \tau_z(\varphi)) \\ \tau_x(\Downarrow\varphi) &= \neg \exists z \tau_z(\varphi) \\ \tau_x(\zeta Y.\varphi) &= \exists z \exists y (x = y \wedge \tau_z(\varphi)) \end{aligned}$$

where  $z \neq x, y$  is always chosen completely fresh; that is, we also assume its corresponding  $\mathcal{L}^+$  variable also does not occur anywhere in  $\varphi$ . Note that we thus use an unbounded number of variables in general. The proof of Theorem 10 appears in the appendix.

In the other direction, we translate  $\mathcal{L}^{FO}$  into  $\mathcal{L}^+$ . The translation  $\widehat{\alpha}$  of  $\alpha$  is given recursively. For atomic:

$$\begin{aligned} x_i = x_j &\mapsto \uparrow(X_i \wedge X_j) \\ R(x_i, x_j) &\mapsto \uparrow(X_i \wedge \exists R.X_j) \\ A(x_i) &\mapsto \uparrow(A \wedge X_i) \end{aligned}$$

For the Boolean connectives and the existential quantifier, suppose the translations  $\widehat{\alpha}$  and  $\widehat{\beta}$  of  $\alpha$  and  $\beta$ , respectively, are already defined. Then:

$$\begin{aligned} (\alpha \wedge \beta) &\mapsto \widehat{\alpha} \wedge \widehat{\beta} \\ \neg\alpha &\mapsto \zeta X_j. \downarrow(X_j \wedge \widehat{\alpha}) \\ \exists x_i \alpha &\mapsto \uparrow \zeta X_i. \widehat{\alpha} \end{aligned}$$

where  $X_j$  is a completely fresh monadic variable, i.e., not occurring anywhere in  $\widehat{\alpha}$ . We then claim:

$$\mathcal{M}, f \models \alpha \quad \text{implies} \quad [[\widehat{\alpha}]]_{g_f} = M \tag{2}$$

$$\mathcal{M}, f \not\models \alpha \quad \text{implies} \quad [[\widehat{\alpha}]]_{g_f} = \emptyset. \tag{3}$$

For the special case that  $\alpha$  has a single free variable  $x$ , we obtain, analogously to Theorem 10:

$$\mathcal{M}, f \models \alpha \quad \text{iff} \quad f(x) \in [[\zeta X.\widehat{\alpha}]]_{\emptyset}. \tag{4}$$

**Theorem 11.** First-order logic embeds into  $\text{SMPL}^+$ . That is, (2) and (3) above hold.

**Corollary 12.**  $\text{SMPL}^+$  and first-order logic are equally expressive as concept languages. Consequently, the problem of deciding whether two concepts in  $\text{SMPL}^+$  are equal is undecidable.

Perhaps surprisingly, adding only this limited  $\zeta$ -abstraction takes us from a decidable system of relatively low complexity to a highly expressive, undecidable system equivalent to first-order logic. There have been claims to the effect that natural language may well have at least the expressive power of first-order logic (Purdy, 1991; Pratt-Hartmann, 2004), and perhaps more (Rescher, 1962; Barwise and Cooper, 1981). So depending on one's perspective, Corollary 12 may be interpreted as either a feature or a bug.

**Remark 13.** One might conclude from Corollary 12 that there is no significant difference between  $\text{SMPL}^+$  and first-order logic. From the abstract perspective of what concepts these two systems can in principle express, this is true. However, it belies potentially significant *syntactic* differences between the two. As an example of this, for problems of concept learning it may be important that some concepts are shorter or simpler to express in one language than in another (see, e.g., Feldman 2016).

Another notable difference has to do with the formal-language-theoretic complexity of the representation languages, construed as classes of expressions. There is a precise sense in which  $\mathcal{L}^+$ , the language of  $\text{SMPL}^+$ , is easy to describe, and specifically it is easy to *generate*. In Definition 4, we presented  $\mathcal{L}^+$  with a very simple context-free grammar. This immediately shows that there is a compact and simple procedure for generating  $\mathcal{L}^+$  expressions. However, it is known that the first-order encodings of concepts—that is, the first-order formulas with exactly one free variable—is not a context-free set (van Benthem, 1987a; Marsh and Partee, 1987). So there can be no analogous generating procedure for these expressions.

This observation is of course closely related to the fact that the syntax of first-order logic allows for expressions like  $R(x, y) \wedge A(x)$ , which can then be turned into concepts, viz. unary predicates, by quantifying, e.g.,  $\exists y (R(x, y) \wedge A(x))$ . Disallowing expressions like  $R(x, y) \wedge A(x)$  was precisely one of the motivations behind the formulation of  $\text{SMPL}$  (and  $\text{SMPL}^+$ ) in the first place.

## 5 Conclusion and Further Directions

The goal of this paper has been to uncover some of the fundamental logical properties of the system  $\text{SMPL}$ , and its extension  $\text{SMPL}^+$ , as introduced and motivated by Pietroski (2018). As we have seen, both can be

understood precisely as (modal/description or hybrid) logical systems, with formal connections to existing proposals in the literature. The results here also help clarify further features of these representation languages that may be relevant in assessing their suitability for describing fundamental conceptual structure. We noted the conspicuous absence of `Some_not` from the expressive repertoire of SMPL. If we think something like SMPL describes a primitive core of conceptual thought, then perhaps the fact that `Some_not` is not lexicalized has something to do with its absence from this core repertoire. Meanwhile, the stronger system, SMPL<sup>+</sup>, takes us all the way to first-order expressible concepts, though it does so in a way that is syntactically simpler, viz. context-free generation (recall Remark 13).

We see the present contribution as only a first step in a more thorough investigation of systems related to SMPL and SMPL<sup>+</sup>. It is easy to imagine a number of different amendments, both weakenings and strengthenings. As an example of the latter, in the area of finite model theory researchers have found it useful to include devices for reasoning about the *transitive closure* of a relation. This might be by adding a fixpoint operator, or a transitive closure or reachability operator directly (see Grädel et al. 2007). Such a device could also be considered a part of core conceptual structure in humans. For instance, as soon as a person has grasped the concept of *parent* it seems only a small step to understand its transitive closure *ancestor*. More broadly, the idea of reachability along a relation seems to be a relatively fundamental cognitive construct. Modest extensions with basic numerical primitives would also be natural to consider (see, e.g., Moss 2016; van Benthem and Icard 2021). In the other direction, it could be enlightening to study even smaller fragments of these languages, including fragments that lie in between SMPL and SMPL<sup>+</sup>.

In sum, we hope to have given evidence that bringing these two lines of research together—philosophical and computational logic on the one hand, and cognitively motivated semantics and philosophy of language on the other—promises new insights and syntheses that may be worth more than the sum of their parts.

**Acknowledgements:** We would like to thank audiences at Indiana University and Stanford University for useful comments on this material. Thanks especially to Paul Pietroski for many discussions and very helpful comments and contributions on an earlier draft.

## References

- Abiteboul, S., Hull, R., and Vianu, V. (1995). *Foundations of Databases*. Addison-Wesley.
- Barwise, J. (1977). An introduction to first-order logic. *Studies in Logic and the Foundations of Mathematics*, 90:5–46.
- Barwise, J. and Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics & Philosophy*, 4:159–219.
- van Benthem, J. (1985). *Modal Logic and Classical Logic*. Bibliopolis.
- van Benthem, J. (1987a). Logical syntax. *Theoretical Linguistics*, 14(2-3):119–142.
- van Benthem, J. (1987b). Meaning: Interpretation and inference. *Synthese*, 73(3):451–470.
- van Benthem, J. and Icard, T. F. (2021). Interleaving logic and counting. ILLC Prepublication Series.
- Blackburn, P. and Seligman, J. (1995). Hybrid languages. *Journal of Logic, Language & Information*, 4(3):251–272.
- Brandom, R. B. (2000). *Articulating Reasons: An Introduction to Inferentialism*. Harvard University Press.
- Buch, S., Fei-Fei, L., and Goodman, N. (2021). Neural event semantics for grounded language understanding. In *Transactions of the Association for Computational Linguistics (ACL)*, volume 9.
- Carey, S. (2009). *The Origin of Concepts*. Oxford University Press.

- Chierchia, G. (1984). *Topics in the Syntax and Semantics of Infinitives and Gerunds*. PhD thesis, University of Massachusetts, Amherst.
- Chomsky, N. (1957). *Syntactic Structures*. Mouton & Co.
- Chomsky, N. (1959). On certain formal properties of grammars. *Information & Control*, 2:137–167.
- Davidson, D. (1966). The logical form of action sentences. In Rescher, N., editor, *The Logic of Decision and Action*. University of Pittsburgh Press.
- Feldman, J. (2016). The simplicity principle in perception and cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 7(5):330–340.
- Fodor, J. (1975). *The Language of Thought*. Harvard University Press.
- Goranko, V. (1996). Hierarchies of modal and temporal logics with reference pointers. *Journal of Logic, Language & Information*, 5(1):1–24.
- Grädel, E., Kolaitis, P. G., Libkin, L., Marx, M., Spencer, J., Vardi, M. Y., Venema, Y., and Weinstein, S. (2007). *Finite Model Theory and its Applications*. Springer.
- Grätzer, G. (2008). *Universal Algebra*. Springer.
- Hamilton, W. (1860). *Lectures on Logic*, volume 1. Blackwood.
- Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, University of Massachusetts, Amherst.
- Hemaspaandra, E. (1996). The price of universality. *Notre Dame Journal of Formal Logic*, 37(2):174–203.
- Hoeksema, J. (1999). Blocking effects and polarity sensitivity. In Gerbrandy, J., Marx, M., de Rijke, M., and Venema, Y., editors, *JFAK: Essays Dedicated to Johan van Benthem on the Occasion of his 50th Birthday*. Vossiuspers, Amsterdam: Amsterdam University.
- Holliday, W. H. and Icard, T. F. (2010). Moorean phenomena in epistemic logic. In Beklemishev, L., Goranko, V., and Shehtman, V., editors, *Advances in Modal Logic*, volume 8, pages 178–199. College Publications.
- Holliday, W. H. and Icard, T. F. (2018). Axiomatization in the meaning sciences. In Ball, D. and Rabern, B., editors, *The Science of Meaning: Essays on the Metatheory of Natural Language Semantics*, pages 73–97. Oxford University Press.
- Horn, L. (1989). *A Natural History of Negation*. University of Chicago Press.
- Icard, T. F. and Moss, L. S. (2014). Recent progress on monotonicity. *Linguistic Issues in Language Technology*, 9(7):167–194.
- Joshi, A. K., Shanker, K. V., and Weir, D. (1991). The convergence of mildly context-sensitive grammar formalisms. In Shieber, S. M., Sells, P., and Wasow, T., editors, *Foundational Issues in Natural Language Processing*, pages 31–81. MIT Press.
- Kamp, H. (1981). A theory of truth and semantic representation. In Groenendijk, J., Janssen, T., and Stokhof, M., editors, *Formal methods in the Study of Language*, page 277–322. Mathematical Centre Tracts 135, Amsterdam: Mathematisch Centrum.
- Keenan, E. (2004). Excursions in natural logic. In Casadio, C., Scott, P. J., and Seely, R. A. G., editors, *Language and Grammar: Studies in Mathematical Linguistics and Natural Language*, pages 31–52. CSLI, Stanford.

- Keenan, E. L. and Stavi, J. (1985). A semantic characterization of natural language determiners. *Linguistics & Philosophy*, 9(3):253–326.
- Knowlton, T., Pietroski, P., Halberda, J., and Lidz, J. (2021). The mental representation of universal quantifiers. *Linguistics and Philosophy*. forthcoming.
- Kurtonina, N. and de Rijke, M. (1999). Expressiveness of concept expressions in first-order description logics. *Artificial Intelligence*, 107:303–333.
- Liang, P., Jordan, M. I., and Klein, D. (2013). Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.
- Lidz, J. (2018). The explanatory power of linguistic theory. In Hornstein, N., Lasnik, H., Patel-Grosz, P., and Yang, C., editors, *Syntactic Structures after 60 Years*, pages 85–91. De Gruyter Mouton.
- Malink, M. (2013). *Aristotle’s Modal Syllogistic*. Harvard University Press.
- Marsh, W. and Partee, B. H. (1987). How non-context free is variable binding? In Savitch, W. J., Bach, E., and Safran-Naveh, W. M. G., editors, *The Formal Complexity of Natural Language*. Springer Studies in Linguistics and Philosophy.
- Montague, R. (1974). *Formal Philosophy*. Yale University Press.
- Moss, L. S. (2008). Completeness theorems for syllogistic fragments. In Hamm, F. and Kepser, S., editors, *Logics for Linguistics Structures*, pages 143–173. De Gruyter.
- Moss, L. S. (2010). Natural logic and semantics. In Aloni, M., Bastiaanse, H., de Jager, T., and Schulz, K., editors, *Logic, Language, and Meaning*, pages 84–93. Springer.
- Moss, L. S. (2016). Syllogistic logic with cardinality comparisons. In Bimbó, K., editor, *J. Michael Dunn on Information Based Logics*, pages 391–415. Springer.
- Muskens, R. A. (2007). Higher order modal logic. In Blackburn, P., van Benthem, J. F. A. K., and Wolter, F., editors, *Handbook of Modal Logic*, pages 621–653. Elsevier.
- Papadimitriou, C. H. (1994). *Computational Complexity*. Addison-Wesley.
- Parsons, T. (1990). *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press.
- Partee, B. (1975). Montague grammar and transformational grammar. *Linguistic Inquiry*, 6(2):203–300.
- Piantadosi, S. T. (2021). The computational origin of representation. *Minds and Machines*, 31(1):1–58.
- Pietroski, P. M. (2005). *Events and Semantic Architecture*. Oxford University Press.
- Pietroski, P. M. (2018). *Conjoining Meanings: Semantics without Truth Values*. Oxford University Press.
- Pietroski, P. M. (2019). Semantic types: Two is better than too many. In *New Frontiers in Artificial Intelligence: JSAI-isAI International Workshops*, pages 148–163.
- Pratt-Hartmann, I. (2004). Fragments of language. *Journal of Logic, Language & Information*, 13(2):207–233.
- Pratt-Hartmann, I. and Moss, L. S. (2009). Logics for the relational syllogistic. *Review of Symbolic Logic*, 2(4):647–683.
- Pullum, G. (1983). Context-freeness and the computer processing of human languages. In *Proceedings of the 21st annual meeting on Association for Computational Linguistics (ACL ’83)*, pages 1–6.

- Purdy, W. C. (1991). A logic for natural language. *Notre Dame Journal of Formal Logic*, 32:409–425.
- Quine, W. V. O. (1948). On what there is. *The Review of Metaphysics*, 2(1):21–38.
- Rescher, N. (1962). Plurality quantification. *Journal of Symbolic Logic*, 27:373–374.
- Ristad, S. (1993). *The Language Complexity Game*. MIT Press.
- Sánchez-Valencia, V. (1991). *Studies on Natural Logic and Categorical Grammar*. PhD thesis, Universiteit van Amsterdam.
- Sbardolini, G. (2021). Assertion, rejection, and semantic universals. In Ghosh, S. and Icard, T., editors, *Logic, Rationality, and Interaction. LORI 2021*, volume 13039 of *Lecture Notes in Computer Science*. Springer.
- Shepard, R. N., Hovland, C. I., and Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs*, 75(13):1–42.
- Steinert-Threlkeld, S. and Szymanik, J. (2019). Learnability and semantic universals. *Semantics & Pragmatics*, 12.
- Szymanik, J. (2016). *Quantifiers and Cognition: Logical and Computational Perspectives*. Springer.
- Tarski, A. (1944). The semantic conception of truth. *Philosophy & Phenomenological Research*, 4:341–375.
- Zylberberg, A., Dehaene, S., Roelfsema, P. R., and Sigman, M. (2011). The human Turing machine: a neural framework for mental programs. *Trends in Cognitive Science*, 15(7):293–300.

## Appendix

In this technical appendix we give detailed proofs of all results in the main text.

### Proof of Theorem 2

To establish Theorem 2 we need to show how any question about satisfiability of a propositional formula  $\alpha$  can be reduced to a question about whether  $\text{tr}(\alpha)$  has a model in which it is non-empty. Let us abbreviate the latter property by saying that  $\text{tr}(\alpha)$  “has a model.” The first point to notice is that  $\text{tr}(\alpha)$  is always built up using monadic predicates,  $\wedge$ , and  $\Downarrow$ ; in particular it never includes any subformula  $\exists R.\varphi$ . Such formulas have a special feature:

**Lemma 14.** Suppose  $\varphi \in \mathcal{L}$  is built up using only monadic predicates in  $\mathcal{A}$ ,  $\wedge$ , and  $\Downarrow$ . Then if  $\varphi$  has a model, it has a model with just one point.

*Proof.* Let  $(M, \llbracket \cdot \rrbracket)$  be a model such that  $\llbracket \varphi \rrbracket \neq \emptyset$ . Pick a point  $a \in \llbracket \varphi \rrbracket$  and consider a new model  $(\{a\}, \llbracket \cdot \rrbracket_a)$  where  $\llbracket A \rrbracket_a = \llbracket A \rrbracket \cap \{a\}$  for all  $A \in \mathcal{A}$ . We then show that  $\llbracket \varphi \rrbracket_a = \{a\}$  by induction on  $\varphi$ .  $\dashv$

To finish the proof of Theorem 2, consider  $\alpha \in \mathcal{L}^{prop}$ . We know that  $\alpha$  is satisfiable iff  $\text{tr}(\alpha)$  has a model: if  $\alpha$  is satisfiable, then by (1) we know  $\text{tr}(\alpha)$  has a (one-point) model; and if  $\text{tr}(\alpha)$  has a model, by Lemma 14 it has a one-point model, which by (1) again means that  $\alpha$  is satisfiable. Because the length of  $\text{tr}(\alpha)$  involves only a polynomial increase in length, we have achieved a polynomial reduction of propositional satisfiability to the problem whether a formula in  $\mathcal{L}$  has a model.

Finally, to check whether two concepts are distinct is at least as difficult as checking whether a given concept has a model. In particular, to check that  $\varphi \in \mathcal{L}$  has a model it suffices to know that  $\varphi \neq \perp$ , i.e., there is a model in which  $\llbracket \varphi \rrbracket \neq \llbracket \perp \rrbracket = \emptyset$ .

## Soundness Proof (Theorem 6)

For every model  $\mathcal{M}$ , if  $\mathcal{M}$  satisfies the hypotheses of one of our rules (the sentence(s) above the line in Figure 2), then  $\mathcal{M}$  also satisfies the conclusion (below). For the rules pertaining to  $\wedge$ , this amounts to basic facts about set intersections. We shall check this *soundness* fact for the rest of the rules.

Fix the model  $\mathcal{M}$ . We write  $M$  for the domain of this model.

First, assume that  $\mathcal{M} \models \psi \leq \varphi$ . This means that  $[[\psi]] \subseteq [[\varphi]]$ . To see that  $[[\Downarrow\varphi]] \subseteq [[\Downarrow\psi]]$ , we may assume that  $[[\Downarrow\varphi]] \neq \emptyset$ . Thus, there is some  $x \in M$  such that  $[[\varphi]] = \emptyset$ . For this same  $x$  (or indeed for any  $x$ ),  $[[\psi]] = \emptyset$  as well. Thus  $[[\Downarrow\psi]] = M$ . So we are done.

Second, we check that  $[[\varphi]] \subseteq [[\top]] \cap [[\Uparrow\varphi]]$ . Again, we may assume that  $[[\varphi]] \neq \emptyset$ . In this case,  $[[\Uparrow\varphi]] = M$ , and  $[[\top]] = M$  always. So  $[[\top]] \cap [[\Uparrow\varphi]] = M$ , and this is a superset of every set.

Third, we check that  $[[\varphi \wedge \Downarrow\varphi]]$  is independent of  $\varphi$ ; indeed, it is always  $\emptyset$ . For if  $[[\varphi]] = \emptyset$ , then this is clear. And if  $[[\varphi]] \neq \emptyset$ , then  $[[\Downarrow\varphi]] = \emptyset$ . So in this case,  $[[\varphi \wedge \Downarrow\varphi]] = [[\varphi]] \cap \emptyset = \emptyset$ .

Fourth, we check that  $[[\Downarrow\Downarrow\Downarrow\varphi]]$  and  $[[\Downarrow\varphi]]$  are the same set. If  $[[\varphi]] = \emptyset$ , then  $[[\Downarrow\varphi]] = M$ . So  $[[\Downarrow\Downarrow\Downarrow\varphi]] = \emptyset$  (since  $M \neq \emptyset$ ), and thus  $[[\Downarrow\Downarrow\Downarrow\varphi]] = M$ . In the other case,  $[[\varphi]] \neq \emptyset$ . This time,  $[[\Downarrow\varphi]] = \emptyset$ . So  $[[\Downarrow\Downarrow\Downarrow\varphi]] = M$ , and  $[[\Downarrow\Downarrow\Downarrow\varphi]] = \emptyset$ . Either way, we have shown what we want.

In the fifth rule, we assume that  $[[\varphi]] \subseteq [[\psi]]$ , and we prove that  $[[\exists R.\varphi]] \subseteq [[\exists R.\psi]]$ . For this, let  $x \in [[\exists R.\varphi]]$ . Then there is some  $y$  such that  $\langle x, y \rangle \in [[R]]$  and  $y \in [[\varphi]]$ . Since  $y \in [[\varphi]]$ , we also have  $y \in [[\psi]]$ . And thus  $y$  is a witness showing that  $x \in [[\exists R.\psi]]$ .

In the final rule, suppose that  $x \in [[\exists R.\varphi]]$ . Let  $y \in [[\varphi]]$  be such that  $\langle x, y \rangle \in [[R]]$ . Then  $y$  shows that  $[[\varphi]] \neq \emptyset$ . Thus,  $[[\Uparrow\varphi]] = M$ . And from this, we know that  $[[\exists R.\varphi]] \subseteq [[\Uparrow\psi]]$ .

At this point, we have checked the soundness of the individual rules. We show by induction on formal proofs that if  $\Gamma \vdash \varphi = \psi$ , then for all models  $\mathcal{M}$ , such that  $\mathcal{M} \models \Gamma$ ,  $[[\varphi]] = [[\psi]]$ . The base case of the induction is when  $\Gamma$  itself contains  $\varphi = \psi$ ; this is immediate. We have induction steps for the rules in Figure 2—these follow from our work just above—and for the (CASES) rule. For this, assume that we have a derivation  $\Gamma \vdash \varphi = \psi$  justified by (CASES) at the root. We have  $\Gamma \cup \{\chi = \perp\} \vdash \varphi$  and also  $\Gamma \cup \{\chi \neq \perp\} \vdash \varphi = \psi$ . Let  $\mathcal{M} \models \Gamma$ . We show that  $[[\varphi]] = [[\psi]]$  in  $\mathcal{M}$ . There are two cases:  $[[\chi]] = \emptyset$ , and  $[[\chi]] \neq \emptyset$ . In the first case, we recall our assumption that  $\Gamma \cup \{\chi = \perp\} \vdash \varphi = \psi$  by a proof shorter than the given proof. So by induction hypothesis,  $\Gamma \cup \{\chi = \perp\} \models \varphi = \psi$ . Our model  $\mathcal{M}$  satisfies all sentences in  $\Gamma \cup \{\chi = \perp\}$ , hence it does satisfy  $\varphi = \psi$ . In the other case, when  $[[\chi]] \neq \emptyset$ , we have  $[[\Uparrow\chi]] = M$ . So  $\mathcal{M}$  will satisfy all sentences in  $\Gamma \cup \{\chi \neq \perp\}$  in this case. And again, the induction hypothesis implies that  $\mathcal{M} \models \varphi = \psi$ .

## Completeness Proof (Theorem 6)

We first record some useful facts about the proof calculus.

**Lemma 15.** The following are all guaranteed by our calculus:

- |  |  |
|--|--|
| 1. $\perp \leq \varphi$  | 6. $\Uparrow\Downarrow\varphi = \Downarrow\Uparrow\varphi = \Downarrow\Downarrow\Downarrow\varphi$ . |
| 2. If $\varphi \leq \psi$ and $\psi \leq \chi$ , then $\varphi \leq \chi$ .                          | 7. $\Uparrow\Uparrow\varphi = \Uparrow\varphi$ .   |
| 3. If $\varphi \leq \psi_1$ and $\varphi \leq \psi_2$ , then $\varphi \leq (\psi_1 \wedge \psi_2)$ . | 8. $\Uparrow\top = \top$ .   |
| 4. $\top \wedge \varphi = \varphi$ .   | 9. $\Downarrow\top = \perp$ .  |
| 5. If $\varphi \leq \psi$ , then $\Uparrow\varphi \leq \Uparrow\psi$ .                               | 10. $\Uparrow\perp = \perp$ .  |

*Proof.* For (1), we use all three rules for  $\wedge$ , and the rule that allows us to replace  $\perp$  with  $\psi \wedge \Downarrow\psi$  for any  $\psi$



whatsoever (including  $\varphi$ ):

$$\begin{aligned}
\perp \wedge \varphi &= (\varphi \wedge \Downarrow\varphi) \wedge \varphi \\
&= (\Downarrow\varphi \wedge \varphi) \wedge \varphi \\
&= \Downarrow\varphi \wedge (\varphi \wedge \varphi) \\
&= \Downarrow\varphi \wedge \varphi \\
&= \varphi \wedge \Downarrow\varphi \\
&= \perp
\end{aligned}$$

For (2),  $\varphi \wedge \chi = (\varphi \wedge \psi) \wedge \chi = \varphi \wedge (\psi \wedge \chi) = \varphi \wedge \psi = \varphi$ .

For (3), let  $\varphi \wedge \psi_1 = \varphi$  and  $\varphi \wedge \psi_2 = \varphi$ . Then  $\varphi \wedge (\psi_1 \wedge \psi_2) = (\varphi \wedge \psi_1) \wedge \psi_2 = \varphi \wedge \psi_2 = \varphi$ .

For (4),  $\top \wedge \varphi \leq \varphi$  because  $(\top \wedge \varphi) \wedge \varphi = \top \wedge (\varphi \wedge \varphi) = \top \wedge \varphi$ . And in the other direction  $\varphi \leq \top$  and  $\varphi \leq \varphi$ , so  $\varphi \leq \top \wedge \varphi$  by part (3).

For (5), assume that  $\varphi \leq \psi$ . We know that  $\Downarrow\psi \leq \Downarrow\varphi$ . So  $\Downarrow\Downarrow\varphi \leq \Downarrow\Downarrow\psi$ . Since  $\Downarrow\Downarrow\varphi = \Uparrow\varphi$ , and similarly for  $\psi$ , we see that  $\Uparrow\varphi \leq \Uparrow\psi$ .

(6) is an immediate consequence of the definition of  $\Uparrow\varphi$  as  $\Downarrow\Downarrow\varphi$ .

For (7),  $\Uparrow\Uparrow\varphi = \Downarrow\Downarrow(\Downarrow\Downarrow\varphi) = \Downarrow(\Downarrow\Downarrow\Downarrow\varphi) = \Downarrow\Downarrow\varphi = \Uparrow\varphi$ .

For (8), use (6) to calculate:  $\Uparrow\top = \Uparrow\Downarrow\perp = \Downarrow\perp = \top$ . Another proof:  $\top \leq \Uparrow\top \leq \top$ .

For (9), first note that  $\top \wedge \Downarrow\top = \perp$ , by definition of  $\perp$  and also using the law  $\varphi \wedge \Downarrow\varphi = \psi \wedge \Downarrow\psi$ . In addition, by (4),  $\top \wedge \Downarrow\top = \Downarrow\top$ . Thus  $\Downarrow\top = \top \wedge \Downarrow\top = \perp$ .

For (10),  $\Uparrow\perp = \Downarrow\Downarrow\perp = \Downarrow\top = \perp$ . We used the definition of  $\top$  as  $\Downarrow\perp$  and also part (9). Another proof:  $\perp \leq \top$ , and so  $\top = \Uparrow\top \leq \Uparrow\perp$ , using part (8). But then  $\Uparrow\perp = \Uparrow\perp \wedge \top = \top$ , using part (4) and the definition of  $\leq$ .  $\dashv$

**Proof of Theorem 6 and Corollary 7.** At this point we fix a set  $\Gamma$  and an equation  $\varphi^* = \psi^*$  such that  $\Gamma \not\vdash \varphi^* = \psi^*$ . We show that there is a model  $\mathcal{N}$  of  $\Gamma$  where  $\llbracket \varphi^* \rrbracket \neq \llbracket \psi^* \rrbracket$ . Going forward, we shall assume that  $\Gamma$  is a *finite* set, so that we can establish the polynomial-time decidability of the consequence relation. That is, we shall find a model  $\mathcal{N}$  whose size is polynomial in the size of  $\Gamma \cup \{\varphi^*, \psi^*\}$ . The same proof, minus all of the finiteness considerations, proves the completeness of the logic. Since the completeness is easier, we omit the details.

**Definition 5.** Fix  $\Gamma$ ,  $\varphi^*$  and  $\psi^*$ . We take the set of *relevant predicates* to be those  $\chi$  which occur in  $\Gamma$ ,  $\varphi^*$ , or  $\psi^*$ . On the assumption that  $\Gamma$  is finite, there are only finitely many relevant predicates, and indeed the number of them is polynomial in the size of  $\Gamma \cup \{\varphi^* = \psi^*\}$ . Every sub-concept of a relevant predicate is itself relevant. List the relevant predicates as  $\chi_1, \dots, \chi_{K-1}$ .

**Expanding  $\Gamma$**  We expand  $\Gamma$  to a larger set of equations which also does not derive  $\varphi^* = \psi^*$ , and which has the following additional property: for all relevant predicates  $\chi$ , either  $\Gamma$  contains  $\chi = \perp$ , or  $\Gamma$  contains  $\chi \neq \perp$ . We do this by a step-by-step construction, building sets  $\Gamma_n$  for  $0 \leq n \leq K$ . We start with  $\Gamma_0 = \Gamma$ . For each  $n$ , let  $\chi_n$  be the  $n$ th relevant predicate. If  $\Gamma_n \cup \{\chi_n = \perp\} \not\vdash \varphi^* = \psi^*$ , we set  $\Gamma_{n+1}$  to be  $\Gamma_n \cup \{\chi_n = \perp\}$ . Otherwise,  $\Gamma_n \cup \{\chi_n = \perp\} \vdash \varphi^* = \psi^*$ , and we must have  $\Gamma_n \cup \{\chi_n \neq \perp\} \not\vdash \varphi^* = \psi^*$ . [For if not,  $\Gamma_n \vdash \varphi^* = \psi^*$  by (CASES).] We set  $\Gamma_{n+1}$  to be  $\Gamma_n \cup \{\chi_n \neq \perp\}$  in this case. This builds sets  $\Gamma = \Gamma_0 \subseteq \Gamma_1 \subseteq \dots \subseteq \Gamma_K$ .

**Replace  $\Gamma$  by  $\Gamma_K$**  Note that  $\Gamma$  and  $\Gamma_K$  have the same predicates. So the notion of a relevant predicate is the same if we expand  $\Gamma$  to  $\Gamma_K$ . Moreover,  $\Gamma_K \not\vdash \varphi^* = \psi^*$ . The rest of this proof produces a model  $\mathcal{N}$  of  $\Gamma_K$  where  $\llbracket \varphi^* \rrbracket \neq \llbracket \psi^* \rrbracket$ , and such a model  $\mathcal{N}$  is a model of  $\Gamma$  as well. To save on notation, we simply replace our original set  $\Gamma$  with  $\Gamma_K$ . In other words, we assume that the original set  $\Gamma$  came to us expanded as in the last paragraph.

**A point on  $\Gamma$**  We claim that  $\Gamma \not\vdash \top \leq \perp$ . For if  $\Gamma \vdash \top \leq \perp$ , then for all  $\chi$ ,  $\Gamma \vdash \perp \leq \chi \leq \top \leq \perp$ , and so  $\Gamma \vdash \chi = \perp$ . In particular, for the concepts  $\varphi^*$  and  $\psi^*$  which we fixed above, we would have  $\Gamma \vdash \varphi^* = \perp = \psi^*$ , contrary to what we assumed at the outset.

**The model  $\mathcal{N}$**  Let

$$N = \{\varphi : \varphi \text{ is a relevant predicate, and } \Gamma \vdash \varphi \neq \perp\}. \quad (5)$$

We interpret our language on this set  $N$  as follows: For a monadic predicate  $A$ ,

$$\llbracket A \rrbracket = \{\varphi \in N : \Gamma \vdash \varphi \leq A\}. \quad (6)$$

For each binary  $R$ , we set

$$\llbracket R \rrbracket = \{\langle \varphi, \psi \rangle \in N \times N : \Gamma \vdash \varphi \leq \exists R.\psi\}. \quad (7)$$

This turns our set  $N$  into a model  $\mathcal{N}$ . Please note that we need not have  $\exists R.\psi$  in  $N$  in order to have  $\langle \varphi, \psi \rangle \in \llbracket R \rrbracket$  in the model.

**Lemma 16** (Truth Lemma for  $\mathcal{N}$ ). For all relevant  $\psi$  (not necessarily in  $N$ ),

$$\llbracket \psi \rrbracket = \{\varphi \in N : \Gamma \vdash \varphi \leq \psi\}. \quad (8)$$

*Proof.* By induction on  $\psi$ . For  $\psi$  a basic predicate  $A$ , the statement in (8) is the definition in (6).

Consider a conjunction  $\psi$ , say  $\psi_1 \wedge \psi_2$ , where  $\psi \in N$  and therefore  $\psi_1$  and  $\psi_2$  also belong to  $N$ . We reason as follows:

$$\begin{aligned} \llbracket \psi_1 \wedge \psi_2 \rrbracket &= \llbracket \psi_1 \rrbracket \cap \llbracket \psi_2 \rrbracket \\ &= \{\varphi \in N : \Gamma \vdash \varphi \leq \psi_1\} \cap \{\varphi \in N : \Gamma \vdash \varphi \leq \psi_2\} \\ &= \{\varphi \in N : \Gamma \vdash \varphi \leq \psi_1 \text{ and } \Gamma \vdash \varphi \leq \psi_2\} \\ &= \{\varphi \in N : \Gamma \vdash \varphi \leq \psi_1 \wedge \psi_2\} \end{aligned}$$

where the last line follows from Lemma 15, part (3).

The next induction step is for a relevant predicate  $\Downarrow\psi$ . Assume (8) for  $\psi$ . Since  $\Downarrow\psi$  is relevant, so is  $\psi$ . This time we prove that

$$\llbracket \Downarrow\psi \rrbracket = \{\varphi \in N : \Gamma \vdash \varphi \leq \Downarrow\psi\}. \quad (9)$$

The two cases are:  $\Gamma \vdash \psi = \perp$ , and  $\Gamma \vdash \psi \neq \perp$ . (Notice that we used the fact that  $\Gamma$  contains either  $\psi = \perp$  or  $\psi \neq \perp$ . We arranged this by expanding  $\Gamma$ .)

In the first case,  $\Gamma \vdash \psi = \perp$ . We claim that  $\llbracket \psi \rrbracket = \emptyset$ . [Here is the proof: If  $\chi \in \llbracket \psi \rrbracket$ , then we use the induction hypothesis to see that  $\Gamma \vdash \tau \leq \uparrow\chi \leq \uparrow\psi \leq \uparrow\perp = \perp$ . And this contradicts our earlier assumption that  $\Gamma \not\vdash \tau \leq \perp$ .] Since  $\llbracket \psi \rrbracket = \emptyset$ ,  $\llbracket \Downarrow\psi \rrbracket = N$ . We evaluate the set on the right in (9). Let  $\varphi \in N$ . Then  $\Gamma \vdash \varphi \leq \tau = \Downarrow\perp = \Downarrow\psi$ . This shows that the set on the right in (9) is all of  $N$ , as desired.

In the second case,  $\psi \in N$ , and indeed  $\psi \in \llbracket \psi \rrbracket$  by induction hypothesis. And so  $\llbracket \Downarrow\psi \rrbracket = \emptyset$ . We claim that the set on the right in (9) is empty. For assume not, and let  $\varphi \in N$  have  $\Gamma \vdash \varphi \leq \Downarrow\psi$ . Since  $\varphi \in N$ ,  $\Gamma \vdash \tau \leq \uparrow\varphi$ . We also have  $\Gamma \vdash \tau \leq \uparrow\psi$ , and therefore  $\Downarrow\uparrow\psi \leq \Downarrow\tau = \perp$ . Suppose towards a contradiction that  $\Gamma \vdash \varphi \leq \Downarrow\psi$ . Then

$$\Gamma \vdash \tau \leq \uparrow\varphi \leq \uparrow\Downarrow\psi = \Downarrow\uparrow\psi \leq \perp.$$

But recall that we are assuming about  $\Gamma$  that  $\Gamma \not\vdash \tau \leq \perp$ . This contradiction confirms that the set on the right in (9) is empty in this case.

The last induction step is for  $\exists R$ . Assume that  $\exists R.\psi$  is relevant. So  $\psi$  also is relevant. We have (8) for  $\psi \in N$ , and prove that

$$\llbracket \exists R.\psi \rrbracket = \{\varphi \in N : \Gamma \vdash \varphi \leq \exists R.\psi\}. \quad (10)$$

First, let  $\varphi \in \llbracket \exists R.\psi \rrbracket$ . Then there is some  $\chi \in \llbracket \psi \rrbracket$  such that  $\langle \varphi, \chi \rangle \in \llbracket R \rrbracket$ . Then  $\chi \in N$  and  $\Gamma \vdash \varphi \leq \exists R.\chi$ . By induction hypothesis on  $\psi$ ,  $\Gamma \vdash \chi \leq \psi$ . Hence by the logic,  $\Gamma \vdash \exists R.\chi \leq \exists R.\psi$ . Thus,  $\Gamma \vdash \varphi \leq \exists R.\psi$ . This is half of (10).

In the other direction, let  $\varphi \in N$  have  $\Gamma \vdash \varphi \leq \exists R.\psi$ . Since  $\varphi \in N$ ,  $\Gamma \vdash \tau \leq \uparrow\varphi$ . By the monotonicity law in the logic,  $\Gamma \vdash \uparrow\varphi \leq \uparrow\exists R.\psi$ . Using the logic,  $\Gamma \vdash \uparrow\exists R.\psi \leq \uparrow\psi$ . Putting these together,  $\Gamma \vdash \tau \leq \uparrow\psi$ . Since  $\psi$  is relevant, we now have  $\psi \in N$ . We have  $\Gamma \vdash \psi \leq \psi$ , and so by induction hypothesis,  $\psi \in \llbracket \psi \rrbracket$ . Since  $\varphi$  and  $\psi$  belong to  $N$  and  $\llbracket R \rrbracket$  is given by (7), we have  $\langle \varphi, \psi \rangle \in \llbracket R \rrbracket$ . Thus,  $\varphi \in \llbracket \exists R.\psi \rrbracket$ , as desired.

This completes the proof.  $\dashv$

**Lemma 17.**  $\mathcal{N}$  satisfies every equation in  $\gamma = \delta$  in  $\Gamma$ . That is,  $\mathcal{N} \models \Gamma$ .

*Proof.* Fix such an equation, and note that  $\gamma$  and  $\delta$  are relevant by Definition 5. Let  $\varphi \in \llbracket \gamma \rrbracket$ . Then  $\varphi \in N$ , and by the Truth Lemma for  $\mathcal{N}$ ,  $\Gamma \vdash \varphi \leq \gamma$ . But also  $\Gamma \vdash \gamma \leq \delta$ , and so  $\Gamma \vdash \varphi \leq \delta$ . Since  $\varphi \in N$ , we use the Truth Lemma for  $\mathcal{N}$  again, this time to see that  $\varphi \in \llbracket \delta \rrbracket$ . This for all  $\varphi$  shows that  $\llbracket \gamma \rrbracket \subseteq \llbracket \delta \rrbracket$ . The converse is similar, and we conclude that  $\mathcal{N} \models \gamma = \delta$ .  $\dashv$

**Lemma 18.** For all predicates  $\gamma$ , if  $\Gamma \vdash \gamma = \perp$ , then  $\llbracket \gamma \rrbracket = \emptyset$ . For all relevant predicates  $\gamma$ , if  $\Gamma \vdash \gamma \neq \perp$ , then  $\llbracket \gamma \rrbracket \neq \emptyset$ .

*Proof.* The first assertion just comes from the fact that  $\llbracket \perp \rrbracket = \emptyset$  in every model of  $\Gamma$ . And since we know from Lemma 17 that  $\mathcal{N}$  is a model of  $\Gamma$ , we see from soundness that  $\llbracket \gamma \rrbracket = \llbracket \perp \rrbracket = \emptyset$ .

For the second assertion, let  $\gamma$  be relevant with  $\Gamma \vdash \gamma \neq \perp$ . Then  $\gamma \in N$ . By the Truth Lemma for  $\mathcal{N}$  and the fact that  $\Gamma \vdash \gamma \leq \gamma$ , we see that  $\gamma \in \llbracket \gamma \rrbracket$ . In particular,  $\llbracket \gamma \rrbracket \neq \emptyset$ .  $\dashv$

In the next lemma, recall that our standing assumption in this proof is that  $\Gamma \not\vdash \varphi^* = \psi^*$ .

**Lemma 19.**  $\mathcal{N} \not\models \varphi^* = \psi^*$ .

*Proof.* Before we start, let us recall the definition of  $N$  in (5) and also Definition 5. Note that both  $\varphi^*$  and  $\psi^*$  are relevant. Recall our construction began by arranging that for relevant  $\chi$ , either  $\Gamma$  contains (and thus derives)  $\chi = \perp$  or else  $\Gamma \vdash \chi \neq \perp$ . We thus have four cases, depending on whether  $\Gamma \vdash \varphi^* = \perp$  or  $\Gamma \vdash \varphi^* \neq \perp$ , and similarly for  $\psi$ . In two of these cases, we shall show that  $\mathcal{N} \not\models \varphi^* = \psi^*$ , and in the other two we derive a contradiction to the standing assumption in this completeness theorem that  $\Gamma \vdash \varphi^* \neq \psi^*$ .

First, if  $\Gamma \vdash \varphi^* = \perp$  and also  $\Gamma \vdash \psi^* = \perp$ , then we easily have our contradiction  $\Gamma \vdash \varphi^* = \psi^*$ .

Second, suppose that  $\Gamma \vdash \varphi^* = \perp$  but that  $\Gamma \vdash \psi^* \neq \perp$ . In this case, Lemma 18 shows that in  $\mathcal{N}$ ,  $\llbracket \varphi^* \rrbracket = \emptyset$  and  $\llbracket \psi^* \rrbracket \neq \emptyset$ . This tells us that  $\mathcal{N} \models \varphi^* \neq \psi^*$ , as desired.

Mutatis mutandis, we obtain the same conclusion  $\mathcal{N} \models \varphi^* \neq \psi^*$  in the case that  $\Gamma \vdash \varphi^* \neq \perp$  but  $\Gamma \vdash \psi^* = \perp$ .

Finally, suppose that  $\Gamma \vdash \varphi^* \neq \perp$  and also  $\Gamma \vdash \psi^* \neq \perp$ . Thus, both  $\varphi^*$  and  $\psi^*$  belong to the set  $N$  which underlies our model. Since  $\Gamma \vdash \varphi^* \leq \varphi^*$ , the Truth Lemma implies that  $\varphi^* \in \llbracket \varphi^* \rrbracket$ . Similarly  $\psi^* \in \llbracket \psi^* \rrbracket$ . Suppose towards a contradiction that  $\mathcal{N} \models \varphi^* = \psi^*$ . Since  $\llbracket \varphi^* \rrbracket = \llbracket \psi^* \rrbracket$ ,  $\varphi^* \in \llbracket \psi^* \rrbracket$  and  $\psi^* \in \llbracket \varphi^* \rrbracket$ . By the Truth Lemma again,  $\Gamma \vdash \varphi^* \leq \psi^* \leq \varphi^*$ . This contradicts  $\Gamma \not\vdash \varphi^* = \psi^*$ .  $\dashv$

**Concluding the proof of completeness** We have shown that if  $\Gamma \not\vdash \varphi^* = \psi^*$ , then there is a model of  $\Gamma$  where  $\varphi^* = \psi^*$  is false. This is the completeness of the proof system.

**On complexity** Our foregoing work also shows that if  $\Gamma$  has any model whatsoever in which  $\varphi^* = \psi^*$  fails, then it has such a model whose universe is a subset of  $S$ , where  $S$  is the set of predicates that appear in  $\Gamma \cup \{\varphi^* = \psi^*\}$ . And the size of  $S$  is polynomially related to the size of  $\Gamma \cup \{\varphi^* = \psi^*\}$ . This is behind the complexity assertion that the relation “ $\Gamma \not\vdash \varphi^* = \psi^*$ ” is in the class NP of problems decidable in non-deterministic polynomial time.

## Proof of Theorem 10

Recall that we assume in our language  $\mathcal{L}^+$  that we have a predicate variable  $X$  for every first-order variable  $x \in \mathcal{L}^{FO}$ . We use this correspondence freely in what follows.

For the translations  $\tau_x$  as defined in Section 4.2, we show Theorem 10 by induction on the structure of  $\varphi$ . For the first base case we simply have  $\mathcal{M}, f \models \tau_x(A)$  iff  $f(x) \in \llbracket A \rrbracket$ . For the second base case we have  $\mathcal{M}, f \models \tau_x(Y)$  iff  $\mathcal{M}, f \models x = y$  iff  $f(x) = f(y)$  iff  $f(x) \in \{g_f(Y)\}$  iff  $f(x) \in \llbracket Y \rrbracket_{g_f}$ . The case of conjunction

is straightforward, as are the next three cases. For the final case:

$$\begin{aligned}
\mathcal{M}, f \models \tau_x(\zeta Y.\varphi) \quad & \text{iff} \quad \mathcal{M}, f \models \exists z \exists y (x = y \wedge \tau_z(\varphi)) \\
& \text{iff} \quad \text{there is } a \text{ s.t. } \mathcal{M}, f[z \mapsto a, y \mapsto f(x)] \models \tau_z(\varphi) \\
& \text{iff} \quad \text{there is } a \text{ s.t. } f[z \mapsto a, y \mapsto f(x)](z) \in \llbracket \varphi \rrbracket_{g_f[z \mapsto a, y \mapsto f(x)]} \\
& \text{iff} \quad \text{there is } a \text{ s.t. } f[z \mapsto a, y \mapsto f(x)](z) \in \llbracket \varphi \rrbracket_{g_f[Y \mapsto f(x)]} \\
& \text{iff} \quad \llbracket \varphi \rrbracket_{g_f[Y \mapsto f(x)]} \neq \emptyset \\
& \text{iff} \quad f(x) \in \llbracket \zeta Y.\varphi \rrbracket_{g_f}
\end{aligned}$$

The third equivalence is by the induction hypothesis, and fourth is by the fact that the assignment to  $Z$  is not included in  $g_f[Y \mapsto f(x)]$  since the variable  $z$  was chosen fresh. That is,  $Z$  did not occur in  $\varphi$ , and so the assignment  $\llbracket \varphi \rrbracket_{g_f[z \mapsto a, y \mapsto f(x)]} = \llbracket \varphi \rrbracket_{g_f[Z \mapsto a, Y \mapsto f(x)]}$  is the same as  $\llbracket \varphi \rrbracket_{g_f[Y \mapsto f(x)]}$ .

### Proof of Theorem 11

We show both simultaneously by induction on formulas in  $\mathcal{L}^{FO}$ . Consider the second base case. We have  $\mathcal{M}, f \models R(x_i, x_j)$  implies  $\langle f(x_i), f(x_j) \rangle \in \llbracket R \rrbracket$ , which in turn implies  $\llbracket X_i \rrbracket_{g_f} \cap \llbracket \exists R.X_j \rrbracket_{g_f} \neq \emptyset$ , and thus  $\llbracket \uparrow(X_i \wedge \exists R.X_j) \rrbracket_{g_f} = M$ . For (3) we have  $\mathcal{M}, f \not\models R(x_i, x_j)$  implies  $\langle f(x_i), f(x_j) \rangle \notin \llbracket R \rrbracket$ , which gives  $\llbracket X_i \rrbracket_{g_f} \cap \llbracket \exists R.X_j \rrbracket_{g_f} = \emptyset$ , and thus  $\llbracket \uparrow(X_i \wedge \exists R.X_j) \rrbracket_{g_f} = \emptyset$ . The other two base cases are analogous.

Conjunction is straightforward, so consider negation. For (2) we have:

$$\begin{aligned}
\mathcal{M}, f \models \neg \alpha \quad & \text{implies} \quad \mathcal{M}, f \not\models \alpha \\
& \text{implies} \quad \llbracket \widehat{\alpha} \rrbracket_{g_f} = \emptyset \\
& \text{implies} \quad \llbracket \zeta Y.\downarrow(Y \wedge \widehat{\alpha}) \rrbracket_{g_f} = M
\end{aligned}$$

where the last implication holds because  $Y$  is chosen fresh and thus is not in  $\text{dom}(g_f)$ : if  $\llbracket \widehat{\alpha} \rrbracket_{g_f} = \emptyset$  then every point  $a$  is such that  $\{a\} \cap \llbracket \widehat{\alpha} \rrbracket_{g_f} = \emptyset$ . And for (3) we have:

$$\begin{aligned}
\mathcal{M}, f \not\models \neg \alpha \quad & \text{implies} \quad \mathcal{M}, f \models \alpha \\
& \text{implies} \quad \llbracket \widehat{\alpha} \rrbracket_{g_f} = M \\
& \text{implies} \quad \llbracket \zeta Y.\downarrow(Y \wedge \widehat{\alpha}) \rrbracket_{g_f} = \emptyset
\end{aligned}$$

where the reasoning in the last step is as in the previous case.

Consider (2) for existential quantification:

$$\begin{aligned}
\mathcal{M}, f \models \exists x_i \alpha \quad & \text{implies} \quad \text{there is } a \text{ s.t. } \mathcal{M}, f[x_i \mapsto a] \models \alpha \\
& \text{implies} \quad \text{there is } a \text{ s.t. } \llbracket \widehat{\alpha} \rrbracket_{g_f[x_i \mapsto a]} = M \\
& \text{implies} \quad \llbracket \zeta X_i.\widehat{\alpha} \rrbracket_{g_f} \neq \emptyset \\
& \text{implies} \quad \llbracket \uparrow \zeta X_i.\widehat{\alpha} \rrbracket_{g_f} = M
\end{aligned}$$

And for (3) we have:

$$\begin{aligned}
\mathcal{M}, f \not\models \exists x_i \alpha \quad & \text{implies} \quad \text{for all } a: \mathcal{M}, f[x_i \mapsto a] \not\models \alpha \\
& \text{implies} \quad \text{for all } a: \llbracket \widehat{\alpha} \rrbracket_{g_f[x_i \mapsto a]} = \emptyset \\
& \text{implies} \quad \llbracket \zeta X_i.\widehat{\alpha} \rrbracket_{g_f} = \emptyset \\
& \text{implies} \quad \llbracket \uparrow \zeta X_i.\widehat{\alpha} \rrbracket_{g_f} = \emptyset
\end{aligned}$$

This completes the proof.