# Kripke Semantics for a Logical Framework

# Kripke Semantics for a Logical Framework

Alex K. Simpson[*]

Department of Computer Science, University of Edinburgh,
JCMB, The King's Buildings, Edinburgh, EH9 3JZ.
Alex.Simpson@dcs.ed.ac.uk

### Abstract

We present a semantics (using Kripke lambda models) for a logical framework (minimal implicational predicate logic with quantification over all higher types). We apply the semantics to obtain straightforward adequacy proofs for encodings of logics in the framework.

## 1 Introduction

There has been much recent interest in the development and use of logical frameworks. A logical framework is a formal system within which many different logics can be easily represented. It is hoped that such frameworks will facilitate the rapid development of proof assistants for the wide variety of different logics used in computer science and other fields. In this paper we give a semantic analysis (using Kripke lambda models) of the use of minimal implicational predicate logic (with quantification over all higher types) as a logical framework. We choose this framework because it is relatively straightforward to give it a useful semantics.

The use of such a logic as a framework is not new. Similar logics have been used for this purpose by Paulson [17] and Felty and Miller [5]. Also, the earlier Edinburgh Logical Framework (LF) of Harper, Honsell and Plotkin [9], although a dependently-typed lambda-calculus, is closely related to minimal implicational predicate logic by the Curry-Howard correspondence between propositions and types.

There are two main ingredients to the encoding of a logic in a framework: the representation of syntax and the representation of logical consequence. In our framework, syntax is encoded by terms of the simply-typed lambda calculus. The encoding of syntax in such a way is, in general, straightforward and of little theoretical interest. We shall therefore simplify matters by giving as few details as possible and abusing notation as

[*]Supported by SERC grant no. 90311820.

much as possible. One might prefer to assume that the syntax of the logic being encoded is already given in the typed lambda calculus (as advocated by Martin-Löf [16], Pfenning and Elliot [18] and others).

The consequence relation of a logic is represented through an axiomatization of its properties in the framework. The desired connection is one of the following form:

$$\phi_1, \ldots, \phi_n \vdash_L \phi \quad \text{iff} \quad \mathcal{A}, \mathit{true}(\phi_1^*), \ldots, \mathit{true}(\phi_n^*) \vdash_F \mathit{true}(\phi^*),$$

where $\vdash_L$ is the consequence relation of the encoded logic, $\vdash_F$ is the consequence relation of the framework, $\mathcal{A}$ is the axiomatization, *true* is a predicate in the framework and $(\cdot)^*$ is the mapping from formulae in the encoded logic to their representing terms in the framework. An axiomatization satisfying the above equivalence is said to be *adequate*.

To encode a logic in a framework one must provide an axiomatization and prove its adequacy. The left-to-right implication of adequacy (which we call *fullness*) is usually quite easy to prove. One takes some proof system for the encoded logic and shows that each proof can be mimicked by one in the framework. The right-to-left implication (which we call *faithfulness*) is trickier. To show it proof-theoretically one must show that any derivation in the framework is (essentially) the representation of a derivation in some proof system for the encoded logic. This is usually done by an analysis of normal forms for derivations in the framework, which can be quite involved (see, e.g., Harper *et al* [9] and Paulson [17]).

The goal of this paper is to show how faithfulness can be easily established semantically. Intuitively, faithfulness should not be a difficult property to demonstrate. The encoding of logical consequence amounts to a meta-axiomatization of its properties. Faithfulness should follow from the "truth" of the meta-axioms. However, it is not always clear how to read a meta-axiom. For example, the meta-axiom:

$$\forall \phi\!:\!o.\, \forall \psi\!:\!o.\, (\mathit{true}(\phi) \supset \mathit{true}(\psi)) \supset \mathit{true}(\phi \Rightarrow \psi)$$

(here $o$ is the type of sentences of the encoded logic, $\Rightarrow$ is implication in the encoded logic and $\supset$ is implication in the framework) could be understood as expressing that the "Deduction Theorem" holds for the encoded logic; alternatively it might be interpreted "classically" as:

$$\forall \phi\!:\!o.\, \forall \psi\!:\!o.\, (\mathit{true}(\phi) \wedge \neg \mathit{true}(\psi)) \vee \mathit{true}(\phi \Rightarrow \psi).$$

In order to understand such statements unambiguously, we need a model of the framework to interpret them in.

In this paper we provide a notion of model for the framework (using the Kripke lambda models of Mitchell and Moggi [15]). Faithfulness is then proved by constructing particular models of an encoding within which the truth (or soundness) of the meta-axiomatization can be verified.

A by-product of this approach to proving faithfulness is that the models we construct give an interpretation to the meta-logic of the framework. We shall consider two different sorts of model: term models constructed (essentially) from the syntax of the encoded logic, and "semantic" models built out of models of the encoded logic. Both kinds of model enable formulae of the framework to be understood as expressing meta-propositions about the encoded logic. Although we do not systematically consider the different meta-interpretations induced by different models, it is important to note that the model theory allows a meta-formula to be given a formal meaning substantiating its informal reading.

The structure of the paper is as follows. In Section 2 we present the framework we use. Then in Section 3 we give some example encodings of logics in the framework. The semantics of the framework is given in section 4. This is used in Section 5 to prove the faithfulness of the encodings presented in Section 3. In Section 6 we begin to build a general theory relating properties of encodings (such as faithfulness) with their semantic counterparts. Finally, in section 7 we conclude with a discussion of related work and possible directions for future research.

## 2 The framework

The framework we use is minimal implicational predicate logic with universal quantification over all higher types. It is very similar to the logic programming language considered by Miller in [13]. It is also closely related to the frameworks considered by Paulson [17] and Felty and Miller [5] (the main differences are to do with the treatment of predicates outwith the type system and, in the case of Paulson's work, our restriction to predicative quantification).

We use $A$, $B$ and $C$ to range over (simple) types, $M$ and $N$ to range over terms (of the simply-typed $\lambda$-calculus), and $\Phi$ and $\Psi$ to range over formulae (lower case Greek letters will be reserved for formulae of the encoded logics).

We assume given four countably infinite, disjoint sets: a set of *type constants*, a set of *predicate symbols*, a set of *term constants* and a set of *variables*. We use $P, \ldots$ to range over the predicate symbols, $c, \ldots$ to range over the term constants and $x, \ldots$ to range over the variables.

A theory is generated by a *presentation* which is a quadruple, $(\mathcal{T}, \mathcal{P}, \Sigma, \mathcal{A})$, where each of $\mathcal{T}$, $\mathcal{P}$, $\Sigma$ and $\mathcal{A}$ are sets as specified below. Mostly (but not exclusively) we consider *finite* presentations, i.e. those in which all four sets are finite.

$\mathcal{T}$ is a subset of the set of type constants. Types are generated from this set by the grammar:

$$A ::= \alpha \mid A \to B$$

where $\alpha$ ranges over elements of $\mathcal{T}$. As usual, when brackets are omitted, "$\to$" associates to the right. (We shall also adopt this convention with implication connectives, both in the framework and in the encoded logics.) $\mathcal{P}$ is a set of *predicate declarations* of the form $P:\langle A_1, \ldots, A_n\rangle$ (where $n$ is possibly zero) such that each predicate symbol, $P$, appears only once in the set. $\Sigma$ is a set of *constant declarations* of the form $c:A$ such that each term constant, $c$, appears only once in the set. The requirements on $\mathcal{A}$ are given below. Henceforth everything will be parameterized over $\mathcal{T}$ and $\mathcal{P}$ and these sets will usually be left implicit. Thus we often refer to the presentation as $(\Sigma, \mathcal{A})$.

A *context*, $\Gamma$, is a finite set of *variable declarations* of the form $x:A$ such that each variable, $x$, appears only once in the set. The abstract syntax of terms and formulae is given by the following grammar.

$$\begin{array}{rcl} M & ::= & c \mid x \mid \lambda x\!:\!A.\, M \mid M(N) \\ \Phi & ::= & P(M_1, \ldots, M_n) \mid \Phi \supset \Psi \mid \forall x\!:\!A.\, \Phi \end{array}$$

We write $N[M/x]$ and $\Phi[M/x]$ for the substitution of $M$ for all free occurrences of $x$ in $N$ and $\Phi$ respectively. Lambda-terms and quantified formulae are considered identified up to $\alpha$-equivalence.

The term calculus is just the simply-typed lambda calculus (for which a good reference is [14]). We write $\Gamma \rhd_\Sigma M:A$ to mean that $M$ is term over $\Sigma$ with type $A$ in context $\Gamma$. We shall only be concerned with $\beta\eta$-equality, $=_{\beta\eta}$, between terms. We note (but shall not use) that equality between terms is decidable. A term, $M$, such that $\Gamma \rhd_\Sigma M:A$ is said to be in *long-$\beta\eta$ normal form* (with respect to $\Gamma$ and $\Sigma$) if it has the form:

$$\lambda x_1\!:\!A_1. \ldots \lambda x_n\!:\!A_n.\, h(M_1)\ldots(M_m)$$

where: $n, m \geq 0$; $h$ is either a variable or a constant; $\Gamma, x_1 : A_1, \ldots, x_n : A_n \rhd_\Sigma h(M_1)\ldots(M_m) : \alpha$ for some type constant $\alpha$; and each $M_i$ $(1 \leq i \leq m)$ is in long-$\beta\eta$ normal form with respect to $\Gamma, x_1\!:\!A_1, \ldots, x_n\!:\!A_n$ and $\Sigma$. Clearly any $M$ in long-$\beta\eta$ normal form with respect to $\Gamma$ and $\Sigma$ is also in long-$\beta\eta$ normal form with respect to $\Gamma' \supseteq \Gamma$ and $\Sigma' \supseteq \Sigma$. The crucial property of long-$\beta\eta$ normal forms is the following (see [12]): if $\Gamma \rhd_\Sigma M:A$ then there is a unique term, $\beta\eta(M)$, in long-$\beta\eta$ normal form (with respect to $\Gamma$ and $\Sigma$) such that $M =_{\beta\eta} \beta\eta(M)$.

In Figure 1 we give a formal system for deriving judgements of the form $\Gamma \rhd_\Sigma \Phi$ *prop*. When $\Gamma \rhd_\Sigma \Phi$ *prop* is derivable we say that $\Phi$ is *well-formed* in $\Gamma$ and $\Sigma$. When $\Gamma$ is the empty set it is omitted from such statements. $\Sigma$ will be omitted when it can be understood from the context.

$$\frac{\Gamma \rhd_\Sigma M_1 : A_1 \quad \dots \quad \Gamma \rhd_\Sigma M_n : A_n \quad P : \langle A_1, \dots, A_n \rangle \in \mathcal{P}}{\Gamma \rhd_\Sigma P(M_1, \dots, M_n)\ \mathit{prop}}$$

$$\frac{\Gamma \rhd_\Sigma \Phi\ \mathit{prop} \quad \Gamma \rhd_\Sigma \Psi\ \mathit{prop}}{\Gamma \rhd_\Sigma \Phi \supset \Psi\ \mathit{prop}} \qquad \frac{\Gamma, x : A \rhd_\Sigma \Phi\ \mathit{prop}}{\Gamma \rhd_\Sigma \forall x : A.\ \Phi\ \mathit{prop}}$$

Figure 1: Well-formedness rules for formulae.

$$\text{Ax} \quad \frac{\Phi \in \mathcal{A}}{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi} \qquad \text{Ass} \quad \frac{\Phi \in \mathcal{H}}{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi}$$

$$\text{Sub} \quad \frac{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi[M/x] \quad M =_{\beta\eta} N \quad \Gamma \rhd_\Sigma \Phi[N/x]\ \mathit{prop}}{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi[N/x]}$$

$$\supset \text{I} \quad \frac{\Gamma ; \mathcal{H}, \Phi \vdash_{(\Sigma, \mathcal{A})} \Psi}{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi \supset \Psi} \qquad \supset \text{E} \quad \frac{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi \supset \Psi \quad \Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi}{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Psi}$$

$$\forall \text{I} \quad \frac{\Gamma, x : A ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi}{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \forall x : A.\ \Phi} \qquad \forall \text{E} \quad \frac{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \forall x : A.\ \Phi \quad \Gamma \rhd_\Sigma M : A}{\Gamma ; \mathcal{H} \vdash_{(\Sigma, \mathcal{A})} \Phi[M/x]}$$

Restriction on $\forall$I: $x$ does not occur free in $\mathcal{H}$.

Figure 2: Rules for meta-logical consequence.

The fourth component of the presentation, $\mathcal{A}$, is a set of formulae, the *axioms*, such that each formula in $\mathcal{A}$ is well-formed in $\Sigma$.

Logical consequence for $(\Sigma, \mathcal{A})$:

$$\Gamma; \mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi$$

relates $\Gamma$, $\mathcal{H}$ and $\Phi$ where $\mathcal{H}$ is a set of formulae, the *hypotheses*, and each formula in $\mathcal{H} \cup \{\Phi\}$ is well-formed in $\Gamma$ and $\Sigma$. This relation is given by the formal system of Figure 2.

In the sequel we shall require the following elementary derived result about consequence.

**Lemma 2.1 (Weakening)** *If* $\Gamma; \mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi$ *and all formulae in* $\mathcal{H}'$ *are well-formed in* $\Gamma \cup \Gamma'$ *then* $\Gamma, \Gamma'; \mathcal{H}, \mathcal{H}' \vdash_{(\Sigma,\mathcal{A})} \Phi$*.*

The easy proof, by induction on the structure of derivations, is omitted.

## 3 Encoding logics in the framework

In this section we give some example encodings illustrating the use of the framework. These encodings will later serve as examples for demonstrating our semantic approach to proving adequacy. Many other examples (in the context of LF) which we could equally well have chosen are detailed in [3].

The first example is a very simple encoding of the implicational fragment of minimal propositional logic. We use $\phi$, $\psi$, ... to range over the formulae of the language, which are given by the grammar:

$$\phi \quad ::= \quad a_i \mid \phi \Rightarrow \psi$$

where $\{a_i\}$ is a countable set of propositional constants indexed by natural numbers $i$. The consequence relation of the logic, $\vdash_m$, is just the restriction of intuitionistic consequence to the fragment. The standard Hilbert-style axiomatization of the consequence relation is well known (see, e.g., Hindley and Seldin [11, p. 193]).

The presentation encoding minimal implicational logic is $(\mathcal{T}_m, \mathcal{P}_m, \Sigma_m, \mathcal{A}_m)$ of Figure 3. There are two type constants: a type of natural numbers, $N$, and a type of formulae, $o$. It is easy to show that the long-$\beta\eta$ normal forms (in the empty context) of $N$ are just the numerals (terms of the form $s^n(0)$). The constructor $a : N \to o$ produces a new propositional constant $a_i$ for every numeral $i$. It is again easy to show that the long-$\beta\eta$ normal forms of type $o$ are in one-to-one correspondence with the formulae of the logic. We do not formalize this correspondence as it is completely routine. Further

$$
\begin{aligned}
\mathcal{T}_m &= \{N, o\} \\
\mathcal{P}_m &= \{\mathit{true}\colon\langle o\rangle\} \\
\Sigma_m &= \{0 : N,\ s : N \to N,\ a : N \to o,\ \Rightarrow : o \to o \to o\} \\
\mathcal{A}_m &= \{\forall p\colon o.\ \forall q\colon o.\ \mathit{true}(p \Rightarrow q \Rightarrow p), \\
&\quad \forall p\colon o.\ \forall q\colon o.\ \forall r\colon o.\ \mathit{true}((p \Rightarrow q \Rightarrow r) \Rightarrow (p \Rightarrow q) \Rightarrow p \Rightarrow r), \\
&\quad \forall p\colon o.\ \forall q\colon o.\ \mathit{true}(p \Rightarrow q) \supset \mathit{true}(p) \supset \mathit{true}(q)\}
\end{aligned}
$$

Figure 3: Presentation of propositional minimal implicational logic.

$$
\begin{aligned}
\mathcal{T}_T &= \{\iota, o\} \\
\mathcal{P}_T &= \{\mathit{true}\colon\langle o\rangle\} \\
\Sigma_T &= \{\Rightarrow : o \to o \to o,\ \neg : o \to o,\ \forall : (\iota \to o) \to o,\ = : \iota \to \iota \to o, \\
&\quad P_1^{a_1} : \overbrace{\iota \to \ldots \iota}^{a_1 \text{ times}} \to o,\ \ldots\ , P_h^{a_h} : \overbrace{\iota \to \ldots \iota}^{a_h \text{ times}} \to o, \\
&\quad f_1^{a'_1} : \overbrace{\iota \to \ldots \iota}^{a'_1 \text{ times}} \to \iota,\ \ldots\ , f_k^{a'_k} : \overbrace{\iota \to \ldots \iota}^{a'_k \text{ times}} \to \iota\} \\
\mathcal{A}_T &= \{\forall p\colon o.\ \forall q\colon o.\ (\mathit{true}(p) \supset \mathit{true}(q)) \supset \mathit{true}(p \Rightarrow q), \\
&\quad \forall p\colon o.\ \forall q\colon o.\ \mathit{true}(p \Rightarrow q) \supset (\mathit{true}(p) \supset \mathit{true}(q)), \\
&\quad \forall p\colon o.\ \forall q\colon o.\ \mathit{true}(p) \supset \mathit{true}(\neg p) \supset \mathit{true}(q), \\
&\quad \forall p\colon o.\ \forall q\colon o.\ (\mathit{true}(p) \supset \mathit{true}(q)) \supset (\mathit{true}(\neg p) \supset \mathit{true}(q)) \supset \mathit{true}(q), \\
&\quad \forall p\colon \iota \to o.\ (\forall x\colon \iota.\ \mathit{true}(p(x))) \supset \mathit{true}(\forall x.\ p(x)), \\
&\quad \forall p\colon \iota \to o.\ \mathit{true}(\forall x.\ p(x)) \supset (\forall x\colon \iota.\ \mathit{true}(p(x))), \\
&\quad \forall x\colon \iota.\ \mathit{true}(x = x), \\
&\quad \forall p\colon \iota \to o.\ \forall x\colon \iota.\ \forall y\colon \iota.\ \mathit{true}(x = y) \supset \mathit{true}(p(x)) \supset \mathit{true}(p(y)), \\
&\quad \mathit{true}(T_1),\ \ldots\ , \mathit{true}(T_l)\}
\end{aligned}
$$

Figure 4: Presentation of an arbitrary classical first-order theory.

we take the liberty (for the sake of notational convenience) of identifying long-$\beta\eta$ normal forms of type $o$ with their corresponding formulae.

We now turn to the axiomatization of consequence given by $\mathcal{A}_m$. Its correctness is summed up by:

**Proposition 3.1 (Adequacy for minimal logic)** *The following are equivalent:*

1. $\phi_1, \ldots, \phi_n \vdash_m \phi$.

2. $true(\phi_1), \ldots, true(\phi_n) \vdash_{(\Sigma_m, \mathcal{A}_m)} true(\phi)$.

We call the property that statement 1 implies statement 2 *fullness* and its converse *faithfulness*. The fullness of the encoding is rather straightforward to establish. The axioms in $\mathcal{A}_m$ follow closely the standard Hilbert-style axiomatization of minimal logic. The first two of our axioms give the usual axiom schemas for minimal logic, and the last of our axioms corresponds to the *modus ponens* rule. Accordingly, it is easy to show that any derivation of $\phi$ from $\phi_1$, ..., $\phi_n$ in the Hilbert system can be mimicked by a derivation of $true(\phi_1), \ldots, true(\phi_n) \vdash_{(\Sigma_m, \mathcal{A}_m)} true(\phi)$ in the framework. The details are entirely routine. Thus the encoding is indeed full.

The faithfulness of the encoding could also be established by proof-theoretical means. To do so would require a normal-form result for the framework (see the discussion in Section 1). However, it is intuitively obvious that the encoding is faithful. All the meta-axioms in the presentation are true with respect to their evident interpretation. To formalize such an argument we must build a model of the whole framework extending the evident interpretation to arbitrary types. We shall give a rigorous proof along these lines in Section 5.

Our second example is an arbitrary finitely axiomatized theory in classical first-order logic. Suppose we have a finite first-order signature consisting of predicate symbols $\{P_1^{a_1}, \ldots, P_h^{a_h}\}$ (where $a_i$ is the arity of $P_i$) and function symbols $\{f_1^{a'_1}, \ldots, f_k^{a'_k}\}$. We now use $\phi$, $\psi$, ... to range over formulae in the resulting first-order language. (For simplicity, we consider just the connectives for negation and implication and the universal quantifier.) Let $T = \{T_1, \ldots, T_l\}$ be a finite set of sentences in this language. We consider the consequence relation, $\vdash_T$, between first-order formulae defined by $\phi_1, \ldots, \phi_n \vdash_T \phi$ if and only if there is a natural deduction derivation (in Prawitz' system for classical natural deduction [19]) of $\phi$ from the axioms in $T$ in which all open assumptions are contained in the set $\{\phi_1, \ldots, \phi_n\}$. This is the so-called "truth" consequence relation of first-order logic (see Avron [2]). An alternative, semantic, characterization of the consequence relation is that $\phi_1, \ldots, \phi_n \vdash_T \phi$ holds if and only if, for all first-order models $\mathcal{M}$ of $T$ and all environments $\rho$ mapping variables to elements of the domain of $\mathcal{M}$, if

$\mathcal{M} \models_\rho \phi_1$ and ... and $\mathcal{M} \models_\rho \phi_n$ then $\mathcal{M} \models_\rho \phi$ (the reader is referred to [23] for the definition of classical satisfaction). We shall use this characterization later.

The encoding of the first-order theory is given by the presentation $(\mathcal{T}_T, \mathcal{P}_T, \Sigma_T, \mathcal{A}_T)$ in Figure 4. Given a finite set of first-order variables, $X$, we write $X : \iota$ for the context $\{x : \iota \mid x \in X\}$. The set of first-order terms with all variables in $X$ is in one-to-one correspondence with the set of framework terms of type $\iota$ in long-$\beta\eta$ normal form with respect to $X : \iota$. Similarly, the set of first-order formulae with all free variables in $X$ is in one-to-one correspondence with the set of long-$\beta\eta$ normal forms of type $o$ with respect to $X : \iota$. Again, these correspondences are straightforward to establish, so we omit details and henceforth identify framework terms in long-$\beta\eta$ form with their associated first-order terms and formulae. For example, we write $\forall x.\ \phi$ for the framework term $\forall(\lambda x : \iota.\ \phi)$. Such conventions are used freely in the axiomatization of consequence.

**Proposition 3.2 (Adequacy for first-order logic)** *Let $X$ be a non-empty finite set of variables containing all free variables in $\phi_1, \ldots, \phi_n, \phi$. Then the following are equivalent:*

1. $\phi_1, \ldots, \phi_n \vdash_T \phi$.
2. $X : \iota;\ true(\phi_1), \ldots, true(\phi_n) \vdash_{(\Sigma_T, \mathcal{A}_T)} true(\phi)$.

Once again, fullness is easy to establish. The axiomatization is designed to easily mimic the standard natural deduction rules for first-order classical logic. The only interesting point is the requirement that $X$ be non-empty even when $\phi_1, \ldots, \phi_n, \phi$ are closed. The reason is that the assumption that the type $\iota$ is non-empty is not built into the framework, whereas the assumption that the domain of quantification is non-empty is (rightly or wrongly) built into first-order logic. (However, if the first-order signature contains a constant then the assumption that $X$ is non-empty is no longer required.) A detailed proof of fullness is given by Harper *et al* [9] for a closely related encoding in the context of LF. They also prove the faithfulness of their encoding using a normal-form result for LF. Again, we shall show in Section 5 that a straightforward semantic proof of faithfulness is possible.

For our last example, we present an an encoding of the minimal normal modal logic K. We now use $\phi$, $\psi$, ... to range over formulae given by the grammar:

$$\phi \quad ::= \quad a_i \mid \phi \Rightarrow \psi \mid \neg\phi \mid \Box\phi$$

We consider the "truth" consequence relation, $\vdash_K$, for the logic (again see Avron [2]). A simple semantic characterization of the consequence relation can be given in terms of Kripke models $(W, R, \models)$ where: $W$ is a set (of "worlds"); $R$ is a binary relation on $W$ (the "visibility" relation); and $\models$ is a binary ("satisfaction") relation between worlds and

$$
\begin{aligned}
\mathcal{T}_K &= \{N, o, W\} \\
\mathcal{P}_K &= \{R{:}\langle W, W\rangle,\ \mathit{sat}{:}\langle W, o\rangle\} \\
\Sigma_K &= \{0 : N,\ s : N \to N,\ a : N \to o,\ \Rightarrow\ : o \to o \to o,\ \neg : o \to o,\ \Box : o \to o\} \\
\mathcal{A}_K &= \{\forall x{:}W.\, \forall p{:}o.\, \forall q{:}o.\, (\mathit{sat}(x,p) \supset \mathit{sat}(x,q)) \supset \mathit{sat}(x, p \Rightarrow q), \\
&\quad \forall x{:}W.\, \forall p{:}o.\, \forall q{:}o.\, \mathit{sat}(x, p \Rightarrow q) \supset (\mathit{sat}(x,p) \supset \mathit{sat}(x,q)), \\
&\quad \forall x{:}W.\, \forall p{:}o.\, \forall q{:}o.\, \mathit{sat}(x,p) \supset \mathit{sat}(x,\neg p) \supset \mathit{sat}(x,q), \\
&\quad \forall x{:}W.\, \forall p{:}o.\, \forall q{:}o.\, (\mathit{sat}(x,p) \supset \mathit{sat}(x,q)) \supset (\mathit{sat}(x,\neg p) \supset \mathit{sat}(x,q)) \supset \mathit{sat}(x,q), \\
&\quad \forall x{:}W.\, \forall p{:}o.\, (\forall y{:}W.\, R(x,y) \supset \mathit{sat}(y,p)) \supset \mathit{sat}(x, \Box p), \\
&\quad \forall x{:}W.\, \forall p{:}o.\, \mathit{sat}(x, \Box p) \supset \forall y{:}W.\, R(x,y) \supset \mathit{sat}(y,p)\}
\end{aligned}
$$

Figure 5: Presentation of the modal logic K.

formulae satisfying well-known inductive properties (see [4]). The consequence relation is defined by: $\phi_1, \ldots, \phi_n \vdash_K \phi$ if and only if, for all models $(W, R, \models)$, for all $w \in W$, if $w \models \phi_1$ and ... and $w \models \phi_n$ then $w \models \phi$.

The encoding is given by the presentation $(\mathcal{T}_K, \mathcal{P}_K, \Sigma_K, \mathcal{A}_K)$ of Figure 5. As with minimal logic, we have a type $N$ of natural numbers (to generate propositional constants) and a type $o$ of formulae. Again, the formulae of modal logic are in one-to-one correspondence with the terms of type $o$ in long-$\beta\eta$ form with respect to the empty context and we identify such terms with their associated formulae. However, this time we have another type, $W$, intended to stand for the set of worlds in a model. Similarly $R$ and $\mathit{sat}$ are intended to represent the visibility and satisfaction relations respectively. So the axiomatization of consequence is most easily understood semantically. However, it can also be seen as representing (a variant of) the Fitch-style natural deduction proof system for K, see Fitting's book [6, Chapter 4, Sections 12, 15 and 16]. It is also closely related to Fitting's prefixed tableau systems [6, Chapter 8]. It is possible to give other presentations of K based on, for example, its standard Hilbert-style proof system (see Avron's encoding of S4 in LF [3]). However, we believe that our semantic-based encoding is rather natural. It will also serve to illustrate nicely the semantic approach to proving faithfulness.

**Proposition 3.3 (Adequacy for K)** *The following are equivalent:*

- $\phi_1, \ldots, \phi_n \vdash \phi$.
- $x{:}W;\ \mathit{sat}(x, \phi_1), \ldots, \mathit{sat}(x, \phi_n) \vdash_{(\Sigma_K, \mathcal{A}_K)} \mathit{sat}(x, \phi)$.

Once again, fullness is routine. Although the encoding is not based on any proof system, it is very easy to show that the standard Hilbert system for K can be mimicked within it. Other than the routine verification of the axioms of classical propositional logic, it suffices to show that the system derives the characteristic axiom of K and is closed under necessitation (restricted to theorems). For the characteristic axiom we must show that, for any $\phi$, $\psi$:

$$x : W;\ \vdash_{(\Sigma_K, \mathcal{A}_K)} sat(x, \Box(\phi \Rightarrow \psi) \Rightarrow \Box\phi \Rightarrow \Box\psi).$$

This is a straightforward, but worthwhile, exercise. For necessitation, we must show that if $x : W;\ \vdash_{(\Sigma_K, \mathcal{A}_K)} sat(x, \phi)$ then $x : W;\ \vdash_{(\Sigma_K, \mathcal{A}_K)} sat(x, \Box\phi)$. But this is an easy consequence of the "$\Box$-introduction" axiom.

Again, the faithfulness of the encoding is more interesting. One possible way to obtain a syntactic proof is as follows. First, define a natural deduction proof system for K based on the proof system adumbrated in the encoding. Then, using a normal form result for the framework, show that every framework derivation does indeed correspond to a derivation in the new system. Lastly, justify the correctness of the new proof system by proving it sound with respect to the semantics of K.[1] Fortunately, the middle ground can be cut out completely. Rather than providing a new proof system, it is possible to justify the soundness of the framework encoding directly. Full details are given in Section 5.

## 4 Semantics

As the meta-logic is intuitionistic with quantification over all higher types, we seek a semantics in terms of Kripke models in which all typed lambda terms can be interpreted at each world. The Kripke lambda models of Mitchell and Moggi [15] are thus a natural choice. This section is rather technical. For a more discursive presentation of Kripke lambda models the reader is urged to consult [15]. However, although we follow their paper quite closely, the reader is advised that some of our notation and terminology differs from that of Mitchell and Moggi.

An (extensional, Kripke, *$\mathcal{T}$-$\mathcal{P}$-)prestructure* is a sextuple:

$$(W, \leq, \{[\![A]\!]_w\}, \{[\![P]\!]_w\}, \{\epsilon_w^{AB}\}, \{i_{ww'}^{A}\})$$

where:

- $W$ is a set of *worlds* partially ordered by $\leq$.

[1] Avron's proof of faithfulness for his twin judgement encodings of modal logic in LF proceeded in this way [1].

- $\{[\![A]\!]_w\}$ is a family of sets, $[\![A]\!]_w$, indexed by types, $A$, and worlds, $w$.
- $\{[\![P]\!]_w\}$ is a family of relations, $[\![P]\!]_w \subseteq [\![A_1]\!]_w \times \ldots \times [\![A_n]\!]_w$, indexed by predicate symbols, $P$, with declarations, $P{:}\langle A_1, \ldots, A_n\rangle$, in $\mathcal{P}$ and worlds, $w$.
- $\{\epsilon_w^{AB}\}$ is a family of functions, $\epsilon_w^{AB} : [\![A \to B]\!]_w \times [\![A]\!]_w \longrightarrow [\![B]\!]_w$, indexed by pairs of types, $A$, $B$, and worlds, $w$.
- $\{i^A_{ww'}\}$ is a family of functions, $i^A_{ww'} : [\![A]\!]_w \to [\![A]\!]_{w'}$, indexed by types, $A$, and pairs of worlds, $w \leq w'$.

subject to the conditions given below. In these (and henceforth) we adopt the following notational conventions. When $f \in [\![A \to B]\!]_w$ and $a \in [\![A]\!]_w$, we write $f(a)$ for $\epsilon_w^{AB}(f, a)$. When $a_w \in [\![A]\!]_w$ and $w \leq w'$, we write $a_{w'}$ for $i^A_{ww'}(a_w)$.

The conditions are:

**identity:** For all worlds $w$, $i^A_{ww}$ is the identity.

**composition:** For all $w \leq w' \leq w''$, $i^A_{w'w''} \circ i^A_{ww'} = i^A_{ww''}$.

**naturality:** For all $w \leq w'$, $i^B_{ww'} \circ \epsilon_w^{AB} = \epsilon_{w'}^{AB} \circ (i^{A\to B}_{ww'} \times i^A_{ww'})$.

**extensionality:** If $f_w, g_w \in [\![A \to B]\!]_w$ and, for all $w' \geq w$, for all $a \in [\![A]\!]_{w'}$, $f_{w'}(a) = g_{w'}(a)$ then $f_w = g_w$.

**persistency:** If $[\![P]\!]_w(a_{1w}, \ldots, a_{nw})$ then, for all $w' \geq w$, $[\![P]\!]_{w'}(a_{1w'}, \ldots, a_{nw'})$.

Thus a prestructure is an "extensional Kripke applicative structure" in the terminology of [15], together with an extra parameter, $\{[\![P]\!]_w\}$, used for interpreting the predicates of the logic.

A *partial element*, $p$, of type $A$ in a prestructure is given by an upper-closed subset $\mathrm{dom}(p) \subseteq W$, its *domain*, and a family of elements, $\{p_w\}$, indexed by worlds $w \in \mathrm{dom}(p)$ such that for all $w' \geq w \in \mathrm{dom}(p)$, $p_w \in [\![A]\!]_w$ and $i^A_{ww'}(p_w) = p_{w'}$. Given $p_w \in [\![A]\!]_w$, we write $p$ for the induced partial element of type $A$ with domain $\{w' \mid w \leq w'\}$ given by the elements $p_{w'} \in [\![A]\!]_{w'}$. A *global element* is a partial element, $p$, for which $\mathrm{dom}(p) = W$.

A $\Sigma$-*structure* is a 9-tuple:

$$(W, \leq, \{[\![A]\!]_w\}, \{[\![P]\!]_w\}, \{[\![c]\!]_w\}, \{\epsilon_w^{A_1A_2}\}, \{i^A_{ww'}\}, \{K_w^{AB}\}, \{S_w^{ABC}\})$$

given by a prestructure together with $\{[\![c]\!]_w\}$, $\{K_w^{AB}\}$ and $\{S_w^{ABC}\}$ where:

- $\{[\![c]\!]_w\}$ is a family of global elements, $[\![c]\!]$, of type $A$ indexed by constants, $c$, with declarations, $c{:}A$, in $\Sigma$.

- $K^{AB}$ is a global element of type $A \to B \to A$ such that, for all worlds $w$, for all $a \in [\![A]\!]_w$, for all $b \in [\![B]\!]_w$, $K^{AB}_w(a)(b) = a$.
- $S^{ABC}$ is a global element of type $(A \to B \to C) \to (A \to B) \to A \to C$ such that, for all worlds $w$, for all $f \in [\![A \to B \to C]\!]_w$, for all $g \in [\![A \to B]\!]_w$, for all $a \in [\![A]\!]_w$, $S^{ABC}_w(f)(g)(a) = f(a)(g(a))$.

Henceforth we refer to a $\Sigma$-structure as $(W, \leq)$ leaving the other components implicit.

An *environment*, $\rho$, is a function from variables to partial elements. We say that *$\rho$ interprets $\Gamma$ at $w$* if, for all $x : A \in \Gamma$, $\rho(x)$ is a partial element of type $A$ with $w \in \mathrm{dom}(\rho(x))$. Clearly if $\rho$ interprets $\Gamma$ at $w$ and $w' \geq w$ then $\rho$ interprets $\Gamma$ at $w'$ too. Also any environment interprets the empty context at any world. Given an environment $\rho$ and an element $a_w \in [\![A]\!]_w$ we write $\rho[x := a]$ for the environment that agrees with $\rho$ on variables other than $x$ and which assigns the induced partial element $a$ to $x$. If $\rho$ interprets $\Gamma$ at $w$ and $a_w \in [\![A]\!]_w$ then clearly $\rho[x := a]$ interprets $\Gamma, x\!:\!A$ at $w$.

If $M$ has type $A$ in $\Gamma$, and $\rho$ interprets $\Gamma$ at $w$, then the interpretation, $[\![M]\!]^\rho_w \in [\![A]\!]_w$, of $M$ by $\rho$ at $w$ is defined inductively on the structure of $M$ by:

$$\begin{array}{rcl}
[\![c]\!]^\rho_w & = & [\![c]\!]_w \\
[\![x]\!]^\rho_w & = & \rho(x)_w \\
[\![\lambda x\!:\!A.\, M]\!]^\rho_w & = & \text{the unique } f_w \in [\![A \to B]\!]_w \text{ (where } A \to B \text{ is the type of} \\
 & & \lambda x\!:\!A.\, M \text{ in } \Gamma\text{) such that, for all } w' \geq w\text{, for all } a_{w'} \in [\![A]\!]_{w'}, \\
 & & f_{w'}(a_{w'}) = [\![M]\!]^{\rho[x:=a]}_{w'} \\
[\![M(N)]\!]^\rho_w & = & [\![M]\!]^\rho_w([\![N]\!]^\rho_w)
\end{array}$$

As in [15], the existence of the $f_w$ required in the $\lambda x : A.\, M$ clause is given by the $S$ and $K$ combinators, and its uniqueness is guaranteed by extensionality. Clearly if $M$ is well-typed in the empty context then the value of $[\![M]\!]^\rho_w$ is independent of $\rho$, so we just write $[\![M]\!]_w$.

We now give some lemmas concerning the interpretation of terms in $\Sigma$-structures.

**Lemma 4.1** *If $\Gamma \rhd_\Sigma M\!:\!A$, $\rho$ interprets $\Gamma$ at $w$ and $w \leq w'$, then $i^A_{ww'}([\![M]\!]^\rho_w) = [\![M]\!]^\rho_{w'}$.*

**Lemma 4.2** *If $\Gamma, x\!:\!A \rhd_\Sigma M\!:\!B$, $\Gamma \rhd_\Sigma N\!:\!A$ and $\rho$ interprets $\Gamma$ at $w$, then $[\![M[N/x]]\!]^\rho_w = [\![M]\!]^{\rho[x:=[\![N]\!]^\rho]}_w$.*

**Lemma 4.3** *If $\Gamma \rhd_\Sigma M : A$, $\Gamma \rhd_\Sigma N : A$, $M =_{\beta\eta} N$, and $\rho$ interprets $\Gamma$ at $w$, then $[\![M]\!]^\rho_w = [\![N]\!]^\rho_w$.*

The first two lemmas are proved by straightforward inductions on the structure of $M$. The third is proved by an induction on derivations (in the usual formal system for $\beta\eta$-equality) of $M =_{\beta\eta} N$, using Lemma 4.2 in the verification $\beta$-equality.

If $\Phi$ is well-formed in $\Gamma$, and $\rho$ interprets $\Gamma$ at $w$, then the "forcing" relation $w \models_\rho \Phi$ is defined inductively on the structure of $\Phi$ by:

$$\begin{array}{lll} w \models_\rho P(M_1, \ldots, M_n) & \text{iff} & [\![P]\!]_w([\![M_1]\!]^\rho_w, \ldots, [\![M_n]\!]^\rho_w) \\ w \models_\rho \Phi \supset \Psi & \text{iff} & \text{for all } w' \geq w \text{, if } w' \models_\rho \Phi \text{ then } w' \models_\rho \Psi \\ w \models_\rho \forall x\!:\!A.\, \Phi & \text{iff} & \text{for all } w' \geq w \text{, for all } a_{w'} \in [\![A]\!]_{w'} \text{, } w' \models_{\rho[x:=a]} \Phi \end{array}$$

If $\mathcal{H}$ is a set of formulae, each well-formed in $\Gamma$, and $\rho$ interprets $\Gamma$ at $w$ then we write $w \models_\rho \mathcal{H}$ to mean that $w \models_\rho \Phi$, for all $\Phi \in \mathcal{H}$. If $\Phi$ is well-formed in the empty context then whether $w \models_\rho \Phi$ holds or not is independent of $\rho$, so we write $w \models \Phi$. We write $(W, \leq) \models \Phi$ to mean, for all $w \in W$, $w \models \Phi$.

The lemmas below give basic properties of the forcing relation.

**Lemma 4.4** *If $w \models_\rho \Phi$ and $w \leq w'$ then $w' \models_\rho \Phi$.*

**Lemma 4.5** *If $\Gamma \rhd_\Sigma M : A$ and $\rho$ interprets $\Gamma$ at $w$, then $w \models_\rho \Phi[M/x]$ if and only if $w \models_{\rho[x:=[\![M]\!]^\rho]} \Phi$.*

Both these lemmas are proved by induction on the structure of $\Phi$. The base case of the first uses Lemma 4.1 together with the persistency property of the structure. The base case of the second uses Lemma 4.2.

A $(\Sigma, \mathcal{A})$-*model* is a $\Sigma$-structure, $(W, \leq)$, such that, for all $\Phi \in \mathcal{A}$, $(W, \leq) \models \Phi$.

**Theorem 4.6 (Soundness and completeness)** *The two statements below are equivalent.*

1. $\Gamma; \mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi$.
2. *For all $(\Sigma, \mathcal{A})$-models $(W, \leq)$, for all $w \in W$, for all $\rho$ interpreting $\Gamma$ at $w$, if $w \models_\rho \mathcal{H}$ then $w \models_\rho \Phi$.*

Soundness is proved by a straightforward induction on the derivation of $\Gamma; \mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi$. The verification of the $\forall$E rule uses Lemma 4.5. The verification of the Sub rule uses Lemmas 4.3 and 4.5. Completeness is proved by a term model construction similar to that used in [15] but interpreting the logic as well. Rather than directly constructing the term model required for completeness, we give instead a general term model construction of which the desired model will be a special case. The general construction will also be used in Section 5 for the faithfulness proofs.

The idea behind the term model construction is to build a $\Sigma$-structure out of those terms well-typed in a limited number of contexts. One extreme will be when only closed terms are allowed. Such a model will be used in Section 5. The other extreme arises when terms typable in arbitrary contexts are allowed. In this case we obtain the model used to prove completeness. In Section 5 we shall also have use for a model lying somewhere between the two extremes. Some open terms will be allowed, but only those typable in specific contexts.

However, placing restrictions on the free variables allowed in terms raises the problem of how to obtain an extensional model. This problem already occurs with classical Henkin models of simply-typed lambda calculus. In general an equivalence relation must be placed on terms to obtain the *extensional collapse* of the underlying (non-extensional) applicative structure [14, p. 421].

To motivate the following definitions, consider the Henkin model obtained by the extensional collapse of the closed term applicative structure This is only guaranteed to exist if there are no term constants with second-order or higher types (again see [14, p. 421]). The reason is that the equivalence relation on terms is only guaranteed to agree with $=_{\beta\eta}$ on base types, so term constants of second-order or higher type might not respect the equivalence relation. However, if we allow free variables of base type (zero-order) then the equivalence relation agrees with $=_{\beta\eta}$ also on first-order types and it is possible to have term constants of second-order type. In general, if we have free variables at type level $n$ then $=_{\beta\eta}$ agrees with the extensional equivalence at level $n+1$ and the extensional collapse works for signatures containing types of level $n+2$ or lower. However, rather than work with type orders (which do not distinguish between distinct base types) everything can be stated with respect to specific types. We shall work with three sets of types: a primitive set of "quantifier" types classifying the variables that must be allowed to appear in terms (somewhere in the Kripke structure); a derived set of "equality" types whose extensional equivalence will be $=_{\beta\eta}$; and a derived set of "context" types containing all the types allowed in the signature.

For any type $A$, define the set $\mathrm{Args}(A)$ of *argument types* of $A$ by:

$$\begin{aligned}\mathrm{Args}(\alpha) &= \emptyset\\ \mathrm{Args}(A \to B) &= \{A\} \cup \mathrm{Args}(B).\end{aligned}$$

The set of quantifier types can be any set of types closed under arguments, i.e., for any quantifier type $A$, the set $\mathrm{Args}(A)$ must contain only quantifier types. We shall use $Q$ to range over quantifier types. The *equality types* are now generated by the grammar:

$$E \quad ::= \quad \alpha \mid Q \to E.$$

An easy induction on $A$ shows that if $\mathrm{Args}(A)$ is contained in the set of quantifier types then $A$ is an equality type. It follows that every quantifier type is an equality type. The *context types* are generated by the grammar:

$$K \quad ::= \quad \alpha \mid E \to K.$$

Clearly every equality type is a context type.

Now suppose we have a presentation $(\mathcal{T}, \mathcal{P}, \Sigma, \mathcal{A})$ such that, for all predicate declarations $P : \langle A_1, \ldots, A_n \rangle \in \mathcal{P}$, the types $A_1$, …, $A_n$ are all equality types and, for all constant declarations $c : A \in \Sigma$, $A$ is a context type. We shall construct a $\Sigma$-structure from the following ingredients. First, we require a partially ordered set $(W, \leq)$. This will be the partially ordered set of worlds underlying the eventual $\Sigma$-structure. Second, we require a function Ctxt mapping $W$ to contexts in which the type of every variable is a context type, such that:

1. $w \leq w'$ implies $\mathrm{Ctxt}(w) \subseteq \mathrm{Ctxt}(w')$, and

2. for all quantifier types $Q$, for all $w \in W$, there exists $w' \geq w$ such that $\mathrm{Ctxt}(w') \supseteq \mathrm{Ctxt}(w), x : Q$ for some $x$ not occurring in $\mathrm{Ctxt}(w)$.

Define

$$(\![A]\!)_w \quad = \quad \{M \mid \mathrm{Ctxt}(w) \rhd_\Sigma M : A\}.$$

Third, we require, for every $P : \langle A_1, \ldots, A_n \rangle \in \mathcal{P}$, a $W$-indexed family of relations $(\![P]\!)_w \subseteq (\![A_1]\!)_w \times \ldots \times (\![A_n]\!)_w$ respecting $=_{\beta\eta}$ such that $w \leq w'$ implies $(\![P]\!)_w \subseteq (\![P]\!)_{w'}$ (it is clear that condition 1 above implies $(\![A_i]\!)_w \subseteq (\![A_i]\!)_{w'}$).

The elements of $[\![A]\!]_w$ will be equivalence classes of $(\![A]\!)_w$ under an equivalence relation $\sim^A_w$. This is defined inductively on the structure of $A$ by:

$$\begin{array}{rll} M \sim^\alpha_w M' & \text{iff} & M =_{\beta\eta} M' \\ M \sim^{A\to B}_w M' & \text{iff} & \text{for all } w' \geq w, \text{ for all } N, N' \in (\![A]\!)_{w'}, \ N \sim^A_{w'} N' \text{ implies} \\ & & \quad M(N) \sim^B_{w'} M'(N'). \end{array}$$

Symmetry and transitivity are easily shown by induction on types. Reflexivity requires more work. Essentially it follows from the "Basic Lemma" for admissible Kripke logical relations over Kripke applicative structures with acceptable meaning functions (cf. [14, Lemma 3.2.5, p. 418]); however, Kripke logical relations have not been treated in such generality in the literature, so it is worthwhile giving the argument.

First, note that $\sim^A$ is monotone: if $M \sim^A_w M'$ and $w \leq w'$ then $M \sim^A_{w'} M'$ (by an easy induction on the structure of $A$).

**Lemma 4.7** *Suppose $M, M' \in (\![K]\!)_w$ (where $K$ is a context type) then $M =_{\beta\eta} M'$ implies $M \sim^K_w M'$. Further, if $K$ is an equality type then $M \sim^K_w M'$ implies $M =_{\beta\eta} M'$.*

**Proof.** By induction on the structure of $K$. For base types the proposition is trivial. For $E \to K'$ suppose that $M =_{\beta\eta} M'$. Take any $w' \geq w$ and $N, N' \in (\![E]\!)_{w'}$ such that $N \sim^{E}_{w'} N'$. Then, by the induction hypothesis, $N =_{\beta\eta} N'$ and so $M(N) =_{\beta\eta} M'(N')$. Thus, again by the induction hypothesis, $M(N) \sim^{K'}_{w'} (M')(N')$. This shows that indeed $M \sim^{E\to K'}_{w} M'$.

If $K$ is a non-atomic equality type then it has form $Q \to E$. Suppose that $M \sim^{Q\to E}_{w} M'$. Let $w' \geq w$ be such that $\mathrm{Ctxt}(w') \supseteq \mathrm{Ctxt}(w), x\!:\!Q$ (as in requirement 2 above). Then, by the induction hypothesis, $x \sim^{Q}_{w'} x$ so $M(x) \sim^{E}_{w'} M'(x)$. Again by the induction hypothesis, $M(x) =_{\beta\eta} M'(x)$. So indeed $M =_{\beta\eta} M'$. □

Thus for equality types $E$, $M \sim^{E}_{w} M'$ if and only if $M =_{\beta\eta} M'$, hence the reason for their name.

**Lemma 4.8** *If $M \sim^{A}_{w} M'$ and $M' =_{\beta\eta} M'' \in (\![A]\!)_{w}$ then $M \sim^{A}_{w} M''$.*

**Proof.** By induction on the structure of $A$. When $A$ is atomic the proposition is trivial. If $A$ is $B \to C$ then suppose $M \sim^{B\to C}_{w} M'$ and $M' =_{\beta\eta} M'' \in (\![B \to C]\!)_{w}$. Take any $w' \geq w$ and $N, N' \in (\![B]\!)_{w'}$ such that $N \sim^{B}_{w'} N'$. Then $M(N) \sim^{C}_{w'} M'(N')$. However, $M'(N') =_{\beta\eta} M''(N') \in (\![C]\!)_{w'}$. So, by the induction hypothesis, $M(N) \sim^{C}_{w'} M''(N')$. Thus indeed $M \sim^{B\to C}_{w} M''$. □

**Corollary 4.9** *If $M$ and $M'$ both have type $B$ in $Ctxt(w), x : A$ and $M[N/x] \sim^{B}_{w} M'[N'/x]$ then $(\lambda x\!:\!A.\, M)(N) \sim^{B}_{w} (\lambda x\!:\!A.\, M')(N')$.*

**Proof.** Apply Lemma 4.8 twice (using the symmetry of $\sim^{B}_{w}$). □

The next lemma is essentially the "Basic Lemma" of logical relations applied to $\sim^{A}_{w}$. We just outline the well-known argument, as all the interesting groundwork has been done.

**Lemma 4.10** *If $N_1 \sim^{A_1} N_1'$, ..., $N_k \sim^{A_k} N_k'$ and $Ctxt(w), x_1\!:\!A_1, \ldots, x_k\!:\!A_k \rhd_{\Sigma} M : A$ then $M[N_1, \ldots, N_k/x_1, \ldots, x_k] \sim^{A}_{w} M[N_1', \ldots, N_k'/x_1, \ldots, x_k]$.*

**Proof.** By induction on the structure of $M$. The case for a variable $x_i$ $(1 \leq i \leq k)$ is trivial. For a variable in $\mathrm{Ctxt}(w)$ or a constant in $\Sigma$ the proposition follows from Lemma 4.7 (as its type must be a context type). Applications follow easily from the definition of $\sim^{B\to C}_{w}$ for function types. Abstractions also follow from this definition using the monotonicity of $\sim^{A_1}$, ..., $\sim^{A_k}$ and Corollary 4.9. □

The reflexivity of $\sim^{A}_{w}$ is an immediate consequence of this lemma. We write $[M]_w$ for the equivalence class of $M \in (\![A]\!)_{w}$ under $\sim^{A}_{w}$. Note that it now follows from Lemma 4.8 that, for $M, M' \in (\![A]\!)_{w}$, $M =_{\beta\eta} M'$ implies $M \sim^{A}_{w} M'$.

We define the remaining components of the $\Sigma$-structure, $(W, \leq)$, by:

$$[\![A]\!]_w \quad = \quad (\![A]\!)_w / \sim^{A}_{w}$$

$$\begin{array}{rcl}
[\![P]\!]_w([M_1]_w, \ldots, [M_n]_w) & \text{iff} & (\![P]\!)_w(M_1, \ldots, M_n) \\
[\![c]\!]_w & = & [c]_w \\
\epsilon_w^{AB}([M]_w, [N]_w) & = & [M(N)]_w \\
i_{ww'}^A([M]_w) & = & [M]_{w'} \\
K_w^{AB} & = & [\lambda x\!:\!A.\, \lambda y\!:\!B.\, x]_w \\
S_w^{ABC} & = & [\lambda x\!:\!A \to B \to C.\, \lambda y\!:\!A \to B.\, \lambda z\!:\!A.\, x(z)(y(z))]_w
\end{array}$$

**Proposition 4.11** *$(W, \leq)$ is indeed a $\Sigma$-structure.*

**Proof.** For $P : \langle A_1, \ldots, A_n \rangle$, $[\![P]\!]_w$ is well-defined because $A_1$, …, $A_n$ are all equality types and so $\sim_w^{A_i}$ is just $=_{\beta\eta}$ and $(\![P]\!)_w$ respects $=_{\beta\eta}$. That the evaluation maps $\epsilon_w^{AB}$ are well-defined follows from the definition of $\sim_w^{A \to B}$. The coercions $i_{ww'}^A$ are well-defined because of the monotonicity of $\sim^A$. The other components are obviously well-defined.

It remains to check the various conditions. Identity, composition and naturality are all immediate. Extensionality is a straightforward consequence of the definition of $\sim_w^{A \to B}$. Persistency follows from the monotonicity requirement on $(\![P]\!)_w$. Lastly, $K^{AB}$ and $S^{ABC}$ satisfy the required equalities because $\sim_w^A$ contains the restriction of $=_{\beta\eta}$ to terms in $(\![A]\!)_w$. □

We conclude the presentation of the $\Sigma$-structure $(W, \leq)$ with two basic propositions relating terms and their interpretations.

**Proposition 4.12** *If $N_1 \in (\![A_1]\!)_w$, …, $N_k \in (\![A_k]\!)_w$ and $x_1 : A_1, \ldots, x_k : A_k \rhd_\Sigma M : A$ then $[\![M]\!]_w^{[x_1 := [N_1], \ldots, x_k := [N_k]]} = [M[N_1, \ldots, N_k / x_1, \ldots, x_k]]_w$.*

**Proof.** A straightforward induction on the structure of $M$. □

**Proposition 4.13** *If $N_1 \in (\![A_1]\!)_w$, …, $N_k \in (\![A_k]\!)_w$ and $\Phi$ is well-formed in the context $x_1\!:\!A_1, \ldots, x_k\!:\!A_k$ then $w \models_{[x_1 := [N_1], \ldots, x_k := [N_k]]} \Phi$ if and only if $w \models \Phi[N_1, \ldots, N_k / x_1, \ldots, x_k]$.*

**Proof.** Immediate from Proposition 4.12 and Lemma 4.5. □

We now prove of the completeness direction of Theorem 4.6. The result will not actually be required in the paper. However, the model we use for proving completeness will be reused in Section 6. The model is one extreme (and simple) case of the term model construction in Section 4. Let the set of quantifier types be the set of all types. Consider the partial order $(W_{(\Sigma,\mathcal{A})}, \leq_{(\Sigma,\mathcal{A})})$ defined by:

$$\begin{array}{rcl}
W_{(\Sigma,\mathcal{A})} & = & \{(\Gamma, \mathcal{H}) \mid \text{every formula in } \mathcal{H} \text{ is well-formed in } \Gamma\} \\
(\Gamma, \mathcal{H}) \leq_{(\Sigma,\mathcal{A})} (\Gamma', \mathcal{H}') & \text{iff} & \Gamma \subseteq \Gamma' \text{ and } \mathcal{H} \subseteq \mathcal{H}'.
\end{array}$$

The function Ctxt from $W_{(\Sigma,\mathcal{A})}$ to context is just the first projection. This clearly satisfies the two conditions above. The interpretation of predicates is:

$$(\![P]\!)_{(\Gamma,\mathcal{H})}(M_1, \ldots, M_n) \quad \text{iff} \quad \Gamma; \mathcal{H} \vdash_{(\Sigma,\mathcal{A})} P(M_1, \ldots, M_n).$$

This respects $=_{\beta\eta}$ because of the Sub rule, and is monotonic because of weakening (Lemma 2.1). We can now construct the $\Sigma$-structure $(W_{(\Sigma,\mathcal{A})}, \leq_{(\Sigma,\mathcal{A})})$ as above. For simplicity we shall just write $[M]$ for the equivalence class of a term $M$ at $(\Gamma,\mathcal{H})$. Note that, as every type is an equality type, the equivalence relation is always $=_{\beta\eta}$ (restricted to terms well-typed in the appropriate context).

Given a context $\Gamma$ we construct a canonical environment $\rho_\Gamma$ as follows. If, for some type $A$, the declaration $x : A$ is in $\Gamma$ then $\rho_\Gamma(x)$ is the partial element with domain $\{(\Gamma',\mathcal{H}) \mid \Gamma' \supseteq \Gamma\}$ given by $\rho_\Gamma(x)_{(\Gamma',\mathcal{H})} = [x]$. Otherwise, if there is no such $A$, then $\rho_\Gamma(x)$ is the trivial partial element with domain $\emptyset$. Clearly, for all $\Gamma' \supseteq \Gamma$, the environment $\rho_\Gamma$ interprets $\Gamma$ at $(\Gamma',\mathcal{H})$.

The crucial property of $(W_{(\Sigma_{\mathcal{A}})}, \leq_{(\Sigma,\mathcal{A})})$ is:

**Lemma 4.14** $\Gamma;\mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi$ *if and only if* $(\Gamma,\mathcal{H}) \models_{\rho_\Gamma} \Phi$.

**Proof.** By induction on the structure of $\Phi$. The cases are:

$P(M_1,\ldots,M_n)$. Immediate from the definitions of $[\![P]\!]_{(\Gamma,\mathcal{H})}$ and $(\![P]\!)_{(\Gamma,\mathcal{H})}$.

$\Phi \supset \Psi$. Suppose that $\Gamma;\mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi \supset \Psi$. Take any $\Gamma' \supseteq \Gamma$ and $\mathcal{H}' \supseteq \mathcal{H}$ such that $(\Gamma',\mathcal{H}') \models_{\rho_\Gamma} \Phi$. Then clearly $(\Gamma',\mathcal{H}') \models_{\rho_{\Gamma'}} \Phi$. So, by the induction hypothesis, $\Gamma';\mathcal{H}' \vdash_{(\Sigma,\mathcal{A})} \Phi$. Therefore $\Gamma';\mathcal{H}' \vdash_{(\Sigma,\mathcal{A})} \Psi$. Then, again by the induction hypothesis, $(\Gamma',\mathcal{H}') \models_{\rho_{\Gamma'}} \Psi$, so $(\Gamma',\mathcal{H}') \models_{\rho_\Gamma} \Psi$ (because all the free variables in $\Psi$ are in $\Gamma$). Thus we have shown that $(\Gamma,\mathcal{H}) \models_{\rho_\Gamma} \Phi \supset \Psi$ as required.

Conversely, suppose $(\Gamma,\mathcal{H}) \models_{\rho_\Gamma} \Phi \supset \Psi$. Now $\Gamma;\mathcal{H},\Phi \vdash_{(\Sigma,\mathcal{A})} \Phi$, so, by the induction hypothesis, $(\Gamma,\mathcal{H} \cup \{\Phi\}) \models_{\rho_\Gamma} \Phi$. Then, by the assumption, $(\Gamma,\mathcal{H} \cup \{\Phi\}) \models_{\rho_\Gamma} \Psi$. Whence, by the induction hypothesis, $\Gamma;\mathcal{H},\Phi \vdash_{(\Sigma,\mathcal{A})} \Psi$. Thus indeed $\Gamma;\mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi \supset \Psi$.

$\forall x\!:\!A.\,\Phi$. Suppose $\Gamma;\mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \forall x : A.\ \Phi$. Take any $\Gamma' \supseteq \Gamma$, $\mathcal{H}' \supseteq \mathcal{H}$ and $[M] \in [\![A]\!]_{(\Gamma',\mathcal{H}')}$. Then $\Gamma' \rhd_\Sigma M : A$ and $\Gamma';\mathcal{H}' \vdash_{(\Sigma,\mathcal{A})} \forall x\!:\!A.\,\Phi$, so $\Gamma';\mathcal{H}' \vdash_{(\Sigma,\mathcal{A})} \Phi[M/x]$. Now, by the induction hypothesis, $(\Gamma',\mathcal{H}') \models_{\rho_{\Gamma'}} \Phi[M/x]$. So, by Proposition 4.13, $(\Gamma',\mathcal{H}') \models_{\rho_{\Gamma'}[x:=[M]]} \Phi$. Now $(\Gamma',\mathcal{H}') \models_{\rho_\Gamma[x:=[M]]} \Phi$, because all the free variables in $\Phi$ are in $\Gamma', x\!:\!A$. We have therefore shown that $(\Gamma,\mathcal{H}) \models_{\rho_\Gamma} \forall x\!:\!A.\,\Phi$ as required.

Conversely, suppose that $(\Gamma,\mathcal{H}) \models_{\rho_\Gamma} \forall x\!:\!A.\,\Phi$. Now $[x] \in [\![A]\!]_{(\Gamma\cup\{x:A\},\mathcal{H})}$. Therefore $(\Gamma \cup \{x\!:\!A\},\mathcal{H}) \models_{\rho_\Gamma[x:=[x]]} \Phi$, and so $(\Gamma \cup \{x\!:\!A\},\mathcal{H}) \models_{\rho_{\Gamma\cup\{x:A\}}} \Phi$ (by the definition of $\rho_{\Gamma\cup\{x:A\}}$). So, by the induction hypothesis, $\Gamma, x : A;\mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi$. Thus indeed $\Gamma;\mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \forall x\!:\!A.\,\Phi$.

□

**Corollary 4.15** $(W_{(\Sigma,\mathcal{A})}, \leq_{(\Sigma,\mathcal{A})})$ *is a* $(\Sigma,\mathcal{A})$*-model.*

**Proof.** Immediate. □

Completeness now follows easily. Suppose that statement 2 of Theorem 4.6 holds. Then in particular $(\Gamma, \mathcal{H}) \models_{\rho(\Gamma,\mathcal{H})} \Phi$. So, by Lemma 4.14, $\Gamma; \mathcal{H} \vdash_{(\Sigma,\mathcal{A})} \Phi$ as required.

## 5 Semantic proofs of faithfulness

The semantics presented above provides the tool for proving the faithfulness properties deferred from Section 3. The proofs all have a common flavour. In each case we define a $\Sigma$-structure (or family of $\Sigma$-structures) and show that it is indeed a model of the relevant encoding. The faithfulness of the encoding will then follow almost immediately from the definition of the model and the soundness direction of Theorem 4.6. A more uniform analysis of the requirements on the model will be given in Section 6.

We begin with the encoding, $(\mathcal{T}_m, \mathcal{P}_m, \Sigma_m, \mathcal{A}_m)$, of minimal implicational logic given in Figure 3. Define $W_m$ to be the set of all finite sets of formulae of minimal implicational logic. We use $\Delta$ to range over $W_m$. We shall define a $\Sigma_m$-structure over the discrete partial order $(W_m, =)$. This will be a term model of the sort constructed in Section 4. The set of quantifier types is just the empty set (which is trivially closed under arguments). Clearly the only argument type to a predicate, $o$, is an equality type and it is easily checked that every type in $\Sigma_m$ is contained in the derived set of context types. The context function, $\mathrm{Ctxt}_m$, maps every element of $W_m$ to the empty set. This clearly satisfies the two required properties. Thus the set $(\![A]\!)_\Delta$ is just the set of closed terms of type $A$. The only predicate, *true*, is interpreted by:

$$(\![\mathit{true}]\!)_\Delta(M) \quad \text{iff} \quad \Delta \vdash_m \beta\eta(M)$$

where we are making use of the equivalence between long-$\beta\eta$ normal forms of type $o$ and formulae. Again this satisfies the required properties: it respects $=_{\beta\eta}$ by definition, and monotonicity is a vacuous requirement over the discrete partial order. These ingredients are all we need to define the $\Sigma_m$-structure $(W_m, =)$ as in Section 4.

In Section 1 we claimed that we would build models of the framework out of the syntax of the encoded logic. Although $(W_m, =)$ is built from terms of the framework, the crucial property is that we have only used closed terms, so $[\![o]\!]_\Delta$ is the set of equivalence classes of closed terms of type $o$ under $=_{\beta\eta}$. Therefore $[\![o]\!]_\Delta$ is in one-to-one correspondence with the set of formulae of minimal implicational logic (by taking $\beta\eta$-normal forms as the representatives of their equivalence classes). This correspondence enables us to interpret framework predicates over type $o$ as expressing (intuitionistic) properties of formulae of minimal implicational logic. To make this explicit, and for notational convenience, we henceforth identify $[\![o]\!]_\Delta$ with the set of formulae.

**Proposition 5.1** *$(W_m, =)$ is a $(\Sigma_m, \mathcal{A}_m)$-model.*

**Proof.** We must show that $(W_m, =)$ validates the three axioms in $\mathcal{A}_m$. Consider, for example, the third axiom:

$$\forall p\!:\!o.\, \forall q\!:\!o.\, \mathit{true}(p \Rightarrow q) \supset \mathit{true}(p) \supset \mathit{true}(q)$$

Suppose we have a world $\Delta$ and an environment $\rho$ with $\rho(p)_\Delta = \phi$, $\rho(q)_\Delta = \psi$, such that $\Delta \models_\rho \mathit{true}(p \Rightarrow q)$ and $\Delta \models_\rho \mathit{true}(p)$. Then, by Proposition 4.13, $\Delta \models \mathit{true}(\phi \Rightarrow \psi)$ and $\Delta \models \mathit{true}(\phi)$. So, by the definition of $(\![\mathit{true}]\!)_\Delta$, $\Delta \vdash_m \phi \Rightarrow \psi$ and $\Delta \vdash_m \phi$, whence, by *modus ponens* (for minimal logic), $\Delta \vdash_m \psi$. Thus $\Delta \models \mathit{true}(\psi)$ and so, again by Proposition 4.13, $\Delta \models_\rho \mathit{true}(q)$. It follows that $(W_m, =)$ does indeed validate the third axiom. The other two axioms are easier. □

Note how the proof follows the informal justification of the correctness of the axioms in $\mathcal{A}_m$.

It is now a simple step to prove the faithfulness direction of Proposition 3.1. Suppose that $\mathit{true}(\phi_1), \ldots, \mathit{true}(\phi_n) \vdash_{(\Sigma_m, \mathcal{A}_m)} \mathit{true}(\phi)$. Consider the world $\Delta = \{\phi_1, \ldots, \phi_n\}$ in $W_m$. Then clearly $\Delta \models \mathit{true}(\phi_1)$ and ... and $\Delta \models \mathit{true}(\phi_n)$. So, by the soundness direction of Theorem 4.6, $\Delta \models \mathit{true}(\phi)$. But then, by the definition of $(\![\mathit{true}]\!)_\Delta$, $\phi_1, \ldots, \phi_n \vdash_m \phi$ as required.

It is interesting to investigate the interpretation of formulae of the framework induced by the model $(W_m, =)$. Because the partial order is discrete, the model validates any instance of Peirce's law, i.e., for any framework sentences $\Phi$ and $\Psi$,

$$(W_m, =) \models ((\Phi \supset \Psi) \supset \Phi) \supset \Phi,$$

so meta-implication is given a classical interpretation. Therefore, the schema for Peirce's law could be added to $\mathcal{A}_m$ without losing adequacy.

A more interesting axiom that one might like to add to $\mathcal{A}_m$ is:

$$\forall p\!:\!o.\, \forall q\!:\!o.\, (\mathit{true}(p) \supset \mathit{true}(q)) \supset \mathit{true}(p \Rightarrow q). \tag{1}$$

Intuitively, this expresses the true fact that the Deduction Theorem holds for $\vdash_m$. Adding (1) to the encoding would enable derivations in the natural deduction system for minimal logic to be mimicked easily in the framework. However, $(W_m, =)$ does not validate (1). Consider the world $\emptyset$ and interpret $p$ by $a_0$ and $q$ by $a_1$. Then it is the case that $\emptyset \models \mathit{true}(a_0) \supset \mathit{true}(a_1)$, because the meta-implication is interpreted classically and $\not\vdash_m a_0$, but it is not the case that $\emptyset \models \mathit{true}(a_0 \Rightarrow a_1)$, because $\not\vdash_m a_0 \Rightarrow a_1$. The problem is that, because of its classical implication, $(W_m, =)$ does not interpret (1) as expressing the Deduction Theorem (see the discussion in Section 1). In fact, the reading of (1) as expressing the Deduction Theorem for minimal logic is inconsistent with a classical meta-implication: if Peirce's law and (1) are added together to $\mathcal{A}_m$ then the resulting encoding is not adequate (it is easy to derive $\mathit{true}(((a_0 \Rightarrow a_1) \Rightarrow a_0) \Rightarrow a_0))$.

However, if just (1) alone is added to $\mathcal{A}_m$ then adequacy is retained. To show this we need a $\Sigma_m$-structure in which meta-implication is interpreted intuitionistically. The modification required is very slight. We take the term model defined over $(W_m, \subseteq)$ generated by the same (empty) set of quantifier types, $\mathrm{Ctxt}_m$ and $(\![\mathit{true}]\!)_\Delta$ (it is easily checked that the required properties hold). The proof of Proposition 5.1 applies *mutatis mutandis* to $(W_m, \subseteq)$ so this is indeed a $(\Sigma_m, \mathcal{A}_m)$-model.

**Proposition 5.2** $(W_m, \subseteq) \models \forall p\!:\!o.\, \forall q\!:\!o.\, (\mathit{true}(p) \supset \mathit{true}(q)) \supset \mathit{true}(p \Rightarrow q)$.

**Proof.** Suppose we have a world $\Delta$ and an environment $\rho$ with $\rho(p)_\Delta = \phi$, $\rho(q)_\Delta = \psi$, such that $\Delta \models_\rho (\mathit{true}(p) \supset \mathit{true}(q))$. Then, by Proposition 4.13, $\Delta \models \mathit{true}(\phi) \supset \mathit{true}(\psi)$. However $\Delta \cup \{\phi\} \supseteq \Delta$ is a world such that $\Delta \cup \{\phi\} \models \mathit{true}(\phi)$. So $\Delta \cup \{\phi\} \models \mathit{true}(\psi)$. But then $\Delta, \phi \vdash_m \psi$ (by the definition of $(\![\mathit{true}]\!)_{\Delta \cup \{\phi\}}$). So $\Delta \vdash_m \phi \Rightarrow \psi$, by the Deduction Theorem for $\vdash_m$. Therefore $\Delta \models \mathit{true}(\phi \Rightarrow \psi)$. So, again by Proposition 4.13, $\Delta \models_\rho \mathit{true}(p \Rightarrow q)$ as required. □

Note that once again the proof follows the intuitive explanation of the correctness of (1) using the Deduction Theorem. Further, the non-triviality of the partial order was essential to the proof. The faithfulness of the encoding now follows by the same argument as before.

We now turn to the encoding, $(\mathcal{T}_T, \mathcal{P}_T, \Sigma_T, \mathcal{A}_T)$, of a first-order theory $T$ given in Figure 4. This time we have axiom (1) built into the encoding. However, this time it is not inconsistent with a classical meta-implication, because the implication of the encoded logic is also classical. This fact enables a very simple proof of faithfulness based on interpretations in the classical full type hierarchy.

Let $\mathcal{M} = (D, \{P_i^{A_i}\}, \{f_i^{a_i}\})$ be any first-order model of $T$. We shall construct a $\Sigma_T$ model, $(\{w_\mathcal{M}\}, =)$, over the trivial partial order. Thus essentially we shall be working with a standard Henkin model [14, §2.4]. None of the extra generality of Kripke lambda models will be required. However, we shall see later that the present construction has its limitations.

Types are interpreted by (as there is only one world we omit the subscript):

$$\begin{aligned} [\![\iota]\!] &= D \\ [\![o]\!] &= \{\mathrm{tt}, \mathrm{ff}\} \\ [\![A \to B]\!] &= [\![B]\!]^{[\![A]\!]} \end{aligned}$$

where $[\![B]\!]^{[\![A]\!]}$ is the full set-theoretic function space. The predicate is interpreted by:

$$[\![\mathit{true}]\!](b) \quad \text{iff} \quad b = \mathrm{tt}.$$

The constants are interpreted, following the semantics of first-order logic, by:

$$\begin{aligned}
[\![\Rightarrow]\!](b_1)(b_2) &= \begin{cases} \mathrm{tt} & \text{if } b_1 = \mathrm{ff} \text{ or } b_2 = \mathrm{tt} \\ \mathrm{ff} & \text{otherwise} \end{cases} \\
[\![\neg]\!](b) &= \begin{cases} \mathrm{tt} & \text{if } b = \mathrm{ff} \\ \mathrm{ff} & \text{if } b = \mathrm{tt} \end{cases} \\
[\![\forall]\!](f) &= \begin{cases} \mathrm{tt} & \text{if, for all } d \in D,\ f(d) = \mathrm{tt} \\ \mathrm{ff} & \text{otherwise} \end{cases} \\
[\![=]\!](d_1)(d_2) &= \begin{cases} \mathrm{tt} & \text{if } d_1 = d_2 \\ \mathrm{ff} & \text{otherwise} \end{cases} \\
[\![P_i^{a_i}]\!](d_1)\dots(d_{a_i}) &= \begin{cases} \mathrm{tt} & \text{if } P_i^{a_i}(d_1,\dots,d_{a_i}) \\ \mathrm{ff} & \text{otherwise} \end{cases} \\
[\![f_i^{a'_i}]\!](d_1)\dots(d_{a'_i}) &= f(d_1,\dots,d_{a'_i}).
\end{aligned}$$

The components $\epsilon^{AB}$, $K^{AB}$ and $S^{ABC}$ are given their standard interpretations in the full type hierarchy [14, p. 384]. (The coercions $i^A_{w_\mathcal{M} w_\mathcal{M}}$ are of course trivial.)

Let $X$ be a finite set of first-order variables. Let $\rho$ be any function from $X$ to $D$. If $t$ is a first-order term with all free variables in $X$ then we write $t^\rho_\mathcal{M}$ for its induced interpretation in $\mathcal{M}$. Clearly $\rho$ as defined is an environment interpreting $X:\iota$ (at $w_\mathcal{M}$) in $(\{w_\mathcal{M}\}, =)$. By the correspondence between framework terms and first-order terms, $t$ is also a term of type $\iota$ in long-$\beta\eta$ form with respect to $X:\iota$.

**Lemma 5.3** $t^\rho_\mathcal{M} = [\![t]\!]^\rho$.

**Proof.** An easy induction on the structure of $t$. □

Now let $\phi$ be any first-order formulae with all free variables in $X$. The satisfaction of the first-order formula $\phi$ by $\mathcal{M}$ is related to the interpretation of the term $\phi$ in $(\{w_\mathcal{M}\}, =)$ by:

**Lemma 5.4** *$\mathcal{M} \models_\rho \phi$ if and only if $[\![\phi]\!]^\rho = tt$.*

**Proof.** A straightforward induction on the structure of $\phi$ using Lemma 5.3 for atomic formulae. □

**Proposition 5.5** *$(\{w_\mathcal{M}\}, =)$ is a $(\Sigma_T, \mathcal{A}_T)$-model.*

**Proof.** We must show that $(\{w_\mathcal{M}\}, =)$ validates the $8+l$ axioms in $\mathcal{A}_T$. For the axioms $\mathit{true}(T_1), \dots, \mathit{true}(T_l)$, this is immediate by Lemma 5.4. We consider two interesting cases from the other eight axioms.

1. $\forall p\colon o.\,\forall q\colon o.\,(true(p) \supset true(q)) \supset true(p \Rightarrow q)$.

   Suppose we an environment $\rho$ with $\rho(p) = b$ and $\rho(q) = b'$ such that $w_{\mathcal{M}} \models_{\rho} true(p) \supset true(q)$. Then either $b = \mathrm{ff}$ or $b' = \mathrm{tt}$. So $[\![p \Rightarrow q]\!]^{\rho} = \mathrm{tt}$. Therefore $w_{\mathcal{M}} \models_{\rho} true(p \Rightarrow q)$ as required.

2. $\forall p\colon \iota \to o.\,(\forall x\colon \iota.\,true(p(x))) \supset true(\forall x.\,p(x))$.

   Suppose we have an environment $\rho$ with $\rho(p) = f$ such that $w_{\mathcal{M}} \models_{\rho} \forall x\colon \iota.\,true(p(x))$. Then given any $d \in [\![\iota]\!]$, we have that $w_{\mathcal{M}} \models_{\rho[x:=d]} true(p(x))$, so $f(d) = \mathrm{tt}$. But then $[\![\forall x\;p(x)]\!]^{\rho} = [\![\forall]\!](f) = \mathrm{tt}$. Thus indeed $w_{\mathcal{M}} \models_{\rho} true(\forall x.\,p(x))$.

□

We can now establish the faithfulness direction of Proposition 3.2. Suppose then that $X\colon\iota;\; true(\phi_1), \ldots, true(\phi_n) \vdash_{(\Sigma_T, \mathcal{A}_T)} true(\phi)$. Let $\mathcal{M}$ be any model of $T$ and let $\rho$ be any first-order environment such that $\mathcal{M} \models_{\rho} \phi_1$ and ... and $\mathcal{M} \models_{\rho} \phi_n$. Then, by Lemma 5.4, $w_{\mathcal{M}} \models_{\rho} true(\phi_1)$ and ... and $w_{\mathcal{M}} \models_{\rho} true(\phi_n)$ (in the model $(\{w_{\mathcal{M}}\}, =)$). So, by the soundness direction of Theorem 4.6, $w_{\mathcal{M}} \models_{\rho} true(\phi)$. But then, again by Lemma 5.4, $\mathcal{M} \models_{\rho} \phi$. Therefore, by the semantic characterization of $\vdash_T$, we have established that indeed $\phi_1, \ldots, \phi_n \vdash_T \phi$.

Although we restricted attention to a finite first-order signature and a finite set of axioms, nothing in the argument has relied upon this fact. However, the given argument only extends to infinite first-order signatures and axiom sets by taking the evident infinite presentation of the theory. Often, however, one has an infinite axiomatization by way of a finite number of axiom schemas. In such cases one would like to encode the logic by a finite presentation.

A case in point is Peano Arithmetic. This has no finite first-order axiomatization but it can be given a finite presentation in the framework. The presentation, which we call $(\Sigma_A, \mathcal{A}_A)$, is just the presentation for the first-order theory (in the language of arithmetic) given by Peano's six axioms, augmented with the framework axiom:

$$\forall p\colon \iota \to o.\; true(p(0) \Rightarrow (\forall x.\,p(x) \Rightarrow p(s(x))) \Rightarrow \forall x.\,p(x)) \tag{2}$$

expressing the induction schema. A similar encoding of arithmetic (in the context of LF) was proved faithful by Harper *et al* [9]. If the $\Sigma_A$-structures $(\{w_{\mathcal{M}}\}, =)$ (constructed as above where $\mathcal{M}$ is a model of Peano Arithmetic) were $(\Sigma_A, \mathcal{A}_A)$-models then we should have a simple semantic proof of faithfulness. Unfortunately, in general they are not! To see this, consider any model $\mathcal{M}$ of Peano Arithmetic such that

$$w_{\mathcal{M}} \models \forall p\colon \iota \to o.\; true(p(0) \Rightarrow (\forall x.\,p(x) \Rightarrow p(s(x))) \Rightarrow \forall x.\,p(x)).$$

Then, for all $f \in [\![\iota \to o]\!]$, $[\![true(p(0) \Rightarrow (\forall x.\,p(x) \Rightarrow p(s(x))) \Rightarrow \forall x.p(x))]\!]^{[p:=f]} = \mathrm{tt}$. But this says that $\mathcal{M}$ satisfies the full second-order induction axiom, which means that $\mathcal{M}$

must be isomorphic to the standard model of arithmetic. Therefore the above argument for faithfulness does not go through.

The example illustrates nicely the limitations of building a model of the framework out of a model of the encoded logic. In the first-order models the variable $p$ ranges over all subsets of the domain, whereas the desired semantic reading of (2) is for $p$ to range only over the arithmetically definable subsets. It may be possible to patch the model construction by considering only a subset of all functions at higher type, as in Henkin's general models of higher order logic [10]. However, we have no definite ideas about how to do this and the details would almost certainly be complicated.

In order to give a proof of the faithfulness of the encoding, we resort instead to a term model construction. This time the set of quantifier types is necessarily non-empty, for otherwise the type of $\forall$ would not be a context type. In fact we can take the set of quantifier types to be just $\{\iota\}$. The model will be constructed over of the partial order $(W_A, \leq_A)$ where:

$$\begin{aligned} W_A \quad &= \quad \{(X,\Delta) \mid X \text{ is a finite set of variables and } \Delta \text{ is a} \\ &\qquad\qquad \text{finite set of formulae with all free variables in } X\} \\ (X,\Delta) \leq_A (X',\Delta') \quad &\text{iff} \quad X \subseteq X' \text{ and } \Delta \subseteq \Delta'. \end{aligned}$$

The context function, $\mathrm{Ctxt}_A$, maps $(X,\delta)$ to $X\!:\!\iota$. This clearly satisfies the two properties. demanded of it. The predicate, *true*, is interpreted by:

$$([\mathit{true}])_{(X,\Delta)}(M) \quad \text{iff} \quad \Delta \vdash_A \beta\eta(M)$$

where $M$ is a term of type $o$ in the context $X\!:\!\iota$, so $\beta\eta(M)$ is indeed (according to our correspondence) a formula (with all free variables in $X$). We now define the $\Sigma_A$-structure $(W_A, \leq_A)$ as is Section 4.

Once more, the restriction to terms typed in contexts of the form $X : \iota$ means that the sets $[\![\iota]\!]_{(X,\Delta)}$ and $[\![o]\!]_{(X,\Delta)}$ correspond to sets of terms and formulae of the encoded logic. Again we take the notational liberty of working with $[\![\iota]\!]_{(X,\Delta)}$ and $[\![o]\!]_{(X,\Delta)}$ as if they were respectively the set of arithmetic terms with variables in $X$ and the set of arithmetic formula with free variables in $X$. Similarly, because $\iota \to o$ is an equality type, we can work with $[\![\iota \to o]\!]_{(X,\Delta)}$ as if it were the set of terms of type $\iota \to o$ in long-$\beta\eta$ normal form with respect to $X\!:\!\iota$.

**Proposition 5.6** *$(W_A, \leq_A)$ is a $(\Sigma_A, \mathcal{A}_A)$-model.*

**Proof.** We must show that $(W_A, \leq_A)$ validates all the axioms in $\mathcal{A}$. For (1) this is shown as in the proof of Proposition 5.2. We consider just two of the other axioms:

1. $\forall p\!:\!\iota \rightarrow o.\,(\forall x\!:\!\iota.\, \mathit{true}(p(x))) \supset \mathit{true}(\forall x.\, p(x))$.

   Suppose we have a world $(X,\Delta)$ and an environment $\rho$ with $\rho(p)_{(X,\Delta)} = \lambda x : \iota.\, \phi$ such that $(X,\Delta) \models_\rho \forall x : \iota.\, \mathit{true}(p(x))$. Then $(X,\Delta) \models \forall x : \iota.\, \mathit{true}(\phi)$, by Proposition 4.13. Let $x$ be any variable not in $X$. Then $x \in [\![\iota]\!]_{(X \cup \{x\},\Delta)}$ and so $(X \cup \{x\},\Delta) \models_{[x:=x]} \mathit{true}(\phi)$. Therefore $(X \cup \{x\},\Delta) \models \mathit{true}(\phi)$, by Proposition 4.13. So $\Delta \vdash_A \phi$ where $x$ does not appear free in $\Delta$. But then, by the $\forall$-introduction rule of natural deduction, $\Delta \vdash_A \forall x.\, \phi$. So $(X,\Delta) \models \mathit{true}(\forall x.\, \phi)$. Thus, again by Proposition 4.13, $(X,\Delta) \models_\rho \mathit{true}(\forall x.\, p(x))$ as required.

2. $\forall p\!:\!\iota \rightarrow o.\, \mathit{true}(p(0) \Rightarrow (\forall x.\, p(x) \Rightarrow p(s(x))) \Rightarrow \forall x.\, p(x))$.

   Suppose we have a world $(X,\Delta)$ and an environment $\rho$ with $\rho(p)_{(X,\Delta)} = \lambda x : \iota.\, \phi$. Now, by the induction schema, $\Delta \vdash_A \phi[0/x] \Rightarrow (\forall x.\, \phi \Rightarrow \phi[s(x)/x]) \Rightarrow \forall x.\, \phi$. So $(X,\Delta) \models \mathit{true}(\phi[0/x] \Rightarrow (\forall x.\, \phi \Rightarrow \phi[s(x)/x]) \Rightarrow \forall x.\, \phi)$. But then, by Proposition 4.13, $(X,\Delta) \models_\rho \mathit{true}(p(0) \Rightarrow (\forall x.\, p(x) \Rightarrow p(s(x))) \Rightarrow \forall x.\, p(x))$ as required.

□

Note that this time the partial order is important. The formula set component of the partial order is required to validate (1) (as in the proof of Proposition 5.2), and the variable set component is required for "$\forall$-introduction".

For faithfulness, suppose that $X : \iota;\ \mathit{true}(\phi_1), \ldots, \mathit{true}(\phi_n) \vdash_A \mathit{true}(\phi)$ where all the free variables in the formulae are contained in $X$. Consider the world $(X,\Delta)$ where $\Delta = \{\phi_1, \ldots, \phi_n\}$. Let $\rho$ be any environment mapping each $x$ in $X : \iota$ to (the partial element generated by) $x$ (at $(X,\Delta)$). Then clearly $(X,\Delta) \models_\rho \mathit{true}(\phi_1)$ and ... and $(X,\Delta) \models_\rho \mathit{true}(\phi_n)$. Therefore $(X,\Delta) \models_\rho \mathit{true}(\phi)$. So indeed $\Delta \vdash_A \phi$.

Clearly the above argument applies equally well in the general case of an arbitrary first-order theory axiomatized by a finite number of axiom schemas. So the general proof of faithfulness we gave using models constructed out of first-order models can be bettered by using term models. What then is the point of considering the original more semantic modelsΓ One reason for considering such models is to give, when possible, a truly semantic interpretation of the encoding. Further, such models are rather simpler than the term models. A more pragmatic reason for considering them is afforded by encodings, such as our presentation of the modal logic K in Figure 5, based directly on the semantics of the encoded logic. In the case of this encoding it is not clear how a proof using term models would proceed (as the encoding is not based on any proof system). However, it is quite straightforward to give a proof of faithfulness based on models of the framework built out of Kripke models of modal logic.

Given a Kripke model, $\mathcal{M} = (W, R, \models)$, we construct once more a model $(\{w_{\mathcal{M}}\}, =)$

in the full type hierarchy. The base types are interpreted by:

$$\begin{aligned}[\![N]\!] &= \mathbf{N}\\ [\![o]\!] &= \wp(W)\\ [\![W]\!] &= W\end{aligned}$$

where $\mathbf{N}$ is the set of natural numbers. (Function types are again interpreted by the full function space.) The predicates are interpreted by:

$$\begin{aligned}[\![R]\!](w,v) &\quad \text{iff} \quad wRv\\ [\![\mathit{sat}]\!](w,V) &\quad \text{iff} \quad w \in V\end{aligned}$$

The constants are interpreted by:

$$\begin{aligned}[\![0]\!] &= 0\\ [\![s]\!](n) &= n+1\\ [\![a]\!](n) &= \{w \mid w \models a_n\}\\ [\![\Rightarrow]\!](V)(V') &= (W\backslash V) \cup V'\\ [\![\neg]\!](V) &= W\backslash V\\ [\![\Box]\!](V) &= \{w \mid \text{for all } v \in W,\, wRv \text{ implies } v \in V\}\end{aligned}$$

The definitions above follow the standard interpretation of modal formulae in the Boolean algebra $(\wp(W), \subseteq)$ induced by the Kripke model $\mathcal{M}$.

**Lemma 5.7** *$w \models \phi$ if and only if $w \in [\![\phi]\!]$.*

**Proof.** By a straightforward induction on the structure of $\phi$. □

**Proposition 5.8** *$(\{w_{\mathcal{M}}\}, \supseteq)$ is a $(\Sigma_K, \mathcal{A}_K)$-model.*

**Proof.** We must show that $(\{w_{\mathcal{M}}\}, \supseteq)$ validates the six axioms. We just consider one case: $\forall x : W.\ \forall p{:}o.\ (\forall y{:}W.\ R(x,y) \supset \mathit{sat}(y,p)) \supset \mathit{sat}(x, \Box p)$. Suppose we have an environment $\rho$ with $\rho(x) = w$ and $\rho(p) = V$ such that $w_{\mathcal{M}} \models_\rho \forall y{:}W.\ R(x,y) \supset \mathit{sat}(y,p)$. Then, for all $v \in W$, $wRv$ implies $v \in V$. Thus $w \in [\![\Box]\!](V)$. So $w_{\mathcal{M}} \models_\rho \mathit{sat}(x, \Box p)$ as required. □

To establish faithfulness, suppose that $x{:}W;\ \mathit{sat}(x,\phi_1), \ldots, \mathit{sat}(x,\phi_n) \vdash_{(\Sigma_K,\mathcal{A}_K)} \mathit{sat}(x,\phi)$. Now take any Kripke model $\mathcal{M} = (W, R, \models)$ and any $w \in W$ such that $w \models \phi_1$ and $\ldots$ and $w \models \phi_n$. Then, by Lemma 5.7, $w_{\mathcal{M}} \models_{[x:=w]} \mathit{sat}(x,\phi_1)$ and $\ldots$ and $w_{\mathcal{M}} \models_{[x:=w]} \mathit{sat}(x,\phi_n)$ (in the model $(\{w_{\mathcal{M}}\}, =)$). So, by the soundness direction of Theorem 4.6, $w_{\mathcal{M}} \models_{[x:=w]} \mathit{sat}(x,\phi)$. Whence, again by Lemma 5.7, $w \models \phi$. This shows that indeed $\phi_1, \ldots, \phi_n \vdash_K \phi$ (by the semantic characterization of $\vdash_K$).

## 6 Towards a general theory

In this section we consider more abstractly the theory underlying the above applications. We give a general definition of what it means to encode a logic in the framework. This enables us to give semantic characterizations of the properties of fullness and faithfulness. We then discuss how well the general theory applies to the examples we have considered.

In order to give a general definition of encoding, we need first a general notion of logic. We consider a logic abstractly as a consequence relation. A *consequence relation*, $\vdash$, over a set $\mathcal{L}$ (of *sentences*) is a binary relation between finite subsets of $\mathcal{L}$ and $\mathcal{L}$ satisfying the following three conditions.

**reflexivity:** $\phi \vdash \phi$.

**weakening:** If $\phi_1, \ldots, \phi_n \vdash \phi$ then $\phi_1, \ldots, \phi_n, \psi \vdash \phi$.

**transitivity:** If $\phi_1, \ldots, \phi_n \vdash \phi$ and $\phi_1, \ldots, \phi_n, \phi \vdash \psi$ then $\phi_1, \ldots, \phi_n \vdash \psi$.

A consequence relation, $\vdash$, determines a closure operator, $\mathit{Th} : \wp(\mathcal{L}) \to \wp(\mathcal{L})$ by:

$$\mathit{Th}(\Delta) \quad = \quad \{\phi \mid \text{there exist } \phi_1, \ldots, \phi_n \in \Delta \text{ such that } \phi_1, \ldots, \phi_n \vdash \phi\}$$

where, in contrast to earlier, we use $\Delta$ to range over arbitrary subsets of $\mathcal{L}$. Further, *Th* is continuous (it preserves directed unions). In fact consequence relations are in one-to-one correspondence with continuous closure operators. We call those sets in the image of the closure operator *theories* and we write *Theories* for the set of all theories. Note that $\mathcal{L}$ itself is a theory. (Thus we are not requiring all theories to be "consistent".)

The definition of consequence relation (in the form of continuous closure operator) goes back to Tarski [22]. For discussion on the appropriateness of the notion of consequence relation as an abstract notion of logic see Scott [21], Avron [2] and Ryan and Sadler [20].

The definition of consequence relation using only finite sets in the antecedent amounts to an assumption of compactness. Many model-theoretic logics have non-compact consequence relations. To treat these one must consider relations between $\wp(\mathcal{L})$ and $\mathcal{L}$. The correct axioms require infinitary forms of weakening and transitivity giving relations corresponding to arbitrary closure operators on $\wp(\mathcal{L})$. However, for our purposes compact consequence relations suffice, for it is impossible to adequately encode a non-compact consequence relation in the framework (because of the compactness of the logic of the framework).

An *encoding* of a consequence relation $\vdash$ over $\mathcal{L}$ is given by a presentation $(\mathcal{T}, \mathcal{P}, \Sigma, \mathcal{A})$ containing the following ingredients:

1. A distinguished type constant $o$ in $\mathcal{T}$ together with a function:

$$(\cdot)^* : \mathcal{L} \to \{M \mid \rhd_\Sigma M : o,\ M \text{ is in long-}\beta\eta \text{ normal form}\}$$

   mapping formulae of the object-logic to their representing terms in the framework.

2. A distinguished predicate, *true*, with declaration $true{:}\langle o\rangle \in \mathcal{P}$.

One might prefer to ask that $(\cdot)^*$ be injective or even bijective, however such requirements make no difference to the theory. The encoding is said to be *full* if

$$\phi_1, \ldots, \phi_n \vdash \phi \quad \text{implies} \quad true(\phi_1^*), \ldots, true(\phi_n^*) \vdash_{(\Sigma,\mathcal{A})} true(\phi^*);$$

it is said to be *faithful* if

$$true(\phi_1^*), \ldots, true(\phi_n^*) \vdash_{(\Sigma,\mathcal{A})} true(\phi^*) \quad \text{implies} \quad \phi_1, \ldots, \phi_n \vdash \phi.$$

An encoding that is both full and faithful is said to be *adequate*.

At this point it is worth considering how well our earlier examples fit into the general scheme. Only the encoding of minimal implicational logic requires no change. The encoding of the modal logic $K$ has no *true* predicate. However, it is easy to massage the encoding into the scheme. One way of doing so is to include the *true* predicate together with a new constant $w_0 : W$ in $\Sigma_K$ and extend the axioms with:

$$\begin{array}{l} \forall p{:}o.\ sat(w_0, p) \supset true(p) \\ \forall p{:}o.\ true(p) \supset sat(w_0, p) \end{array}$$

Semantically, this amounts to considering consequence over pointed Kripke models. For the encoding of first-order logic, the problem is that the consequence relation we consider is between open formulae, whereas in our general definition of encoding we consider only closed terms in the framework. An obvious way of fitting first-order logic into the general scheme is to consider instead the consequence relation over sentences. However, it would be more interesting to generalize the notions of consequence relation and encoding to cater explicitly for free variables and substitution. One proposal along these lines has been recently suggested by Gardner [7]. However, for simplicity, we work with the less complicated "propositional" notion of consequence relation.

There are other changes that one might make to the notion of encoding. Gardner gives a general definition of encoding adopting fullness as one of the requirements [8]. However, we prefer to keep the definition of encoding as general as possible considering the two halves of adequacy as extra conditions (which in practice an encoding must satisfy). This approach allows us to analyse semantic properties corresponding to the fullness and faithfulness of an encoding. It must be said that neither one of fullness and faithfulness is particularly interesting without the other. A trivial full encoding is always given by the axiom $\forall p\!:\!o.\ \mathit{true}(p)$. A trivial faithful encoding is given by an empty $\mathcal{A}$. Further, some faithful encodings do not extend to adequate encodings. For an example consider the encoding of minimal implicational logic obtained by replacing the entire set $\mathcal{A}_m$ with $\{\forall p\!:\!o.\ \mathit{true}(p \Rightarrow p) \supset \mathit{true}(p)\}$ (the faithfulness of this is an easy semantic exercise).

The two halves of adequacy each correspond to conditions on the class of $(\Sigma, \mathcal{A})$-models. Given a $(\Sigma, \mathcal{A})$-model, $(W, \leq)$, define $\mathit{form} : W \to \wp(\mathcal{L})$ by:

$$\mathit{form}(w) \quad = \quad \{\phi \mid w \models \mathit{true}(\phi^*)\}$$

We write $\mathit{form}(W)$ for the set $\{\mathit{form}(w) \mid w \in W\}$.

**Theorem 6.1 (Semantic characterization of fullness)** *The following are equivalent:*

1. *The encoding is full.*

2. *For all $(\Sigma, \mathcal{A})$-models $(W, \leq)$, $\mathit{form}(W) \subseteq \mathit{Theories}$.*

**Proof.** Suppose the encoding is full. Let $(W, \leq)$ be any $(\Sigma, \mathcal{A})$-model, and take any $w \in W$. We must show that $\mathit{form}(w) \in \mathit{Theories}$. Suppose that $\phi_1, \ldots, \phi_n \vdash \phi$ for some $\phi_1, \ldots, \phi_n \in \mathit{form}(w)$. By fullness, $\mathit{true}(\phi_1^*), \ldots, \mathit{true}(\phi_n^*) \vdash_{(\Sigma,\mathcal{A})} \mathit{true}(\phi^*)$. But, by the definition of $\mathit{form}(w)$, $w \models \mathit{true}(\phi_1^*)$ and … and $w \models \mathit{true}(\phi_n^*)$. So, by the soundness of the meta-logic, $w \models \mathit{true}(\phi^*)$. But then $\phi \in \mathit{form}(w)$. Thus $\mathit{form}(w) = \mathit{Th}(\mathit{form}(w))$. So indeed $\mathit{form}(w) \in \mathit{Theories}$.

Conversely, suppose that, for all $(\Sigma, \mathcal{A})$-models $(W, \leq)$, $\mathit{form}(W) \subseteq \mathit{Theories}$. Suppose further that $\phi_1, \ldots, \phi_n \vdash \phi$. We must show that $\mathit{true}(\phi_1^*), \ldots, \mathit{true}(\phi_n^*) \vdash_{(\Sigma,\mathcal{A})} \mathit{true}(\phi^*)$. For this we use the model, $(W_{(\Sigma,\mathcal{A})}, \leq_{(\Sigma,\mathcal{A})})$, used in the proof of completeness in Section 4. By Corollary 4.15, we know this is indeed a $(\Sigma, \mathcal{A})$-model. Consider the world $w = (\emptyset, \{\mathit{true}(\phi_1^*), \ldots, \mathit{true}(\phi_n^*)\})$. By the initial supposition, $\mathit{form}(w) \in \mathit{Theories}$. But clearly $\{\phi_1, \ldots, \phi_n\} \subseteq \mathit{form}(w)$, so $\phi \in \mathit{form}(w)$ (as any theory is closed under consequence). Therefore $w \models \mathit{true}(\phi^*)$. So, by the definition of $(W_{(\Sigma,\mathcal{A})}, \leq_{(\Sigma,\mathcal{A})})$, $\mathit{true}(\phi_1^*), \ldots, \mathit{true}(\phi_n^*) \vdash_{(\Sigma,\mathcal{A})} \mathit{true}(\phi^*)$ as required. □

**Theorem 6.2 (Semantic characterization of faithfulness)** *The following are equivalent:*

1. *The encoding is faithful.*

2. *There exists a $(\Sigma, \mathcal{A})$-model, $(W, \leq)$, such that $\mathit{Theories} \subseteq \mathit{form}(W)$.*

**Proof.** Suppose the encoding is faithful. We will show that the $(\Sigma, \mathcal{A})$-model $(W_{(\Sigma,\mathcal{A})}, \leq_{(\Sigma,\mathcal{A})})$ has the required property. Take any $T \in \mathit{Theories}$. We must show that there exists $(\Gamma, \mathcal{H}) \in W_{(\Sigma,\mathcal{A})}$ with $\mathit{form}((\Gamma, \mathcal{H})) = T$. For this we take the world $w = (\emptyset, \{\mathit{true}(\psi^*) \mid \psi \in T\})$. Clearly (from the definition of $(W_{(\Sigma,\mathcal{A})}, \leq_{(\Sigma,\mathcal{A})})$) $T \subseteq \mathit{form}(w)$. We now show that $\mathit{form}(w) \subseteq T$. Suppose that $\phi \in \mathit{form}(w)$. Then $w \models \mathit{true}(\phi^*)$, so $\{\mathit{true}(\psi^*) \mid \phi \in T\} \vdash_{(\Sigma,\mathcal{A})} \mathit{true}(\phi^*)$. By the (evident) compactness of $\vdash_{(\Sigma,\mathcal{A})}$, there exist $\phi_1, \ldots, \phi_n \in T$ such that $\mathit{true}(\phi_1^*), \ldots, \mathit{true}(\phi_n^*) \vdash_{(\Sigma,\mathcal{A})} \mathit{true}(\phi^*)$. Therefore, by faithfulness, $\phi_1, \ldots, \phi_n \vdash \phi$. Thus indeed $\phi \in T$ (as $T$ is closed under consequence).

Conversely, suppose that there exists a $(\Sigma, \mathcal{A})$-model, $(W, \leq)$, such that $\mathit{Theories} \subseteq \mathit{form}(W)$. Suppose further that $\mathit{true}(\phi_1^*), \ldots, \mathit{true}(\phi_n^*) \vdash_{(\Sigma,\mathcal{A})} \mathit{true}(\phi^*)$. We must show that $\phi_1, \ldots, \phi_n \vdash \phi$. Let $w \in W$ be such that $\mathit{form}(w) = \mathit{Th}(\{\phi_1, \ldots, \phi_n\})$ (such a $w$ is guaranteed to exist by the assumed property of $(W, \leq)$). Clearly $w \models \mathit{true}(\phi_1^*)$ and ... and $w \models \mathit{true}(\phi_n^*)$. So, by soundness, $w \models \mathit{true}(\phi^*)$. But then $\phi \in \mathit{form}(w)$ so $\phi \in \mathit{Th}(\{\phi_1, \ldots, \phi_n\})$. Thus $\phi_1, \ldots, \phi_n \vdash \phi$ as required. □

Although pleasing, the theorem characterizing fullness is not particularly useful. As we have seen, fullness is easily established proof theoretically. The universal quantification over models prevents Theorem 6.1 from being a viable alternative. On the other hand, Theorem 6.2 *is* intended to be useful. In order to establish faithfulness it is enough to construct a $(\Sigma, \mathcal{A})$-model with the required property.

Unfortunately, none of the earlier proofs of faithfulness are applications of Theorem 6.2 to the letter. There are two possible remedies. One is to modify the models used to prove faithfulness. The other is to modify Theorem 6.2. In fact a simple modification suggests itself from the proof. Call a theory, $T$, *finitely generated* if there exist $\phi_1, \ldots, \phi_n$ such that $T = \mathit{Th}(\{\phi_1, \ldots, \phi_n\})$. Then it is clear that an encoding is faithful if and only if there exists a $(\Sigma, \mathcal{A})$-model, $(W, \leq)$, such that $\mathit{form}(W)$ contains all finitely generated theories. Both the models $(W_m, =)$ and $(W_m, \subseteq)$ used in the proof of faithfulness for minimal implicational logic satisfy this modified condition. So those proofs can be seen as applications of (the modified) Theorem 6.2. However, we stated Theorem 6.2 in its form above to emphasize the duality with Theorem 6.1. Also, it is trivial to modify the definition of $W_m$ (allowing arbitrary sets of formulae) so that Theorem 6.2 applies as stated. A similar situation occurs with the term model we used to prove the faithfulness of the encoding of Peano Arithmetic.

The other proofs of faithfulness (using the structures built from models of the encoded logic) are further from Theorem 6.2. To accommodate these we can adapt the theorem as follows:

**Proposition 6.3** *The following are equivalent:*

1. *The encoding is faithful.*

2. *There exists a family $F \subseteq$ Theories such that every finitely generated theory can be obtained as $\bigcap F'$ for some $F' \subseteq F$ and, for every $T \in F$, there exists a $(\Sigma, \mathcal{A})$-model, $(W, \leq)$, and a world, $w \in W$, such that form$(w) = T$.*

**Proof.** That 1 implies 2 is immediate from Theorem 6.2 (taking $F = \mathit{Theories}$). For the converse, suppose that statement 2 holds and that $\mathit{true}(\phi_1^*), \ldots, \mathit{true}(\phi_n^*) \vdash_{(\Sigma,\mathcal{A})} \mathit{true}(\phi^*)$. Take some $F' \subseteq F$ such that $\bigcap F' = \mathit{Th}(\{\phi_1, \ldots, \phi_n\})$. Now, for each $T \in F'$, there exists a model, $(W, \leq)$, and a world, $w$, such that $\mathit{form}(w) = T$. Then, as in the proof of Theorem 6.2, $\phi \in T$. So $\phi \in \bigcap F'$. Therefore $\phi_1, \ldots, \phi_n \vdash \phi$ as required. □

The proposition is more general than Theorem 6.2, but also less natural.

It is perhaps more interesting to consider again how Theorem 6.2 can be applied as stated. For example, consider the proof of faithfulness for the encoding of the first-order theory $T$ using $\Sigma_T$-structures of the form $(\{\mathcal{M}\}, =)$. These different $\Sigma_T$-structures can be gathered together in one all encompassing $\Sigma_T$-structure as follows. First (to avoid the size problem) we consider only those first-order models of $T$ whose domain is contained in some fixed infinite set (the Löwenheim-Skolem Theorem guarantees that we can do so). Let $I$ be the set of such models. We can define a $\Sigma_T$-structure over the partial order $(\wp(I), \supseteq)$ by taking the interpretations of types at a world $J \subseteq I$ to be the $J$-indexed product of the interpretations in the various $\mathcal{M} \in J$. Predicates, constants and combinators are interpreted pointwise. The coercions, $i^A_{JJ'}$ are the evident projections. We do not go into further details of the construction. However, Theorem 6.2 holds as stated because, for any first-order theory $T' \supseteq T$, we have that $T' = \mathit{form}(\{\mathcal{M} \mid \mathcal{M}$ is a model of $T'\})$.

## 7 Conclusions and related work

In this paper we presented a semantics for a logical framework using the Kripke lambda models of Mitchell and Moggi. Our main use of the semantics was to give easy faithfulness proofs for encodings of logics in the framework. Then we gave the beginnings of a general theory relating properties of encodings to their semantics.

It is worth comparing our use of Kripke lambda models with their use by Mitchell and Moggi in [15]. In both cases the rôle of the partial order is to model intuitionistic entailment. But there are differences in emphasis due to the different logical languages considered. In [15], although it is remarked that the models interpret full intuitionistic predicate logic with quantification over all higher types, only the interpretation of equations is given explicitly. This is because their interest is in obtaining a completeness theorem for the usual equational consequence relation of the typed lambda-calculus once empty types are permitted. In this paper we too are not making full use of the scope of the models. We consider only a fragment of the full intuitionistic logic, and we have no equality predicate in the logic. The absence of equality means that the definition of model could be simplified in various ways. For example, it would be possible to insist that the coercions, $i^A_{ww'}$, are injections. However, such restrictions are unnatural. Furthermore, we prefer to keep the definition of model in its full generality to allow the logic to be extended with equality (and also the other connectives) if desired.

Because of the complexity of Kripke lambda models, the reader might doubt that our proofs of faithfulness are easier than the usual syntactic ones. We believe that the proofs using the models constructed over the full type hierarchy are self-evidently simple. (Also, in the case of our encoding of K, we have argued that a syntactic proof of faithfulness would be quite involved.) However, the term model constructions certainly are complex. Nevertheless, we were able to give the general construction just once (in Section 4). Given this, the applications of the construction in Section 5 were quite straightforward. Irrespective of the question of simplicity there are two other reasons for preferring semantic proofs of faithfulness. One is that the same model can be reused to prove the faithfulness of different encodings of the same logic (as in, e.g. the minimal logic examples involving Pierce's Law and the Deduction Theorem in Section 5). The other is the fact, mentioned in Section 1, that the models of the encodings give a meta-logical interpretation to formulae of the framework.

If one is primarily interested in term models, it is possible to work with a considerably simpler notion of Kripke lambda model. Miller has given a semantics for a logic programming language (very similar to our framework) in terms of Kripke models built directly out of contexts (for the partial order) and terms [13]. Miller's models are simpler for two reasons. First, due to their concrete nature, the models provide the application, coercion and combinator components for free. Second, Miller does not require the models to be extensional. The term models we consider can be reformulated as Miller models without having to go through all the rigmarole of extensional collapse.

Nonetheless, there are reasons for preferring extensional models. For one, they are more "semantic": extensional models give a unique interpretation to terms as (intuitionistic) functions. Also, if one were to extend the framework with an equality predicate, extensionality would be required to validate $\eta$-conversion. Technically, even without equality, we believe that the extensional collapse technique is interesting enough to make the consideration of extensional models worthwhile.

One disappointing fact about our proofs of faithfulness in Section 5 is that we never needed a model that was both genuinely semantic and genuinely intuitionistic. Our "semantic" models were standard classical models of type theory, and our "intuitionistic" models (i.e. ones over non-trivial partial orders) were term models. It would be interesting to find a use for a proper hybrid model (perhaps by giving a truly semantic model to an encoding of a non-classical logic).

As to future work, there are several interesting lines of development. One is to generalize the theory of Section 6 to more interesting notions of logic (e.g. the consequence relations introduced by Gardner [7]). Another is to analyse notions of derived and admissible rule and see whether the semantics of an encoding has anything to say about how well these are represented by the encoding. It would also be interesting to develop a notion of Kripke model for the more intricate dependent type theory of LF.

## Acknowledgements

I gratefully acknowledge the support of my supervisors, Gordon Plotkin and David Pym. I would also like to thank Philippa Gardner for discussions on this work. Philippa Gardner, Fausto Giunchiglia and Andrew Wilson made useful comments on earlier drafts of the paper.

## References

[1] A. Avron. Modal logics in the Edinburgh LF. In *Workshop on General Logic*, number ECS-LFCS-88-52 in LFCS Report Series. LFCS, Department of Computer Science, University of Edinburgh, 1988.

[2] A. Avron. Simple consequence relations. *Information and Computation*, 92:105–139, 1991.

[3] A. Avron, I. Mason F. Honsell, and R. Pollack. Using typed lambda calculus to implement formal systems on a machine. *Journal of Automated Reasoning*, 9:309–354, 1992.

[4] B. Chellas. *Modal Logic*. Cambridge University Press, 1980.

[5] A. Felty and D. Miller. Specifying theorem provers in a higher-order logic-programming language. In *Proceedings of Ninth International Conference on Automated Deduction*, pages 61–80. Springer-Verlag, 1988.

[6] M. C. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel Publishing Co., Dortrecht, 1983.

[7] P. A. Gardner. Equivalences between logics and their representing type theories. Technical Report ECS-LFCS-92-251, LFCS, Deptartment of Computer Science, University of Edinburgh, 1992. Submitted for publication.

[8] P. A. Gardner. A new type theory for representing logics. In A. Voronkov, editor, *Logic Programming and Automated Reasoning*, number 698 in Lecture Notes in Artificial Intelligence. Springer Verlag, 1993.

[9] R. Harper, F. Honsell, and G. D. Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinary*, 40(1):143–184, 1992.

[10] L. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91, 1950.

[11] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and the $\lambda$-Calculus.* London Mathematical Society, Student Texts 1. Cambridge University Press, 1986.

[12] G. Huet. A unification algorithm for typed $\lambda$-calculus. *Theoretical Computer Science*, 1:27 – 57, 1975.

[13] D. Miller. Abstract syntax and logic programming. In A. Voronkov, editor, *Proceedings of the Second Russian Conference on Logic Programming*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 1991.

[14] J. C. Mitchell. Type systems for programming languages. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume II, pages 365 – 458. Elsevier Science Publishers, 1990.

[15] J. C. Mitchell and E. Moggi. Kripke-style models for typed lambda calculus. *Journal of Pure and Applied Logic*, 51:99–124, 1991.

[16] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory.* Oxford University Press, 1990.

[17] L. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5:363–396, 1989.

[18] F. Pfenning and C. Elliot. Higher-order abstract syntax. In *Proccedings of ACM-SIGPLAN Conference on Programming Language Design and Implementation*, 1988.

[19] D. Prawitz. *Natural Deduction - A proof theoretical study.* Almquist and Wiksell, Stockholm, 1965.

[20] M. Ryan and M. Sadler. Valuation systems and consequence relations. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 1. Clarendon Press, Oxford, 1992.

[21] D. S. Scott. On engendering an illusion of understanding. *Journal of Philosophy*, 68:787–807, 1971.

[22] A. Tarski. *Logic, Semantics, Metamathematics*. Oxford University Press, 1956.

[23] D. van Dalen. *Logic and Structure*. Springer-Verlag, 1983. Second edition.