Functorial Kripke-Beth-Joyal models of the $\lambda\Pi$-calculus II: the LF logical framework

David J. Pym* Mark A. Price†
Hewlett Packard Labs University of Bath

January 11, 2007

Abstract

Kripke models.

1 Introduction

The order needs to be more logical. It needs to become the following:

- *Introduction*
- *Judged Systems*
- *Models*
- *Meta-theorems about General Picture*
- *Vigano Intro*
- *Taking Vigano through square*
- *Meta-theorems about Vigano*

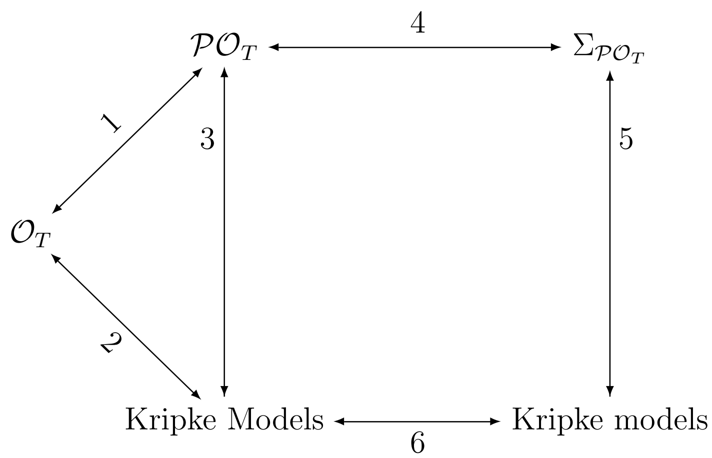
*The research reported herein, and in the two associated papers described in the introduction, was begun whilst the author was associated with The University of Edinburgh, Scotland, U.K.; it was continued whilst the author was associated with The University of Birmingham, England, U.K., and then with Queen Mary & Westfield College, University of London. The author has since acted in a supervisory role while Price has completed the research. The partial support of the UK EPSRC is gratefully acknowledged.

†The research reported herein, and in the two associated papers described in the introduction has been completed by the author as part of his Ph.D. research at the University of Bath under the supervision of Pym. The support of the UK EPSRC for this research is gratefully acknowledged.

This paper, “Functorial Kripke-Beth-Joyal models of the $\lambda\Pi$-calculus II: the LF logical framework” (henceforth abbreviated to II), is second in a sequence of three connected works, beginning with “Functorial Kripke-Beth-Joyal models of the $\lambda\Pi$-calculus: type theory and internal logic” (henceforth abbreviated to III) [PP06] and concluding with “Functorial Kripke-Beth-Joyel models of the $\lambda\Pi$-calculus III: logic programming and its semantics” (henceforth abbreviated to III) [Pym01]. It is concerned with the basic model theory of the LF logical framework. Here we are concerned with logical frameworks in the original sense introduced by Avron, Harper, Mason and Plotkin in [HHP87, HHP93, AHMP92, Pym90] and developed from the point of view of encoding or representation, in [AHMP98].

This paper builds on the work of I, in which we present the model theory of $\lambda\Pi$ and its internal logic. In I, we present a categorical semantics for $\lambda\Pi$ and prove that this semantics together with a suitable notion of satisfaction was sound and complete. We also gave a categorical semantics to the internal logic of $\lambda\Pi$ using the propositions-as-types correspondence an isomorphism of models is induced.

In this paper, we generalise this result and show how a ‘generic’ logic can be represented in $\lambda\Pi$ using the judgements-as-types principle which is a generalization of the propositions-as-types correspondence. The following diagram helps to explain the layout and purpose of this paper:



Here $\mathcal{O}_T$ corresponds to a logic which is defined using the labelled deductive systems of Basin, Matthews and Viganò [BMV96, BMV97, BMV98, Vig00]. Labelled deductive systems are defined in a way that is neither purely syntactic or purely semantic. A formula is labelled with a world and relations between worlds are used to define connectives. The Kripke semantics of a particular logic provide the information to describe the relation on worlds which defines a connective. Labelled deductive systems provide a means to classify a wide range of logics. We are using them as a way of characterizing a class of logics which can then be suitably encoded into a logical framework.

Logics which can be defined in terms of a labelled deductive system are by no means all of the logics which can be written down. However they do capture a lot of logics in a systematic way and this is why they have been used as a means to describe a ‘generic’ logic.

The work in this paper needs to be done in a systematic and non ad-hoc way. One way of avoiding an ad-hoc approach is to use labelled deductive systems and this has been taken here.

From the labelled deductive system, a proof system is extracted. This proof system will either be a Hilbert-type system, natural deduction system or a hybrid of the two.[1] This extraction provides $\mathcal{PO}_T$ in the above diagram and we will study arrows 1 and 2. We are looking for soundness and completeness results for the correspondence 1 and their semantic equivalents for the correspondence 2, adequacy and faithfulness. Understanding correspondence 2 will allow us to see in what sense our Kripke models are related to the traditional Kripke semantics.

So far, we have just been concerned with attempting to classify, systematically, a large range of logics. Now, we are interested in how to encode a judged Hilbert-type system, natural deduction system or a hybrid system, $\mathcal{PO}_T$, into LF. The correspondence 3 involves proving soundness and completeness for a proof system obtained from a labelled deductive system with respect to a Kripke model. The Kripke model presented here is similar to the Kripke models presented in I. We want this to be the case because we wish to consider them as objects in a category of (Kripke) models and wish to study (generalized) isomorphisms (arrows) between them.

The correspondence 4 is the encoding or representation of the object-logic in LF. We are interested what effect the properties of the representation have on the soundness and completeness results for both the proof system and the encoded logic. The representation can either be faithful or adequate. Adequacy being the strongest condition guarantees that everything which can be proved in the proof system can be proved in the encoded system and vice versa. *State completeness induced result here*

5 is the soundness and completeness result for the encoded logic. This is shown with respect to the Kripke model presented in I. We are interested in what effect a completeness result here has on the completeness result for $\mathcal{PO}_T$.

Correspondence 6 is the morphism between the Kripke models for $\mathcal{PO}_T$ and the Kripke models for $\Sigma_{\mathcal{O}_T}$. We showed in I that this morphism is an isomorphism if $\mathcal{PO}_T$ is the $\{\forall, \supset\}$-fragment of minimal first-order logic and 4 was the propositions-as-types correspondence. In general we investigate what type of morphism is induced from different choices of 4.

The third paper in this sequence, "Functorial Kripke-Beth-Joyal models of the $\lambda\Pi$-calculus III: logic programming and its semantics" [Pym01] (henceforth abbreviated to III), provides ...

[1]Indeed, it seems that, [Pfe95] not withstanding, LF does not provide a satisfactory metatheory for presentations of logical systems based on sequent calculi [Gen34].

2 Introduction to the LF logical framework

2.1 Introduction

In this section, we provide an introduction to the idea of logical frameworks, including both their logical and computational motivations.

Logically, the idea of a logical framework can be seen as arising from Martin-Löf's intuitionistic theory of iterated inductive definitions [ML71, ML75], in which form and inductive definitional status in the natural deductive rules is considered. In other words, Martin-Löf considers a formal metatheory of inference rules. This theory is further developed to provide justification for logical rules by extending Kant's notion of a judgement [Kan00] in [ML82].

Computationally, the need for a formal account of the relationship between a logic and its metatheory arises from the desire, in computer science, to manipulate representations of logics and other formal systems. Here we are mainly concerned with logics[2] In order to represent a logic to a machine, the logic must be described in a programming language or metalogic. Moreover, if we are to understand the resulting program, we must have a fixed mechanism for describing logics in the metalogic.

2.2 The notion of a framework

In order to describe a framework, we must [IP98] have methods of:

1. Characterizing the class of (object-)logics to be represented;
2. Describing a metalogic or language, together with its metalogical status *vis-à-vis* the class of object-logics;
3. Characterizing the representation mechanism.

We remark that these components are not entirely independent of each other.

The above prescription can be summarized by the slogan

$$Framework = Language + Representaion$$

In § 2.3, we describe the LF logical framework, for which $\lambda\Pi$ is the language and judgements-as-types the representation mechanism.

2.3 The LF logical framework

In this section, we provide an overview of the LF logical framework. In the sequel, we provide a detailed account of the framework from both proof-theoretic and model-theoretic points of view.

[2]Our conception of logic here is a broad one. For example, in the sequel, we shall consider the linear λ-calculus with equality judgements. Such judgements can be considered to be propositions (*cf.* [MM91]).

One representation mechanism is that of judgements-as-types [HHP93], which originates from Martin-Löf's [ML85] development of Kant's [Kan00] notion of a judgement. The two higher-order judgements, the hypothetical $J \vdash K$ and the general $\Lambda x \in J \, . \, K(x)$, correspond to ordinary and dependent function spaces respectively. The methodology of judgements-as-types is that judgements are represented as the type of their proofs. A logical system $\mathcal{L}$ is represented by a signature which assigns kinds and types to a finite set of constants that represent its syntax, its judgements and its rule schemes. An object-logic's rules and proofs are seen as primitive proofs of hypothetico-general judgements, $\Lambda_{x \in C} \, . \, J(x) \vdash K$. Representation theorems relate consequence in an object-logic $\vdash_{\mathcal{L}}$ to consequence in an encoded logic $\vdash_{\Sigma_{\mathcal{L}}}$.

The judgements-as-types notion of representation, described informally for LF in [HHP93], begins with Kant's formulation of logic [Kan00], as developed by Martin-Löf [ML85]. We contend that it is important to formulate this idea in two steps — identifiable formally for LF in [HHP93] only for particular cases of (classical) first- and higher-order natural deduction — as follows:

1. Consider object-logics as systems for deriving not propositions but rather judged propositions;

2. Consider a correspondence between judged propositions and types in the language of the framework constructed over a signature containing type-constructors corresponding to each judgement form of the object-logic.

With this formulation, LF's representation of object-logics now goes as follows:

An object-consequence, in logic $\mathcal{L}$, is written

$$\delta : (X, \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_\mathsf{m}(\phi_m) \vdash_{\mathcal{L}} \mathsf{j}(\phi)),$$

where j_i and j are judgements, X is the set of variables that occur in the formulae and δ is a proof-object. This object-consequence corresponds, in the language of the framework, to a meta-consequence

$$\Gamma_X, y_1 : \mathsf{j}_1(\phi_1), \ldots, y_m : \mathsf{j}_m(\phi_m) \vdash_{\Sigma_L} M_\delta : \mathsf{j}(\epsilon_X(\phi)),$$

where Γ_X corresponds to the set X of variables, each y_i corresponds to a place-holder for the interpretation of a proof of each $\mathsf{j}_i(\phi_i)$. M_δ is a $\lambda\Pi$-term corresponding to the proof-object δ and ϵ_X is the encoding function for the language.

The propositions-as-types correspondence for the $\{\forall, \supset\}$-fragment of minimal first-order logic is the special case in which each $\mathsf{j}_i (= \mathsf{j}) = \mathsf{proof}$.

We sketch the encoding function as follows: For any $x \in X$, $\epsilon_X(x) = x$ and for any logical connective $\phi \# \psi$, $\epsilon_X(\phi \# \psi) = \# \epsilon_X(\phi) \epsilon_X(\psi)$, where $\#$ on the right hand side is a constant in $\lambda\Pi$.

Roughly speaking, LF is concerned with those Hilbert and natural deduction systems for which the correspondence is uniform [HST94][3] The basic idea is that an encoding $\Sigma_{\mathcal{L}}$ of a

[3]This notion appears to require some adaptation for our formulation.

logic $\mathcal{L}$ is uniform if there is a surjection from consequences

$$\delta : (X, \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_\mathsf{m}(\phi_m) \vdash_{\mathcal{L}} \mathsf{j}(\phi)),$$

in $\mathcal{L}$, to consequences

$$\Gamma_X, y_1 : \mathsf{j}_1(\phi_1), \ldots, y_m : \mathsf{j}_m(\phi_m) \vdash_{\Sigma_{\mathcal{L}}} M_\delta : \mathsf{j}(\epsilon_X(\phi)),$$

in $\Sigma_{\mathcal{L}}$.

One property of this form of representation is that the encoded version of an object-logic inherits the structural properties, such as weakening and/or contraction, of the language of the framework. For example, suppose that $\Sigma_{\mathcal{L}}$ is a uniform encoding of $\mathcal{L}$, and that

$$\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{\mathcal{L}}} M_\delta : \mathsf{j}(\epsilon_X(\phi))$$

is the image of the object-consequence

$$\delta : (X, \Delta \vdash_{\mathcal{L}} \mathsf{j}(\phi)),$$

where δ should be read as the realizer of the consequence.

In $\lambda\Pi$, weakening is admissible, so that if

$$\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{\mathcal{L}}} M_\delta : \mathsf{j}(\epsilon_X(\phi))$$

is provable, then so is

$$\Gamma_X, \Gamma_\Delta, \Gamma_\Theta \vdash_{\Sigma_{\mathcal{L}}} M_\delta : \mathsf{j}(\epsilon_X(\phi))$$

(provided $\Gamma_X, \Gamma_\Delta, \Gamma_\Theta$ is well-formed). By uniformity of $\Sigma_{\mathcal{L}}$,

$$\Gamma_X, \Gamma_\Delta, \Gamma_\Theta \vdash_{\Sigma_{\mathcal{L}}} M_\delta : \mathsf{j}(\epsilon_X(\phi))$$

is then the image of an object-consequence

$$\delta' : (X, \Delta, \Theta \vdash_{\mathcal{L}} \mathsf{j}(\phi)).$$

Consequently, LF is unable to uniformly encode relevant, or substructural [SHD93, Rea88], logics such as intuitionistic linear logic [Gir87]. A framework, also based on judgements-as-types notion of representation, which is able to uniformly encode intuitionistic linear logic has been presented in [IP02].

While there are many examples of representation theorems [AHMP92], the results are 'ad-hoc' and the proof showing the relationship between derivations in $\lambda\Pi$ and derivations in the object-logic is often fiddly and technically difficult. Intuitively, this proof should follow from the truth of the derivation in $\lambda\Pi$ and thus be straightforward. The set-up we are creating will allow this intuition to be exploited and enable the reader to prove representation theorems in a straightforward manner.

3 Labelled Deductive Systems

3.1 Introduction

We wish to study the representation of the largest class of logics we can. It is by no means clear how to write down a generic logic; in fact the question 'what is a logic?' is still an open problem. To this end, we need to make a choice about the class of logics we wish to use. In making this choice a lot of factors need to be taken into consideration:

- Is the characterization of the class of logics uniform?
- Are the class of logics suitable for representation in a logical framework?
- Do we capture the traditional logics, *e.g.* classical, intuitionistic, modal etc?
- How big is the class of logics?

It seems that our class of logics has to start with the traditional logics and provide a suitable generalization of them. We also have to take into account what logics can be represented in a logical framework; this is straightforward, the logical rules and axioms have to give a proof system which is either Hilbert-style, natural deduction or hybrid of both.

We choose to take the labelled deductive systems of Basin, Matthews and Viganò [BMV96, BMV97, BMV98, Vig00]. This class of logics certainly describe the traditional logics as well as providing a uniform characterization of many more. It is also the case that these systems have a give rise to a natural deduction proof system.

The labelled deductive systems mix semantics and syntax. The syntax comes equipped with a Kripke semantics. To be able to represent these systems in a logical framework we need to translate a labelled system into a judged system. This is done in § **??**. In this section, we are considered with defining the systems and looking at some examples of labelled deductive systems.

The key idea behind a lablled deductive system is that connectives can be characterized by their relational properties, given by a Kripke semantics. To access these relational properties, each formula is labelled with a world at which it holds. It is then possible to separate connectives into two classes; local and global.[4] A local connective can only act on formulae which all hold at the same world while a global connective can act on formulae at different but related worlds.

We cannot stress enough that the choice of labelled deductive systems is just to map out a class of logics from the space of all possible logics. There are other characterizations of classes of logics which could have been chosen but it appears that labelled deductive systems provide the most suitable characterization according to the criteria set out above.

[4]Basin, Matthews and Viganò use non-local, but it seems that global is a better term.

3.2 A labelled deductive system

All logical systems are built up inductively from an alphabet to a language, which is then provided with axioms and rules. A labelled deductive system is the same and so we begin by defining an alphabet.

Definition 3.1 (alphabet) *An alphabet is a quintuple $A = (S, V, E, C_L, C_G)$ of sets of symbols as follows:*

- *S is a finite set of symbols with natural number arities;*
- *$V \subset S$ is a distinguished subset of S which contains variables;*
- *E is a finite set of expression symbols;*
- *$C_L \subset E$ is a distinguished subset of E which contains local connectives;*
- *$C_G \subset E$ is a distinguished subset of E which contains global connectives.* □

We remark that the term "connective" in Definition 3.1 should be interpreted broadly; for example, the assignment operator := of Hoare's logic [AHMP92] should be considered as a member of this class of symbols.

We now show how to generate the different syntactic categories of the logic from the alphabet.

Definition 3.2 (syntactic categories) *Let $A = (S, V, E, C_L, C_G)$ be an alphabet. The syntactic categories generated by A are inductively defined as follows:*

- *The nullary symbols are syntactic categories;*
- *Let $c_1, \ldots, c_m$ be syntactic categories and let $s \in S$ be an m-ary symbol, then $sc_1 \ldots c_m$ is a syntactic category.*

The syntactic categories containing variables are those formed solely from elements of V. We will distinguish a finite (possibly empty) set of nullary symbols $\{o_1, \ldots, o_m\}$ as the syntactic category of propositions. □

Following Martin-Löf [ML75], Aczel [Acz78] and Gardner [Gar92], we define the expressions of our logical syntax via a notion of arity.

Definition 3.3 (arities and levels) *An arity a is of the form $(a_1, \ldots, a_m) \xrightarrow{s}$, where, for $0 \leq i \leq m$ is itself an arity and s is a syntactic category. Associated with each such arity is a level, defined as follows:*

$$level(a) = \begin{cases} 0 & \text{if } m = 0 \\ 1 + max_{0 \leq i \leq m}(level(a_i)) & \text{if } m > 0 \end{cases}$$

We refer to $a_1, \ldots, a_m$ as the domain arities of a. □

Every n-ary connective $\# \in C = C_L \cup C_G$ has an arity

$$\underbrace{(\ldots, o_i, \ldots)}_{n-\text{times}} \longrightarrow o_j,$$

where each o_i and o_j is one of the distinguished syntactic classes of propositions.

Each n-ary expression symbol $e \in E \backslash C$ has arity

$$(\ldots, \iota_j, \ldots) \longrightarrow s,$$

where each ι_j is one of the distinguished classes containing variables and s is a syntactic category.

We also single out constants as those expressions which have arity equal to the syntactic category containing variables and level 0. Predicates are those expressions which have arity equal to $(s_1, \ldots, s_m) \longrightarrow o$, where each s_i is a syntactic category containing variables and o is a syntactic category of propositions with level 1. Finally, a function is an expression with arity $(s_1, \ldots, s_m) \longrightarrow s_n$, where each s_i is a syntactic category containing variables with level 1.

We now define the set of expressions generated by an alphabet $A = (S, V, E, C_L, C_G)$. We assume a countable set of variables, V_s, for each syntactic category $s \in V$.

Definition 3.4 (expressions) *Let $A = (S, V, E, C_L, C_G)$ be an alphabet. The set of expressions generated by A is inductively defined as follows:*

variables *If $x \in V_s$, then x is an expression with arity s of level 0;*

applications *If $e \in E$ with arity $(a_1 \ldots, a_m) \longrightarrow s$ of level $l \geq 1$ and if $e_1, \ldots, e_m$ are expressions of arity $a_1 \ldots, a_m$ respectively, then $ee_1 \ldots e_m \in E$ of arity s;*

abstractions *If e is an expression with arity s of level 0 and if $x_1, \ldots, x_m$ are distinct variables with arities $a_1 \ldots, a_m$ of level 0 respectively, then $(x_1, \ldots, x_m)e$ is an expression with arity $(a_1, \ldots, a_m) \longrightarrow s$.*

The logical expressions are those with level 0 arities. The term expressions are those logical expressions which inhabit syntactic categories containing variables and the proposition expressions are those logical expressions which inhabit one of the $o_1, \ldots, o_m$. □

Here application produces new expressions, while abstraction produces expressions which can be applied to other expressions of arity of level > 1.

Substitution has to take place simultaneously over all expressions which have been applied to the initial expression. Given an expression $ee_1 \ldots e_m$, we substitute t for x in the following way: $ee_1 \ldots e_m[t/x] = ee_1[t/x] \ldots e_m[t/x]$.

The previous definitions have provided us with a language for a logical system, however it is yet to be labelled. Intuitively, the logical expressions will correspond with the term expressions of a logic and propositions will corresponds with it's propositions. So we need to label each of the logical expressions and propositions with a world. We introduce W as a countable set of worlds or labels and R as a binary relation over W. The relation is only needed for global connectives. We make this precise in the following definition.

Definition 3.5 (labelled and relational formulae) *Let W be a countable set of worlds (labels) and R a binary relation over W. If $w \in W$ and ϕ is a proposition (or logical expression) then $w : \phi$ is a labelled formula. If w_1 and w_2 are worlds then w_1Rw_2 is a relational formula.* □

The above definition is a modification of Definition 2.1.3 in [Vig00]. We have moved from the term labels to worlds to stress the semantic content implicate in the definition. We now need to define the semantic content used to classify (and define) connectives before providing the natural deduction rules for these systems together with a Kripke semantics.

For # to be a local connective, the truth of the formula $w : \#\phi_1 \ldots \phi_n$ depends on the truth of each $w : \phi_1, \ldots, w : \phi_n$. In other words, a local connective can only be applied to formulae which all hold at the same world.

For # to be a global connective, the truth of the formula $w : \#\phi_1 \ldots \phi_n$ depends on the truth of each $w_1 : \phi_1, \ldots, w_n : \phi_n$ and we must have that $Rww_1 \ldots w_n$ holds.

Since we have an underling Kripke model, we can also give a truth relation with respect to the model $\mathcal{M}$. We have that for a local connective #, we have

$$\models_{\mathcal{M}} w : \#\phi_1 \ldots \phi_n \text{ iff } \models_{\mathcal{M}} w : \phi_1 \text{ and } \ldots \text{ and } \models_{\mathcal{M}} w : \phi_n,$$

while for a global connective #, there are two cases. This depends on the metalevel quantification in the evaluation clause of #, which can be either universal or existential. If # is universal we have

$$\models_{\mathcal{M}} w : \#\phi_1 \ldots \phi_n \text{ iff for all } w_1, \ldots, w_n((\models_{\mathcal{M}} Rww_1 \ldots w_n \text{and } \models_{\mathcal{M}} w_1 : \phi_1 \text{ and } \ldots$$

$$\text{and } \models_{\mathcal{M}} w_{n-1} : \phi_{n-1}) \text{ imply } \models_{\mathcal{M}} w_n : \phi_n)$$

and if # is existential we have

$$\models_{\mathcal{M}} w : \#\phi_1 \ldots \phi_n \text{ iff there exist } w_1, \ldots, w_n(\models_{\mathcal{M}} Rww_1 \ldots w_n \text{ and } \models_{\mathcal{M}} w_1 : \phi_1 \text{ and } \ldots$$

$$\text{and } \models_{\mathcal{M}} w_{n-1} : \phi_{n-1} \text{ and } \models_{\mathcal{M}} w_n : \phi_n.$$

Since global connectives also have a relation on worlds, we also need to define this with respect to the truth relation.

$$\models_{\mathcal{M}} Rww_1 \ldots w_n \text{ iff } (w, w_1, \ldots, w_n) \in \mathcal{R}$$

where $\mathcal{R}$ is an $n+1$-ary relation on $\mathcal{M}$.

Finally, we need to have rules and axioms which allow us to create a proof system. We now define a labelled natural deduction system. We also need to have a consequence relation, we define $\vdash$ as a subset of the powerset of propositions cross the powerset of propositions.

Definition 3.6 (labelled natural deduction systems) *Let $A = (S, V, E, C_L, C_G)$ be an alphabet and let $\vdash$ be a consequence relation over A. A labelled natural deduction system for $\vdash$ is given by the following:*

- *A set of axioms;*
- *For each local connective $\# \in C_L$, an introduction rule schema of the form*

$$\cfrac{\begin{array}{ccc} & [w : \psi_{i,1}] \cdots [w : \psi_{i,h_i}] & \\ \vdots & \vdots & \vdots \\ w : \phi_1 & w : \phi_i & w : \phi_p \end{array}}{w : \#(\phi_1 \ldots \phi_n)} \; \#I$$

- *For each local connective $\# \in C_L$, an elimination rule schema of the form*

$$\cfrac{\begin{array}{cccc} & [\Gamma_1] & & [\Gamma_p] \\ & \vdots & \cdots & \vdots \\ w : \#(\phi_1 \ldots \phi_n) & w' : \chi_1 & & w' : \chi_p \end{array}}{w' : \tau} \; \#e$$

where the p minor premises of the form $w : \chi_i$ are derived from the set of assumptions Γ_i, for $1 \leq i \leq p$.

- *For each universal global connective $\# \in C_G$, an introduction rule schema of the form*

$$\cfrac{\begin{array}{c} [w_1 : \phi_1] \cdots [w_{n-1} : \phi_{n-1}][Rww_1 \ldots w_n] \\ \vdots \\ w_n : \phi_n \end{array}}{w : \#\phi_1 \ldots \phi_n} \; \#I$$

- *For each universal global connective $\# \in C_G$, an elimination rule schema of the form*

$$\frac{w : \#\phi_1 \ldots \phi_n \quad w_1 : \phi_1 \ldots w_{n-1} : \phi_{n-1} \quad Rww_1 \ldots w_n}{w_n : \phi_n} \; \#E$$

- *For each existential global connective $\# \in C_G$, an introduction rule schema of the form*

$$\frac{w_1 : \phi_1 \ldots w_m : \phi_m \quad Rww_1 \ldots w_m}{w : \#\phi_1 \ldots \phi_m} \; \#I$$

- *For each existential global connective, an elimination rule schema of the form*

$$\cfrac{w : \#\phi_1 \ldots \phi_m \qquad \begin{array}{c}[w_1 : \phi_1] \cdots [w_m : \phi_m][Rww_1 \ldots w_m] \\ \vdots \\ w' : \psi\end{array}}{w' : \psi}\ \#E$$

We shall refer to the labelled natural deduction system $\mathcal{L}$ for $\vdash$ over A. □

It is common to require natural deduction systems to enjoy the symmetry between the introduction and elimination rules for each connective and that these rules be, in a suitable sense, inverses of one another [Sun01]. We need to ensure that our formulation does not permit examples of the "*tonk*" form, [Hod01] which introduce inconsistency. Avoiding "*tonk*" means we can only infer a formula from what we necessarily had to know to infer the formula [Hod01]. The problem is that the introduction rules for *tonk* are

$$\frac{\phi}{\phi \textit{ tonk } \psi}\ \textit{tonk I} \quad \text{and} \quad \frac{\phi \textit{ tonk } \psi}{\psi}\ \textit{tonk E},$$

from which it follows that, for all ϕ and ψ, $\phi \vdash \psi$. This pair of rules violates the condition that introduction and elimination rules be inverses of one another.

To avoid "*tonk*", we define the following property which our system must have, which corresponds to normalization. This condition is sufficient to prevent consistency for each of the different classes of connectives.

Definition 3.7 (local reduction property) *Given a proof Π in our natural deduction system which contains an application of the introduction rule $\#I$ followed immediately by an application of the elimination rule $\#E$ (for the same connective) which takes the result of the introduction rule as its major premise, then the introduction and elimination rule can be eliminated, leading to a more direct proof of the conclusion.* □

Intuitively, this corresponds to the philosophical idea give by [Hod01]; avoiding "*tonk*" means that we can only infer a formula from what we necessarily had to know to infer the formula.

We show why a system with "*tonk*" fails. Assume we have a proof containing an application of *tonk I* followed immediately by *tonk E*.

$$\begin{array}{c}\vdots \\ \cfrac{\cfrac{A}{A \textit{ tonk } B}\ \textit{tonk I}}{B}\ \textit{tonk E} \\ \vdots\end{array}$$

If we were to eliminate this step, we would not obtain a more direct proof of the conclusion since the only way we could have proved B from A above would have been to use the "*tonk*" introduction and elimination rules.

So far apart from a relation on worlds, we have not really given much semantic information to the labelled deductive system. In the definitions above, each global connective comes with a relation on worlds. The natural question is to ask how do each of these relations interact. This interaction is given by a Horn relational theory. This is a theory generated by sets of rules of the form

$$\frac{R_i w_1^1 \dots w_n^1 \cdots R_i w_1^m \dots w_n^m}{R_i w_1 \dots w_n}$$

The different rules needed for generating different modal logics come from correspondence theory.

To write down a labelled natural deduction system, a choice of base system is required when is then extended by the rules of the relational Horn theory. To describe any propositional modal logic, the base system $N(\mathcal{B})$ is taken to be the rules required to give the modal logic K. The connectives used in the definition of K are all local except for $\Box$ which is global. The rules are the following:

$$\frac{\begin{array}{c}[w\!:\!\phi \supset \bot] \\ \vdots \\ w'\!:\!\bot\end{array}}{w\!:\!\phi}\ \bot\ E \qquad \frac{\begin{array}{c}[w\!:\!\phi] \\ \vdots \\ w\!:\!\psi\end{array}}{w\!:\!\phi \supset \psi}\ \supset\ I \qquad \frac{w\!:\!\phi \supset \psi \quad w\!:\!\phi}{w\!:\!\psi}\ \supset\ E,$$

$$\frac{\begin{array}{c}[wRw'] \\ \vdots \\ w'\!:\!\phi\end{array}}{w\!:\!\Box\phi}\ \Box\ I \qquad \frac{w\!:\!\Box\phi \quad wRw'}{w'\!:\!\phi}\ \Box\ E.$$

So if we wish to extend this system to $S5$ using an appropriate relational Horn theory $N(\mathcal{T})$, we have to add the following rules:

$$\frac{}{wRw}\ reflexivity \qquad \frac{wRw'}{w'Rw}\ symmetry \qquad \frac{wRw' \quad w'Rw''}{wRw''}\ transitivity$$

3.3 Soundness and completeness of labelled natural deduction systems

A labelled natural deductive system comes with an implicit Kripke model $\mathcal{M}$ and we will show soundness and completeness for the system with respect to this model. We will begin by showing that it is sound and complete for any propositional modal logic. This result is taken straight from [Vig00] and we consider a base system $N(\mathcal{K})$ as defined above together with a relational Horn theory $N(\mathcal{T})$. We only sketch the main details of the results, we do not deal with the case where Skolem functions are involved. To deal with this case, we need to extend Definition 3.8 to include constants which will deal with Skolem functions and we also need to consider the Horn relational theory as corresponding to a collection of restricted convergency axioms. The details for this can be found in [Vig00]. We present the soundness and completeness to demonstrate that the labelled natural deductive systems do characterize a sensible class of logics.

We follow [Vig00] and start by defining a Kripke frame for our propositional modal logic $N(\mathcal{L})$.

Definition 3.8 (Viganò) *A* <u>*Kripke frame*</u> *for* $N(\mathcal{L})$ *is a pair* $(\mathcal{W}, \mathcal{R})$*, where* $\mathcal{W}$ *is a non-empty set of worlds and* $\mathcal{R} \subset \mathcal{W} \times \mathcal{W}$*. A* <u>*Kripke model*</u> *for* $N(\mathcal{L})$ *is a triple* $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \mathcal{B})$*, where* $(\mathcal{W}, \mathcal{R})$ *is a frame for* $N(\mathcal{L})$*, and the valuation* $\mathcal{B}$ *maps an element of* $\mathcal{W}$ *and a propositional variable to a truth value (*0 *or* 1*). We say that a frame* $(\mathcal{W}, \mathcal{R})$ *and a model* $(\mathcal{W}, \mathcal{R}, \mathcal{B})$ *have some property of binary relations (*e.g. *transitivity) iff* $\mathcal{R}$ *has that property.* □

Definition 3.9 (Viganò) *Given a set of labelled formulae* Γ *and a set of relational formulae* Δ*, we call the ordered pair* (Γ, Δ) *a* <u>*proof context*</u>*. When* $\Gamma_1 \subset \Gamma_2$ *and* $\Delta_1 \subset \Delta_2$*, we write* $(\Gamma_1, \Delta_1) \subset (\Gamma_2, \Delta_2)$ *and say that* (Γ_1, Δ_1) *is included in* (Γ_2, Δ_2)*. When* $w : \phi \in \Gamma$*, we write* $w : \phi \in (\Gamma, \Delta)$ *irrespective of* Δ*, and when* $wRw' \in \Delta$*, we write* $wRw' \in (\Gamma, \Delta)$ *irrespective of* Γ*. Finally, we say that a label* w *occurs in* (Γ, Δ)*, in symbols* $w \lhd (\Gamma, \Delta)$*, if there exists a* ϕ *such that* $w : \phi \in \Gamma$ *or there exists a* w' *such that* $wRw' \in \Delta$ *or* $w'Rw \in \Delta$*.* □

Since we have a fixed base system, we are only concerned with showing the truth of the connectives $\supset$ and $\Box$. Thus we have the following definition of truth.

Definition 3.10 (Viganò) <u>*Truth*</u> *for a relational or labelled formula* ϕ *in a model* $\mathcal{M}$*,* $\models_{\mathcal{M}} \phi$*, is the smallest relation satisfying:*

$$
\begin{array}{ll}
\models_{\mathcal{M}} wRw' & \text{iff } (w, w') \in \mathcal{R}; \\
\models_{\mathcal{M}} w : p & \text{iff } \mathcal{B}(w, p) = 1; \\
\models_{\mathcal{M}} w : \phi \supset \psi & \text{iff } \models_{\mathcal{M}} w : \phi \text{ implies } \models_{\mathcal{M}} w : \psi; \\
\models_{\mathcal{M}} w : \Box \phi & \text{iff for all } w', \models_{\mathcal{M}} wRw' \text{ implies } \models_{\mathcal{M}} w' : \phi.
\end{array}
$$

When $\models_{\mathcal{M}} \phi$*, we say that* ϕ *is true in* $\mathcal{M}$*. By extension:*

$$
\begin{array}{ll}
\models_{\mathcal{M}} \Gamma & \text{means that } \models_{\mathcal{M}} w : \phi \text{ for all } w : \phi \in \Gamma; \\
\models_{\mathcal{M}} \Delta & \text{means that } \models_{\mathcal{M}} wRw' \text{ for all } wRw' \in \Delta; \\
\models_{\mathcal{M}} (\Gamma, \Delta) & \text{means that } \models_{\mathcal{M}} \Gamma \text{ and } \models_{\mathcal{M}} \Delta; \\
\Delta \models_{\mathcal{M}} wRw' & \text{means that } \models_{\mathcal{M}} \Delta \text{ implies } \models_{\mathcal{M}} wRw'; \\
\Delta \models wRw' & \text{means that } \Delta \models_{\mathcal{M}} \text{ for all } \mathcal{M}; \\
\Gamma, \Delta \models_{\mathcal{M}} w : \phi & \text{means that } \models_{\mathcal{M}} (\Gamma, \Delta) \text{ implies } \models_{\mathcal{M}} w : \phi; \\
\Gamma, \Delta \models w : \phi & \text{means that } \Gamma, \Delta_{\mathcal{M}} w : \phi \text{ for all } \mathcal{M}'
\end{array}
$$

□

The truth relation above is the same as the usual truth function for modal logics if we remove the world label and consider the relation as being true at a particular world.

We are now in a position to state and prove the soundness result for labelled propositional modal logics.

Lemma 3.11 (soundness (Viganò)) *Let $N(\mathcal{L}) = N(\mathcal{K}) + N(\mathcal{T})$ be a labelled propositional modal logic. Then we have that*

(i) $\Delta \vdash_{N(\mathcal{L})} wRw'$ *implies* $\Delta \models wRw'$*, and*

(ii) $\Gamma, \Delta \vdash_{N(\mathcal{L})} w\!:\!\phi$ *implies* $\Gamma, \Delta \models w\!:\!\phi$.

Proof Let $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \mathcal{N})$ be an arbitrary model for $N(\mathcal{L})$. We prove (i) by induction on the structure of the derivation of wRw' from Δ. The base case, where $wRw' \in \Delta$ is trivial. There is one induction step for each Horn relational rule of $N(\mathcal{T})$, and we just prove transitivity; the other cases are similar.

For transitivity, we assume that $\mathcal{R}$ is transitive and consider an application of the rule *trans*,

$$\dfrac{\begin{array}{cc}\Pi_1 & \Pi_2 \\ wRw' & w'Rw''\end{array}}{wRw''}\, trans\,,$$

where Π_1 and Π_2 are the derivations $\Delta_1 \vdash_{N(\mathcal{L})} wRw'$ and $\Delta \vdash_{N(\mathcal{L})} w'Rw''$, with $\Delta = \Delta_1 \cup \Delta_2$. By the induction hypotheses, $\Delta_1 \vdash_{N(\mathcal{L})} wRw'$ implies $\Delta_1 \models wRw'$, and $\Delta_2 \vdash_{N(\mathcal{L})} w'Rw''$ implies $\Delta_2 \models w'Rw''$. Assume that $\models_{\mathcal{M}} \Delta$. Then, from the induction hypotheses we obtain $\models_{\mathcal{M}} wRw'$ and $\models_{\mathcal{M}} w'Rw''$, *i.e.* $(w, w') \in \mathcal{R}$ and $(w', w'') \in \mathcal{R}$. Since $\mathcal{R}$ is transitive, we conclude $\models_{\mathcal{M}} wRw''$ by Definition 3.10.

We prove (ii) by induction on the structure of the derivation of $w\!:\!\phi$ from Γ and Δ. The base case, where $w\!:\!\phi \in \Gamma$ is trivial. There is one step for each inference rule of $N(\mathcal{K})$, and we prove the rules $\bot$ and $\Box$.

Consider an application of the rule $\bot E$,

$$\dfrac{\begin{array}{c}[w\!:\!\phi \supset \bot] \\ \Pi \\ w'\!:\!\bot\end{array}}{w\!:\!\phi}\, \bot E$$

where Π is the derivation $\Gamma_1, \Delta \vdash_{N(\mathcal{L})} w'\!:\!\bot$, with $\Gamma_1 = \Gamma \cup \{w\!:\!\phi \supset \bot\}$. By the induction hypothesis $\Gamma_1, \Delta \vdash_{N(\mathcal{L})} w' : \bot$ implies $\Gamma_1, \Delta \models w' : \bot$. We assume $\models_{\mathcal{M}} (\Gamma, \Delta)$ and prove $\models_{\mathcal{M}} w : \phi$. Since $\not\models w' : \phi$ for any w', form the induction hypothesis we obtain $\not\models \Gamma_1$, and therefore $\not\models w\!:\!\phi \supset w\!:\!\bot$, *i.e.* $\models w\!:\!\phi$ and $\not\models w\!:\!\bot$ by Definition 3.10.

Consider an application of the rule $\supset I$,

$$\dfrac{\begin{array}{c}[w\!:\!\phi]\\ \Pi\\ w\!:\!\psi\end{array}}{w\!:\!\phi\supset\psi}\supset I$$

where Π is the derivation $\Gamma_1, \Delta \vdash_{N(\mathcal{L})} w : \psi$, with $\Gamma_1 = \Gamma \cup \{w : \phi\}$. By the induction hypothesis, $\Gamma_1, \Delta \vdash_{N(\mathcal{L})} w\!:\!\phi$ implies $\Gamma_1, \Delta \models w\!:\!\phi$. Assume $\models_{\mathcal{M}} (\Gamma, \Delta)$. Since $\not\models_{\mathcal{M}} w\!:\!\psi$ for any w, from the induction hypothesis $\not\models_{\mathcal{M}} \Gamma_1$, and therefore $\not\models_{\mathcal{M}} w\!:\!\phi$, *i.e.* $\models_{\mathcal{M}} w\!:\!\phi\supset\psi$ by Definition 3.10.

Consider an application of the rule $\supset E$,

$$\dfrac{\begin{array}{c}\Pi_1\\ w\!:\!\phi\supset\psi\end{array}\qquad\begin{array}{c}\Pi_2\\ w\!:\!\phi\end{array}}{w\!:\!\psi}\supset E$$

where Π_1 and Π_2 are the derivations $\Gamma_1, \Delta \vdash_{N(\mathcal{L})} w : \phi\supset\psi$ and $\Gamma_2, \Delta \vdash_{N(\mathcal{L})} w : \phi$, with $\Gamma = \Gamma_1 \cup \Gamma_2$. By the induction hypotheses, $\Gamma_1, \Delta \vdash_{N(\mathcal{L})} w\!:\!\phi\supset\psi$ implies $\Gamma_1, \Delta \models w\!:\!\phi\supset\psi$ and $\Gamma_2, \Delta \vdash_{N(\mathcal{L})} w\!:\!\phi$ implies $\Gamma_2, \Delta \models w\!:\!\phi$. We assume $\models_{\mathcal{M}} (\Gamma, \Delta)$. From the induction hypotheses, we have that $\models_{\mathcal{M}} w\!:\!\phi\supset\psi$ and $\models_{\mathcal{M}} w\!:\!\psi$ and thus $\models_{\mathcal{M}} w\!:\!\psi$ by Definition 3.10.

Consider an application of the rule $\Box I$,

$$\dfrac{\begin{array}{c}[wRw']\\ \Pi\\ w'\!:\!\phi\end{array}}{w\!:\!\Box\phi}\Box I$$

where Π is the derivation $\Gamma, \Delta_1 \vdash_{N(\mathcal{L})} w' : \phi$, with $\Delta_1 = \Delta \cup \{wRw'\}$. By the induction hypothesis, $\Gamma, \Delta_1 \vdash_{N(\mathcal{L})} w' : \phi$ implies $\Gamma, \Delta_1 \models w' : \phi$. Assume $\models_{\mathcal{M}} (\Gamma, \Delta)$. Considering the restriction on the application of $\Box I$, we can extend Δ to $\Delta' = \Delta \cup \{wRw''\}$ for an arbitrary $w'' \not\lhd (\Gamma, \Delta)$, and assume $\models_{\mathcal{M}} \Delta'$. Since $\models_{\mathcal{M}} \Delta'$ implies $\models_{\mathcal{M}} \Delta_1$, from the induction hypothesis we obtain $\models_{\mathcal{M}} w' : \phi$, that is $\models_{\mathcal{M}} w'' : \phi$ for an arbitrary $w'' \not\lhd (\Gamma, \delta)$ such that $\models_{\mathcal{M}} wRw''$. We conclude $\models_{\mathcal{M}} w\!:\!\Box\phi$ by Definition 3.10.

Consider an application of the rule $\Box E$,

$$\dfrac{\begin{array}{c}\Pi_1\\ w\!:\!\Box\phi\end{array}\qquad\begin{array}{c}\Pi_2\\ wRw'\end{array}}{w'\!:\!\phi}\Box E$$

where Π_1 and Π_2 are the derivations $\Gamma, \Delta_1 \vdash_{N(\mathcal{L})} w : \Box\phi$ and $\Delta_2 \vdash_{N(\mathcal{L})} wRw'$, with $\Delta = \Delta_1 \cup \Delta_2$. Assume $\models_{\mathcal{M}} (\Gamma, \delta)$. Then, from the induction hypotheses we obtain $\models_{\mathcal{M}} w\!:\!\Box\phi$ and $\models_{\mathcal{M}} wRw'$, and thus $\models_{\mathcal{M}} w'\!:\!\phi$ by Definition 3.10. $\Box$

Before we can move on to completeness for a labelled propositional modal logic a few more definitions and results are needed. The proof of completeness follows a Henkin-style proof an so we need to have a definition of a maximally consistent extension of our proof context. We does this in the usual way.

Definition 3.12 (Viganò) *Let $N(\mathcal{L}) = N(\mathcal{K}) + N(\mathcal{T})$ be consistent system,* i. e. $\nvdash_{N(\mathcal{L})} w : \bot$ *for every w. A proof context (Γ, Δ) is $N(\mathcal{L})$-consistent iff $\Gamma, \Delta \nvdash_{N(\mathcal{L})} w : \bot$ for every w. (Γ, Δ) is $N(\mathcal{L})$-inconsistent iff it is not $N(\mathcal{L})$-consistent.* □

We will abuse by notation by talking about consistent and inconsistent proof contexts in general rather than $N(\mathcal{L})$-consistent and inconsistent proof contexts. To keep the notation down, we introduce negation, which is defined as follows:

$$\begin{aligned} \models_{\mathcal{M}} w : \neg\phi \text{ iff } & \not\models_{\mathcal{M}} w : \phi \\ \text{iff } & \models_{\mathcal{M}} w : \phi \text{ implies } \models_{\mathcal{M}} w : \bot \end{aligned}$$

with inference rules

$$\dfrac{\begin{array}{c}[w : \phi] \\ \vdots \\ w : \bot\end{array}}{w : \neg\phi}\ \neg I \qquad \dfrac{w : \neg\phi \quad w : \phi}{w : \bot}\ \neg E$$

this is a derived rule from $N(\mathcal{K})$ and so we can just add it to the system without worrying about loss of soundness. We can now formulate the following result for labelled propositional modal logics.

Proposition 3.13 (Viganò) *If (Γ, Δ) is consistent, then for every w and every ϕ, either $(\Gamma \cup \{w : \phi\}, \Delta)$ is consistent or $(\Gamma \cup \{\neg\phi\}, \Delta)$ is consistent.*

We now define the deductive closure of the relational Horn theory,

Definition 3.14 (Viganò) *Let $N(\mathcal{L}) = N(\mathcal{K}) + N(\mathcal{T})$ be a labelled propositional modal logic, let $\Delta_{N(\mathcal{L})}$ be the deductive closure of Δ under $N(\mathcal{L})$,* i.e.

$$\Delta_{N(\mathcal{L})} =_{def} \{wRw' | \Delta \vdash_{N(\mathcal{L})} wRw'\}.$$

□

From this definition and Proposition 3.13, we can deduce that $\Gamma, \Delta \vdash_{N(\mathcal{L})} \phi$ iff $\Gamma, \Delta_{N(\mathcal{L})} \vdash_{N(\mathcal{L})} \phi$, and that $\Delta_{N(\mathcal{L})}$ might be empty when Δ is empty and for example, $N(\mathcal{L})$ is $N(\mathcal{K})$.

Definition 3.15 (Viganò) *A proof context (Γ, Δ) is maximally consistent iff* *(i)* *it is consistent,*

(ii) $\Delta = \Delta_{N(\mathcal{L})}$, *and*

(iii) *for every* w *and every* ϕ, *either* $w:\phi \in \Gamma$ *or* $w:\phi \notin \Gamma$. $\square$

We now begin to prove the results that lead up to completeness.

Lemma 3.16 (Viganò) *Every consistent proof context* (Γ, Δ) *can be extended to a maximally consistent proof context* (Γ^*, Δ^*).

The usual proof of this result for unlabelled modal logics uses witnesses, here we use witness words which is the appropriate concept for labelled systems.

Proof We extend the language of $N(\mathcal{L})$ with countably many new constants for witness worlds. Let w range over a worlds, v range over the witness worlds and u range over both. We enumerate the formulae by $l_1, l_2, \ldots$ and when l_i is $u:\phi$, we write $\neg l_i$ for $u:\neg\phi$. Starting from $(\Gamma_0, \Delta_0) = (\Gamma, \Delta)$, we inductively build a sequence of consistent proof contexts by defining $(\Gamma_{i+1}, \Delta_{i+1})$ to be:

- (Γ_i, Δ_i), if $(\Gamma_i \cup \{l_{i+1}\}, \Delta)$ is inconsistent; else
- $(\Gamma_i \cup \{l_{i+1}\}, \Delta_i)$, if l_{i+1} is not $u:\neg\phi$; else
- $(\Gamma_i \cup \{u:\neg\phi, v:\phi\}, \Delta_i \cup \{uRv\})$ for a $v \lhd (\Gamma_i \cup \{u:\neg\Box\phi\}, \Delta_i)$, if l_{i+1} is $u:\neg\Box\phi$.

Every (Γ_i, Δ_i) is consistent. To show this we show that if $(\Gamma_i \cup \{u:\neg\Box\phi\}, \Delta_i)$ is consistent, then so is $(\Gamma_i \cup \{u:\neg\phi, v:\neg\phi\}, \Delta_i \cup \{uRv\}, \Delta_i)$; the other cases follow by construction. We proceed by contraposition. Suppose that

$$\Gamma \cup \{u:\neg\Box\phi, v:\neg\phi\}, \Delta_i \cup \{uRv\} \vdash_{N(\mathcal{L})} u_j:\bot$$

where $v \not\lhd (\Gamma_i \cup \{u:\neg\Box\phi\}, \Delta_i)$. Then by $\bot E$,

$$\Gamma_i \cup \{u:\neg\Box\phi\}\Delta_i \cup \{uRv\} \vdash_{N(\mathcal{L})} v:\phi,$$

and $\Box I$ yields

$$\Gamma_i \cup \{u:\neg\phi\}, \Delta_i \vdash_{N(\mathcal{L})} u:\Box\phi.$$

Since also

$$\Gamma_i \cup \{u:\neg\Box\phi\}, \Delta_i \vdash_{N(\mathcal{L})} u:\bot,$$

by $\neg E$ we have

$$\Gamma_i \cup \{u:\neg\Box\phi\}, \Delta_i \vdash_{n(\mathcal{L})} u:\bot,$$

i.e. $(\Gamma_i \cup \{u:\neg\phi\}, \Delta_i\}, \Delta_i)$ is inconsistent. Contradiction.

Now define

$$\Gamma^* = \bigcup_{i \geq 0} \Gamma_i \text{ and } \Delta^* = \bigcup_{i \geq 0} (\Delta_i)_{N(\mathcal{L})}.$$

We show that (Γ^*, Δ^*) is maximally consistent by proving that it satisfies the conditions in Definition 3.15. For (i), note that

$$\text{if } (\bigcup_{i\geq 0}\Gamma_i, \bigcup_{i\geq 0}\Delta_i) \text{ is consistent, then so is } (\bigcup_{i\geq 0}\Gamma_i, \bigcup_{i\geq 0}(\Delta_i)_{N(\mathcal{L})}).$$

Now suppose that (Γ^*, Δ^*) is inconsistent. Then for some finite (Γ', Δ') included in (Γ^*, Δ^*) there exists a u such that $\Gamma', \Delta' \vdash_{N(\mathcal{L})} u : \bot$. Every labelled formulae $l \in (\Gamma', \Delta')$ is in some (Γ_j, Δ_j). For each $l \in (\Gamma', \Delta')$, let i_l be the least j such that $l \in (\Gamma_j, \Delta_j)$ and $i = max\{i_l | l \in (\Gamma', \Delta')\}$. Then $(\Gamma', \Delta') \subset (\Gamma_i, \Delta_i)$, and (Γ_i, Δ_i) is inconsistent, which is not the case. Condition (ii) is satisfied by the definition of Δ^*. For (iii), suppose that $l_{i+1} \notin (\Gamma^*, \Delta^*)$. Then $l_{i+1} \notin (\Gamma_{i+1}, \Delta_{i+1})$ and $(\Gamma_i \cup \{l_{i+1}\}, \Delta_i)$ is inconsistent. Thus, by Proposition 3.13, $(\Gamma_i \cup \{\neg l_{i+1}, \Delta_i)$ is consistent, and $\neg l_{i+1}$ is consistently added to some (Γ_j, Δ_j) during the construction, and therefore $\neg l_{i+1} \in (\Gamma^*, \Delta^*)$. □

We now state and prove some properties of maximally consistent proof contexts.

Proposition 3.17 (Viganò) *Let* (Γ^*, Δ^*) *be a maximally consistent proof context. Then*

(i) $\Delta^* \vdash_{N(\mathcal{L})} u_i R u_j$ *iff* $u_i R u_j \in \Delta^*$.

(ii) $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} u : \phi$ *iff* $u : \phi \in \Gamma^*$.

(iii) $u : \phi \supset \psi \in \Gamma^*$ *iff* $u : \phi \in \Gamma^*$ *implies* $u : \psi \in \Gamma^*$.

(iv) $u_i : \Box\phi \in \Gamma^*$ *iff* $u_i R u_j \in \Delta^*$ *implies* $u_j : \phi \in \Gamma^*$ *for all* u_j.

Proof (i) and (ii) follow immediately by definition and Proposition 3.13. We only prove (iv); (v) follows analogously. From left-to-right direction, suppose that $u_i : \Box\phi \in \Gamma^*$. Then, by (iii), $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} u_i : \Box\phi$, and, by $\Box E$, we have $\Delta^* \vdash_{N(\mathcal{L})} u_i R u_j$ implies $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} u_j : \phi$ for all u_j. By (i) and (ii), conclude that $u_i R u_j \in \Delta^*$ implies $u_j : \phi \in \Gamma^*$ for all u_j. For the converse, suppose that $u_i : \Box\phi \notin \Gamma^*$. Then $u_i : \Box\phi \in \Gamma^*$, and, by the construction of (Γ^*, Δ^*), there exists a u_j such that $u_i R u_j \in \Delta^*$ and $u_j : \phi \in \Gamma^*$. □

We are now in a position to define the canonical model.

Definition 3.18 (Viagnò) *Given a maximally consistent proof context* (Γ^*, Δ^*), *we define the* <u>*canonical model*</u> $\mathcal{M}^C = (\mathcal{W}^C, \mathcal{R}^C, \mathcal{B}^C)$ *for the proof system* $N(\mathcal{L})$ *as follows:*

- $\mathcal{W}^C = \{u | u \lhd (\Gamma^*, \Delta^*)\}$;
- $(u_i, u_j) \in \mathcal{R}^C$ *iff* $u_i R u_j \in \Delta^*$;
- $\mathcal{B}^C(u, p) = 1$ *iff* $u : p \in \Gamma^*$. □

We now have two more propositions and then we get completeness.

Proposition 3.19 (Viganò) $u_i R u_j \in \Delta^*$ *iff* $\Delta^* \models_{\mathcal{M}^C} u_i R u_j$.

Proposition 3.20 (Viganó) $u\!:\!\phi \in (\Gamma^*, \Delta^*)$ *iff* $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} u\!:\!\phi$.

Proof We proceed by induction on number of times $\supset$ and $\Box$ occur in $u : \phi$. We begin by proving the base case where $u : \phi$ is $u_i : \Box\psi$, with the other base cases being analogous. For the left-to-right direction, assume $u_i\!:\!\Box\psi \in \Gamma^*$. Then, by Proposition 3.17, $u_i R u_j \in \Delta^*$ implies $u_j \psi \in \Gamma^*$, for all u_j. Proposition 3.19 and the induction hypothesis yield $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} u_j : \psi$ for all u_j such that $\Delta^* \models_{\mathcal{M}^C} u_i R u_j$, *i.e.* $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} u_i : \Box\psi$ by Definition 3.10. For the converse assume $u_i : \neg\Box\psi \in \Gamma^*$. Then, by Proposition 3.17, $u_i R u_j \in \Delta^*$ and $u_j\!:\!\neg\psi \in \Gamma^*$, for some u_j. Proposition 3.19 and the induction hypothesis yield $\Delta^* \models_{\mathcal{M}^C} u_i R u_j$ and $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} u_j\!:\!\neg\psi$, *i.e.* $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} u_i\!:\!\neg\Box\psi$ by Definition 3.10. □

Lemma 3.21 (completeness (Viganò)) *Let* $N(\mathcal{L}) = N(\mathcal{K}) + N(\mathcal{T})$ *be a labelled propositional modal logic. Then the following holds:*

(i) $\Delta \models wRw'$ *implies* $\Delta \vdash_{N(\mathcal{L})} wRw'$*, and*

(ii) $\Gamma, \Delta \models w\!:\!\phi$ *implies* $\Gamma, \Delta \vdash_{N(\mathcal{L})} w\!:\!\phi$*.*

Proof *(i)* If $\Delta \nvdash_{N(\mathcal{L})} w_i R w_j$, then $w_i R w_j \notin \Delta^*$, and thus $\Delta^* \not\models_{\mathcal{M}^C} w_i R w_j$ by Proposition 3.19. *(ii)* If $\Gamma \cup \{w : \neg\phi\}, \Delta \vdash_{N(\mathcal{L})} w_i : \bot$, and then $\Gamma, \Delta \vdash_{N(\mathcal{L})} w : \phi$. Therefore, by Proposition 3.16, $(\Gamma \cup \{w\!:\!\neg\phi\}, \Delta)$ is included in a maximally consistent proof context $((\Gamma \cup \{w : \neg\phi)^*, \Delta^*)$. Then, by Proposition 3.20, $\Gamma \cup \{w : \neg\phi\})^*, \Delta^* \models_{N(\mathcal{L})} w : \neg\phi$, *i.e.* $(\Gamma \cup \{w\!:\!\neg\phi\})^*, \Delta^* \not\models_{\mathcal{M}^C} w\!:\!\phi$, and thus $\Gamma, \Delta \not\models_{\mathcal{M}^C} w\!:\!\phi$. □

Theorem 3.22 (soundness and completeness) *Let* $N(\mathcal{L} = N(\mathcal{K}) + N(\mathcal{T})$ *be a labelled propositional modal logic. Then the following holds:*

(i) $\Delta \models wRw'$ *iff* $\Delta \vdash_{N(\mathcal{L})} wrw'$*, and*

(ii) $\Gamma, \Delta \models w\!:\!\phi$ *iff* $\Gamma, \Delta \vdash_{N(\mathcal{L})} w\!:\!\phi$*.*

Thus we have soundness and completeness for a labelled propositional modal logic with respect to it's underlying Kripke model. We now show that a labelled propositional non-classical logic is sound and complete with respect to it's underlying Kripke model. We have dealt with soundness and completeness for two different classes of logics because the second case is more general. Here we have arbitrary connectives with relations and the logics do not correspond as cleanly as the propositional modal logics do to the underlying Kripke model. A propositional non-classical logic $N(\mathcal{L})$ comprises of a base $N(\mathcal{B})$ and a Horn relational theory $N(\mathcal{T})$.

Definition 3.23 (Viganò) *A* <u>*(Kripke) frame*</u> *for* $N(\mathcal{L})$ *is a tuple* $(\mathcal{W}, \wr, \overline{\mathcal{R}}, *)$*, where* $\mathcal{W}$ *is a non-empty set of worlds,* $\wr \in \mathcal{W}$ *is the actual world.* $\overline{\mathcal{R}} = \{\mathcal{R}_i | i \in I\}$ *is the set of relations over* $\mathcal{W}$ *corresponding to* $\overline{\mathcal{R}}$*, and* $*$ *is a function of type* $\mathcal{W} \supset \mathcal{W}$*. A* <u>*(Kripke) model*</u>

$\mathcal{M} = (\mathcal{W}, \wr, \overline{\mathcal{R}}, *, \mathcal{B})$ for $N(\mathcal{L})$ consists of a frame and a function $\mathcal{B}$ mapping elements of $\mathcal{W}$ and propositional variables to truth values (0 or 1), where

$$\models_{\mathcal{M}} w\!:\!p \textit{ iff } \mathcal{B}(w,p) = 1.$$

$\models_{\mathcal{M}}$ is extended to labelled formulae with local and global connectives and to relational formulae as above, and when $\models_{\mathcal{M}} \phi$, for ϕ a labelled formula or a relational formula, we say that ϕ is true in $\mathcal{M}$. By extension:

$\models_{\mathcal{M}} \Gamma$	*means that* $\models_{\mathcal{M}} w\!:\!\phi$ *for all* $w\!:\!\phi \in \Gamma$;
$\models_{\mathcal{M}} \Delta$	*means that* $\models_{\mathcal{M}} R_i w w_1 \ldots w_n$ *for all* $R_i w w_1 \ldots w_n \in \Delta$;
$\models_{\mathcal{M}} (\Gamma, \Delta)$	*means that* $\models_{\mathcal{M}} \Gamma$ *and* $\models_{\mathcal{M}} \Delta$;
$\Delta \models_{\mathcal{M}} R_i w w_1 \ldots w_n$	*means that* $\models_{\mathcal{M}} \Delta$ *implies* $\models_{\mathcal{M}} R_i w w_1 \ldots w_n$;
$\Delta \models R_i w w_1 \ldots w_n$	*means that* $\Delta \models_{\mathcal{M}} R_i w w_1 \ldots w_n$ *for all* $\mathcal{M}$;
$\Gamma, \Delta \models_{\mathcal{M}} w\!:\!\phi$	*means that* $\models_{\mathcal{M}} (\Gamma, \Delta)$ *implies* $\models_{\mathcal{M}} w\!:\!\phi$;
$\Gamma, \Delta \models_{\mathcal{M}} w\!:\!\phi$	*means that* $\Gamma, \Delta \models_{\mathcal{M}} w\!:\!\phi$ *for all* $\mathcal{M}$.

□

Some logics require truth to be monotonic, *e.g.* intuitionistic logic. To capture this, we define a partial order on worlds $\sqsubseteq$, where for intuitionistic logic $\sqsubseteq$ coincides with the accessibility relation. We define the atomic monotony condition, which we require $\mathcal{B}$ to satisfy. For any w_i and w_j and for any propositional variable p,

$$\text{if } \models_{\mathcal{M}} w_i\!:\!p \text{ and } \models_{\mathcal{M}} w_i \sqsubseteq w_j, \text{ then } \models_{\mathcal{M}} w_j\!:\!p. \tag{1}$$

We can generalize this condition, by induction, to any arbitrary formulae. However, this does not work when we mix classical and intuitionistic implication, for example. This problem is solved by introducing the notion of a persistent formula. An example of an intuitionistic/classical hybrid logic can be found in [nDCH96].

Definition 3.24 (Viganò) *A formula ϕ of an intuitionistic/classical hybrid logic is persistent iff*

(i) it is atomic, or

(ii) it is of the form $B \longrightarrow C$ or $\neg B$, where $\neg$ and $\longrightarrow$ are intuitionistic, or

(iii) it is of the form $B \wedge C$ or $B \vee C$, and B and C are both persistent. □

We now introduce the generalized version of (1), which is proved by induction.

Proposition 3.25 (Viganò) *For any w_i and w_j, and for any persistent formula ϕ,*

$$\textit{if } \models_{\mathcal{M}} w_i\!:\!\phi \textit{ and } \models_{\mathcal{M}} w_i \sqsubseteq w_j, \textit{ then } \models_{\mathcal{M}} w_j\!:\!\phi.$$

□

We also need a similar condition for relational formulae. In this case we require that for any $n+1$-ary relation R_i that for all $j < n$,

$$\text{if } \models_{\mathcal{M}} R_i w_0 \dots w_{j-1} w_j w_{j+1} \dots w_n \text{ and } \models_{\mathcal{M}} w \sqsubseteq w_j,$$

$$\text{then } \models_{\mathcal{M}} R_i w_0 \dots w_{j-1} w w_{j+1} \dots w_n$$

and

$$\text{if } \models_{\mathcal{M}} R_i w_0 \dots w_{n-1} w_n \text{ and } \models_{\mathcal{M}} w_n \sqsubseteq w, \text{ then } \models_{\mathcal{M}} R_i w_0 \dots w_{n-1} w.$$

We can now prove soundness:

Lemma 3.26 (soundness (Viganò)) *Let $N(\mathcal{L}) = N(\mathcal{B}) + N(\mathcal{T})$ be a propositional non-classical logic. Then the following holds:*

(i) $\Delta \vdash_{N(\mathcal{L})} R_i w w_1 \dots w_n$ *implies* $\Delta \models R_i w w_1 \dots w_n$, *and*

(ii) $\Gamma, \Delta \vdash_{N(\mathcal{L})} w : \phi$ *implies* $\Gamma, \Delta \models w : \phi$.

Proof Let $\mathcal{M} = (\mathcal{W}, \wr, \mathcal{R}^u, \mathcal{R}^e, *, \mathcal{B})$ be an arbitrary model for $N(\mathcal{L})$. We prove (i) by induction on the structure of the derivation of the relational formula $R_i w w_1 \dots w_n$ from Δ. The base case $R_i w w_1 \dots w_n \in \Delta$ is trivial and there is one step case for each Horn relational rule of $N(\mathcal{T})$. We prove the case where the rule involves applications of the following rules for a ternary relation R^u,

$$\dfrac{\begin{array}{cc}\Pi_1 & \Pi_2 \\ R^u abx & R^u xcd\end{array}}{R^u bcf(a,b,c,d,x)}\, assoc\ 1 \qquad \textit{and} \qquad \dfrac{\begin{array}{cc}\Pi_1 & \Pi_2 \\ R^u abx & R^u xcd\end{array}}{R^U af(a,b,c,d,x)d}\, assoc\ 2$$

where Π_1 is the derivation $\Delta_1 \vdash_{N(\mathcal{L})} R^u abx$, and Π_2 is the derivation $\Delta_2 \vdash_{N(\mathcal{L})} R^u xcd$, with $\Delta = \Delta_1 \cup \Delta_2$. Assume that $\mathcal{R}^u$ is associative and that $\models_{\mathcal{M}} \Delta$. Then from the induction hypotheses we obtain $\models_{\mathcal{M}} R^u abx$ and $\models_{\mathcal{M}} R^u xcd$, and we conclude $\models_{\mathcal{M}} R^u bcf(a,b,c,d,x)$ and $\models_{\mathcal{M}} R^u af(a,b,c,d,x)d$.

We prove (ii) by induction on the structure of the derivation $w : \phi$ from Γ and Δ. The base case $w : \phi \in \Gamma$, is trivial, and there is one induction step for each inference rule of $N(\mathcal{B})$. We deal with applications of $\mathcal{C}^u I$, $\mathcal{C}^u E$, $\mathcal{C}^e I$, and $\mathcal{C}^e E$, where $\mathcal{C}^e$ is an existential global connective and $\mathcal{C}^u$ is a universal global connective.

Consider an application of the rule $\mathcal{M}^u I$,

$$\dfrac{\begin{array}{c}[w_1 : \phi_1] \cdots [w_{u-1} : \phi_{u-1}][R^u w w_1 \dots w_u] \\ \Pi_1 \\ w : \mathcal{M}^u \phi_1 \dots \phi_i\end{array}}{}\, \mathcal{C}^u I$$

where Π_1 is the derivation $\Gamma_1, \Delta \vdash_{N(\mathcal{L})} w_u : \phi_u$, with $\Gamma_1 = \Gamma \cup \{w_1 : \phi_1, \dots, w_{u-1} : \phi_{u-1}\}$ and $\Delta_1 = \Delta \cup \{R^u w w_1 \dots w_n\}$. The induction hypothesis is $\Gamma_1, \Delta_1 \vdash_{N(\mathcal{L})} w_u : \phi_u$ implies

$\Gamma_1, \Delta_1 \models w_u : \phi_u$. Assume $\models_{\mathcal{M}} (\Gamma, \Delta)$. Considering the restriction on the application of $\mathcal{M}^u I$, we can extend Γ and Δ to $\Gamma' = \Gamma \cup \{w'_1 : \phi_1, \ldots, w'_{u-1} : \phi_{u-1}\}$ and $\Delta' = \Delta \cup \{R^u w w'_1 \ldots w'_u\}$ for arbitrary $w'_1, \ldots, w'_u \not\lhd (\Gamma, \Delta)$, and assume $\models_{\mathcal{M}} \Gamma'$ and $\models_{\mathcal{M}} \Delta'$. Since $\models_{\mathcal{M}} \Gamma'$ implies $\models_{\mathcal{M}} \Gamma_1$ and $\models_{\mathcal{M}} \Delta'$ implies $\models_{\mathcal{M}} \Delta_1$, from the induction hypothesis we obtain $\models_{\mathcal{M}} w'_u : \phi_u$ for arbitrary $w'_1, \ldots, w'_u \not\lhd (\Gamma, \Delta)$ such that $\models_{\mathcal{M}} R^u w w'_1 \ldots w'_u$ and $\models_{\mathcal{M}} w'_1 : \phi_1, \ldots, \models_{\mathcal{M}} w'_{u-1} : \phi_{u-1}$. We conclude $\models_{\mathcal{M}} w : \mathcal{C}^u \phi_1 \ldots \phi_u$ from the definition of $\models_{\mathcal{M}}$.

Consider a application of the rule $\mathcal{C}^u E$,

$$\dfrac{\begin{matrix}\Pi_0 \\ w : \mathcal{C}^u \phi_1 \ldots \phi_u\end{matrix} \qquad \begin{matrix}\Pi_1 \\ w_1 : \phi_1\end{matrix} \quad \cdots \quad \begin{matrix}\Pi_{u-1} \\ w_{u-1} : \phi_{u-1}\end{matrix} \qquad \begin{matrix}\Pi_u \\ R^u w w_1 \ldots w_u\end{matrix}}{w_u : \phi_u} \, \mathcal{C}^u E$$

where Π_0 is the derivation $\Gamma_0, \Delta_0 \vdash_{N(\mathcal{L})} w : \mathcal{C}^u \phi_1 \ldots \phi_u$; Π_i is the derivation $\Gamma_i, \Delta_i \vdash_{N(\mathcal{L})} w_i : \phi_i$; Π_u is the derivation $\Delta_u \vdash_{N(\mathcal{L})} R^u w w_1 \ldots w_u$; $\Gamma = \bigcup_{0 \leq i \leq u} \Gamma_i$ and $\Delta = \bigcup_{0 \leq i \leq n} \Delta_i$. Assume $\models_{\mathcal{M}} (\Gamma, \Delta)$. Then, from the induction hypotheses we obtain $\models_{\mathcal{M}} w : \mathcal{C}^u \phi_1 \ldots \phi_u, \models_{\mathcal{M}} w_1 : \phi_1, \ldots, \models_{\mathcal{M}} w_{u-1} : \phi_{u-1}$, and $\models_{\mathcal{M}} R^u w w_1 \ldots w_u$, and thus $\models_{\mathcal{M}} w_u : \phi_u$ from the definition of $\models_{\mathcal{M}}$.

Consider an application of the rule $\mathcal{C}^e I$,

$$\dfrac{\begin{matrix}\Pi_1 \\ w_1 : \phi_1\end{matrix} \quad \cdots \quad \begin{matrix}\Pi_e \\ w_e : \phi_e\end{matrix} \quad \begin{matrix}\Pi_{e+1} \\ R^e w w_1 \ldots w_e\end{matrix}}{w : \mathcal{C}^e \phi_1 \ldots \phi_e} \, \mathcal{C}^e I$$

where Π_i is the derivation $\Gamma_i, \Delta_i \vdash_{N(\mathcal{L})} w_i : \phi_i$, for $1 \leq i \leq e$; Π_{e+1} is the derivation $\Delta_{e+1} \vdash_{N(\mathcal{L})} R^e w w_1 \ldots w_e$; $\Gamma = \bigcup_{1 \leq i \leq e} \Gamma_i$ and $\Delta = \bigcup_{1 \leq i \leq e+1} \Delta_i$. Assume $\models_{\mathcal{M}} (\Gamma, \Delta)$. Then, from the induction hypotheses we obtain $\models_{\mathcal{M}} w_1 : \phi_1, \ldots, \models_{\mathcal{M}} w_e : \phi_e$ and $\models_{\mathcal{M}} R^e w w_1 \ldots w_e$, and thus $\models_{\mathcal{M}} w : \mathcal{C}^e \phi_1 \ldots \phi_e$ from the definition of $\models_{\mathcal{M}}$.

For $\mathcal{C}^e E$, let Π be the derivation

$$\dfrac{\begin{matrix}\Pi_1 \\ w : \mathcal{C}^e \phi_1 \ldots \phi_e\end{matrix} \qquad \begin{matrix}[w_1 : \phi_1] \cdots [w_e : \phi_e][R^e w w_1 \ldots w_e] \\ \Pi_2 \\ y : \psi\end{matrix}}{y : \psi} \, \mathcal{C}^e E$$

That is, Π is $\Gamma, \Delta \vdash_{N(\mathcal{L})} y : \psi$, where, by the restriction on $\mathcal{C}^e E$, the labels $w_1, \ldots, w_e$ do not occur in (Γ, Δ) and are different from w and y. Moreover, Π_1 is the derivation $\Gamma, \Delta \vdash_{N(\mathcal{L})} w : \mathcal{C}^e \phi_1 \ldots \phi_e$, and Π_2 is the derivation $\Gamma \cup \{w_1 : \phi_1, \ldots, w_e : \phi_e\}, \Delta \cup \{R^e w w_1 \ldots w_e\} \vdash_{N(\mathcal{L})} w' : \psi$. By the induction hypothesis for Π_1, we have that $\Gamma, \Delta \models_{\mathcal{M}} w : \mathcal{C}^e \phi_1 \ldots \phi_e$, and thus, from the definition of $\models_{\mathcal{M}}$, there exist $y_1, \ldots, y_e$ such that $\models_{\mathcal{M}} y_1 : \phi_1, \ldots, \models_{\mathcal{M}} y_e : \phi_e$ and $\models_{\mathcal{M}} R^e w y_1 \ldots y_e$. We can extend Γ and Δ to $\Gamma' = \Gamma \cup \{w'_1 : \phi_1, \ldots, w'_e : \phi_e\}$ and $\Delta' = \Delta \cup \{R^e w w'_1 \ldots w'_e\}$ for arbitrary $w'_1, \ldots, w'_e \not\lhd (\Gamma, \Delta)$, and from the induction hypothesis for Π_2 we conclude $\Gamma, \Delta \models_{\mathcal{M}} y : \psi$. $\square$

We now prove completeness by a Henkin-style proof. To obtain the necessary counter-model, we extend the canonical model by 'theory – counter-theory' pairs. This method of extension is general enough for us to prove the necessary completeness results for the class of logics we are dealing with.

Definition 3.27 *For any system $N(\mathcal{L}) = N(\mathcal{B}) + N(\mathcal{T})$, let $\Delta_{N(\mathcal{L})}$ be the* <u>*deductive closure*</u> *of Δ under $N(\mathcal{L})$,* i.e.

$$\Delta_{N(\mathcal{L})} = \{R_i w w_1 \dots w_n | \Delta \vdash_{N(\mathcal{L})} R_i w w_1 \dots w_n\}.$$

□

We proceed by defining a maximal proof context.

Definition 3.28 *A proof context (Γ, Δ) is* <u>*maximal*</u> *with respect to $a:\phi$ iff*

(i) $\Delta = \Delta_{N(\mathcal{L})}$, *and*

(ii) $y:\psi \notin (\Gamma, \Delta)$ *iff* $\Gamma \cup \{y:\psi\}, \Delta \vdash_{N(\mathcal{L})} a:\phi$. □

We now need to extend a proof context which provides the countrapositives to completeness can be extended to a maximal proof context.

Lemma 3.29 (Viganò) *If $\Gamma, \Delta \nvdash_{N(\mathcal{L})} a:\phi$, then (Γ, Δ) can be extended to a proof context (Γ^*, Δ^*) that is maximal with respect to $a:\phi$.*

Proof We extend the language of $N(\mathcal{L})$ to include countably many constants for witness worlds. We let t range over the constants for witness worlds and w range over worlds and the new constants. Let $l_1, l_2, \dots$ be an enumeration of all labelled formulae in the extended language. We build an sequence of proof contexts from $(\Gamma_0, \Delta_0) = (\Gamma, \Delta)$ by defining $(\Gamma_{i+1}, \Gamma_{i+1})$ as follows:

- if $\Gamma_i \cup \{l_{i+1}\}, \Delta_i \vdash_{N(\mathcal{L})} a:\phi$, then $(\Gamma_{i+1}, \Delta_{i+1}) = (\Gamma, \Delta)$
- if $\Gamma_i \cup \{l_{i+1}\}, \Delta_i \nvdash_{N(\mathcal{L})} a:\phi$, then
  - if l_{i+1} is $w : \mathcal{C}^e \phi_1 \dots \phi_e$, then we add witnesses to its truth, *i.e.* for $t_1, \dots, t_e$ / $\lhd(\Gamma_i \cup \{w:\mathcal{C}^e \phi_1 \dots \phi_e\}, \Delta_i)$,

$$\begin{aligned} \Gamma_{i+1} &= \Gamma_i \cup \{w:\mathcal{C}^e \phi_1 \dots \phi_e, t:\phi_1, \dots, w_e:\phi_e\} \\ \Delta_{i+1} &= \Delta_i \cup \{R^e w t_1 \dots t_e\} \end{aligned}$$

  - if l_{i+1} is not $w:\mathcal{C}^e \phi_1 \dots \phi_e$, then $(\Gamma_{i+1}, \Delta_{i+1}) = (\Gamma_{i+1} \cup \{l_{i+1}, \Delta_i)$

Every (Γ_i, Δ_i) is such that $\Gamma_i, \Delta_i \not\vdash_{N(\mathcal{L})} w:\phi$ to show this we show that

$$\text{if } \Gamma_i, \Delta_i \not\vdash_{N(\mathcal{L})} w:\phi \text{ then } \Gamma_{i+1}, \Delta_{i+1} \not\vdash_{N(\mathcal{L})} w:\phi.$$

The only non-trivial case is the addition of witness to the truth of $w:\mathcal{C}^e\phi_1 \ldots \phi_e$. Suppose that

$$\Gamma_i \cup \{w:\mathcal{C}^e\phi_1 \ldots \phi_e, t_1:\phi_1, \ldots, t_e:\phi_e\}, \Delta_i \cup \{R^e wt_1 \ldots t_e\} \vdash_{N(\mathcal{L})} w:\phi$$

where $t_1, \ldots, t_e \not\lhd (\Gamma_i \cup \{w:\mathcal{C}^e\phi_1 \ldots \phi_e\}, \Delta_i)$. Then we can apply $\mathcal{C}^e E$, and thus

$$\Gamma_i \cup \{w:\mathcal{C}^e\phi_1 \ldots \phi_e\}, \Delta_i \vdash_{N(\mathcal{L})} w:\phi.$$

Contradiction.

Now define

$$\Gamma^* = \bigcup_{i \geq 0} \Gamma_i \text{ and } \Delta^* = \bigcup_{i \geq 0} (\Delta_i)_{N(\mathcal{L})}.$$

Then, $(\Gamma, \Delta) \in (\Gamma^*, \Delta^*)$ and $a:\phi \notin (\Gamma^*, \Delta^*)$. Moreover, (Γ^*, Δ^*) is maximal with respect to $a:\phi$. Condition (i) in Definition 3.28 is satisfied by the definition of Δ^*, and we show that condition (ii) holds as well. $\Gamma^* \cup \{y:\psi\}, \Delta^* \not\vdash_{N(\mathcal{L})} a:\psi$ implies $y:\psi \in \Gamma^*$ by construction. For the converse, assume that $y:\psi \in \Gamma^*$. If $\Gamma^* \cup \{y:\psi\}, \Delta^* \vdash_{N(\mathcal{L})} a:\phi$, then, since $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} y:\psi$, by transitivity of derivations we have that $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a:\phi$. Contradiction. **Proof**

If we have that $\Delta \not\vdash_{N(\mathcal{L})} R_i ww_1 \ldots w_n$, then we can extend Δ to $\Delta^* = \Delta_{N(\mathcal{L})}$, so that we have $R_i ww_1 \ldots w_n \notin \Delta^*$. We can do this since in the definition of deductive closure we have that $\Delta^* \vdash_{N(\mathcal{L})} R_i ww_1 \ldots w_n$ iff $R_i ww_1 \ldots w_n \in \Delta^*$.

Lemma 3.30 (Viganò) *Let (Γ^*, Δ^*) be maximal with respect to $a:\phi$. Then we have:*

(i) $\Delta^* \vdash_{N(\mathcal{L})} R_i ww_1 \ldots w_n$ *iff* $R_i ww_1 \ldots w_n \in \Delta^*$.

(ii) $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w:\psi$ *iff* $w:\psi \in \Gamma^*$.

(iii) $w:\mathcal{C}^u\phi_1 \ldots \phi_u \in \Gamma^*$ *iff* $R^u ww_1 \ldots w_u \in \Delta^*$ *and* $w_1:\phi_1 \in \Gamma^*$ *and ... and* $w_{u-1}:\phi_{u-1} \in \Gamma^*$ *imply* $w_u:\phi_u \in \Gamma^*$*, for all* $w_1, \ldots, w_u$.

(iv) $w:\mathcal{C}^e\phi_1 \ldots \phi_e \in \Gamma^*$ *ifff* $R^e ww_1 \ldots w_e \in \Delta^*$ *and* $w_1:\phi_1 \in \Gamma^*$ *and ... and* $w_e:\phi_e \in \Gamma^*$*, for some* $w_1, \ldots, w_e$.

(v) $w:\phi_1 \wedge \phi_2 \in \Gamma^*$ *iff* $w:\phi_1 \in \Gamma^*$ *and* $w:\phi_2 \in \Gamma^*$.

(vi) $w:\phi_1 \vee \phi_2 \in \Gamma^*$ *iff* $w:\phi_1 \in \Gamma^*$ *or* $w:\phi_2 \in \Gamma^*$.

(vii) $w:\phi_1 \supset \phi_2 \in \Gamma^*$ *iff* $w:\phi_1 \in \Gamma^*$ *implies* $w:\phi_2 \in \Gamma^*$.

Proof The proof of (i) is straightforward.

(ii) Suppose that $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w : \psi$. If $w : \psi \notin \Gamma^*$, then, since (Γ^*, Δ^*) is maximal with respect to $a:\phi, \Gamma^* \cup \{w:\psi\}, \Delta^* \vdash_{N(\mathcal{L})} w:\phi$, and thus, by transitivity of derivations, $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a:\phi$. Contradiction. The converse holds by definition.

(iii) Suppose that $w\!:\!\mathcal{C}^u\phi_1\ldots\phi_u \in \Gamma^*$. Then $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w\!:\!\mathcal{C}^u\phi_1\ldots\phi_u$ by (ii). Now if $R^u ww_1\ldots w_u \in \Delta^*$ and $w_1\!:\!\phi_1 \in \Gamma^*$ and $\ldots$ and $w_{u-1}\!:\!\phi_{u-1} \in \Gamma^*$, we conclude $w_u\!:\!\phi_2 \in \Gamma^*$ by (i), (ii) and $\mathcal{C}^u E$. For the converse, assume that $w\!:\!\mathcal{C}^u\phi_1\ldots\phi_u \notin \Gamma^*$, and prove that there exist $w_1,\ldots,w_u$ such that $R^u ww_1\ldots w_u \in \Gamma^*$ and $w_1\!:\!\phi_1 \in \Gamma^*$ and $\ldots$ and $w_{u-1}\!:\!\phi_{u-1} \in \Gamma^*$ imply $w_{u-1} \notin \Gamma^*$. By (i) and (ii), the assumption yields

$$\Gamma^* \cup \{w\!:\!\mathcal{C}^e\phi_1\ldots\phi_u\}, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi.$$

Now if for all $w_1,\ldots,w_u$,

$$\Gamma^* \cup \{w_1\!:\!\phi_1,\ldots,w_{u-1}\!:\!\phi_{u-1}\}, \Delta^* \cup \{R^u ww_1\ldots w_u\} \vdash_{N(\mathcal{L})} w_u\!:\!\phi_u,$$

then, by $\mathcal{C}^u I$, we have $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w\!:\!\mathcal{C}^u\phi_1\ldots\phi_u$, and thus $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi$ by transitivity of derivations. Contradiction.

(iv) Suppose that $R^e ww_1\ldots w_e \in \Delta^*$ and $w_1\!:\!\phi_1 \in \Gamma^*$ and $\ldots$ and $w_{e-1}\!:\!\phi_{e-1} \in \Gamma^*$ imply $w_e\!:\!\phi_e \notin \Gamma^*$, for all $w_1,\ldots,w_e$. Then, by (i) and (ii), we have

$$\Gamma^* \cup \{w_1\!:\!\phi_1,\ldots,w_{u-1}\!:\!\phi_{u-1}\}, \Delta^* \cup \{R^u ww_1\ldots w_u\} \vdash_{N(\mathcal{L})} w_u\!:\!\phi_u,$$

for all $w_1,\ldots,w_e$. Now, if $w\!:\!\mathcal{C}^e\phi_1\ldots\phi_e \in \Gamma^*$, then, by (ii), $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w\!:\!\mathcal{C}^e\phi_1\ldots\phi_e$, and thus $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi$ by $\mathcal{C}^e E$. Contradiction. For the converse suppose that $w\!:\!\mathcal{C}^e\phi_1\ldots\phi_e \notin \Gamma^*$. Then

$$\Gamma^* \cup \{w\!:\!\mathcal{C}^e\phi_1\ldots\phi_e\}, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi.$$

If for some $w_1,\ldots,w_e, R^e ww_1\ldots w_e \in \Delta^*$ and $w_1 : \phi_1 \in \Gamma^*$ and $\ldots$ and $w_e : \phi_e \in \Gamma^*$, then, by (i), (ii) and $\mathcal{C}^e I$, we have $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w : \mathcal{C}^e\phi_1\ldots\phi_e$, and thus $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a : \phi$, by transitivity of derivations. Contradiction.

The proofs of (v) and (vi) are straightforward and so we only prove (vi). Suppose that $w\!:\!\phi_1 \vee \phi_2 \in \Gamma^*$. If $w\!:\!\phi_1 \notin \Gamma^*$ and $w\!:\!\phi_2 \notin \Gamma^*$, then

$$\Gamma^* \cup \{w\!:\!\phi_1\}, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi \text{ and } \Gamma^* \cup \{w\!:\!\phi_2\}, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi,$$

and thus $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a : \phi$ by (ii) and $\wedge E$. Contradiction. For the converse suppose that $w\!:\!\phi_i \in \Gamma^*$ for $i = 1$ or $i = 2$. If $w\!:\!\phi_1 \vee \phi_2 \notin \Gamma^*$, then

$$\Gamma^* \cup \{w\!:\!\phi_1 \vee \phi_2\}, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi.$$

By (ii) and $\vee Ii$ for $i = 1$ or $i = 2$, the assumption yields $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w\!:\!\phi_1 \vee \phi_2$, and thus $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi$, by transitivity of derivations. Contradiction.

(vii) Suppose that $w\!:\!\phi_1 \supset \phi_2 \in \Gamma^*$ and $w\!:\!\phi_1 \in \Gamma^*$. If $w\!:\!\phi_2 \notin \Gamma^*$, then

$$\Gamma^* \cup \{w\!:\!\phi_2\}, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi.$$

By (ii) and $\supset I$, the assumptions yield $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w : \phi_2$, and thus $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a : \phi$ by transitivity of derivations. Contradiction. For the converse suppose that $w\!:\!\phi_1 \in \Gamma^*$ implies $w\!:\!\phi_2 \in \Gamma^*$. If $w\!:\!\phi_1 \supset \phi_2 \notin \Gamma^*$, then

$$\Gamma^* \cup \{w\!:\!\phi_1 \supset \phi_2\}, \Delta^* \vdash_{N(\mathcal{L})} a\!:\!\phi.$$

By (ii) and $\supset I$, the assumptions yield $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} w:\phi_1 \supset \phi_2$, and thus $\Gamma^*, \Delta^* \vdash_{N(\mathcal{L})} a:\phi$ by transitivity of derivations. Contradiction. □

We can now define the canonical model.

Definition 3.31 (Viganò) *Given a proof context* (Γ^*, Δ^*) *maximal with respect to* $a:\phi$*, we define the* <u>*canonical model*</u> $\mathcal{M}^C = (\mathcal{W}^C, \wr^C, \mathcal{R}^{u^c}, \mathcal{R}^{e^C}, *^C, \mathcal{B}^C)$ *for the system* $N(\mathcal{L})$ *as follows:*

- $\mathcal{W}^C = \{w | w \lhd (\Gamma^*, \Delta^*)\}$*, where* $\wr^C = 0$ *and* $w^{*^C} = w^*$*;*
- $(w, w_1, \ldots, w_u) \in \mathcal{R}^{u^C}$ *iff* $R^u w w_1 \ldots w_u \in \Delta^*$*, and* $(w, w_1 \ldots, w_e) \in \mathcal{R}^{e^C}$ *iff* $R^e w w_1 \ldots w_e \in \Delta^*$*;*
- $\mathcal{B}^C(w, p) = 1$ *iff* $w:p \in \Gamma^*$*.* □

The usual definition of $\mathcal{R}^{u^C}$ as $(w, w_1, \ldots, w_u) \in \mathcal{R}^{u^C}$ iff $\{\phi_u | \mathcal{C}^u \phi_1 \ldots \phi_u \in w, \phi_1 \in w, \ldots, \phi_{u-1} \in w_{u-1}\} \subseteq w_u$, does not apply since this definition does not imply $\vdash_{N(\mathcal{L})} R^u w w_1 \ldots w_u$. This means that completeness would not hold since we would have cases where $\nvdash_{N(\mathcal{L})} R^u w w_1 \ldots w_u$ but $(w, w_1, \ldots, w_U) \in \mathcal{C}^{u^C}$ and thus $\models_{\mathcal{M}^c} R^u w w_1 \ldots w_u$. So we define $(w, w_1 \ldots, w_u) \in \mathcal{C}^{u^C}$ iff $R^u w w_1 \ldots w_u \in \Delta^*$, which implies the usual definition. We thus have

Proposition 3.32 (Viganò) $R_i w w_1 \ldots w_n \in \Delta^*$ *iff* $\Delta^* \models_{\mathcal{C}^C} R_i w w_1 \ldots w_n$*.*

We need one more lemma then we can prove completeness.

Lemma 3.33 (Viganò) $w:\psi \in (\Gamma^*, \Delta^*)$ *iff* $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} w:\psi$*.*

Proof We proceed by induction on the number of local and global operators that occur in ψ, and we only prove the case where $w : \psi$ is $w : \mathcal{C}^u \phi_1 \ldots \phi_u$; the other cases follow analogously.

For the left-to-right implication, assume that $w : \mathcal{C}^u \phi_1 \ldots \phi_u \in \Gamma^*$. Then, by Lemma 3.30. $R^u w w_1 \ldots w_u \in \Delta^*$ and $w_1 : \phi_1 \in \Gamma^*$ and … and $w_{u-1} \in \Gamma^*$ imply $w_u : \phi_u \in \Gamma^*$, for all $w_1, \ldots, w_u$. Propositions 3.32 and the induction hypotheses yield $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} w_u : \phi_u$ for all $w_1, \ldots, w_u$ such that $\Delta^* \models_{\mathcal{M}^C} R^u w w_1 \ldots w_u$ and $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} w_1 : \phi_1$ and … and $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} w_{u-1} : \phi_{u-1}$, *i.e.* $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} w : \mathcal{C}^u \phi_1 \ldots \phi_u$ from the definition of truth.

For the right-to-left implication, assume that $w : \mathcal{C}^u \phi_1 \ldots \phi_u \notin \Gamma^*$. Then, by Lemma 3.30, we have that $R^u w w_1 \ldots w_u \in \Delta^*$ and $w_1 : \phi_1 \in \Gamma^*$ and … and $w_{u-1} : \phi_{u-1} \in \Gamma^*$ and $w_u : \phi_u \notin \Gamma^*$, for some $w_1, \ldots, w_u$. Proposition 3.32 and the induction hypotheses yield $\Delta^* \models_{\mathcal{M}^C} R^u w w_1 \ldots w_u$ and $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} w)1 : \phi_1$ and … and $\Gamma^*, \Delta^* \models_{\mathcal{M}^C} w_{u-1} : \phi_{u-1}$ and $\Gamma^*, \Delta^* \not\models_{\mathcal{M}^C} w_u : \phi_u$, *i.e.* $\Gamma^*, \Delta^* \not\models_{\mathcal{M}^C} w : \mathcal{C}^u \phi_1 \ldots \phi_u$ from the definition of truth. □

Now we can prove completeness.

Lemma 3.34 (completeness (Viganò)) *For the system* $N(\mathcal{L})$*, the following holds:*

(i) $\Delta \models R_i w w_1 \ldots w_n$ *implies* $\Delta \vdash_{N(\mathcal{L})} R_i w w_1 \ldots w_n$*, and*

(ii) $\Gamma, \Delta \models w:\phi$ *implies* $\Gamma, \Delta \vdash_{N(\mathcal{L})} w:\phi$.

Proof (*i*) If $\Delta \not\vdash_{N(\mathcal{L})} R_i w w_1 \dots w_n$, then $R_i w w_1 \dots w_n \notin \Delta^*$, and thus $\Delta^* \not\models_{\mathcal{M}^C} R_i w w_1 \dots w_n$, by Proposition 3.32. Hence $\Delta \not\models_{\mathcal{M}^C} R_i w w_1 \dots w_n$.

(*ii*) If $\Gamma, \Delta \not\vdash_{N(\mathcal{L})} w:\phi$, then we extend (Γ, Δ) to a proof context (Γ^*, Δ^*) maximal with respect to $w:\phi$. Then, by Lemma 3.33, $\Gamma^*, \Delta^* \not\models_{\mathcal{M}^C} w:\phi$, and thus, $\Gamma, \Delta \not\models_{\mathcal{M}^C} w:\phi$. □

We summarize the results in the following theorem.

Theorem 3.35 (soundness and completeness) *For the system* $N(\mathcal{L})$*, the following holds:*

(i) $\Delta \models R_i w w_1 \dots w_n$ *iff* $\Delta \vdash_{N(\mathcal{L})} R_i w w_1 \dots w_n$*, and*

(ii) $\Gamma, \Delta \models w:\phi$ *iff* $\Gamma, \Delta \vdash_{N(\mathcal{L})} w:\phi$.

4 Judged Proof Systems

4.1 Introduction

We introduce the idea of a judged proof system by first studying how to capture the various modal logics. This provides the motivation for further work on judgements. We note that the labelled systems discussed in the previous chapter capture a greater range of modal logics than tht next section, since our discussion here requires that we have an axiom corresponding to the geometric condition on the Kripke frame. There are more geometric conditions than known corresponding axioms. However the judged systems are more general than this and as we will show in Section **??** can capture labelled systems.

To conclude this section, we provide the definitions of both a Hilbert-style and natural deduction judged proof system. This then leads into the next section where we provide a semantics for any judged proof system which is a hybrid of Hilbert-style and natural deduction.

4.2 Judged Proof Systems for Modal Logic

Let us suppose, that for some language L, we have a Hilbert-type or natural deduction system $\mathcal{L}$. The basic ides of a proof in such a system is that of a labelled tree. The labels of the tree are formulae of L. Successor nodes are generated by the axioms and inference rules of $\mathcal{L}$ subject to the following condition

(*) The formula which labels a node which is not a leaf must follow from the formulae which label its successors by one of the inference rules of $\mathcal{L}$.

Such systems are said to be pure [Avr91, AHMP98] if the condition $(*)$ has the following localness property: that, at a given node, it can be checked by examining just that node and its successors. Examples of rules which fail to have this localness property are the □-rule in the standard Hilbert-type presentation of $S4$, which requires the global condition that the premiss be a theorem, and $\Box I$ in Prawit's natural deduction system for $S4$ [Pra65], which requires the global condition that all of the hypotheses have $\Box$ outermost.[5] Systems which require such global conditions in order to determine local correctness are said to be impure. From the point of view of logical frameworks, impure systems are problematic for at least the following two reason: firstly, from a structural point of view, the formal description of global conditions may require *ad hoc* additions either to the language of the framework or to the representation mechanism or to both; secondly, checking such global conditions can be computationally expensive.

It is common practice to give both Hilbert-type sand natural deduction presentations of logics as systems for deriving formulae that are bare propositions ϕ. In such formulations, Hilbert-type and natural deduction inference rules can be considered to have the form

$$\forall\Gamma_1 \ldots \Gamma_m \qquad \frac{\Delta_1(\Gamma_1) \vdash^{i_1} \ldots \Delta_m(\Gamma_m) \vdash^{i_m} \phi_m}{(\bigoplus_{i=1}^{m} \Gamma_i) \vdash^{i} \phi} \, C$$

where each $\Delta_i(\Gamma_i)$ denotes a context, *i.e.*, a collection of bare propositions, which includes the components of Γ, $(\bigoplus_{i=1}^{m} \Gamma_i)$ denotes a combination of components of the Γ_is and C is a possible side condition, concerning things like occurrences of variables or occurrences of modalities. Typically, these side-conditions are global conditions which must be checked at the application of the rule.

However, by moving from bare propositions ϕ to judged propositions $\mathsf{j}(\phi)$, with a given logic exploiting possibly many different judgements, we find that global correctness conditions can be rendered local.[6] The technique is best understood by considering a (quite general) example, from which the general situation should be clear.

Our subsequent informal discussion applies to Hilbert-type presentations of minimal, intuitionistic and classical first-order and higher-order predicate logics and minor variations thereon, as well as to the family K, KT, $K4$, $KT4$ (or $S4$), $KT45$ (or $S5$) and KL of modal logics, as discussed in [AHMP98]. It also applies to the following systems from [AHMP92], and some minor variations thereon: Kleene's three-valued logic; classical first-order logic with (a version of) Hilbert's choice operator, classical λ-calculus; call-by-value λ-calculus; and, with care, Hoare's logic.

Consider any system $\mathcal{L}$, over a language L, drawn from the collection described above, in particular suppose we have the following $\Box$ rule:

$$\frac{\phi}{\Box\phi} \, \Box \quad (\phi \text{ depends on no assumptions})$$

[5]In a Hilbert-type system, any rule of proof, *i.e.* a rule with no side-formulae, is also an example. Rules with side-formula are called rules of derivation.

[6]In fact, we can make a stronger claim: that presentations based on bare propositions are formally inadequate. However, the development of the supporting argument is beyond our present scope.

Here the condition that ϕ depends on no assumptions, *i.e.* that ϕ be a theorem, is a global one on the Hilbert-type proof of ϕ. To check that the rule $\Box$ has been used correctly, we must check that all of the formulae at the leaves of the Hilbert-type proof of ϕ are themselves theorems.

By reconstructing $\mathcal{L}$ (*cf.* propositional $S4$) as a judged logic with two judgements, true and valid, we can render the check for theoremhood a local one. The presence of the judgement valid allows us to separate the rules of a system into two groups. The first group contains the usual rules of classical logic and allows the inference just of the propositions judged true. The second group consists of the rules for $\Box$ which can be use to derive valid propositions only in valid contexts. All axioms are judged valid.

Each of true and valid is a symbol with propositional arity, so that the pairs $\langle \phi, \mathsf{true} \rangle$ and $\langle \phi, \mathsf{valid} \rangle$ are formulae of judged $\mathcal{L}$. The nodes of a proof in judged $\mathcal{L}$ are thus labelled with such pairs. Following [AHMP98], we say that such a tree is a judged $\mathcal{L}$-proof if the following conditions are satisfied:

- The tree is a legitimate proof-tree in the system $\mathcal{L}$', which is the system obtained from $\mathcal{L}$ by transforming all rules of proof into rules of derivation (by adding side-formulae in the obvious way);
- A node which is not a leaf is labelled valid if and only if all of its successors are so labelled;
- Any node derived by a rule of proof of $\mathcal{L}$ is labelled valid;
- Axioms of $\mathcal{L}$ are labelled valid.

The first group of rules includes the following judged version of the $\Box$ rule:

$$\frac{\langle \phi, \mathsf{valid} \rangle}{\langle \Box(\phi), \mathsf{valid} \rangle} \Box \ .$$

Rules from the first group are accessible to valid propositions via the following connecting rule:

$$\frac{\langle \phi, \mathsf{valid} \rangle}{\langle \phi, \mathsf{true} \rangle} C \ .$$

A straightforward argument by induction on the structure of $\mathcal{L}$-proofs, judged $\mathcal{L}$-proofs and $\mathcal{L}'$-proofs leads to the following lemma:

Lemma 4.1 (judged systems [AHMP98]) *The erasing of the labelling judgement is a compositional bijection between:*

1. *$\mathcal{L}'$-proofs and judged $\mathcal{L}$-proofs in which all nodes are labelled* valid*;*
2. *$\mathcal{L}$-proofs and judged $\mathcal{L}$-proofs in which all leaves which are not axioms are labelled* true*.*

Proof We show that there is a bijection between $\mathcal{L}'$-proofs and judged $\mathcal{L}$-proofs by induction over the structure of the tree of $\mathcal{L}'$-proofs.

We assume that the tree has just one node which is both a leaf and a root. There are no rules of proof to transform into a derivation, so there is a bijection regardless of whether the node is an axiom or not.

We now assume that we are at a node which has two leaves. If we have used a rule of proof to join the two nodes, we just add a side condition to the rule and we have a bijection. If we have not used a rule of proof then there is a bijection and we are done.

We assume that we are at a node which has one successor which is not a leaf. If the rule used to join these is a rule of proof then we turn it into a derivation and add the necessary side condition and we have a bijection. If it is not a rule of proof then we just take the obvious bijection.

We assume that we are at a node which has two successors which are not leaves, then we look at the rule used. If it is a rule of proof, then we add the necessary side condition and we have a bijection. Otherwise, we do nothing and have a bijection.

We show that there is a bijection between $\mathcal{L}$-proofs and judged $\mathcal{L}$-proofs by induction over the structure of the tree of $\mathcal{L}$-proofs.

We assume that the tree has just one node which is both a leaf and a root. If it is an axiom, it will be labelled as valid in judged $\mathcal{L}$ and we take the obvious bijection. If it is not an axiom, it is labelled as true in judged $\mathcal{L}$ and again we take the obvious bijection.

We assume that we are at a root which has two leaves which are both axioms. In judged $\mathcal{L}$ they are both labelled valid by assumption. We take the bijection between them.

Staying at the same node, we assume that one is not a bijection and this leaf is now labelled true in judged $\mathcal{L}$ by assumption. Again we have a bijection.

Finally, at this node, we assume that both of the leaves are not axioms and are thus they are both labelled true in judged $\mathcal{L}$ by assumption. We take the same bijection.

We now assume that we are at a node with one successor which is a leaf. If this leaf is not an axiom, it is labelled true in judged $\mathcal{L}$. We apply the induction hypothesis to the subtree below the node we are looking at and the subtree at the successor which is not a leaf to obtain a bijection these. Finally, we need to show a bijection at the node we are considering. We see that in $\mathcal{L}$ it has just one leaf which we map to the lead which is labelled true in judged $\mathcal{L}$ and use the bijections from the induction hypothesis. Similarly, if it is an axiom. $\square$

The formulation of judged systems is such that their metatheory is rather simple and elegant. Specifically, we ensure the purity of the proof systems by allowing, in a judged consequence relation, several consequence relations to be treated simultaneously. Representations of logical systems in this way, using many judgements, are called non-uniform representations by Avron [Avr91]. In our running example of Hilbert-type presentations of modal logics, we can state, informally, a theorem which explains the value of using judged systems in logical frameworks.

Let $\Sigma_{2j(\mathcal{H})}$ be the signature which represents a logic from the family K, KT, $K4$, KT (or $S4$), $KT45$ (or $S5$) and KL, formulated as a judged Hilbert-type systems.

Before we give the signatures for the Hilbert-type systems, we must give the signature for the underlying language and the encoding function for this language.[7] We have the following signature for the underlying language;

$$\Sigma = \begin{cases} o & : Type \\ \neg & : o \longrightarrow o \\ \Box & : o \longrightarrow o \\ \supset & : o \longrightarrow o \longrightarrow o \end{cases}$$

and the encoding function from the language to o is given by

$$\begin{array}{ll} \epsilon_X(x) & = x \text{ if } x \in X \\ \epsilon_X(\neg\phi) & = \neg\epsilon_X(\phi) \\ \epsilon_X(\Box\phi) & = \Box\epsilon_X(\phi) \\ \epsilon_X(\phi \supset \psi) & = \supset \epsilon_X(\phi)\epsilon_X(\psi) \end{array}$$

and thus we can define the signature for each of the Hilbert-type systems. We add the (encoded) rules K, T, 4, 5 and L to get the system we are interested in. We have two judgements which are given by

$$\begin{array}{l} \mathsf{true} : o \longrightarrow Type \\ \mathsf{valid} : o \longrightarrow Type \end{array}$$

and the axioms and rules are given by

$$\begin{array}{ll} A_1 & : \Pi x{:}o\,.\,\Pi y{:}o\,.\,\mathsf{valid}(x \supset (y \supset x)) \\ A_2 & : \Pi x{:}o\,.\,\Pi y{:}o\,.\,\Pi z{:}o\,.\,\mathsf{valid}(x \supset (y \supset z) \supset (x \supset y) \supset (x \supset z)) \\ A_3 & : \Pi x{:}o\,.\,\Pi y{:}o\,.\,\mathsf{valid}((\neg x \supset \neg y) \supset (x \supset y)) \\ K & : \Pi x{:}o\,.\,\Pi y{:}o\,.\,\mathsf{valid}(\Box(x \supset y) \supset (\Box x \supset y)) \\ T & : \Pi x{:}o\,.\,\mathsf{valid}(\Box x \supset x) \\ 4 & : \Pi x{:}o\,.\,\mathsf{valid}(\Box x \supset \Box\Box x) \\ 5 & : \Pi x{:}o\,.\,\mathsf{valid}(\Diamond x \supset \Box\Diamond x) \\ L & : \Pi x{:}o\,.\,\mathsf{valid}(\Box(\Box x \supset x(\supset \Box x) \\ MP_T & : \Pi x{:}o\,.\,\Pi y{:}o\,.\,\mathsf{true}(x \supset y) \longrightarrow \mathsf{true}(x) \longrightarrow \mathsf{true}(y) \\ MP_V & : \Pi x{:}o\,.\,\Pi y{:}o\,.\,\mathsf{valid}(x \supset y) \longrightarrow \mathsf{valid}(x) \longrightarrow \mathsf{valid}(y) \\ Nec & : \Pi x{:}o\,.\,\mathsf{valid}(x) \longrightarrow \mathsf{valid}(\Box x) \\ C & : \Pi x{:}o\,.\,\mathsf{valid}(x) \longrightarrow \mathsf{valid}(x) \end{array}$$

where C is the connecting rule mentioned earlier.

Theorem 4.2 (encoding judged $\mathcal{L}$ [AHMP98]) *For $\mathcal{L}$ one of the family K, KT, $K4$, KT, $KT45$ and KL, there is a compositional bijection between judged $\mathcal{L}$-proofs of*

$$\langle\phi_1, \mathsf{l}_1\rangle, \ldots, \langle\phi_m, \mathsf{l}_m\rangle \vdash_{\mathcal{L}} \langle\phi, \mathsf{l}\rangle$$

[7]These are the same for each of the families of modal logics we are concerned with here except that KL has $\Diamond$, which is given by $\Diamond{:}o \longrightarrow o \longrightarrow o$ in the signature and $\epsilon_X(\Diamond\phi) = \Diamond\epsilon_X(\phi)$.

and canonical $\lambda\Pi$*-terms* M *over the signature* $\Sigma_{2j(\mathcal{H})}$ *such that*

$$\Gamma_X, \gamma_v(\Delta), \gamma_t(\Xi) \vdash_{\Sigma_{2j(\mathcal{L})}} M : \mathsf{j}(\epsilon_X(\phi))$$

where

- $\Delta = \{\phi_i | l_i = \mathsf{valid}\}$,
- $\Xi = \{\phi_i | l_i = \mathsf{true}\}$,
- $\mathsf{j} = \mathsf{true}$ *if* $\mathsf{l} = \mathsf{true}$ *and* valid *otherwise,*

and where Γ_X *and* $\Phi_X = \gamma_v(\Delta), \gamma_t(\Xi)$ *are defined as in § 2.3.*

A canonical term is one of the form x, $\lambda x : A \,.\, M$ or MN which is in β-normal form.

Proof We show that there is a bijection from judged $\mathcal{L}$-proofs to $\lambda\Pi$-terms M by induction over proofs in judged $\mathcal{L}$.

We consider a proof where just an axiom has been used. These are all labelled valid. By assumption, we have a constant in the signature for each axiom. Since we have no assumptions in the proof, we are encoding something of the form

$$X \vdash_{\mathcal{L}} \langle Ax, \mathsf{valid}\rangle$$

where Ax is an axiom and X corresponds to the variables used in the formulae. This corresponds to

$$\Gamma_X \vdash_{\Sigma_{wj(\mathcal{H})}} Ax : \mathsf{valid}(\epsilon_X(Ax))$$

in $\lambda\Pi$ applied to two instantiations of the variables. So taking K as an example, proving the formulae

$$(\Box(\phi \supset \psi) \supset (\Box\phi \supset \Box\psi)$$

amounts to invoking the axiom K and so in judged $\mathcal{L}$ corresponds to the judged $\mathcal{L}$-proof

$$\vdash_{\mathcal{L}} \langle(\Box(\phi \supset \psi) \supset (\Box\phi \supset \Box\psi), \mathsf{valid}\rangle$$

which is encoded as the term

$$x : o, y : o \vdash_{\Sigma_{2j(\mathcal{H})}} K : \Pi x : o \,.\, \Pi y : o \,.\, \mathsf{valid}(\Box(x \supset y) \supset (\Box x \supset \Box y))$$

and $x : o, y : o$ are just the encoding of syntactic variables used in the proof, *i.e.* Γ_X. We then use the application rule together with $\vdash_{\Sigma_{2j(\mathcal{H})}} \epsilon_X(\phi) : o$ and then with $\vdash_{\Sigma_{2j(\mathcal{H})}} \epsilon_X(\psi) : o$ to obtain

$$K\epsilon_X(\phi)\epsilon_X(\psi) : \mathsf{valid}(\Box(\phi \supset \psi) \supset (\Box\phi \supset \Box\psi))$$

which is a canonical term in $\lambda\Pi$.

We now look at how we apply MP_V. This is encoded to the constant

$$MP_V : \Pi x : o \,.\, \Pi y : o \,.\, \mathsf{valid}(x \supset y) \longrightarrow \mathsf{valid}(x) \longrightarrow \mathsf{valid}(y)$$

and we consider that we already have proofs of $x \supset y$ and X which have been encoded to $\Gamma_X, \Gamma_\Xi \vdash_{\Sigma_{2j(\mathcal{H})}} \Theta_1 : \mathsf{valid}(\epsilon_X(\phi) \supset \epsilon_X(\psi))$ and $\Gamma_X, \Gamma_\Xi \vdash_{\Sigma_{2j(\mathcal{H})}} \Theta_2 : \mathsf{valid}(\epsilon_X(\phi))$. Without loss of generality, we take Γ_X to encode all the syntactic variables in both proofs and similarly Γ_Ξ for all the formulae in both proofs. We apply the rule

$$\frac{\Gamma \vdash_\Sigma M : \Pi x : A \, . \, B \quad \Gamma \vdash_\Sigma N : A}{\Gamma \vdash_\Sigma MN : B[N/x]}$$

twice to obtain the term

$$\Gamma_X, \Gamma_\Xi \vdash_{\Sigma_{2j(\mathcal{H})}} MP_V \epsilon_X(\phi)\epsilon_X(\psi) : \mathsf{valid}(\epsilon_X(\phi) \supset \epsilon_X(\psi)) \longrightarrow \mathsf{valid}(\epsilon_X(\phi)) \longrightarrow \mathsf{valid}(\epsilon_X(\psi))$$

to which we apply the same rule twice with $\Gamma_X, \Gamma_\Xi \vdash_{\Sigma_{2j(\mathcal{H})}} \Theta_1 : \mathsf{valid}(\epsilon_X(\phi) \supset \epsilon_X(\psi))$ and $\Gamma_X, \Gamma_\Xi \vdash_{\Sigma_{2j(\mathcal{H})}} \Theta_2 : \mathsf{valid}(\epsilon_X(\phi))$ to obtain

$$\Gamma_X, \Gamma_\Xi \vdash_{\Sigma_{j(\mathcal{H})}} MP_V \epsilon_X(\phi)\epsilon_X(\psi)\Theta_1\Theta_2 : \mathsf{valid}(\epsilon_X(\psi))$$

which is a canonical term in $\lambda\Pi$.

MP_T, Nec and C are handled in the same way except that Nec and C only require one instantiation of a variable.

To show that this is in fact a bijection, we describe an inverse δ_X. We first describe it for the underlying language

$$\begin{array}{ll} \delta_X(x) & = x \text{ if } x \in X \\ \delta_X(\neg\phi) & = \neg\delta_X(\phi) \\ \delta_X(\Box\phi) & = \Box\delta_X(\phi) \\ \delta_X(\supset \phi\psi) & = \delta_X(\phi) \supset \delta_X(\psi) \end{array}$$

and now for the canonical terms of $\lambda\Pi$

$$\begin{array}{ll} \delta_X(\mathsf{valid}(\phi) : o) & = \langle \phi, \mathsf{valid} \rangle \\ \delta_X(\mathsf{true}(\phi) : o) & = \langle \phi, \mathsf{true} \rangle \\ \delta_X(A_i M_1 \dots M_n) & = A_i(\delta_X(M_1) \dots \delta_X(M_n)) \\ \delta_X(NecMN) & = Nec(\delta_X(M))\delta_X(N) \\ \delta_X(MP_T MNPQ) & = MP_T(\delta_X(M)\delta_X(N))\delta_X(P)\delta_X(Q) \\ \delta_X(MP_V MNPQ) & = MP_V(\delta_X(M)\delta_X(N))\delta_X(P)\delta_X(Q) \\ \delta_X(CMN) & = C(\delta_X(M))\delta_X(N). \end{array}$$

Here a rule or axiom followed by a bracketed term means we instantiate the rule or axiom using that term, *i.e.* $A_1(\phi\psi) = \phi \supset (\psi \supset \phi)$ and $A_1(\alpha\beta) = \alpha \supset (\beta \supset \alpha)$. This function is total and well-defined and it is defined so that it is an inverse to ϵ_X, thus giving a bijection between judged $\mathcal{L}$-proofs and canonical $\lambda\Pi$-terms. □

Corollary 4.3 (encoding $\mathcal{L}$ and $\mathcal{L}'$ [AHMP98]) *Suppose* $\{\phi_1, \dots, \phi_m, \phi\} \in \Phi_X(L)$.

1. *There is a compositional bijection between proofs in $\mathcal{L}'$ of $\phi_1, \ldots, \phi_m \vdash \phi$ and canonical $\lambda\Pi$-terms M such that*

$$\Gamma_X, \gamma_v(\{\phi_1, \ldots, \phi_m\}) \vdash_{\Sigma_{2j(\mathcal{H})}} M : \mathsf{valid}(\epsilon_X(\phi)).$$

2. *There is a compositional bijection between proofs in $\mathcal{L}$ of $\phi_1, \ldots, \phi_m \vdash \phi$ and canonical $\lambda\Pi$-terms M such that*

$$\Gamma_X, \gamma_t(\{\phi_1, \ldots, \phi_m\}) \vdash_{\Sigma_{2j(\mathcal{H})}} M : \mathsf{j}(\epsilon_X(\phi)),$$

where j *is* valid*, if either $m = 0$ or no ϕ_i occurs in the proof, and is* true*, otherwise.*

Proof Any canonical $\lambda\Pi$-terms where the final type is of the form $\mathsf{valid}(\epsilon_X(\phi))$ means that it corresponds to a proof in $\mathcal{L}'$ since any $\lambda\Pi$-term which has it's final node as valid must have all successors as valid and by Lemma 4.1 there is a bijection.

If $\mathsf{j} = \mathsf{true}$ then all of it's successors must be valid and so it corresponds to a proof in $\mathcal{L}$ by Lemma 4.1. If $\mathsf{j} = \mathsf{valid}$ and $m = 0$ or no ϕ_i occurs in the proof, then no assumptions have been used and so it corresponds to a proof in $\mathcal{L}$. □

Before embarking upon our technical development of judged logics, it will be useful to consider a range of examples of normative formulations of object-logics and the representation mechanisms that can be used to encode them in LF, judgements-as-types being our leading example.

We identify three typical, though not exhaustive, cases. Again, we draw substantially upon [AHMP98] for background.

1. Proof-trees labelled with multiple judgements, encoded using the judgements-as-types representation mechanism. For examples:

    - Hilbert-type formulations, for both truth and validity, of the modal systems, such as K, $K4$, KT, $S4$, *etc.* discussed above. Two logical judgements, true and valid, are used.
    - Natural deduction formulations, for validity, of the modal logics discussed above. Two logical judgements, true and valid are used.
    - Natural deduction formulations, for truth and validity, of the modal logics discussed above. Three logical judgements, taut, valid and true are used.

2. Proof-trees labelled (degenerately) with a single logical judgement, encoded using the worlds-as-parameters representation mechanism. For example:

    - Hilbert-type formulations, for truth, of K, $K4$, KT, $S4$, *etc.*. In a worlds-as-parameters encoding, worlds are introduced via a sort, the "universe" which, having no constructors, is inhabited only by variables, or "worlds". The use of these parameters is purely syntactic. They permit the representation of the global side-conditions found in the rules of proof, such as "no assumptions", by transferring them to metalogical conditions such as "no free variables". The details can be found in [AHMP98].

- Natural deduction formulations, for truth, of K, $K4$, KT, $S4$, *etc.*.

3. Proof-trees labelled with a single logical judgement but with additional syntactic, structural judgements, such as "closed assumptions", "boxed assumptions" or "boxed fringe" [AHMP98], encoded using a representation mechanism which is a variation on judgements-as-types in which the types which encode a consequence correspond not to propositions that have been judged logically but to propositions that have first been judged logically and then judged structurally.

We now formally define the judged logical system. For the sake of brevity, and due to later results, we use the same definitions as the we used to define a labelled system. The main differences being that we have different inference rules, we do not need as many, and that we have to define judgements and make sure that all our formulae are judged. We begin by modifying Definition 3.1.

Definition 4.4 (alphabet) *An alphabet is a quintuple $A = (S, V, E, C, \mathsf{J})$ of sets of symbols defined as follows:*

- *S is a finite set of symbols with natural number arities;*
- *$V \subseteq S$ is a distinguished subset of S which contains variables;*
- *E is a finite set of expression symbols;*
- *$C \subseteq E$ is a distinguished subset of E which contains connectives;*
- *J is a finite set of judgement symbols.*

The main differences are that we have added a set of judgement symbols and not distinguished the connectives in any way. The development of the judged system now follows the development of the labelled deductive system until Definition 3.4. We now continue from here to define judgements.

An alphabet includes a set of judgement symbols J. We take each $\mathsf{j} \in \mathsf{J}$ to be equipped with an arity of the form $(s_1, \ldots, s_m)$, where each s_i is a syntactic category.

Definition 4.5 (basic judgements) *Let $A = (S, V, E, C, \mathsf{J})$ be an alphabet. The set J of basic judgements generated by A is*

$$\{\mathsf{j}(e_1, \ldots, e_m) | \mathsf{j} \in \mathsf{J} \text{ has arity } (s_1, \ldots, s_m) \text{ and, for } 1 \leq i \leq m, \text{ each } e_i \text{ is in } s_i\}$$

For each syntactic category c, we assume the existence of a judgement null *of arity c.* □

The term basic judgement derives from the work of Martin-Löf [ML82] on the meanings of logical constants and rules of inference, which in term derives from Kant [Kan00]. We let J, K, L, possibly subscripted, range over the basic judgements. Martin-Löf further constructs general judgements, of the form $\Lambda x : J \,.\, K$ and the hypothetical judgements, of the form $J \vdash K$. These higher-order judgements should be read, respectively, as the universal and implication formulae of a metalogic which has basic judgements as its atomic formulae. This metalogic is non-other than the internal logic of the language ($\lambda\Pi$) of the LF logical framework.[8]

Definitions 4.4, 3.2 to 3.4 and 4.5 determine a language L. In order to define a logic $\mathcal{L}$, we must consider consequences and their axiomatization.

For a set S, let $\wp_f(S)$ denote the set of all finite sequences of elements of S.

Definition 4.6 (judged consequence relations) *Let $A = (S, V, E, C, \mathsf{J})$ be an alphabet. A judged consequence relation (JCR) over A is a pair $(\mathsf{J}, \vdash)$, where J is a set of basic judgements over A and $\vdash \subseteq \wp_f(\mathsf{J}) \times \mathsf{J}$ is a binary relation such that:*

1. *Reflexivity: $J \vdash J$, for each basic judgement $J \in \mathsf{J}$;*
2. *Transitivity (cut): If $\Delta \vdash J$ and $\Delta, J, \Delta' \vdash K$, then $\Delta, \Delta' \vdash K$, for each $\Delta, \Delta' \in \wp_f(\mathsf{J})$ and each $J, K \in \mathsf{J}$;*
3. *Weakening: If $\Delta \vdash J$, then $\Delta, \Delta' \vdash J$, for each $\Delta, \Delta' \in \wp_f(\mathsf{J})$ and each $J \in \mathsf{J}$.*

A judged consequence relation $(\mathsf{J}, \vdash)$ is permuting if $\Delta, J, K\Delta' \vdash L$ implies $\Delta, K, J, \Delta' \vdash L$, for each $\Delta, \Delta' \in \wp_f(\mathsf{J})$ and each $J, K, L \in \mathsf{J}$. □

JCRs provide an abstract characterization of the correct consequences of a logical system. Access to consequence is provided either via a class of models and satisfaction relations, the topic of § 5, or via proof systems, to which we now turn.

In the context of LF, we are concerned with two classes of proof system, Hilbert-type systems and natural deduction systems.[9]

The use of multiple judgements means that we can avoid systems which are impure in the sense of (∗). However we now have the problem of non-uniformity in the sense of being of the form of the deductive system defined below. The class of logics which is non-uniform is smaller than the class of logics which are impure and thus we are admitting a much larger class of logics.

We write a JCR as just $\vdash$ where the set of basic judgments J is clear from the context.

Definition 4.7 (Hilbert-type systems) *Let $A = (S, V, E, C, \mathsf{J})$ be an alphabet and let $\vdash$ be a JCR over A. A Hilbert-type system for $\vdash$ is given by the following:*

- *A set of axioms $\mathsf{A} \subseteq \mathsf{J}$;*
- *A set of rules of the form*

$$\frac{J_1 \ldots J_n}{J}$$

  where, for $1 \leq i \leq m$, $J_i \in \mathsf{J}$ and $J \in \mathsf{J}$.

[8]In this meta-logic, we can follow Martin-Löf and define the hypothetico-general judgements, of the Horn form $\Lambda x_1 : J_1 \ldots \Lambda x_m : J_m . K_1, \ldots, K_n \vdash K$, which can be read as meta-logical definitions of (Hilbert-type and natural deduction) inference rules.

[9]Recall that LF is not concerned with sequent calculi.

We shall refer to the Hilbert-type system $\mathcal{L}$ for $\vdash$ over A. □

Definition 4.8 (natural deduction systems) *Let $A = (S, V, E, C, \mathsf{J})$ be an alphabet and let $\vdash$ be a JCR over A. A* <u>*natural deduction system*</u> *for $\vdash$ is given by the following:*

- *A set of axioms $\mathsf{A} \subseteq \mathsf{J}$;*
- *For each connective $\# \in C$, an introduction rule schema of the form*

$$\dfrac{\begin{matrix} & [K_{i,1}] \cdots [K_{i,h_i}] & \\ \vdots & \vdots & \vdots \\ J_1 & J_i & J_p \end{matrix}}{J} \; \# \, I$$

  in which $K_{i,1} = \mathsf{k}_{i,1}(e_1, \ldots, e_n)$, $K_{i,h_i} = \mathsf{k}_{i,h_i}(e_1, \ldots, e_n)$, $J_i = j_i(e_1, \ldots, e_n)$, for $1 \leq 1 \leq p$, and $J = \mathsf{j}(\#(e_1, \ldots, e_n))$, and where each e_i is of the form $e_i(\phi_1, \ldots, \phi_m)$: the inference infers a basic judgement $\mathsf{j}(\#(e_1, \ldots, e_n))$ from p premises $J_1 \ldots, J_p$ and can bind assumptions of the form $K_{i,1}, \ldots, K_{i,h_i}$;

- *For each connective $\# \in C$, an elimination rule schema of the form*

$$\dfrac{\begin{matrix} & [\Gamma_1] & & [\Gamma_p] \\ & \vdots & \cdots & \vdots \\ \mathsf{k}(\#(e_1, \ldots, e_n)) & J & & J \end{matrix}}{J} \; \# \, E$$

  The p minor premises of the form J are derived from the set of assumptions Γ_i, for $1 \leq i \leq p$.

We shall refer to the natural deduction system $\mathcal{L}$ for $\vdash$ over A. □

As with the labelled natural deduction system, we need to ensure that our system is consistent. This is dealt with in exactly the same way as before. We stipulate that our system must have the local reduction property, *i.e.* Definition 3.7. This guarantees that the introduction and elimination rules for "*tonk*" cannot be used in our system and that the introduction and elimination rules for a particular connective are 'inverses' of each other.

We are now able to give a general definition of a judged proof system since we have just defined judged natural deduction and Hilbert-type systems.

Definition 4.9 (Judged Proof System) *A* <u>*judged proof system*</u> *consists of a language L determined by Definitions 4.4, 3.2 to 3.4 and 4.5 together with a judged consequence relation $\vdash$ and a collection of the Hilbert-type and natural deduction inference rules defined above.* □

Note that the above definition does not exclude pure Hilbert-type and natural deduction proof systems. We do however wish to include proof systems which contain Hilbert-type and natural deduction rules. The labelled systems we defined earlier are examples of one of these hybrid proof systems, the rules for labelled formulae are natural deduction rules while the Horn relational theory consists of Hilbert-type rules. When we translate labelled systems into judged systems, we will need to be able to deal with the co-existence of both of these types of rules.

It is important that our account of logics include an account of theories; in particular of arithmetic theories. For example, the theory of Peano arithmetic for classical first-order predicate calculus requires the following extension of the language of first-order predicate calculus:

Expression symbols	0	arity ι
	$succ$	arity $\iota \longrightarrow \iota$
	$+$	arity $(\iota, \iota) \longrightarrow \iota$
	$=$	arity $(\iota, \iota) \longrightarrow o$.

To get Peano arithmetic, we need some axioms and inference rules for these expressions. For example, the substitution rule fo $=$,

$$\frac{=(t,u) \quad \phi(t)}{\phi(u)}\,(sub),$$

the rule of transitivity for $=$,

$$\frac{=(t,u) \quad =(u,v)}{=(t,v)}\,(tran),$$

and the rule of induction,

$$\frac{\phi(0) \qquad \begin{array}{c}[\phi(x)]\\ \vdots \\ \phi(succ(x))\end{array}}{\forall\, x : \iota\,.\,\phi(x)}\,(ind).$$

Definition 4.10 (theories) *Let $A = (S, V, E, C, \mathsf{J})$ be an alphabet, $\vdash$ be a JCR over A together with axioms and inference rules be a proof system. A theory is an extension of the alphabet together with extra axioms and inference rules for this extension. The alphabet is extended by adding extra expression symbols to E and the extra axioms and inference rules which correspond to these extra expressions.* □

The use of multiple judgements allows us to deal with multiple impure rules. For each impure rule, we add a new judgement. We use the same judgement for introduction and elimination rules of the same connective. There are no new rules added for these new judgements, so we can only derive one of these judgements when all of it's premisses are judged by the same judgement. This means that when we have used one of the rules using these new judgements, it must be the case that the leaves of the derivation we are applying this rule to must have been judged by the same judgement. This means that we do not have to do the non-local checking required to deal with the side condition which makes the rule impure in the unjudged case. A consequence of this is that all the axioms must be labelled with these new judgements and not the original judgement of the system, which we usually denote by true. Thus we need to introduce a rule which says that given a formula judged by one of the new judgements, we can deduce that the formula is also judged to be true. Each of this judgements defines a different consequence relation, which together make up the consequence relation of the logic. The consequence relation for true, $\vdash_{\mathsf{true}}$ is the usual consequence relation of classical logic.

5 Kripke Models of Judged Proof Systems

5.1 Introduction

We give a definition of a class of models of object-logics. Starting with the basic indexed structure, we shall add the structure required to interpret additional connectives by the requiring the existence of arrows corresponding to the clauses of a satisfaction relation which defines the connectives. We begin with an informal account of the classes of connectives we consider, before defining our class of object-logic models.

Our satisfaction relation is to be the one of Kripke forcing [Kri63]. Let W be a set of worlds and let R be an $m+1$-ary relation over W and write $R(w, w_1, \ldots, w_m)$ for the value of R at worlds $w, w_1, \ldots, w_m$. Let $\phi = \#(\phi_1, \ldots, \phi_m)$ be a formula with outermost connective $\#$ of arity m.

We distinguish two general classes (*c.f.* § 3.2) of connectives, together with judgements:

1. $\langle \#, \mathsf{j} \rangle$ is local if the meaning of $\langle \phi, \mathsf{j} \rangle$ at world w depends only on the meaning of each $\langle \phi_i, \mathsf{j}_i \rangle$ at each world w_i;

2. $\langle \#, \mathsf{j} \rangle$ is global if the meaning of $\langle \phi, \mathsf{j} \rangle$ at each world w, depends on the meaning of each $\langle \phi_i, \mathsf{j}_i \rangle$ at world w_i, where $R_{\#}(w, w_1, \ldots, w_m)$ holds for some chosen relation $R_{\#}$ over w. We will consider both universal global connectives, in which we permit universal quantification over the worlds occurring in the definiens and existential global connectives, in which we permit existential quantification over the worlds occurring in the definiens.[10]

Following this idea, we define Kripke models of object-logics in our usual indexed categorical setting, following the pattern established for $\lambda\Pi$, by requiring the base category to interpret first-order terms, given by some signature, and by requiring the fibres to carry exactly the structure specified by the satisfaction relation that defines the object-logic.

[10] Clearly, more complex classes are possible.

The use of the satisfaction relation to define the connectives arbitrarily has the consequence that we cannot guarantee that a given occurrence of a connective in a formula can be interpreted in a given model without reference to the interpretation $[\![-]\!]$ of the syntax in a model. However, we can require, at the level of prestructures, that there be enough points to interpret the function symbols of the term language in the base category.

5.2 Kripke prestructures and structures

The basic idea of our functorial treatment of models of the object-logics is one that is familiar from our treatment of models of $\lambda\Pi$and its internal logic in [PP06]. At each world, a prestructure provides, functorially, a functor from environments to values. However, in contrast with the internal logic of $\lambda\Pi$, our formulation of object-logics, specially their proof systems, makes essential use of judged propositions and we should like this to be reflected in their models.

Adumbrating our coming technical development, an example of the desirability of an explicit treatment of judgements can be seen in the case of, say, Hilbert-type presentations of $S4$. We may be interested in a model $\mathcal{K}_{\mathcal{J}}$ which supports the principle

$$\text{“if } \mathsf{valid}(\phi), \text{ then } \mathsf{true}(\phi)''$$

by virtue of the existence of an arrow

$$[\![\langle \Box\phi, \mathsf{true}\rangle]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} \xrightarrow{m} [\![\langle \Box\phi, \mathsf{valid}\rangle]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}},$$

independently of the existence of a proof δ such that

$$\mathcal{K}_{\mathcal{J}}, w, \rho \mid\!\mid\!\!-_T \delta : (\langle \Box\phi, \mathsf{valid}\rangle \vdash_T \langle \Box\phi, \mathsf{true}\rangle).$$

In order to include an explicit account of judgements in our models of object-logics, we modify the category of values, in which propositions are interpreted, to be categories of pairs of judgements and values. Specifically, we replace each category of values V, *i.e.*, each object of $\mathcal{V}$, with $C \otimes \mathsf{J}$, where J is a category of judgements.

Definition 5.1 (category of judgements) *A* *category of judgements* *is defined as follows:*

Objects: Judgements j*, including a terminal object* null*;*

Arrows: Identities and at least $\mathsf{null} \xrightarrow{\mathsf{j}} \mathsf{j}$*, for each judgement* j*.* □

Additionally, we may have arrows $\mathsf{j} \xrightarrow{m} \mathsf{k}$. which may be used in a model to validate principles of the kind discussed above. Note that although we have used the syntax of judgement symbols to describe categories of judgements, we consider any category having the specified structure to be a category of judgements.

A judged proposition $\langle \phi, \mathsf{j} \rangle$ is interpreted in a category of judged values formed by forming a product (in $\mathcal{C}$) of a category of judgements and a category of values. The transition in the semantics from values to judged values is directly, and deliberately, analogous to the transition from proof trees to proof trees labelled with judgements, which is characterized using the techniques of Lemma 4.1.

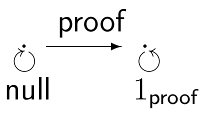
Definition 5.2 (judged values) *A* <u>*category of judged values*</u> *is a category*

$$V \otimes \mathsf{J}$$

where J *is a category of judgements,* V *is a category (of values) and* $\otimes$ *is the product in* $\mathcal{C}$*.*
□

We are always able to construct the category of judged values because both V and J live in the category of small categories and functors which is cartesian closed. Thus there is always a product operation defined in this category which we are able to apply to V and J.

For example, in our judged view of intuitionistic propositional logic, we have just one judgement, proof. In this case, the category of judgements J is exactly



and a judged proposition $\langle \phi, \mathsf{proof} \rangle$ is interpreted as an object $[\![\phi]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} \otimes \mathsf{proof}$ of $V \otimes \mathsf{J}$. A proof δ of $\langle \phi, \mathsf{proof} \rangle$, from judged assumptions Γ, is interpreted as an arrow

$$1 \otimes \mathsf{null} \xrightarrow{[\![\delta]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} \otimes \mathsf{proof}} [\![\phi]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} \otimes \mathsf{proof},$$

where δ denotes the corresponding unjudged proof tree, which can be characterized using the techniques of Lemma 4.1.

When we interpret a logic in a category, we need to make sure that the category has enough structure to interpret each constant of the language and keeps each constant of the same sort distinct. We define this condition:

Definition 5.3 (enough points) *We say that a* <u>*category C has enough points to interpret a language $\mathcal{L}$*</u> *if for each constant $c : S$, there exists an arrow $1 \xrightarrow{c} [\![S]\!]$ and for two constants $c_1 : S \neq c_2 : S$, interpreted $1 \xrightarrow{c_1} [\![S]\!]$ and $1 \xrightarrow{c_2} [\![S]\!]$, we have that for all arrows $f : [\![S]\!] \longrightarrow [\![S']\!]$ that $fc_1 \neq fc_2$.* □

We are now able to define a Kripke prestructure. The definition of a Kripke prestructure here has considerably less structure than the definition of a Kripke prestructure for the $\{\forall, \supset\}$-fragment of minimal first-order logic. This is because we wish our Kripke prestructure to provide a model for a wide class of object-logics and so we have to relax the amount of structure we require in the definition. The conditions required to provide adequate structure to capture a connective # is provided by $\rhd_{\#}$ which contains the necessary semantic information to interpret a connective in the model.

Definition 5.4 (Kripke prestructures of $\mathcal{O}_T$) *A Kripke $\mathcal{O}_T$-prestructure is a functor*

$$\mathcal{J} : [\mathcal{W}, [\mathcal{B}^{op}, \mathcal{V}]],$$

such that:

1. *$\mathcal{W}$ is a small category of worlds;*
2. *$\mathcal{B}$ is a small cartesian closed category which has enough points to interpret $\mathcal{L}$;*
3. *$\mathcal{V}$ is a (sub)category of judged values such that each $\mathcal{J}(W)(D)$, where $W \in \mathcal{W}$ and $D \in \mathcal{B}$, has a terminal object $1_{\mathcal{J}(W)(D)}$ and finite products, preserved on the nose by the inverse image functors;*
4. *The indexed category $\mathcal{J}$ is strict.* □

Structures for $\mathcal{O}_T$ extend prestructures in the usual way, providing a setting in which consequences can be interpreted.

Definition 5.5 (Kripke structures for $\mathcal{O}_T$) *Let $\mathcal{J}$ be a Kripke $\mathcal{O}_T$-prestructure. A Kripke $\mathcal{O}_T$-structure is a functor*

$$\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{B}^{op}, \mathbf{V}]],$$

where the category $\mathbf{V}$ is defined as follows:

Objects: *Categories built of out of $V = \mathcal{J}(W)(Y)$, with*

Objects: *Arrows*

$$\overline{A} \xrightarrow{f_{\overline{A},A}} A$$

in V, where $\overline{A} = A_1 \times \ldots \times A_m$;

Arrows: *Arrows*

$$(\overline{A} \xrightarrow{f_{\overline{A},A}} A) \longrightarrow (\overline{B} \xrightarrow{f_{\overline{B},B}} B)$$

are arrows $\overline{A} \xrightarrow{\mu} \overline{B}$ in V, where $\overline{B} = B_1 \times \ldots \times B_n$.

Arrows: *Functors* $\mathcal{K}_{\mathcal{J}}(W)(f)\colon\mathcal{K}_{\mathcal{J}}(W)(Y) \longrightarrow \mathcal{K}_{\mathcal{J}}(W)(X)$, *where* $X \xrightarrow{f} Y$ *in* $\mathcal{B}$ *and* $\mathcal{K}_{\mathcal{J}}(W)(X)$ *and* $\mathcal{K}_{\mathcal{J}}(W)(Y)$ *are objects of the category* $\mathbf{V}$ *at world* W. *The functor has the following properties:*

1. *The functor* $\mathcal{K}_{\mathcal{J}}(W)(f)$ *takes an object of* $\mathcal{K}_{\mathcal{J}}(W)(Y)$, *the arrow* $f_{\overline{C},C}$, *and returns an object in* $\mathcal{K}_{\mathcal{J}}(W)(X)$, *which is the arrow:*

$$\mathcal{K}_{\mathcal{J}}(W)(f)(f_{\overline{C},C}) = \prod_{i=1}^{n} \mathcal{J}(W)(f)(C_i) \stackrel{\mathcal{J}(W)(f)(f_{\overline{C},C})}{\longrightarrow} \mathcal{J}(W)(f)(C);$$

2. *The functor* $\mathcal{K}_{\mathcal{J}}(W)(f)$ *takes an arrow of* $\mathcal{K}_{\mathcal{J}}(W)(Y)$, $A_1\times\ldots\times A_m \xrightarrow{\mu} B_1\times\ldots\times B_n$, *and returns the arrow* $\nu = \mathcal{J}(W)(f)(\mu)$, *where* $C_1\times\ldots\times C_m \xrightarrow{\nu} D_1\times\ldots\times D_n$, *where* $\mathcal{J}(W)(f)(A_i) = C_i$ *for* $1 \leq i \leq m$ *and* $\mathcal{J}(W)(f)(B_j) = D_j$ *for* $1 \leq j \leq n$. $\square$

In order to define even the structure carried by models of $\mathcal{O}_T$, we require an interpretation $[\![-]\!]_{\mathcal{K}_{\mathcal{J}}}$ of the syntax of $\mathcal{O}_T$ in a structure $\mathcal{K}_{\mathcal{J}}$: the relations $R_{\#}$ and $\rhd_{\#}$ defining a connective $\#$ are predicted on the interpretations of the propositional arguments taken by $\#$.

Definition 5.6 (Kripke models of $\mathcal{O}_T$) *A* *<u>Kripke $\mathcal{O}_T$-model</u>* *consists of a pair* $\langle\mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-,\rho}_{\mathcal{K}_{\mathcal{J}}}\rangle$, *where* $\mathcal{K}_{\mathcal{J}}$ *is a Kripke structure for* $\mathcal{O}_T$ *and the partial function* $[\![-]\!]^{-,\rho}_{\mathcal{K}_{\mathcal{J}}}$ *is an interpretation of* $\mathcal{O}_T$ *in* $\mathcal{K}_{\mathcal{J}}$, *which is defined by induction on the structure of (i) terms, which are interpreted in* $\mathcal{B}$, *and (ii) judged formulae, with variables in the set* $X = \{x_1 : S_1, \ldots, x_m : S_m\}$, *which are interpreted in the fibre over* $[\![S_1]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} \times \ldots \times [\![S_m]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$.[11] *We begin by interpreting sorts:*

- *For each basic sort* S, $[\![S]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$ *is (a choice of) an object of* $\mathcal{B}$.

The terms are interpreted,exploiting the cartesian closed structure of $\mathcal{B}$, *as follows:*

- *For each variable* $x\colon S$, $[\![x]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} = [\![S]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} \xrightarrow{\rho(x)} [\![S]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$;
- *For each constant* $c\colon S$, $[\![c]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} = 1 \xrightarrow{\mu} [\![S]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$;
- *For each function symbol* $f : (S_1, \ldots, S_n) \longrightarrow S$, $[\![f]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} = (\prod_{i=1}^{i=n} [\![S_i]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}) \xrightarrow{\mu} [\![S]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$ *where* μ *is determined by the CCC structure of* $\mathcal{B}$ *in the usual way [LS86];*
- *Term-formation by application is interpreted by function space application in* $\mathcal{B}$;

The predicates symbols are also interpreted in $\mathcal{B}$:

- *For each predicate symbol* $P\colon S_1, \ldots, S_n \longrightarrow T$,

$$[\![P]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} = ([\![S]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}})^{(\prod_{i=1}^{i=n} [\![S_i]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}})},$$

the internal hom in $\mathcal{B}$;

[11] Note that if $x = \emptyset$, then $[\![\mathrm{j}(\phi(X))]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$ is an object of $\mathcal{K}_{\mathcal{J}}(w)([\![1]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}})$.

We now interpret judged formulae.

- *For each basic judgement* $\mathsf{j}(\phi)$*, we have an arrow*

$$1 \xrightarrow{f} [\![\mathsf{f}j(\phi)]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$$

  in $\mathcal{K}_{\mathcal{J}}(W)(X)$*, where* $X = [\![S_1]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}} \times \ldots \times [\![S_n]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$ *and the free variables of* ϕ *are* $x_1 : S_1, \ldots, x_n : S_n$*;*

- *For each local connective* $\#$*, satisfaction relation* $\rhd_{\#}$*,* $[\![\mathsf{j}(\#(\phi_1, \ldots, \phi_m))]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$ *is defined as follows:* $f \in \rhd_{\#}(f_1, \ldots, f_m)$ *if and only if there exists arrows*

$$f_i = [\![\mathsf{j}_i(\phi_i)]\!]^{w,\rho(\prod_{l=1}^{h_i} [\![\mathsf{K}_{i,l}(\phi_i)]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}})}_{\mathcal{K}_{\mathcal{J}}}$$

  for $1 \leq i \leq m$ *at world* w *which imply the existence of an arrow*

$$1 \xrightarrow{f} [\![\mathsf{j}(\#(\phi_1, \ldots, \phi_m))]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}};$$

- *For each global connective* $\#$ *with associated world relation* $R_{\#}$ *and satisfaction relation* $\rhd_{\#}$*,* $[\![\mathsf{j}(\#(\phi_1, \ldots, \phi_m))]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$ *is defined as follows:* $f \in \rhd_{\#}(f_1, \ldots, f_m)$ *if and only if there exist arrows*

$$f_i = [\![\mathsf{j}_i(\phi_i)]\!]^{w_i,\rho(\prod_{l=1}^{h_i} [\![\mathsf{K}_{i,l}]\!]^{w_i,\rho}_{\mathcal{K}_{\mathcal{J}}})}_{\mathcal{K}_{\mathcal{J}}}$$

  for $1 \leq i \leq m$ *where* $R_{\#}(w, w_1, \ldots, w_m)$ *imply the existence of an arrow*

$$1 \xrightarrow{f} [\![\mathsf{j}(\#(\phi_1, \ldots, \phi_m))]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}};$$

  there is an arrow

$$1 \xrightarrow{f} [\![\mathsf{j}(\#(\phi_1, \ldots, \phi_n))]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$$

  if there are arrows

$$f_i = [\![\mathsf{j}_i(\phi_i)]\!]^{w_i,\rho(\prod_{l=1}^{h_i} [\![\mathsf{K}_{i,l}(\phi_i)]\!]^{w_i,\rho}_{\mathcal{K}_{\mathcal{J}}})}_{\mathcal{K}_{\mathcal{J}}}$$

  at w_i*, for* $1 \leq i \leq m$ *such that* $R_{\#}(w, w_1, \ldots, w_m)$ *and* $f \in \rhd_{\#}(f_1, \ldots, f_m)$*;*

- *We require the following syntactic monotonicity condition: if* $[\![X]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$ *is defined, then so is* $[\![X']\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$*, for every subterm or subformula* X' *of* X*;*

- *We require the following accessibility condition: if there is an arrow* $w \xrightarrow{\alpha} w'$ *in* $\mathcal{W}$*, then* $\mathcal{J}(w')([\![\Gamma]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}) = \mathcal{J}(w)([\![\Gamma]\!]^{w',\rho}_{\mathcal{K}_{\mathcal{J}}})$ *and* $\mathcal{J}(w)([\![\Gamma]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}) = \mathcal{J}(w)([\![\Gamma]\!]^{w',\rho}_{\mathcal{K}_{\mathcal{J}}})$*.* □

Note that if $X = \emptyset$, then $[\![\mathsf{j}(\phi(X))]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$ is an object of $\mathcal{K}_{\mathcal{J}}(w)([\![1]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$. Note also the important of the syntactic monotonicity requirement: the definition of the structure in the model corresponding to each connective # relies on the definedness of the interpretations of the propositional argument taken by #.

Also the interpretation of a logical connective takes into account the most general case of a derivation of a judgement. For a Hilbert-style rule, we do not need the assumptions to be there and so the derivation becomes $1 \xrightarrow{f_i} [\![\mathsf{j}_i(\phi_1, \ldots, \phi_m)]\!]^{w,\rho}_{\mathcal{K}_{\mathcal{J}}}$.

5.3 Soundness and Completeness of Kripke Models

We can now complete our semantic view of object-logics by defining the satisfaction of a proposition in a model. The satisfaction relations $\rhd_{\#}$ are part of the semantic definitions of the connectives #. Here we are concerned with the satisfaction relation for the whole logic.

Definition 5.7 (satisfaction) *Let $\mathcal{K}_{\mathcal{J}}$ be a Kripke $\mathcal{O}_T$-model. $\mathcal{K}_{\mathcal{J}}$ satisfies $\mathsf{j}(\phi)$ at world w with respect to environment ρ,*

$$w, \rho \Vdash_T \mathsf{j}(\phi)$$

if and only if there is an arrow

$$1 \xrightarrow{f} [\![\mathsf{j}(\phi)]\!]^{w;\rho}_{\mathcal{K}_{\mathcal{J}}}.$$

If $\Gamma = \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_m(\phi_m)$, then we write $w, \rho \Vdash_T \Gamma$ if, for each $1 \leq i \leq m$, $w, \rho \Vdash_T \mathsf{j}_i(\phi_i)$. We write $w, \rho \Vdash_T (\Gamma \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$ or more commonly $w, \rho\Gamma \Vdash_T \mathsf{j}(\phi)$, if $w, \rho \Vdash_T \Gamma$ implies $w, \rho \Vdash_T \mathsf{j}(\phi)$. □

We prove a result about monotonicity of satisfaction.

Lemma 5.8 (monotonicity of satisfaction) *Let $\mathcal{O}_T$ be an object-logic and $\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-;\rho}_{\mathcal{K}_{\mathcal{J}}} \rangle$ be a Kripke $\mathcal{O}_T$-model. If $w, \rho \Vdash_T \mathsf{j}(\phi)$, $w \xrightarrow{\alpha} w'$ then $w', \rho \Vdash_T \mathsf{j}(\phi)$.*

Proof By the accessibility condition of Definition 5.6, $w \xrightarrow{\alpha} w'$ gives us the existence of an arrow $1 \xrightarrow{f'} [\![\mathsf{j}(\phi)]\!]^{w',\rho}_{\mathcal{K}_{\mathcal{J}}}$ and thus $w', \rho \Vdash_T \mathsf{j}(\phi)$. □

We refer the reader to the work of Aczel [Acz80], where he presents a logical system which is very similar in style to the one presented in Definitions 4.4,3.2 to 3.4 and 4.5 except that it is less general. He only has one judgement true and does not treat logical connectives in general. In fact, his definition of a Frege structure can be considered as a logic according to our definition, since we can just add the extra the extra logical connectives λ and *app* to classical logic. The idea of not being able to define truth internally within a language and the split between the logic and the metalogic can be captured more explicitly in the setup we are using. This link between the work of Aczel and our own is expressed by the comment in the final section that the type theory of Martin-Löf, which we are using here can be modeled in a suitably chosen Frege structure and we are providing the opposite view that Frege structures can be expressed in Martin-Löf type theory. The lambda structure of Aczel can be seen to correspond to the $\lambda\Pi$-calculus that we use as a metalogic. The definition of an iterative set, the smallest collection of objects χ such that every set of χ's is in χ can also be carried over into our setting to allow us to have a concept of an iterative set.

We prove a general soundness and completeness result for a judged proof system. We will return to these results later on and consider when they hold for specific classes of object-logics and so obtain some representation results.

Lemma 5.9 (soundness of $\mathcal{O}_T$ for $\Vdash_T$) *Let $\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-;\rho}_{\mathcal{K}_{\mathcal{J}}} \rangle$ be a Kripke $\mathcal{O}_T$-model and let w be any world in this model. If $\Gamma \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$, then $w, \rho, \Gamma \Vdash_T \mathsf{j}(\phi)$.*

Proof The proof proceeds by induction over the structure of proofs in the object-logic.

We begin with the case where we have an axiom. Here we have $\mathsf{j}(\phi) \in \Gamma$ and so by induction hypothesis, we have an arrow $1 \xrightarrow{f} [\![\mathsf{j}(\phi)]\!]^{w,\rho}_{\mathcal{K}_\mathcal{J}}$.

We now look at an introduction rule. We have that $w, \rho, \Gamma' \Vdash_T \mathsf{j}_i(\phi_i)$ for $1 \leq i \leq n$, where Γ' is Γ plus any hypotheses. By the induction hypothesis, we have arrows $[\![\mathsf{j}_i(\phi_i)]\!]^{w,\rho(\prod_{l=1}^{h_i} [\![\mathsf{K}_{i,l}(\phi_i)]\!]^{w,\rho}_{\mathcal{K}_\mathcal{J}})}_{\mathcal{K}_\mathcal{J}}$ for each $1 \leq i \leq n$. By Definition 5.6, we have that there exists an arrow $1 \xrightarrow{f} [\![\mathsf{j}(\#(\phi_1, \ldots, \phi_n))]\!]^{w,\rho}_{\mathcal{K}_\mathcal{J}}$ with $f \in \rhd_\#(f_1, \ldots, f_n)$ and thus $w, \rho, \Gamma \Vdash_T \mathsf{j}(\#(\phi_1, \ldots, \phi_n))$.

Finally, we consider an elimination rule. We have that $w, \rho, \Gamma \Vdash_T \mathsf{k}(\#(\phi_1, \ldots, \phi_n))$ and $w, \rho, \Gamma' \Vdash_T \mathsf{j}(\phi)$, where Γ' is Γ plus any hypotheses. By the induction hypothesis, we have arrows $1 \xrightarrow{f} [\![\mathsf{k}(\#(\phi_1, \ldots, \phi_n))]\!]^{w,\rho}_{\mathcal{K}_\mathcal{J}}$ and $[\![\mathsf{j}(\phi)]\!]^{w,\rho([\![\mathsf{K}_i]\!]^{w,\rho}_{\mathcal{K}_\mathcal{J}})}_{\mathcal{K}_\mathcal{J}}$ for $1 \leq i \leq n$. Since we are in a Cartesian closed category, we obtain $1 \longrightarrow [\![\mathsf{j}(\phi_1, \ldots, \phi_n)]\!]^{w,\rho}_{\mathcal{K}_\mathcal{J}}$ and thus $w, \rho, \Gamma \Vdash_T \mathsf{j}(\phi)$. □

To prove completeness, we construct a model where there exists a world w_0 such that $\Gamma \not\vdash_T \mathsf{j}(\phi)$ implies $w_0, \rho, \Gamma \not\Vdash_T \mathsf{j}(\phi)$. We begin by constructing a categorical structure which we will prove is a Kripke $\mathcal{O}_T$-structure and then give it an interpretation which gives the result.

Going back to our motivation for constructing Kripke models, we recall that we wish to use Kripke partiality to determine which proof variables are defined at a world and thus which formulae have proofs. We take worlds to be collections of proof variables. Thus we have a proof is true at a particular world if and only if it's proof variable is at that world or all the proof variables in it's assumptions hold at that world.

Informally, our term model has the following structure; a base category which captures terms, here the objects are products of sorts and the arrows are realizations, and fibres over the base category which will capture consequence. The objects of each fibre will be proofs and the arrows proof transformations.

We begin by defining the base category, $\mathcal{B}(A)$.

Definition 5.10 *Let A be an alphabet. The <u>(base) category $\mathcal{B}(A)$ of sorts and realizations</u> is defined as follows:*

Objects: Products of sorts $\prod_{i=1}^m S_i$ such that S_i correspond to the syntactic categories of the language generated by the alphabet;

Arrows: Realizations $X \xrightarrow{\langle t_1, \ldots, t_n \rangle} Y$ such that $x_1 : S_1, \ldots, x_m : S_i \vdash_{\mathcal{O}_T} t_i : T_i$ for all $1 \leq i \leq n$;

- *Identities, written id_X, are $S_1 \times \ldots \times S_m \xrightarrow{\langle x_1, \ldots, x_m \rangle} S_1 \times \ldots \times S_m$.*
- *Composition is defined as follows: if $\sigma = X \xrightarrow{\langle t_1, \ldots, t_n \rangle} Y$ and $\rho = Y \xrightarrow{\langle s_1, \ldots, s_p \rangle} Z$, then $\sigma; \rho = X \xrightarrow{\langle s_1[t_j/y_j]_{j=1}^n, \ldots, s_p[x_j/y_j]_{j=1}^n \rangle} Z$.* □

Here variables are understood as a special case of the identity. A variable is just the identity on a single sort, *i.e.* $S \xrightarrow{\langle x \rangle} S$.

Before we define out Kripke prestructure, we define a strict indexed category over the base category, which we will use to construct the Kripke prestructure.

Definition 5.11 *We define inductively a* <u>*strict indexed category* $\mathcal{E}(A)$</u> *over the base category* $calB(A)$*,*

$$\mathcal{E}(A){:}\mathcal{B}(A)^{op} \longrightarrow \mathcal{C},$$

where $\mathcal{C}$ *denotes the category of small categories and functors, as follows:*

- *For each product of sorts,* $S = S_1 \times \ldots \times S_n$ *in* $\mathcal{B}(A)$*, the category* $\mathcal{E}(A)(S)$ *is defined as follows:*

Objects: *Judged formulae* $\mathsf{j}(P(t_1, \ldots, t_n))$ *such that* $x_1{:}S_1, \ldots, x_n{:}S_n \vdash_{\mathcal{O}_T} \mathsf{j}(P(t_1, \ldots, t_n))$*;*

Arrows: $\mathsf{j}(P(t_1, \ldots, t_n)) \xrightarrow{t} \mathsf{j}(Q(r_1, \ldots, r_m))$*, where the arrow* t *is such that* $S_1 \times \ldots \times S_n \times T \xrightarrow{\langle x,t \rangle} S_1 \times \ldots \times S_n \times U$ *in* $\mathcal{B}(A)$*, where* $\mathsf{j}(P(t_1, \ldots, t_n)){:}T$ *and* $\mathsf{j}(W(r_1, \ldots, r_p)){:}U$*; if* $\mathsf{j}(P(t_1, \ldots, t_n)) \xrightarrow{t} \mathsf{j}(Q(r_1, \ldots, r_m))$ *and* $\mathsf{j}(Q(r_1, \ldots, r_m)) \xrightarrow{u} \mathsf{j}(R(u_1, \ldots, u_p))$*, then* $\mathsf{j}(P(t_1, \ldots, t_n)) \xrightarrow{t;u} \mathsf{j}(R(u_1, \ldots, u_p))$ *is given by* $\mathsf{j}(P(t_1, \ldots, t_n)) \xrightarrow{u[t/U]} \mathsf{j}(R(u_1, \ldots, u_p))$*, where* u *is such that* $S_1 \times S_n \times V \xrightarrow{\langle x,u \rangle} S_1 \times \ldots \times S_n \times U$ *is an arrow in* $\mathcal{B}(A)$*;*

- *For each* $Y \xrightarrow{\sigma} X$ *in* $\mathcal{B}(A)$*,* $\mathcal{E}(A)(Y \xrightarrow{\sigma} X)$ *is a functor* $\mathcal{E}(A)(X) \xrightarrow{\sigma^*} \mathcal{E}(A)(Y)$ *given by* $\sigma^*\mathsf{j}(P(t_1, \ldots, t_n)) = \mathsf{j}(P(t_1, \ldots, t_n))[y_i/x_i]_{i=1}^{m}$*, where* $x_1 : S_1, \ldots, x_n : S_n$ *and* $y_1 : T_1, \ldots, y_m : T_m$ *and* $\sigma^*\mathsf{j}(P(t_1, \ldots, t_n)) \longrightarrow \mathsf{j}(Q(r_1, \ldots, r_p) = \sigma^*\mathsf{j}(P(t_1, \ldots, t_n)) \longrightarrow \sigma^*\mathsf{j}(Q(r_1, \ldots, r_p))$*.* □

Before we construct our Kripke prestructure, we have to define our category of worlds. To fully exploit the Kripke partiality of the model we are constructing, we take worlds to be collections of proof variables, this will then tell us what we can and can not prove in the fibres over the base category.

Definition 5.12 *We define a* <u>*category of proof variables,* $\mathcal{W}$</u>*, as follows:*

Objects: *Sets of proof variables* $\{\zeta_1, \ldots, \zeta_n\}$*;*

Arrows: *Given by subset inclusion.* □

For a proof $\mathsf{j}(\phi)$ to be defined in a suitable fibre, we need to be over the world containing at least the proof variable ζ which corresponds to $\mathsf{j}(\phi)$. This is the heart of our use of Kripke partiality.

Over each world X, we place a base category $\mathcal{B}(A)$. We define a functor $\mathcal{T}(A) : [\mathcal{W}, [\mathcal{B}(A)^{op}, \mathcal{V}]$, which will be our Kripke prestructure. Given an object $S = S_1 \times \ldots \times S_n$ of $\mathcal{B}(A)$, the functor $\mathcal{T}(A)(W)$ gives us the following category:

$$\mathcal{T}(A)(W)(S) = \begin{cases} \textit{Objects} & : \text{Judged formulae } \mathsf{j}(\phi) \text{ such that} \\ & \; x_1{:}S_1, \ldots, x_n{:}S_n \vdash_{\mathcal{O}_T} \mathsf{j}(\phi) \text{ and } \zeta \in W; \\ \textit{Arrows} & : \mathcal{E}(A)(X) \text{ arrows;} \end{cases}$$

where ζ is the proof variable corresponding to $\mathsf{j}(\phi)$.

From this, we build the Kripke structure, following the construction defined in Definition 5.5. This construction will give the term model. ***Need to fill in the details that this thing is indeed a functor category.***

The interpretation required to show that this is a Kripke $\mathcal{O}_T$-model is the obvious one, we interpret the proof object $(X)\ \Gamma \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$ as an arrow $1 \longrightarrow [\![\mathsf{j}(\phi)]\!]^{w,\rho}_{\mathcal{K}_\mathcal{J}}$ in the fibre $\mathcal{K}_\mathcal{J}(W)(T)$, where $T = S_1 \times \ldots \times S_n$ and $X = x_1 : S_1, \ldots, x_n : S_n$ and $\zeta_i \in W$ for each ζ_i corresponding to a proof of $\mathsf{j}_i(\phi_i) \in \Gamma$.

From the definition of satisfaction, we have that $\Gamma \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$ iff $w, \rho, \Gamma \Vdash_T \mathsf{j}(\phi)$. If we take $w_0 = \emptyset$, then we have that $\Gamma \not\vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$ implies $w_0, \rho, \Gamma \not\Vdash_T \mathsf{j}(\phi)$, thus we have the following model existence result.

Lemma 5.13 (model existence) *There exists a Kripke $\mathcal{O}_T$-model $\langle \mathcal{K}_\mathcal{J}, [\![-]\!]^{-,\rho}_{\mathcal{K}_\mathcal{J}} \rangle$ with a world w_0 such that if there exists a proof $\delta : (\Gamma \not\vdash_{\mathcal{O}_T} \mathsf{j}(\phi))$, then*
$w_0, \rho, \Gamma \not\Vdash_T \mathsf{j}(\phi)$.

Proof See the previous construction. □

The term model we have constructed above has the correct property to allow us to prove completeness

Theorem 5.14 (completeness) $\delta : ((X)\ \Gamma \vdash_{\mathcal{O}_T} \mathsf{j}(\phi))$ *if and only if* $w, \rho, \Gamma \Vdash_T \mathsf{j}(\phi)$.

Proof

(only if) This is soundness, Lemma 5.9.

(if) Suppose $\delta : ((X)\Gamma \vdash_{\mathcal{O}_T} \mathsf{j}(\phi))$, then Lemma 5.13 yields a contradiction. □

6 Judgements-as-types

6.1 Introduction

Having described a judged object-logic, we now describe how to encode this logic into a signature in the $\lambda\Pi$-calculus and thus how it is represented in a logical framework. The encoding is the judgements-as-types correspondence, this is a generalization of the propositions-as-types correspondence discussed in I and well known in the literature.

In particular, following Kant [Kan00] and Martin-Löf [ML82], we consider a metatheory of object-logics in which the basic units manipulated by rules of inference are judged propositions rather than the bare propositions. This is the judgements-as-types correspondence.

We describe how to take a proof-object $\delta : ((X)\Delta \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$ and translate it to a $\lambda\Pi$-term $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$. We encode the inference rules of the object-logic as constants in the signature of $\lambda\Pi$ and then construct the term in $\lambda\Pi$. The rules of $\lambda\Pi$ allow us to mimic applications of inference rules in the object-logic and so allow us to construct terms corresponding to proof-objects in the object-logic.

Once we have the encoding defined, we are able to talk about the properties of the map from the syntax of the object-logic to the syntax of $\lambda\Pi$ and it's properties. In particular, we are able to define full and faithful encodings, as well as whether or not an encoding is uniform.

The judgements-as-types correspondence is not the only choice of representation mechanism, worlds-as-parameters is another paradigm we could have employed. The judgements-as-types correspondence has a more solid philosophical background and is a generalization of the propositions-as-types correspondence which is the basis for the majority of proof theoretic study. The use of a type of worlds which is not constructed seems to be less natural than the judgements-as-types correspondence.

We look at the judgements-as-types correspondence in two senses; the first is syntactic and provides us with an encoding mechanism, the second is semantic and provides us with a morphism of models. These two viewpoints are related in the following way: the encoding function will induce the morphism of models and properties of the encoding function will be mirrored in the morphism of models. So a bijective encoding function will induce an isomorphism of models and a surjective encoding function will induce an epimorphism of models. We begin by looking at the syntactic judgements-as-types before setting up the necessary categorical framework to describe the morphism and finally exploring the relationship between the encoding and the induced morphism.

6.2 Syntactic judgements-as-types: Encoding logics in LF

We recall that an object-logic $\mathcal{O}_T$ is an alphabet $A = (S, V, E, C, J)$, made up of symbols, variable symbols, expressions, connectives and judgements and a collection of inference rules. We encode all of these as constants contained in the signature of $\lambda\Pi$. The idea behind the encoding of the an inference rule is the following; given a rule

$$\cfrac{\begin{matrix}\vdots & & \vdots\\ \mathsf{j}_1(\phi_1) & \cdots & \mathsf{j}_n(\phi_n)\end{matrix}}{\mathsf{j}(\phi)}$$

we think of it being quantified over all ϕ and each of the derivations of the judgements being given by arrows and the combination of all the derivations being arrows, thus:

$$\prod \phi \quad \cfrac{\begin{matrix}\vdots\cup & & \vdots\cup\\ \mathsf{j}_1(\phi_1) & \overset{\supset}{\cdots} & \mathsf{j}_n(\phi_n)\end{matrix}}{\mathsf{j}(\phi)} \cup$$

With the $\prod$ corresponding to Π in the $\lambda\Pi$-calculus and $\supset$ corresponding to $\longrightarrow$. Usually, the assumptions are understood as being joined by conjunction rather than implication, in the sense of if we have this assumption and this assumption ..., then we have the conclusion. With a bit of thought, it can be seen that using implication here instead of conjunction will capture enough information to allow us to represent the rule in a logical framework. Essentially, we are looking at an uncurried version of the rule. If we wanted to use conjunction here, we would have to use a different type theory as our language since $\lambda\Pi$ does not have conjunction. There are dependent type theories which have conjunction but using them is beyond the scope of this paper since we are interested in studying representation rather than the language used in the logical framework. We make this intuitive notion concrete in the following definition.

Definition 6.1 (encodings) *Let $\mathcal{O}_T$ be an object-logic, presented either* (1) *as a Hilbert-type system, or* (2) *as a natural deduction system. An encoding of $\mathcal{O}_T$ in LF is determined by a pair $\epsilon = (\epsilon_s, \epsilon_\mathsf{j})$ of functions in which ϵ_s maps the syntax of $\mathcal{O}_T$ to $\lambda\Pi$-terms and ϵ_j maps the proofs of $\mathcal{O}_T$ to $\lambda\Pi$-terms. An alphabet $A = (S, V, E, C, \mathsf{J})$ is encoded as follows:*

- *Let $s \in S \backslash V$: if s has arity 0, then s is encoded by a constant $s : \mathrm{Type} \in \Sigma_{\mathcal{O}_T}$ or by a constant $s : s' \in \Sigma_{\mathcal{O}_T}$, where $s' : \mathrm{Type}$ is declared to the left of $s : s'$, if s has arity m, then s is encoded in LF either by a constant*

$$s : (s_1 \longrightarrow \cdots \longrightarrow s_m) \longrightarrow \mathrm{Type} \in \Sigma_{\mathcal{O}_T}$$

  or by a constant

$$s : (s_1 \longrightarrow \cdots \longrightarrow s_m) \longrightarrow s' \in \Sigma_{\mathcal{O}_T}$$

  where $s' : \mathrm{Type} \in \Sigma_{\mathcal{O}_T}$ is declared to the left of $s : s'$ in $s' : \mathrm{Type} \in \Sigma_{\mathcal{O}_T}$.

- *Let $s \in V$: a variable $x \in V_s$ (of arity 0) is encoded in LF by a declaration $x : s \in \Gamma$, where Γ is the context within which the declaration is required.*

- *Let $e \in E$: if e has arity m, then e is encoded in LF by a constant*

$$i_1 \longrightarrow i_j \longrightarrow i_m \longrightarrow i$$

  where i_j is a syntactic category.

- *Each judgement symbol j of arity $(s_1, \ldots, s_m)$ is encoded in LF by a constant*

$$\mathsf{j} : (s_1 \longrightarrow \ldots \longrightarrow \ldots s_m) \longrightarrow \mathrm{Type}$$

Application in $\mathcal{L}$ is encoded in LF by application in $\lambda\Pi$; abstraction in $\mathcal{L}$ is encoded in LF by abstraction in $\lambda\Pi$.

1. *Hilbert-type systems:*

    - *An axiom Ax of the form $\mathsf{j}(e(\phi_1, \ldots, \phi_m))$, is encoded in LF by a constant of the form*

$$Ax : \Pi\phi_1 : o_1 \, . \, \ldots \, . \, \Pi\phi_m \, . \, \mathsf{j}(e(\phi_1, \ldots, \phi_m))$$

      of $\lambda\Pi$, where each o_i is one of the syntactic categories of propositons distinguised in A;

- *A rule R of the form*

$$\frac{\mathsf{j}_1(e_1,\ldots,e_n)\ldots\mathsf{j}_m(e_1,\ldots,e_n)}{\mathsf{j}(e_1,\ldots,e_n)}\,R$$

in which, each e_i, is of the form $e_i(\phi_1,\ldots,\phi_m)$, is encoded in LF by a constant of the form

$$\begin{array}{rcl} R & : & \Pi\phi_1\!:\!o_1\,.\,\ldots\,.\,\Pi\phi_m\!:\!o_m\,.\,\mathsf{j}_1(e_1,\ldots,e_n)\longrightarrow \\ & & (\mathsf{j}_2(e_1,\ldots,e_n)\longrightarrow(\ldots\longrightarrow\mathsf{j}(e_1,\ldots,e_n))) \end{array}$$

of $\lambda\Pi$;

2. *Natural deduction systems:*

- *An axiom Ax of the form $\mathsf{j}(e(\phi_1,\ldots,\phi_m))$, is in LF by a constant of the form*

$$Ax\!:\!\Pi\phi_1\!:\!o_1\,.\,\ldots\,.\,\Pi\phi_m\!:\!o_m\,.\,\mathsf{j}(e(\phi_1,\ldots,\phi_m))$$

of $\lambda\Pi$, where each o_i is one of the syntactic categories of propositions distinguished in A;

- *The #-introduction rule schema of the form*

$$\begin{array}{ccccc} & [K_{i,1}] & \cdots & [K_{i,h_i}] & \\ \vdots & & \vdots & & \vdots \\ J_1 & \cdots & J_i & \cdots & J_p \end{array}$$
$$\frac{\qquad\qquad\qquad\qquad\qquad}{J}\,\#\,I$$

in which $K_{i,h_i}=\mathsf{k}_{i,h_i}(e_1,\ldots,e_n), J_i=\mathsf{j}_i(e_1,\ldots,e_n)$, for $1\leq i\leq p$, and $J=\mathsf{j}(\#(e_1,\ldots,e_n))$, and where each e_i is of the form $e_1(\phi_1,\ldots,\phi_m)$, is encoded in LF by a constant of the form

$$\begin{array}{rcl} \#I & : & \Pi\phi_1\!:\!o_1,\,\ldots\,.\,\Pi\phi_m\!:\!o_m\,.\,(K_{1,1}\longrightarrow\ldots\longrightarrow K_{1,h_1}\longrightarrow J_1) \\ & & \ldots)\longrightarrow(\ldots\longrightarrow J)) \end{array}$$

of $\lambda\Pi$;

- *The #-elimination rule schema of the form*

$$\begin{array}{cccc} & [\Gamma_1] & & [\Gamma_p] \\ & \vdots & \cdots & \vdots \\ \mathsf{k}(\#(e_1,\ldots,e_n)) & J & & J \end{array}$$
$$\frac{\qquad\qquad\qquad\qquad\qquad}{J}\,\#\,E$$

in which the p minor premisses of the form J are derived from the set of assumptions Γ_i, for $1\leq i\leq p$, is encoded in LF by a constant of the form

$$\begin{array}{rcl} \#\,E & : & \Pi\phi_1\!:\!o_m\,.\,\ldots\,.\,\Pi\phi_m\!:\!o_m\,.\,(\mathsf{k}(\#(e_1,\ldots,e_m))\longrightarrow \\ & & ((\Gamma_1\longrightarrow J)\longrightarrow\ldots\longrightarrow(\Gamma_p\longrightarrow J)\ldots\longrightarrow J) \end{array}$$

of $\lambda\Pi$(here, for brevity, we elide the currying of the Γ_i's).

The functions ϵ_s and ϵ_j are then determined as follows:

1. *We define ϵ_s inductively over the syntax of $\mathcal{O}_T$.*
   - $\epsilon_s(x) = x$, *where x is a variable;*
   - $\epsilon_s(ee_1 \dots e_n) = \epsilon_s(e) \dots \epsilon_s(e_n)$, *where each e_i is an expression;*
   - $\epsilon_s(x_1, \dots, x_m)e = \lambda x_1 : s_1 . \dots . \lambda x_m : s_m . \epsilon_s(e)$, *where each $x_1 : s_1$ is a variable x_i belonging to the syntactic category s_i.*
2. *We define ϵ_j inductively over the proof expressions of $\mathcal{O}_T$.*
   - *Given a proof variable ξ for the proof of* $\mathsf{j}(e)$, $\epsilon_j(\xi) = \xi : \mathsf{j}(e)$
   - *Given an application of a rule $\#\, I$ or $\#\, E$ to proofs* $\Pi_1, \dots \Pi_n$, *we have* $\epsilon_j(\#\, I(\Pi_1, \dots, \Pi)n)) = \epsilon_j(\#\, I)\epsilon_j(\Pi_1) \dots \epsilon_j(\Pi_n)$. □

We extend our definition to theories.

Definition 6.2 (encoding of theories) *We extend the encoding function ϵ_j to cover the new inference rules of the theory.* □

Theories in which induction is restricted, *e.g.* Σ_0^1-induction, present a further syntactic challenge for the definition of an object-logic. In an informal meta-theory, we simply restrict our attention to induction formulae of the appropriate class, *e.g.* Σ_0^1-induction. However, our object-logics are intended as the

The next two definitions are closely related to definitions in [HST94] and [Sim93].

Definition 6.3 (full,faithful encodings) *An encoding of an object-logic $\mathcal{O}_T$ is full if the judged consequence $\delta : (X)\mathsf{j}_1(\phi_1), \dots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$ in the object-logic implies $\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \dots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$ and faithful if $\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \dots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$ implies $\delta : (X)\mathsf{j}_1(\phi_1), \dots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$. We call a full and faithful encoding adequate.* □

Definition 6.4 (encoding uniformly) *An encoding of an object-logic $\mathcal{O}_T$ is uniform if the encoding function is surjective. We say an encoding is uniformly full if the encoding is both uniform and full, similarly we say an encoding is uniformly faithful if the encoding is both uniform and faithful.* □

In [HST94], the term 'uniform encoding' is used to denote a stronger property than our 'uniformly faithful', requiring a quantification over all possible signatures $\Sigma_{\mathcal{L}}$ which 'present' the logic $\mathcal{L}$. The details of the approach to the representation of logics described in [HST94] are beyond our present scope.

From now on we will mainly be concerned about uniformly full encodings, since we want every term in $\lambda\Pi$ to correspond to a proof in the logic. Any weaker condition than this would mean that the logical framework would not be fully representing the logic. While we will have encoding functions which are bijections, these will be special cases rather than the norm. We are interested in full encodings because we the proof of fullness is just an induction proof and we are more concerned about proving fullness semantically.

6.3 Semantic judgements-as-types: a morphism of models

Finally, we are now able to set up we shall call the judgements-as-types morphsim, an indexed morphism between suitable Kripke models, induced by the judgements-as-types correspondence. As discussed above, we are concerned with the cases where the encoding function is surjective or bijective. In these cases, the morphism will be an epimorphism or isomorphism respectively.

We begin by recalling briefly several key definitions from [PP06]. We keep our presentation shorter than that given in [PP06] and the reader is referred to this paper if more details are required. As in [PP06], we notice that the definition of a Kripke model of the λΠ-calculus is similar to that of a Kripke model of the object-logic. We exploit this similarity by defining a morphism between them. We recall the definition of a generalized indexed functor.

Definition 6.5 (generalized indexed functors) *Let $\mathcal{F} : \mathcal{A}^{op} \longrightarrow C$ and $\mathcal{G} : \mathcal{B}^{op} \longrightarrow \mathcal{C}$ be strict indexed categories. A generalized indexed functor from $\mathcal{F}$ to $\mathcal{G}$ consists of a triple $\tau = (\beta, \gamma, \epsilon)$ in which $\beta : \mathcal{A} \longrightarrow \mathcal{B}$ and $\gamma : \mathcal{C} \longrightarrow \mathcal{D}$ be functors and $\epsilon : \mathcal{F};\gamma \Longrightarrow \beta^{op};\mathcal{G}$ is a natural transformation. We extend this definition to functors $\mathcal{F} : [\mathcal{W}, [\mathcal{A}^{op}, \mathcal{C}]]$ and $\mathcal{G} : [\mathcal{X}, [\mathcal{B}^{op}, \mathcal{D}]]$. A generalized indexed functor from $\mathcal{F}$ to $\mathcal{G}$ consists of a quadruple*

$$\tau = (\alpha, \beta, \gamma, (\epsilon_w)_{w \in \mathcal{W}}),$$

where $\alpha : \mathcal{W} \longrightarrow \mathcal{X}$, $\beta : \mathcal{A} \longrightarrow \mathcal{B}$, $\gamma : \mathcal{C} \longrightarrow \mathcal{D}$ be functors and, for each object w of $\mathcal{W}$, $\epsilon_w : \mathcal{F}(w);\gamma \Longrightarrow \beta^{op};\mathcal{G}(\alpha(w))$ is a natural transformation such that for $f : v \longrightarrow w$, the diagram

$$\begin{array}{ccccc} v & & \mathcal{F}(v);\gamma & \xRightarrow{\epsilon_v} & \beta^{op};\ \mathcal{G}(\alpha(v)) \\ f \Big\downarrow & \mathcal{F}(f) & \Big\Downarrow & & \Big\Downarrow \ \mathcal{G}(\alpha(f)) \\ w & & \mathcal{F}(w);\gamma & \xRightarrow[\epsilon_w]{} & \beta^{op};\ \mathcal{G}(\alpha(w)) \end{array}$$

commutes. □

We define the notion of an indexed epimorphism and indexed isomorphism since we will need these to describe cases of our morphism. Both of these definitions are straightforward.

Definition 6.6 (indexed epimorphisms) *A generalized indexed functor $\tau = (\beta, \gamma, \epsilon)$ is a generalized indexed epimorphism if β and γ are epimorphisms and ϵ is a natural epimorphism. A generalized indexed functor $\tau = (\alpha, \beta, \gamma, (\epsilon_w)_w)$ is a generalized epimorphism if α, β and γ are epimorphisms and each ϵ_w is a natural epimorphism.* □

Definition 6.7 (indexed isomorphisms) *A generalized indexed functor $\tau = (\beta, \gamma, \epsilon)$ is a generalized indexed isomorphism if β and γ are epimorphisms and ϵ is a natural isomorphism. A generalized indexed functor $\tau = (\alpha, \beta, \gamma, (\epsilon_w)_w)$ is a generalized isomorphism if α, β and γ are isomorphisms and each ϵ_w is a natural isomorphism.* □

Now we define the category where all the models live.

Definition 6.8 (category of models) *We define the* <u>*category $\mathcal{M}$ of models*</u> *as follows:*

Objects: each object of $\mathcal{M}$ is either a Kripke $\Sigma_{\mathcal{O}_T}$-$\lambda\Pi$-model $\mathcal{K}_{\mathcal{J}}$ or a Kripke $\mathcal{O}_T$-model $\mathcal{R}_{\mathcal{S}}$[12]*;*

Arrows: there are four cases:

1. *An arrow*
$$\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}} \rangle \xrightarrow{h} \langle \mathcal{K}'_{\mathcal{J}'}, [\![-]\!]^{-}_{\mathcal{K}'_{\mathcal{J}'}} \rangle$$
is given by an indexed functor $\langle \alpha, \beta, \gamma, (\epsilon_w)_w \rangle : \mathcal{K}_{\mathcal{J}} \longrightarrow \mathcal{K}'_{\mathcal{J}'}$ such that if $\alpha w = w'$, then $h([\![X]\!]^{w}_{\mathcal{K}_{\mathcal{J}}}) = [\![X]\!]^{w'}_{\mathcal{K}'_{\mathcal{J}'}}$;
2. *An arrow*
$$\langle \mathcal{R}_{\mathcal{S}}, [\![-]\!]^{-,\rho}_{\mathcal{R}_{\mathcal{S}}} \rangle \xrightarrow{h} \langle \mathcal{R}'_{\mathcal{S}'}, [\![-]\!]^{-,\rho}_{\mathcal{R}'_{\mathcal{S}'}} \rangle$$
is given by a generalized indexed functor $(\alpha, \beta, \gamma, (\epsilon_w)_w) : \mathcal{R}_{\mathcal{S}} \longrightarrow \mathcal{R}'_{\mathcal{S}'}$, such that if $\alpha x = x'$, then $h([\![X]\!]^{x,\rho}_{\mathcal{R}_{\mathcal{S}}}) = [\![X]\!]^{x',\rho}_{\mathcal{R}'_{\mathcal{S}'}}$;
3. *An arrow*
$$\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}} \rangle \xrightarrow{h} \langle \mathcal{R}_{\mathcal{S}}, [\![-]\!]^{-,\rho}_{\mathcal{R}_{\mathcal{S}}} \rangle$$
is given by a generalized indexed functor $(\alpha, \beta, \gamma, (\epsilon_w)_w) : \mathcal{K}_{\mathcal{J}} \longrightarrow \mathcal{R}_{\mathcal{S}}$ such that if $\alpha w = x$, then $h([\![\{\!\{X\}\!\}_T]\!]^{w}_{\mathcal{K}_{\mathcal{J}}}) = [\![X]\!]^{x,\rho}_{\mathcal{R}_{\mathcal{S}}}$;
4. *An arrow*
$$\langle \mathcal{R}_{\mathcal{S}}, [\![-]\!]^{-,\rho}_{\mathcal{R}_{\mathcal{S}}} \rangle \xrightarrow{h} \langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}} \rangle$$
is given by a generalized indexed functor $(\alpha, \beta, \gamma, (\epsilon_w)_w) : \mathcal{R}_{\mathcal{S}} \longrightarrow \mathcal{K}_{\mathcal{J}}$ such that if $\alpha x = w$, then $h([\![X]\!]^{x,\rho}_{\mathcal{R}_{\mathcal{S}}}) = [\![\{\!\{X\}\!\}_T]\!]^{w}_{\mathcal{K}_{\mathcal{J}}}$.

where $\{\!\{X\}\!\}_T$ is the result of applying the encoding function ϵ to X. □

We show that $\mathcal{M}$ is well-defined.

Lemma 6.9 (M is well-defined) *The category $\mathcal{M}$ described in Definition 6.8 is well-defined.*

Proof We just need to check that arrows compose, since the identity arrow is given by a special case of one or two depending on whether we are dealing with a Kripke $\lambda\Pi$-model or an $\mathcal{O}_T$-model. As long as the encoding function is surjective, we can always find the inverse function needed to define morphisms of type three. This means that we can always compose arrows and obtain arrows of the category. □

We are almost ready to prove the existence of a morphism of models induced by the judgements-as-types correspondence. We just need one final piece of terminology. Let

[12]Notation: to avoid confusion, we use $\mathcal{R}_{\mathcal{S}}$, etc., for object-logic models.

$\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}}\rangle$ and $\langle \mathcal{R}_{\mathcal{S}}, [\![-]\!]^{-,\rho}_{\mathcal{R}_{\mathcal{S}}}\rangle$ be objects of $\mathcal{M}$. A morphism h between them is an epimorphism (isomorphism) of models if it is an indexed epimorphism (isomorphism) such that, abusing notation by allowing X to range over all of the syntax of $\mathcal{O}_T$ and suppressing information about worlds,

$$h([\![\{\!\!\{X\}\!\!\}_T]\!]_{\mathcal{K}_{\mathcal{J}}}) = [\![X]\!]_{\mathcal{R}_{\mathcal{S}}}.$$

Proposition 6.10 (judgements-as-types epimorphism) *Let $\mathcal{O}_T$ be an object-logic as defined in Definition 4.9 and let $\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}}\rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathbf{V}]]$, be a Kripke $\Sigma_{\mathcal{O}_T}$-$\lambda\Pi$-model, where $\Sigma_{\mathcal{O}_T}$ is the $\lambda\Pi$-signature in judgements-as-types correspondence with $\mathcal{O}_T$, where the encoding is uniformly faithful. Then there is a Kripke $\mathcal{O}_T$-model, $\langle \mathcal{R}_{\mathcal{S}}, [\![-]\!]^{-,\rho}_{\mathcal{R}_{\mathcal{S}}}\rangle$, where $\mathcal{R}_{\mathcal{S}} : [\mathcal{X}, [\mathcal{E}^{op}, \mathbf{U}]]$, together with an (indexed) epimorphism of models,*

$$h = (\alpha, \beta, \gamma, (\epsilon_w)_w) : \mathcal{K}_{\mathcal{J}} \longleftarrow \mathcal{R}_{\mathcal{S}},$$

induced by the judgements-as-types correspondence, which respects the interpretation of syntax. Specifically, abusing notation by allowing X to range over all of the syntax of $\mathcal{O}_T$ and suppressing information about worlds, if $[\![X]\!]^{x}_{\mathcal{R}_{\mathcal{S}}}$ and $[\![\{\!\!\{X\}\!\!\}_T]\!]^{W}_{\mathcal{K}_{\mathcal{J}}}$ are defined, then

$$h([\![\{\!\!\{X\}\!\!\}_T]\!]_{\mathcal{K}_{\mathcal{J}}}) = [\![X]\!]_{\mathcal{R}_{\mathcal{S}}}.$$

Proof

- We take $\mathcal{X} = \mathcal{W}$ with $\alpha = 1_{\mathcal{W}}$. Thus α is an isomorphism.
- We take $\mathcal{E}$ to be the subcategory of $\mathcal{D}$ with all pullbacks. This means that $\mathcal{E}$ has pullbacks together with a terminal object and is thus cartesian closed. We take β to be the subcategory restriction, which is trivially an epimorphism.
- We take $\mathbf{U}$ to be a subcategory of $\mathbf{V}$. We take the subcategory which only has objects formed with fibres over $\mathcal{E}$ and arrows of the form $\overline{A} \longrightarrow A$. We take γ to be the subcategory restriction, which is trivially an epimorphism.
- Finally, we have to define the natural transformation ϵ_w for all $w \in \mathcal{W}$. We take this to be the identity, which will give it all the required properties.
- Finally, we need to show that condition on interpretations holds. This follows from the uniformity of the encoding function. We have that every object in $\lambda\Pi$ can be understand as corresponding to the encoding of an element of the language of $\mathcal{O}_T$. Thus, after an induction over the structure of $\lambda\Pi$, we have that

  $$h([\![\{\!\!\{X\}\!\!\}_T]\!]_{\mathcal{K}_{\mathcal{J}}}) = [\![X]\!]_{\mathcal{R}_{\mathcal{S}}}$$

  holds.

□

We call the epimorphism constructed in Proposition 6.10 the judgements-as-types epimorphism. We have the following corollary; if we repeat the above proof with the encoding function as a bijection, *i.e.* the encoding is adequate, then we will obtain an isomorphism of models, which we call the judgements-as-types isomorphism.

Corollary 6.11 (judgements-as-types isomorphism) *Let $\mathcal{O}_T$ be an object-logic as defined in Definition 4.9 and let $\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathbf{V}]]$, be a Kripke $\Sigma_{\mathcal{O}_T}$-$\lambda\Pi$-model, where $\Sigma_{\mathcal{O}_T}$ is the $\lambda\Pi$-signature in judgements-as-types correspondence with $\mathcal{O}_T$, where the encoding is adequate and a bijection. Then there is a Kripke $\mathcal{O}_T$-model, $\mathcal{R}_{\mathcal{S}}, [\![-]\!]^{-,\rho}_{\mathcal{R}_{\mathcal{S}}} \rangle$, where $\mathcal{R}_{\mathcal{S}} : [\mathcal{X}, [\mathcal{E}^{op}, \mathbf{U}]]$, together with an (indexed) isomorphism of models,*

$$h = (\alpha, \beta, \gamma, (\epsilon_w)_w) : \mathcal{K}_{\mathcal{J}} \longleftarrow \mathcal{R}_{\mathcal{S}},$$

induced by the judgements-as-types correspondence, which respects the interpretation of syntax. Specifically, abusing notation by allowing X to range over all of the syntax of $\mathcal{O}_T$ and suppressing information about worlds, if $[\![X]\!]^{x}_{\mathcal{R}_{\mathcal{S}}}$ and $[\![\{\!\{X\}\!\}_T]\!]^{W}_{\mathcal{K}_{\mathcal{J}}}$ are defined, then

$$h([\![\{\!\{X\}\!\}_T]\!]_{\mathcal{K}_{\mathcal{J}}}) = [\![X]\!]_{\mathcal{R}_{\mathcal{S}}}.$$

Next, we show as another corollary of the existence of the judgements-as-types epimorphism, that a model of a representation of an object-logic can be uniformly constructed from a model of the object-logic. This result generalizes one of Simpson [Sim93], and ...

Corollary 6.12 (induced models) *Let $\mathcal{O}_T$ be an object-logic as defined in Definition 4.9 and let $\langle R_S, [\![-]\!]^{-,\rho}_{\mathcal{R}_{\mathcal{S}}} \rangle$, where $\mathcal{R}_{\mathcal{S}} : [\mathcal{X}.[\mathcal{E}^{op}, \mathbf{U}]]$, be a Kripke $\mathcal{O}_T$-model. Let $\Sigma_{\mathcal{O}_T}$ be the $\lambda\Pi$-signature in judgements-as-types correspondence with $\mathcal{O}_T$. Then there is a Kripke $\Sigma_{\mathcal{O}_T}$-$\lambda\Pi$-model, $\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathbf{V}]]$, induced by the correspondence.*

Proof We examine the construction used in Proposition 6.10. We notice that the subcategories we restricted to are Kripke $\lambda\Pi$-models in their own right and thus we can think of a Kripke $\mathcal{O}_T$-model as a Kripke $\lambda\Pi$-model. The judgements-as-types correspondence gives us the corresponding interpretation. □

Finally, we give a general representation theorem.

Theorem 6.13 (representation) *Given a judged object-logic $\mathcal{O}_T$ together with a uniform encoding, which is complete with respect to it's Kripke model and that the encoded logic is complete with respect to it's Kripke model then the encoding is adequate and there is a surjection between proofs*

$$\delta :\ (X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$$

and (proof) terms

$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$$

in $\lambda\Pi$.

Proof To show fullness, an induction over the structure of $\lambda\Pi$ is required, we elide the details since we are more concerned about the proof of faithfulness. The proof of faithfulness takes advantage of all the framework we have just set up. Let M_δ be a proof term in $\mathcal{O}_T$ with derivation

$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$$

in $\lambda\Pi$. Since $\Sigma_{\mathcal{O}_T}$-$\lambda\Pi$ is complete with respect to it's Kripke model, we have that

$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \models_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$$

holds. We use Proposition 6.10 to obtain a Kripke $\mathcal{O}_T$-model where

$$(X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \models_{\mathcal{O}_T} \mathsf{j}(\phi)$$

holds and since $\mathcal{O}_T$ is complete with respect to it's Kripke model, we have that the proof

$$(X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$$

holds in $\mathcal{O}_T$ and we have proved faithfulness.

The surjection between proofs in $\mathcal{O}_T$ and proof terms in $\Sigma_{\mathcal{O}_T}$-$\lambda\Pi$ follows from examining the step in the previous argument between the two Kripke models. Given a derivation of a proof term in $\Sigma_{\mathcal{O}_T}$-$\lambda\Pi$ it corresponds to

$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \models_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$$

in the Kripke model. Since we have an epimorphism between models, by Proposition 6.10, we know that each proof term has come from a proof

$$(X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \models_{\mathcal{O}_T} \mathsf{j}(\phi)$$

in the Kripke model. By completeness, this corresponds to a proof

$$(X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$$

in $\mathcal{O}_T$. Thus every proof term in $\lambda\Pi$ has come from a proof in $\mathcal{O}_T$ and so we have a surjection.
□

We have the following corollary:

Corollary 6.14 (representation) *Given a judged object-logic $\mathcal{O}_T$ together with a bijective encoding function, which is complete with respect to it's Kripke model and that the encoded logic is complete with respect to it's Kripke model then the encoding is adequate and there is a bijection between proofs*

$$\delta :\ (X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$$

and (proof) terms

$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$$

in $\lambda\Pi$.

The proof just uses the fact that an bijective encoding provides an isomorphism of models rather than an epimorphism, Corollary 6.11.

7 Representation Theorems

7.1 Introduction

All the previous work has been done to allow us to prove representation theorems. These are results which tell us which object-logics can be encoded in LF. Rather than dealing with individual object-logics as in the literature, [AHMP92, HHP93], we have dealt with general classes of object-logics. We have done this to avoid being ad-hoc as well as allowing us to develop not just representation theorems but a theory of representation. In the later sections, we deal with more specific classes of object-logics obtained through repeating the previous work with respect to labelled deductive systems.

There are two aspects to a representation theorem, both of which are concerned with the relationship between a proof in the object logic and a derivation of a proof term in $\lambda\Pi$:

1. The first condition is characterized by properties of the encoding. A full encoding means that
$$(X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$$
implies
$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$$
while a faithful encoding means the opposite implication holds. A full and faithful encoding is adequate.

2. The second is a more direct relationship between the proof and the proof terms. We are interested in any properties that a map between them might have, *i.e.* is it a surjection or bijection.

Proving that an encoding is full is just an induction proof and is relatively straightforward. We are not concerned with proving fullness because we are interested in the semantics of the object-logic and $\lambda\Pi$. Analyzing the semantics is done with the aim of providing proofs of faithfulness. The current proofs of faithfulness have been done proof-theoretically. This can be difficult since the proof involves showing that any derivation in $\lambda\Pi$is essentially the representation of a derivation in the object-logic. Faithfulness should be easy to prove semantically, it should follow from the truth of derivations in $\lambda\Pi$. In fact, the proof of Theorem 6.13 shows how straightforward it can be although we did a lot of work to get there.

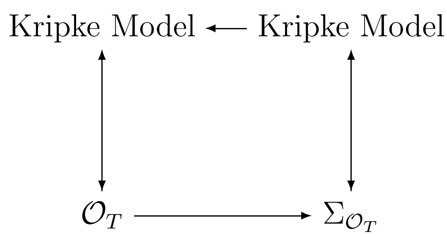
The relationship between proof terms and proofs is again something which is fairly straightforward to prove semantically. The morphism induced between the Kripke models, e.g. Theorem 6.10, provides the strength of this map. This is clear with the way the proof was constructed.

In this section, we will study what properties are required to obtain a faithful encoding and examine what effect different properties of the encoding function have on the above properties. The above analysis and the proof of Theorem 6.13 show us that there is a relationship between the two properties.

7.2 Representation Theorems

We recall that at the end of Section 6.3, we stated the following relationship between an object-logic and it's encoding in $\lambda\Pi$. If the encoding was uniform and the object-logic and encoded logic were complete with respect to their Kripke models then the encoding was adequate and there was a surjection between proofs in the object-logic and their corresponding proof term in $\lambda\Pi$. We gave as a corollary the case where the encoding function was bijective and thus gave a bijection between proofs and proof terms, as well as the adequacy result. Throughout the rest of this section, we will only make reference to this special case when the result either does not hold or is not a corollary of the uniform case. Thus all the theorems stated below carry across to the adequate case unless otherwise stated. We also do not analyse encodings which are not full, the reason for this is that fullness is proved by induction. We are concerned with using the techniques we have developed to look at semantic proofs of faithfulness.

The above representation theorem is quite strong, it requires that both the object-logic and encoded logic be sound and complete with respect to their Kripke models. We are interested in finding the weakest versions we can of the conditions on the object-logic and it's encoding. It is clear that these two conditions are not mutually exclusive. To understand how we can weaken this representation theorem, we need to examine it's proof in detail. The proof of fullness involves tracing our way around the following diagram:



Closer analysis shows that we are starting from the bottom right corner and tracing around the diagram until we reach the bottom left corner via the top edge. This means that we only need to be able to trace the diagram along this route; the completeness conditions mean that we have a equivalence on both sides which is far stronger than we need. We thus weaken these conditions:

Lemma 7.1 (representation) *Given a judged object-logic $\mathcal{O}_T$ with a uniform full encoding. If*

$$(X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \models_{\mathcal{O}_T} \mathsf{j}(\phi)$$

implies

$$(X)\ \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$$

and if

$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$$

implies

$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \models_{\Sigma_{\mathcal{O}_T}} M_\delta : \mathsf{j}(\phi)$$

then the encoding is faithful and there exists a surjection between proofs in $\mathcal{O}_T$ and proof terms M_δ in $\Sigma_{\mathcal{O}_T}$-$\lambda\Pi$.

The above side conditions are the minimum to allow us to travel round the diagram and so is really just a corollary of Theorem 6.13. The conditions on provability and validity for the object-logic amount to being able to either construct a term model or developing prime theories, while the conditions on deriviability and validity in $\lambda\Pi$amount to a proof of soundness. We will proceed to study the proofs of these conditions in detail to see where they fail and what is the minimum condition for them to hold.

8 Discussion

Worlds-as-parameters.

The work of Basin, Matthews and Viganò deals in detail with substructural logics like relevant logic. We have not dealt with this here since while it would be possible to provide a Kripke model for the object-logic, we would be unable to adequately encode a substructural logic into LF. The reason for this is that since the meta-logic provided by LF is intuitionistic, we obtain weakening for free, c.f §2.3. We do not want weakening to be present for a substructural logic.

The logical framework, RLF [IP98] does provide a suitable meta-logic for representing substructural logics. The meta-theoretic analysis of an object-logic in LF can be carried out for an object-logic in RLF but this goes beyond the scope of this paper.

Within the work of Basin, Matthews and Viganò, they deal with quantified modal logics. These are modal logics where the domain of the semantics is either increasing, decreasing or fixed. This requires each connective to come with not just a relation (or relational theory) of worlds but also information about the domain (or domain theory) of the semantics. This then produces a two-dimensional space of possible logics. It appears that the Kripke models used in this paper do not carry information about the domain. To deal with these types of logics, it seems that a new categorical model will be required and possibly a new logical framework.

The labelled deductive systems of Basin, Matthews and Viganò classify a wide range of logics but this is not exhaustive. There are a lot of logics which can not be presented in a labelled deductive system. Within this paper a labelled deductive system was used as a means to describe a 'generic' logic and a natural deduction system, Hilbert-type system or some sort of hybrid of the two was then extracted from it. This extracted proof system was then analysed throughout the rest of the paper. It is a conjecture of the authors that any natural deduction system, Hilbert-type system or hybrid system can be used here, not just one obtained from a labelled deductive system.

Acknowledgements. The authors are grateful to S. Abramsky, M. Collinson, D. Galmiche, R. Harper, F. Honsell, G. Huet, M. Hyland, J. Lipton, S. Matthews, G. McCusker, E. Moggi, G. Plotkin, E. Ritter, E. Robinson, A. Simpson, T. Streicher, P. Taylor and L. Wallen for their comments on various aspects of this work.

References

[Acz78] Peter Aczel. The type theoretic interpretation of constructive set theory. In *Logic Colloquium 1977*, pages 55–66. North-Holland Publishing Company, 1978.

[Acz80] Peter Aczel. Frege structures and the notions of proposition, truth and set. In J. Barwise, H.J.Keisler, and K. Kunen, editors, *The Kleene Symposium*, pages 31–59. North-Holland Publishing Company, 1980.

[AHMP92] Arnon Avron, Furio Honsell, Ian Mason, and Randy Pollack. Using typed lambda calculus to implement formal systems on a machine. *Journal of Automated Reasoning*, 9:309–354, 1992.

[AHMP98] Arnon Avron, Furio Honsell, Marion Miculan, and Cristian Paravano. Encoding modal logics in logical frameworks. *Studia Logica*, 60:161–208, 1998.

[Avr91] Arnon Avron. Simple consequence relations. *Information and Computation*, 92:105–139, 1991.

[BMV96] David Basin, Seán Matthews, and Luca Viganò. Implementing modal and relevance logics in a logical framework. In L.C. Aiello, J. Doyle, and S.C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR '96)*, pages 386–397. Morgan Kaufmann Publishers, San Francisco, CA, 1996.

[BMV97] David Basin, Seán Matthews, and Luca Viganò. Labelled propositional modal logics: theory and practice. *Journal of Logic and Computation*, 7(1):685–717, 1997.

[BMV98] David Basin, Seán Matthews, and Luca Viganò. Natural deduction for non-classical logics. *studia logica*, 60(1):119–160, 1998.

[Gar92] Philippa Gardner. *Representing Logics in Type Theory.* PhD thesis, University of Edinburgh, 1992.

[Gen34] Gerhard Gentzen. Untersuchungen uber das logische schliessen. *Mathematische Zeitschrift*, 39:176–210,405–431, 1934.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[HHP87] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. In *Proceedings 2nd Annual IEEE Symp. on Logic in Computer Science, LICS'87, Ithaca, NY, USA, 22–25 June 1987*, pages 194–204. IEEE Computer Society Press, New York, 1987.

[HHP93] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinery*, 40(1):143–184, 1993.

[Hod01] Wilfred Hodges. Elementary predicate logic. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 1. Kluwer Academic Publishers, 2001.

[HST94] Robert Harper, Donald Sannella, and Andrzej Tarlecki. Structured theory presentations and logic representations. *Annals of Pure and Applied Logic*, 67(1-3):113–160, 1994.

[IP98] Samin S. Ishtiaq and David J. Pym. A relevant analysis of natural deduction. *Journal of Logic and Computation*, 8(6):809–838, 1998.

[IP02] Samin Ishtiaq and David J. Pym. Dependently-typed, bunched λ-calculus. *Journal of Logic and Computation*, 12(6):1061–1104, 2002.

[Kan00] Immanuel Kant. *Immanuel Kants Logik: Ein Hanbuch zu Vorlesungen*. Gottlob Benjamin Jäsche, Königsberg, 1800.

[Kri63] Saul A. Kripke. Semantical analysis of modal logic I. *Zeitschrift für mathematiche Logik und Grundlagen der Mathematik*, 16:83–94, 1963.

[LS86] Joachim Lambek and Phil Scott. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, 1986.

[ML71] Per Martin-Löf. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In J. E. Fenstad, editor, *Second Scandinavian Logic Symposium*. North-Holland, 1971.

[ML75] Per Martin-Löf. An intutionistic theory of types: Predicative part. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquim 1973*, pages 73–188. North-Holland Publishing Company, 1975.

[ML82] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws, 1982.

[ML85] Per Martin-Löf. On the meanings of logical constants and the justification of the logical laws. Technical report, Scuola di Specializzazione in Logica Matematica, Dipartimento di Matematica, Università di Siena, 1985. Technical Report 2.

[MM91] John C. Mitchell and Eugenio Moggi. Kripke-style models for typed lambda calculus. *Annals of Pure and Applied Logic*, 51:99–124, 1991.

[nDCH96] L. Fari nas Del Cerro and A. Herzig. Combining classical and intuitionistic logic. In F. Baader and K.U. Schulz, editors, *Frontiers of Combining Systems*, pages 93–102. Kluwer, 1996.

[Pfe95] Frank Pfenning. Structural cut elimination. In D. Kozen, editor, *Proceedings of the Tenth Annual Symposium on Logic in Computer Science*, pages 156–166. IEEE Computer Society Press, 1995.

[PP06] David J. Pym and Mark A. Price. Functorial kripke-beth-joyal models of the $\lambda\Pi$-calculus I: type theory and internal logic. `www.cs.bath.ac.uk/MPrice/downloads/Kripke1.ps`, 2006.

[Pra65] Dag Prawitz. *Natural Deduction: A Proof-Theoretic Study.* Almquist and Wiksell, 1965.

[Pym90] David J. Pym. *Proofs, Search and Computation in General Logic.* PhD thesis, University of Edinburgh, 1990.

[Pym01] David J. Pym. Functorial kripke-beth-joyal models of the $\lambda\Pi$-calculus III: logic programming and its semantics. `www.cs.bath.ac.uk/~pym/kripke3.ps`, 2001.

[Rea88] Stephen Read. *Relevant Logic.* Blackwell, 1988.

[SHD93] Peter Schröder-Heister and Kosta Dǒsen, editors. *Substructural Logics.* Clarendon Press, 1993.

[Sim93] Alex Simpson. Kripke semantics for a logical framework. `homepages.inf.edu.ac.uk/als/Research/kripke.ps.gz`, 1993.

[Sun01] Göran Sundholm. Systems of deduction. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 2. Kluwer Academic Publishers, 2001.

[Vig00] Luca Viganò. *Labelled Non-Classical Logics.* Kluwer Academic Publishers, 2000.