



Citation for published version:

Price, M 2008, Logical Aspects of Logical Frameworks. Department of Computer Science Technical Report Series, no. CSBU-2008-07, Department of Computer Science, University of Bath, Bath, U. K.

Publication date:
2008

[Link to publication](#)

©The Author December 2008

University of Bath

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**Department of
Computer Science**



Technical Report

PhD. Dissertation: Logical Aspects of Logical Frameworks

Mark Andrew Price

Copyright ©December 2008 by the authors.

Contact Address:

Department of Computer Science
University of Bath
Bath, BA2 7AY
United Kingdom
URL: <http://www.cs.bath.ac.uk>

ISSN 1740-9497

Logical Aspects of Logical Frameworks

submitted by

Mark Andrew Price

for the degree of Doctor of Philosophy

University of Bath

Department of Computer Science

November 2008

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis rests with its author. A copy of this thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and they must not copy it or use material from it except as permitted by law or with the consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signature of Author

Mark Andrew Price

Logical Aspects of Logical Frameworks

Mark Andrew Price

ABSTRACT

This thesis provides a model-theoretic semantic analysis of aspects of the LF logical framework. The LF logical framework is the $\lambda\Pi$ -calculus together with the judgements-as-types representation mechanism.

A denotational semantics is provided for the $\lambda\Pi$ -calculus in terms of Kripke $\lambda\Pi$ -models. These are a generalization of the Kripke lambda models of Mitchell and Moggi to dependent types and are based on the contextual categories of Cartmell and their reformulation by Ritter. We analyse these models in terms of (Kripke) logical relations.

We also present Kripke models of the internal logic of the $\lambda\Pi$ -calculus, the $\{\forall, \supset\}$ -fragment of many-sorted minimal first-order logic, and show that the propositions-as-types correspondence induces an isomorphism between the two classes of Kripke models.

We provide a proof- and model-theoretic account of judged object-logics (logics suitable for representation in LF). We show that the judgements-as-types correspondence induces an epimorphism between these Kripke models and Kripke $\lambda\Pi$ -models; which we use to provide model-theoretic proofs of faithfulness.

We consider a variant of the LF logical framework which uses the worlds-as-parameters representation mechanism. The generality of our account of the judgements-as-types correspondence allows us to treat the worlds-as-parameters representation mechanism as a special case. We interpret the syntactic ‘worlds’ introduced by worlds-as-parameters as worlds in our Kripke models and show that there exists a worlds-as-parameters epimorphism.

We provide a semantic account of proof-search in the $\lambda\Pi$ -calculus by identifying a class of Herbrand models and providing a least fixed-point construction corresponding, as usual, to resolution. Finally, we provide a characterization of abstract logic programming languages, as defined by Miller *et al.*, in the LF logical framework.

ACKNOWLEDGEMENTS

I would like to thank all of the people who have helped and supported me during my time as a PhD student. I owe a particular debt of gratitude to my parents who have supported me throughout my time as a student and have always been on hand to help in a variety of ways. My father deserves a special mention for all the shelves he has put up over the years without which I would have been unable to store all my books needed for studying.

I owe a lot to Theresa for all the support she has given me over the last couple of years and especially for her willingness to become a 'PhD widow' for a while.

This thesis would not have been possible without the support and guidance offered by my supervisor David Pym. Thanks for all the support and encouragement over the years as well as the much needed motivational talks.

I have also received invaluable guidance and supervision from Guy McCusker. Thanks for all the help and particularly encouraging me to clarify my own thoughts before mentioning them to others.

Last and by no means least, a special mention is needed for Matthew Collinson. Thanks for all the help and friendship over the years: from showing me round the lab on my first day to providing academic guidance you have been a good friend throughout.

Contents

1	Introduction	1
2	The $\lambda\Pi$-calculus	4
2.1	A Syntactic Presentation	4
2.2	An Algebraic Presentation	10
3	Kripke Models of the $\lambda\Pi$-calculus	14
3.1	Kripke $\lambda\Pi$ -Prestructures and -Structures	16
3.2	Examples	19
3.2.1	Term Model	19
3.2.2	The $\{\forall, \supset\}$ -fragment of many-sorted minimal first-order logic	26
3.3	Kripke Σ - $\lambda\Pi$ -models	36
3.4	Examples	40
3.4.1	Term Model	40
3.4.2	The $\{\forall, \supset\}$ -fragment of many-sorted first-order logic . . .	40
3.5	Adding Definitional Equality: $\lambda\Pi_{=}$	40
3.6	Satisfaction	44
3.7	Soundness and Completeness of $\lambda\Pi$ and $\lambda\Pi_{=}$	53
3.8	Soundness and Completeness of $\lambda\Pi$ and $\lambda\Pi_{=}$ for \Vdash	60
4	Applicative Structures and Kripke Logical Relations	68
4.1	Applicative Structures	68
4.1.1	Kripke Σ - $\lambda\Pi$ -models with Families	68
4.1.2	Equational Kripke Σ - $\lambda\Pi_{=}$ -Applicative Structures	71
4.2	Kripke Logical Relations	71
4.2.1	Partial Logical Equivalence Relations	75
4.2.2	Kripke Logical Relations on Classical Applicative Structures	75
4.2.3	A Counter-model to Semantic Implication	77
5	The Internal Logic and its Models	78
5.1	The Propositions-as-types Correspondence	78
5.2	The Semantics of the Internal Logic \mathcal{L}_T	81
5.2.1	Kripke Prestructures and Structures for \mathcal{L}_T	82

5.2.2	Kripke Models of \mathcal{L}_T	84
5.2.3	Term Model	86
5.2.4	Satisfaction	90
5.2.5	Soundness and Completeness for \Vdash	91
5.2.6	Soundness and Completeness for $\Vdash\rightarrow$	93
5.3	Propositions-as-types Isomorphism	95
5.4	Kripke Models vs. Classical Models	100
6	Introduction to Logical Frameworks	102
6.1	Kant's Notion of a Judgement	102
6.2	Martin-Löf's Higher-Order Judgements	103
6.3	The Notion of a Logical Framework	106
6.4	The LF Logical Framework	107
7	LF's Object-Logics	110
7.1	Background	110
7.2	The Theory of Judged Object-logics	115
7.2.1	Classical λ -calculus	129
7.2.2	Hoare's Logic	130
7.3	Models of Judged Object-logics	133
7.4	Meta-theorems for Object-logics	148
8	Judgements-as-types Correspondence	157
8.1	Syntactic Judgements-as-types: Encoding Logics in LF	158
8.2	Semantic Judgements-as-types: an Epimorphism of Models	165
8.3	Representation Theorems	169
9	Worlds-as-parameters	170
9.1	Labelled Natural Deduction Systems	172
9.2	Soundness and Completeness of Labelled Natural Deduction Systems	175
9.3	Worlds-as-parameters Encoding	178
10	Introduction to Logic Programming	181
10.1	Logic Programming in $\lambda\Pi$	183
11	Σ-$\lambda\Pi$-Herbrand Models	186
11.1	Herbrand Prestructures and Structures	186
11.2	Σ - $\lambda\Pi$ -Herbrand Models	187
11.3	Least Fixed-Point Construction	188

12 Encoding Sequent Systems in LF	196
12.1 Intuitionistic Logic	197
12.2 Classical Logic	204
12.3 Encoding Higher-Order Intuitionistic Logic in LF	214
13 Representing Abstract Logic Programming Languages in LF	222
13.1 Representing Uniform Proofs in G_3i in LF	226
13.2 Representing Uniform Proofs in G_3HOIL in LF	239
13.3 Representing ALPLs in LF, the Story so Far	244
13.4 Representing Uniform Proofs in G_3c in LF	245
13.5 Summary	264
14 Conclusion	266
Bibliography	267
A Signatures of Object-logics	277
B Examples of Labelled Natural Deductive Systems	290
C Various Sequent Calculi	293

Chapter 1

Introduction

An important aspect of (theoretical) computer science is being able to reason about formal systems in a way which is independent of their implementation. One proposed solution is to represent the formal system in a logical framework (Harper, Honsell & Plotkin 1993).

A logical framework can be seen as arising from Martin-Löf's intuitionistic theory of iterated inductive definitions (Martin-Löf 1971) and (Martin-Löf 1975), in which form and inductive definitional status in the natural deduction inference rules are considered. In other words, Martin-Löf provides a formal metatheory of inference rules.

In order to describe a logical framework, we must (Ishtiaq & Pym 2002) 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

$$\textit{Framework} = \textit{Language} + \textit{Representation}.$$

This thesis provides a proof- and model-theoretic account of the LF logical framework (Harper et al. 1993). LF is the logical framework whose language is the $\lambda\Pi$ -calculus and whose representation mechanism is the judgements-as-types correspondence.

Chapter 2 describes the $\lambda\Pi$ -calculus (Harper et al. 1993) both syntactically and algebraically. Chapter 7 provides a categorical semantics for the $\lambda\Pi$ -calculus in terms of Kripke models. These models are based on the contextual categories

of Cartmell (1986) and their reformulation by Pitts (2000). Our Kripke models are a generalization of the Kripke λ -models of Mitchell & Moggi (1991) to the dependently typed setting.

The work of Mitchell and Moggi also provides the motivation of Chapter 4. We show that their analysis of Kripke λ -models in terms of logical relations can also be carried out for our Kripke models of the $\lambda\Pi$ -calculus.

We conclude our semantic analysis of the $\lambda\Pi$ -calculus and foreshadow the next four chapters in Chapter 5. We describe the internal logic of the $\lambda\Pi$ -calculus and show that it can be represented in the $\lambda\Pi$ -calculus using the propositions-as-types correspondence. We then carry out original analysis to show that the propositions-as-types correspondence induces an isomorphism between Kripke models of the internal logic and Kripke models of the $\lambda\Pi$ -calculus.

Moving on from our analysis of the $\lambda\Pi$ -calculus, we turn our attention to a characterization of a class of object-logics which are suitable for representation in LF. Chapter 6 provides an overview of logical frameworks. This chapter is a summary of existing research on logical frameworks heavily influenced by (Avron, Honsell, Miculan & Paravano 1997).

Chapter 8 provides an account of judged object-logics. This account includes proof- and model-theoretic analysis. We describe judged proof systems which provide a proof-theoretic characterization of judged object-logics. This work is closely related to that of Gardner (1992*b*). We describe Kripke models of judged object-logics and examine under what circumstances soundness and completeness holds. The Kripke models draw heavily on the work of Lawvere (1970).

Having provided a characterization of object-logics which are suitable for encoding in LF, we turn our attention to the representation mechanism. Chapter 8 provides a syntactic and semantic account of the judgements-as-types correspondence. The syntactic account is a presentation of the encoding found in (Harper et al. 1993). The semantic account is original and shows that the judgements-as-types correspondence induces an epimorphism between Kripke models of the object-logic and Kripke models of the $\lambda\Pi$ -calculus. Using an idea of Simpson (1993), we show that the faithfulness of an encoding can be established model-theoretically.

We conclude this section of work by looking at the worlds-as-parameters representation mechanism, Chapter 9. This turns out to be a special case of the judgements-as-types encoding described in Chapter 8. We argue that labelled natural deduction systems (Basin, Matthews & Viganò 1998) are the appropriate proof system to characterize object-logics which are encoded using worlds-as-parameters. They also provide a more systematic account of the satisfaction relation used to define certain connectives in a Kripke model. We show that the labels used in labelled natural deduction systems can be interpreted as worlds in Kripke models of the object-logics, and that the worlds-as-parameters representation mechanism induces an epimorphism between these models and Kripke

models of the $\lambda\Pi$ -calculus where the objects that encode labels are interpreted as worlds. Apart from the labelled natural deduction systems and the worlds-as-parameters representation mechanism, the chapter is original.

We conclude the thesis by providing a model-theoretic account of proof-search in LF. Chapter 10 provides an overview of logic programming and why proof-search is important. Chapter 11 shows that a class of Kripke $\lambda\Pi$ -models are also Herbrand models. We show that there is a least fixed-point operator on Herbrand models arising from resolution in the $\lambda\Pi$ -calculus and that the least fixed-point is a Herbrand model. This work is original and is a generalization of the work of Miller (1989).

The final two chapters examine the relationship between uniform proofs in the object-logic and their representation in LF. Chapter 12 shows how certain sequent calculi can be represented in LF. This work is taken from (Pfenning 2000). We show that his encoding method can be extended to higher-order logics. The logics discussed in this chapter are those given as examples of abstract logic programming languages in (Miller, Nadathur, Pfenning & Šcedrov 1991). Chapter 13 provides an account of the relationship between abstract logic programming languages and their encodings in LF. We provide some conditions on LF terms so that they always represent uniform proof-terms.

This thesis can also be seen as carrying out the research proposal outlined in (Ishtiaq & Pym 2002), which means we have provided LF with a semantics. This research is based on (Pym 2004a), (Pym 2004b) and (Pym 2004c).

It is important to note that while this thesis does apply the results of Chapters 2 to 9 to logic programming (Chapters 10 to 13), this thesis mainly provides a contribution to logic. The majority of this thesis is a mathematical study of logical frameworks. The relationship between logical frameworks and logical systems (logical frameworks provide a basis for a systematic study of the uniformity of logical systems) means that this thesis contributes to our understanding and study of logical systems and hence to logic.

This thesis was typeset with $\text{\LaTeX} 2_{\epsilon}$ ($\text{\LaTeX} 2_{\epsilon}$ 1994) and commutative diagrams were produced with Taylor (1986) diagrams.

Chapter 2

The $\lambda\Pi$ -calculus

We develop two presentations of the $\lambda\Pi$ -calculus: the first is syntactic and the second algebraic. The syntactic presentation is an overview of the presentation of the $\lambda\Pi$ -calculus introduced in Harper, Honsell & Plotkin (1987) and (Harper et al. 1993). The $\lambda\Pi$ -calculus is essentially the dependently typed λ -calculus λP (Barendregt 1991) with signatures. The $\lambda\Pi$ -calculus is in propositions-as-types correspondence (Howard 1980) with the many sorted $\{\forall, \supset\}$ -fragment of minimal first-order logic this is discussed in detail in § 5. As well as defining the syntax, grammar and typing rules, we discuss the important meta-theoretic properties; for example, that weakening, transitivity, strengthening and permutation are admissible, uniqueness of types and kinds, a Church-Rosser property holds and the decidability of the typing relations. Equality is not dealt with in this presentation; it is left until § 3.5. This is done for the sake of simplicity and allows us to highlight the important meta-theoretic properties rather than dealing with the technical issues associated with equality.

The algebraic presentation involves the construction of strict indexed categories out of the syntax of the $\lambda\Pi$ -calculus. It is similar to other algebraic presentations of dependent type theories; for example, (Streicher 1989), (Jacobs 1991), (Hofmann 1996) and (Pitts 2000).

2.1 A Syntactic Presentation

The $\lambda\Pi$ -calculus is a language with entities of three levels: objects; types and families of types; and, kinds. Objects are classified by types; while types and families of types are classified by kinds. The kind **Type** classifies the types; the other kinds classify functions, f , which yield a type $f(M_1) \dots (M_n)$ when applied to objects M_1, \dots, M_n of certain types determined by the kind of f . Any function derivable in this system has a type as domain, while its range can either be a type, if it is an object, or a kind, if it is a family of types. The $\lambda\Pi$ -calculus is therefore predicative.

The theory we shall deal with is a formal system for deriving assertions of one of the following shapes:

$\vdash \Sigma$ sig	Σ is a signature
$\vdash_{\Sigma} \Gamma$ context	Γ is a valid context in Σ
$\Gamma \vdash_{\Sigma} K$ Kind	K is a kind in Γ and Σ
$\Gamma \vdash_{\Sigma} A:K$	A has kind K in Γ and Σ
$\Gamma \vdash_{\Sigma} M:A$	M has type A in Γ and Σ

where the syntax is specified by the following grammar:

Signatures	$\Sigma ::= \langle \rangle \mid \Sigma, c:K \mid \Sigma, c:A$
Contexts	$\Gamma ::= \langle \rangle \mid \Gamma, x:A$
Kinds	$K ::= \mathbf{Type} \mid \Pi x:A. K$
Types	$A ::= c \mid \Pi x:A. B \mid \lambda x:A. B \mid AM$
Objects	$M ::= c \mid x \mid \lambda x:A. M \mid MN$

where we let M and N range over expressions for objects, A and B range over expressions for types and families of types, K for kinds, x and y over variables and c over constants. We also allow f and g to range over variables where the intention is that, in general, these have higher types. We assume α -equivalence throughout.

We define free and bound variables in the usual way (*cf.* (Barendregt 1992)) with Π and λ being the only binding operators. Capture-avoiding substitution is also defined in the usual way. We write $X[M_1, \dots, M_k/x_1, \dots, x_k]$ for the result of simultaneously substituting M_1, \dots, M_k for free occurrences of x_1, \dots, x_k in X , renaming bound variables where necessary.

We refer to the collection of constants declared in a signature Σ by $Dom(\Sigma)$, the collection of variables declared in a context Γ by $Dom(\Gamma)$ and the collection of free variables in an expression E by $FV(E)$.

The inference rules of the $\lambda\Pi$ -calculus appear in Table 1. We shall refer to this system as \mathbf{N} because it is a system of natural deduction. We write \mathbf{N} proves $\Gamma \vdash_{\Sigma} M:A$, *etc.*, to denote that the assertion $\Gamma \vdash_{\Sigma} M:A$, *etc.*, is provable in the system \mathbf{N} and we shall sometimes write simply $\Gamma \vdash_{\Sigma} M:A$ where no confusion can arise. For technical reasons, we also require that the assertion $\Gamma \vdash_{\Sigma} \langle \rangle: \langle \rangle$, where $\langle \rangle$ stands for the unit type and unit object, can be derived from any signature Σ and context Γ .

We have not included the premiss $\Gamma \vdash_{\Sigma} A: \mathbf{Type}$ in each of the abstraction-forming rules, unlike (Harper et al. 1987), (Harper et al. 1993), (Pym 1990) and (Pym 1995). This premiss is inessential for the definition of proofs in the $\lambda\Pi$ -calculus. Certain inductive proofs, however, such as that of Theorem 2.1 in (Harper et al. 1993) which is proven via the correctness of an inductive algorithmic formulation of the calculus, are technically simplified by their inclusion.

Valid Signatures

$$\frac{}{\vdash \langle \rangle \text{ sig}} \quad (2.1)$$

$$\frac{\vdash \Sigma \text{ sig} \quad \vdash_{\Sigma} K \text{ Kind} \quad c \notin \text{Dom}(\Sigma)}{\vdash \Sigma, c: K \text{ sig}} \quad (2.2)$$

$$\frac{\vdash \Sigma \text{ sig} \quad \vdash_{\Sigma} A: \text{Type} \quad c \notin \text{Dom}(\Sigma)}{\vdash \Sigma, c: A \text{ sig}} \quad (2.3)$$

Valid Contexts

$$\frac{\vdash \Sigma \text{ sig}}{\vdash_{\Sigma} \langle \rangle \text{ context}} \quad (2.4)$$

$$\frac{\vdash_{\Sigma} \Gamma \text{ context} \quad \Gamma \vdash_{\Sigma} A: \text{Type} \quad x \notin \text{Dom}(\Gamma)}{\vdash_{\Sigma} \Gamma, x: A \text{ context}} \quad (2.5)$$

Valid Kinds

$$\frac{\vdash_{\Sigma} \Gamma \text{ context}}{\Gamma \vdash_{\Sigma} \text{Type} \text{ Kind}} \quad (2.6)$$

$$\frac{\Gamma, x: A \vdash_{\Sigma} K \text{ Kind}}{\Gamma \vdash_{\Sigma} \Pi x: A. K \text{ Kind}} \quad (2.7)$$

Valid Families

$$\frac{\vdash_{\Sigma} \Gamma \text{ context} \quad c: K \in \Sigma}{\Gamma \vdash_{\Sigma} c: K} \quad (2.8)$$

$$\frac{\Gamma, x: A \vdash_{\Sigma} B: \text{Type}}{\Gamma \vdash_{\Sigma} \Pi x: A. B: \text{Type}} \quad (2.9)$$

$$\frac{\Gamma, x: A \vdash_{\Sigma} B: K}{\Gamma \vdash_{\Sigma} \lambda x: A. B: \Pi x: A. K} \quad (2.10)$$

$$\frac{\Gamma \vdash_{\Sigma} B: \Pi x: A. K \quad \Gamma \vdash_{\Sigma} N: A}{\Gamma \vdash_{\Sigma} BN: K[N/x]} \quad (2.11)$$

$$\frac{\Gamma \vdash_{\Sigma} A: K \quad \Gamma \vdash_{\Sigma} K' \text{ Kind} \quad K =_{\beta\eta} K'}{\Gamma \vdash_{\Sigma} A: K'} \quad (2.12)$$

Table 1: Rules for Typings (continued on the next page)

Valid Objects

$$\frac{\vdash_{\Sigma} \Gamma \text{ context} \quad c:A \in \Sigma}{\Gamma \vdash_{\Sigma} c:A} \quad (2.13)$$

$$\frac{\vdash_{\Sigma} \Gamma \text{ context} \quad x:A \in \Gamma}{\Gamma \vdash_{\Sigma} x:A} \quad (2.14)$$

$$\frac{\Gamma, x:A \vdash_{\Sigma} M:B}{\Gamma \vdash_{\Sigma} \lambda x:A. M: \Pi x:A. B} \quad (2.15)$$

$$\frac{\Gamma \vdash_{\Sigma} M: \Pi x:A. B \quad \Gamma \vdash_{\Sigma} N:A}{\Gamma \vdash_{\Sigma} MN: B[N/x]} \quad (2.16)$$

$$\frac{\Gamma \vdash_{\Sigma} M:A \quad \Gamma \vdash_{\Sigma} A': \mathbf{Type} \quad A =_{\beta\eta} A'}{\Gamma \vdash_{\Sigma} M:A'} \quad (2.17)$$

Table 1: Rules for Typings

We write $A \rightarrow B$ for $\Pi x:A. B$ when x does not occur free in B and $A \rightarrow K$ for $\Pi x:A. K$ when x does not occur free in K . Together with the typing rules:

$$\frac{\Gamma, x:A \vdash_{\Sigma} B: \mathbf{Type} \quad x \notin FV(B)}{\Gamma \vdash_{\Sigma} A \rightarrow B: \mathbf{Type}} \quad (2.18)$$

$$\frac{\Gamma, x:A \vdash_{\Sigma} B: K \quad x \notin FV(B)}{\Gamma \vdash_{\Sigma} \lambda x:A. B: A \rightarrow K} \quad (2.19)$$

$$\frac{\Gamma \vdash_{\Sigma} B: A \rightarrow K \quad \Gamma \vdash_{\Sigma} N:A}{\Gamma \vdash_{\Sigma} BN: K} \quad (2.20)$$

$$\frac{\Gamma, x:A \vdash_{\Sigma} M: B \quad x \notin FV(B)}{\Gamma \vdash_{\Sigma} \lambda x:A. M: A \rightarrow B} \quad (2.21)$$

$$\frac{\Gamma \vdash_{\Sigma} M: A \rightarrow B \quad \Gamma \vdash_{\Sigma} N:A}{\Gamma \vdash_{\Sigma} MN: B} \quad (2.22)$$

this use of \rightarrow constitutes a conservative extension of the language.

For a calculus to be suitable for proof-search each typing rule needs to satisfy the sub-formula property: every type in the conclusion of a rule is a sub-type of an object in the premiss. If the sub-formula property does not hold for a given rule then there is too much non-determinism to carry out proof-search because one has to guess any type which does not have a sub-type in the conclusion. In \mathbf{N} the rule (2.15) does not satisfy the sub-formula property making it a bad

calculus for proof-search. Alternative presentations of the $\lambda\Pi$ -calculus which do satisfy the sub-formula property can be found in (Pym 1990) and (Pym 1995). In § 3.8, we present a calculus **C** which satisfies the sub-formula property and is thus suitable for proof-search.

A term is said to be *well-typed in a signature and context* if it can be shown to either be a kind, have a kind, or have a type in that signature or context. A term is *well-typed* if it is well-typed in some signature and context.

The $\lambda\Pi$ -calculus comes equipped with the intensional $\alpha\beta\eta$ -equality. We write $U = V$ to denote the α -equality of expressions U and V , while we use \equiv to denote their syntactic identity. To denote a definition we write \equiv_{def} . $\beta\eta$ -reduction, written $\rightarrow_{\beta\eta}$, can be defined at the level of types and families of types in the obvious way; the details can be found in (Harper et al. 1993). We write $M =_{\beta\eta} N$ if and only if $M \rightarrow_{\beta\eta}^* P$ and $N \rightarrow_{\beta\eta}^* P$ for some object P , where $*$ denotes transitive closure. We write $NF(U)$ to denote the $\beta\eta$ -normal form of the expression U .

A summary of the major meta-theorems pertaining to **N** and its reduction properties are given by the following theorem:

Theorem 2.1 (Basic Metatheory of the $\lambda\Pi$ -calculus (Harper et al. 1993))

Let Σ be a $\lambda\Pi$ -signature and Γ be a $\lambda\Pi$ -context. Let X range over basic assertions of the form $A:K$ and $M:A$. The following statements hold in the $\lambda\Pi$ -calculus:

1. *Thinning (weakening) is an admissible rule: if **N** proves $\Gamma \vdash_{\Sigma} X$ and **N** proves $\vdash_{\Sigma} \Gamma, \Gamma'$ context, then **N** proves $\Gamma, \Gamma' \vdash_{\Sigma} X$;*
2. *Transitivity is an admissible rule: if **N** proves $\Gamma \vdash_{\Sigma} M:A$ and **N** proves $\Gamma, x:A, \Gamma' \vdash_{\Sigma} X$ then **N** proves $\Gamma, \Gamma' \vdash_{\Sigma} X[M/x]$;*
3. *Strengthening is an admissible rule: if **N** proves $\Gamma, x; \Gamma' \vdash_{\Sigma} X$ and $x \notin FV(\Gamma') \cup FV(X)$ then $\Gamma, \Gamma' \vdash_{\Sigma} X$;*
4. *Permutation is an admissible rule: if **N** proves $\Gamma, x:A, y:B, \Gamma' \vdash_{\Sigma} X$ and if $x \notin FV(B)$, then **N** proves $\Gamma, y:B, x:A, \Gamma' \vdash_{\Sigma} X$;*
5. *Uniqueness of types and kinds: if **N** proves $\Gamma \vdash_{\Sigma} M:A$ and **N** proves $\Gamma \vdash_{\Sigma} M:A'$, then $A =_{\beta\eta} A'$, and similarly for kinds;*
6. *Subject reduction: if **N** proves $\Gamma \vdash_{\Sigma} M:A$ and $M \rightarrow_{\beta\eta} M'$, then $\Gamma \vdash_{\Sigma} M':A$, and similarly for types;*
7. *All well-typed terms are strongly normalizing, i.e., all reduction sequences arrive at a normal form;*
8. *All well-typed terms are Church-Rosser, i.e., if $X \rightarrow_{\beta\eta}^* X'$ and $X \rightarrow_{\beta\eta}^* X''$, then there exists a Y such that $X' \rightarrow_{\beta\eta}^* Y$ and $X'' \rightarrow_{\beta\eta}^* Y$;*

9. Each of the five relations defined by Table 1 is decidable, as is the property of being well-typed;
10. Predicativity: if \mathbf{N} proves $\Gamma \vdash_{\Sigma} M : A$ then the type-free λ -term obtained by erasing all the type information from M can be typed in the Church type-assignment system (cf. Hindley & Seldin (1986), ch. 15, pp. 205-223). ■

The proof of this theorem is rather complicated. The main difficulty lies in proving the Church-Rosser property in the presence of η -conversion. One method due to Salvesen (1990) adopts the methods developed by Van Daalen (1980) in his thesis to this type theory. The essential step in obtaining the proof in the presence of η -conversion is to first reformulate the $\lambda\Pi$ -calculus as a system with equality judgements in which type labels are explicit, *i.e.*, the assertions of equality have shape $\Gamma \vdash_{\Sigma} M = N : A$, *etc.* This step is sufficient to allow the methods of van Daalen to go through. Harper (1988) also considers equational formulations of the $\lambda\Pi$ -calculus as a basis for the construction of environmental models (Meyer 1982) of the type of the LF logical framework. We will discuss Harper's presentation in more detail in § 3.5.

In (Harper et al. 1987) similar properties are proven for \mathbf{N} with β -reduction only. \mathbf{N} was presented without η -reduction because the aforementioned proof was only discovered by Salvesen after the original technical report was published. The conclusion of (Harper et al. 1993) discusses this in detail. The proof of part 8 in (Harper et al. 1993), requires one to first prove decidability. The Church-Rosser property, strong normalization and the presence of type labels are all required to prove decidability. There is an alternative approach due to Coquand (1991) which makes use of an analogy with Kripke worlds, however Coquand does not exploit this by providing a Kripke model and hence a Kripke semantics, in fact he only uses a single world.

In Appendix A of (Felty 1991), Felty presents a system which she calls canonical LF. This is essentially a restriction of the $\lambda\Pi$ -calculus to canonical terms. She defines a term as canonical if it is in β -normal form and every variable is fully applied with respect to Γ . A variable x is fully applied with respect to a context Γ if it occurs in a subterm of the form $xM_1 \dots M_n$, where n is the arity of x . An object P is pre-canonical if its β -normal form is canonical. Felty shows that canonical LF can be embedded into full LF, *i.e.*, the $\lambda\Pi$ -calculus and full LF restricted to pre-canonical terms is equivalent to canonical LF. This system is useful for representing logics since one does not need to restrict the representation theorems to terms in canonical forms. The result of Salvesen (1990) means that it is now simpler to work with the $\lambda\Pi$ -calculus than this system.

2.2 An Algebraic Presentation

We give a brief account of the algebraic organization of the $\lambda\Pi$ -calculus. Presentation of similar systems can be found in, for example, (Cartmell 1986), (Cartmell 1990), (Streicher 1989), (Jacobs 1991), (Hofmann 1996) and (Pitts 2000). We let $|E|$, where E ranges over the grammatical expressions of the $\lambda\Pi$ -calculus ((Harper et al. 1993) and (Ishtiaq & Pym 2002)), denote the equivalence class of E with respect to provable $\alpha\beta\eta$ -equality. Where no confusion can occur, we shall omit the brackets $|-\cdot|$. We begin by defining the base category of contexts and realizations.

Definition 2.2 (category of contexts and realizations)

Let Σ be a $\lambda\Pi$ -signature. The (*base*) *category* $\mathcal{B}(\Sigma)$ of contexts and realizations is defined as follows:

Objects: contexts Γ such that $\vdash_{\Sigma} \Gamma$ context;

Arrows: realizations $\Gamma \xrightarrow{\langle M_1, \dots, M_n \rangle} \Delta$, such that, for each $1 \leq i \leq n$, \mathbf{N} proves $\Gamma \vdash_{\Sigma} M_i : B_i [M_j / y_j]_{j=1}^{i-1}$, where $\Delta = y_1 : B_1, \dots, y_n : B_n$.

- Identities, written $1_{\Gamma(\equiv x_1:A_1, \dots, x_m:A_m)}$, are $x_1 : A_1, \dots, x_m : A_m \xrightarrow{\langle x_1, \dots, x_m \rangle} x_1 : A_1, \dots, x_m : A_m$.
- Composition is defined as follows: if $\sigma = \Gamma \xrightarrow{\langle M_1, \dots, M_n \rangle} \Delta$ and $\rho = \Delta \xrightarrow{\langle N_1, \dots, N_p \rangle} \Theta$ are arrows in $\mathcal{B}(\Sigma)$ then their composition $\sigma; \rho = \Gamma \xrightarrow{\langle N_1[M_j/y_j]_{j=1}^n, \dots, N_p[M_j/y_j]_{j=1}^n \rangle} \Theta$. ■

We now proceed to define the appropriate indexed category. The key idea here is that we have a category over each context which classifies the dependence of the assertions $M : A$ and $A : \text{Type}$ on that context.

Definition 2.3

Let Σ be a $\lambda\Pi$ -signature. We define a *strict indexed category* $\mathcal{E}(\Sigma)$ over the base category $\mathcal{B}(\Sigma)$,

$$\mathcal{E}(\Sigma) : \mathcal{B}(\Sigma)^{op} \rightarrow \mathcal{Cat},$$

where \mathcal{Cat} denotes the category of all small categories and functors, as follows:

- For each object Γ in $\mathcal{B}(\Sigma)$, the category $\mathcal{E}(\Sigma)(\Gamma)$ is defined as follows:

Objects: Types A such that \mathbf{N} proves $\Gamma \vdash_{\Sigma} A : \text{Type}$;

Arrows: $A \xrightarrow{M} B$ such that $\Gamma, x : A \xrightarrow{\langle 1_{\Gamma}, M \rangle} \Gamma, y : B$ is an arrow in $\mathcal{B}(\Sigma)$.

- Identities are arrows $A \xrightarrow{1_A} A$ such that $\Gamma, x : A \xrightarrow{\langle 1_{\Gamma}, 1_A \rangle} \Gamma, x : A$ is an arrow in $\mathcal{B}(\Sigma)$.

- Composition is defined as follows: if $A \xrightarrow{M} B$ and $B \xrightarrow{N} C$ are arrows in $\mathcal{E}(\Sigma)(\Gamma)$ then their composition $A \xrightarrow{M;N} C$ is $A \xrightarrow{N[M/y]} C$, where N is such that $\Gamma, y: B \xrightarrow{\langle 1_\Gamma, N \rangle} \Gamma, z: C$ is an arrow in $\mathcal{B}(\Sigma)$ for appropriate x and y .

- For each arrow $\Gamma' \xrightarrow{\sigma} \Gamma$ in $\mathcal{B}(\Sigma)$, $\mathcal{E}(\Sigma)(\sigma)$ is a functor $\mathcal{E}(\Sigma)(\Gamma) \xrightarrow{\sigma^*} \mathcal{E}(\Sigma)(\Gamma')$ which sends an object A of $\mathcal{E}(\Sigma)(\Gamma)$ to an object $A[\sigma/\vec{x}]$ of $\mathcal{E}(\Sigma)(\Gamma')$, where $A[\sigma/\vec{x}] = A[M_i/x_i]_{i=1}^n$ when $\sigma = \langle M_1, \dots, M_n \rangle$ and $\Gamma = x_1:A_1, \dots, x_n:A_n$. Similarly, σ^* sends an arrow $A \xrightarrow{M} B$ in $\mathcal{E}(\Sigma)(\Gamma)$ to an arrow $A[\sigma/\vec{x}] \xrightarrow{M[\sigma/\vec{x}]} B[\sigma/\vec{x}]$ in $\mathcal{E}(\Sigma)(\Gamma')$, where $M[\sigma/\vec{x}] = M[M_i/x_i]_{i=1}^n$ when $\sigma = \langle M_1, \dots, M_n \rangle$ and $\Gamma = x_1:A_1, \dots, x_n:A_n$. \blacksquare

We note that functoriality for $\mathcal{E}(\Sigma)$ follows from composition in the base category. We compare $\mathcal{E}(\Sigma)$ with Cartmell's (1986) category $\mathcal{R}(U)$ of contexts and realizations of a theory U . Cartmell does not have our indexed structure, he uses pullbacks to obtain an arrow analogous to the functor $\mathcal{E}(\Sigma)(\Gamma) \xrightarrow{\sigma^*} \mathcal{E}(\Sigma)(\Gamma')$. If we were to collapse our indexed structure onto the base category, we would have a contextual category.

In anticipation of constructions to follow later, *q.v.*, § 3.8 and § 4, we observe that we can extend the classifying category to permit, respectively, realizations and families of types in the fibres. We begin by extending our definition to allow us to classify realizations in the fibres.

Definition 2.4 (Contextual Fibres)

The indexed category $\mathcal{E}(\Sigma)$ can be extended to a strict *indexed category* $\mathcal{C}on(\Sigma)$ as follows:

- The base of $\mathcal{C}on(\Sigma)$ is $\mathcal{B}(\Sigma)$, just as in $\mathcal{E}(\Sigma)$;
- The objects of the fibre over Γ are arrows $\Delta \xrightarrow{\sigma} \Theta$, where $\Gamma, \Delta \xrightarrow{\langle 1_\Gamma, \sigma \rangle} \Gamma, \Theta$ is an arrow on $\mathcal{B}(\Sigma)$ (we require that variables are standardized apart);
- The arrows of the fibre of $\mathcal{C}on(\Sigma)$ over Γ are pairs (h, k) such that the diagram

$$\begin{array}{ccc}
 \Delta & \xrightarrow{\sigma} & \Theta \\
 \downarrow h & & \downarrow k \\
 \Psi & \xrightarrow{\sigma'} & \Phi
 \end{array}$$

commutes, where $\Delta \xrightarrow{h} \Psi$ and $\Theta \xrightarrow{k} \Phi$ are objects of $\mathcal{C}on(\Sigma)$ over Γ ;

- For each $\Gamma' \xrightarrow{\sigma'} \Gamma$, $\mathcal{C}on(\Sigma)(\sigma)$ is a functor $\mathcal{C}on(\Sigma)(\Gamma) \xrightarrow{\sigma^*} \mathcal{C}on(\Sigma)(\Gamma')$ defined by the commuting square

$$\begin{array}{ccc}
\sigma^*\Delta & \xrightarrow{\sigma^*\rho} & \sigma^*\Theta \\
\sigma^*h \downarrow & & \downarrow \sigma^*k \\
\sigma^*\Psi & \xrightarrow{\sigma^*\rho'} & \sigma^*\Phi
\end{array}$$

which gives the action on arrows (h, k) as above and σ^* is the usual substitution. ■

Before we can extend the definition of $\mathcal{E}(\Sigma)$ to families of types in the fibres, we need to recall the definition of the category of families of sets.

Definition 2.5 (Families of Sets)

The *category $\mathcal{F}am$ of families of sets* is defined as follows:

Objects: ordered pairs, $F = (B, E)$, where B is a set and $E = (E_b)_{b \in B}$ is a family of sets indexed by elements of B ;

Arrows: if $F = (B, E)$ and $F' = (B', E')$ are objects of $\mathcal{F}am$, then an arrow from F to F' is an ordered pair (β, ϵ) , where $\beta: B \rightarrow B'$ is a function and $\epsilon = (\epsilon_b)_{b \in B}$ is a family of functions $\epsilon_b: E_b \rightarrow E'_{\beta(b)}$.

Given two arrows $(B, E) \xrightarrow{(\beta, \epsilon)} (C, G)$ and $(C, G) \xrightarrow{(\gamma, \delta)} (D, H)$, their composition is a function $\gamma \circ \beta$ and a family of functions $\delta_{\beta(b)} \circ \epsilon_b$ determined by function composition for each $b \in B$. The identity is the identity function paired with a family of identity functions. ■

This definition tells us how to extend $\mathcal{E}(\Sigma)$ to a category $\mathcal{F}am(\Sigma): \mathcal{B}(\Sigma)^{op} \rightarrow \mathcal{F}am$ which will capture families of types in the fibres. Note that whereas $\mathcal{E}(\Sigma)$ is valued in $\mathcal{C}at$, $\mathcal{F}am(\Sigma)$ is valued in $\mathcal{F}am$. This category is well-known, see; for example, (Hofmann 1996) and (Dybjer 1995).

Definition 2.6 (Familial Fibres)

The indexed category $\mathcal{E}(\Sigma)$ can be extended to a *category $\mathcal{F}am(\Sigma): \mathcal{B}(\Sigma)^{op} \rightarrow \mathcal{F}am$* as follows:

- Over each object Γ in $\mathcal{B}(\Sigma)$, there is a family $(\mathbf{Ty}(\Gamma), \mathbf{Tm}(\Gamma, A)_{A \in \mathbf{Ty}(\Gamma)})$, where $\mathbf{Ty}(\Gamma)$ is the set of well-formed types in the context Γ (and signature Σ) and each $\mathbf{Tm}(\Gamma, A)$ is the set of well-formed terms of type A in the context Γ (and signature Σ);

- Each arrow $\Gamma' \xrightarrow{\sigma} \Gamma'$ defines an arrow in \mathcal{Fam} defined by pointwise substitution. ■

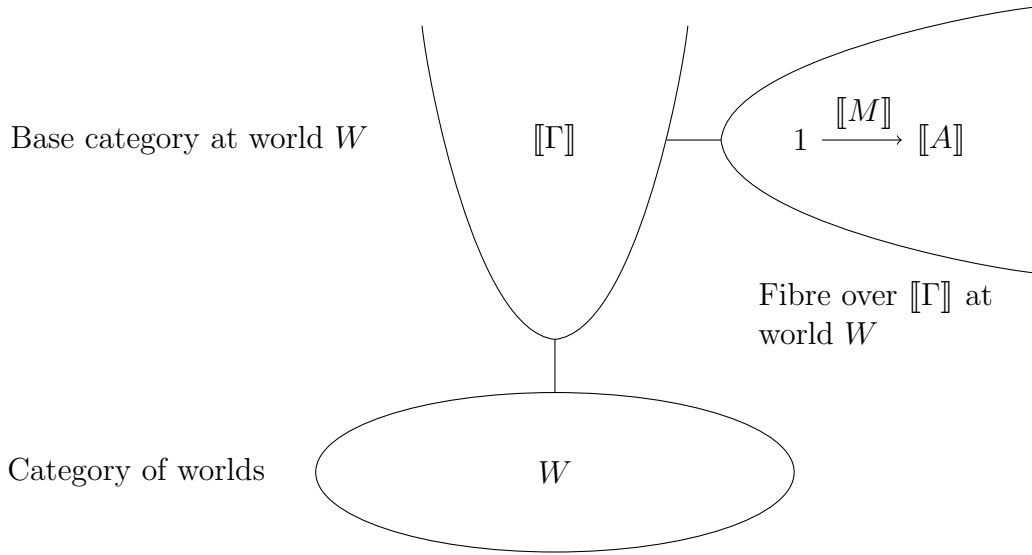
Chapter 3

Kripke Models of the $\lambda\Pi$ -calculus

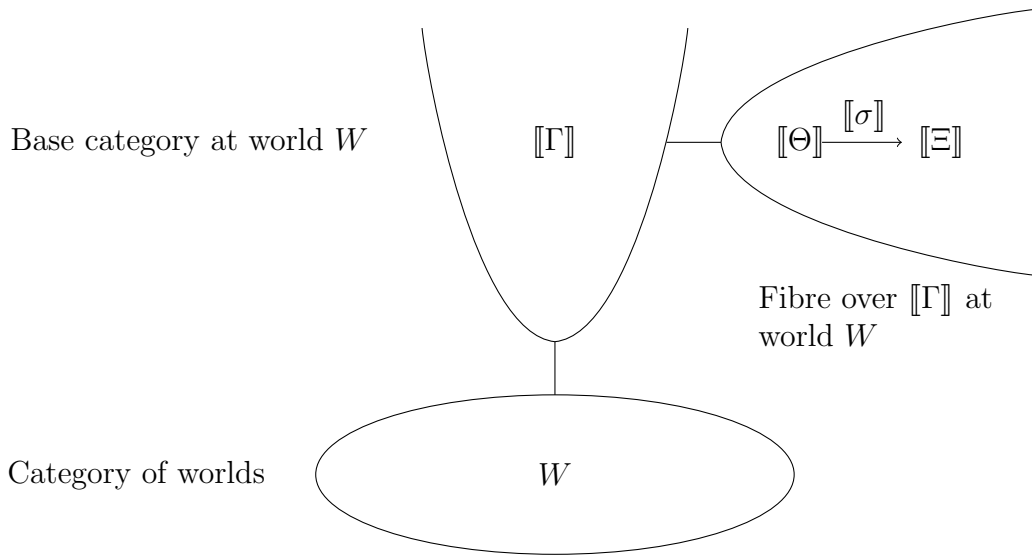
In this chapter, we define a (categorical) model of the $\lambda\Pi$ -calculus. The $\lambda\Pi$ -calculus is in propositions-as-types correspondence with the $\{\forall, \supset\}$ -fragment of minimal first-order logic and so to describe it in terms of Kripke models is a natural choice. Further motivation for the choice of Kripke models arises from the work of Mitchell & Moggi (1991). Their Kripke lambda models provide a general class of models for the simply typed λ -calculus. Our models are generalizations of theirs to the dependently typed setting. Gallier (1997) has also extended Mitchell and Moggi's Kripke lambda models to the second-order λ -calculus. His models, however, are more concrete than ours; that is, not categorical, and so not as general. His models are intended to model inequalities as well as equalities, while our models do not allow us to model inequalities.

It is well-known ((Cartmell 1986), (Streicher 1989), (Cartmell 1990), (Jacobs 1991), (Ritter 1992), (Jacobs 1993), (Hofmann 1996), (Jacobs 1999) and (Pitts 2000)) and that (strict) indexed categories provide a suitable model for dependent type theories, like the $\lambda\Pi$ -calculus. One can view the context Γ in the assertion $\Gamma \vdash_{\Sigma} M : A$ as being an index for $M : A$. Thus $M : A$ depends on Γ for its meaning, and we are led to the technology of an indexed category (Paré & Schumacher 1978). Given an indexed category $\mathcal{C}^{op} \rightarrow \mathcal{C}at$, the context Γ is interpreted as a (choice of) object in the base category \mathcal{C} , the type A is interpreted as an object in the fibre over (the interpretation of) Γ and the assertion $\Gamma \vdash_{\Sigma} M : A$ is interpreted as the arrow from the terminal object in the category over Γ to (the interpretation of) A .

Our Kripke models of the $\lambda\Pi$ -calculus combine the Kripke lambda models of Mitchell and Moggi with an indexed structure. Pictorially, we have:



This picture describe a Kripke $\lambda\Pi$ -prestructure in which we can interpret assertions of the form $\Gamma \vdash_{\Sigma} M : A$ at a given world W . We wish to also be able to interpret realizations; that is, assertions of the form $\Gamma \vdash_{\Sigma} \Delta \overset{\sigma}{\rightarrow} \Delta$. We achieve this by moving to a Kripke $\lambda\Pi$ -prestructure.



A Kripke model of the $\lambda\Pi$ -calculus is defined to be a Kripke $\lambda\Pi$ -structure together with an interpretation function.

After examples, we turn to soundness and completeness of Kripke models of the with respect to the $\lambda\Pi$ -calculus. These results are fairly routine. Since our Kripke models of the $\lambda\Pi$ -calculus are able to interpret realizations, we define a calculus of realizers, \mathbf{C} , and provide soundness and completeness results.

The majority of the work in this chapter involves adapting existing material and results for our purposes. So while it is known that the results hold, no one

has actually worked out the details in quite this way before. The Kripke Σ - $\lambda\Pi$ -models are similar to the Kripke resource models of (Ishtiaq & Pym 2002). The calculus of realizers, Definition 3.43 is taken from (Galmiche & Pym 2000).

3.1 Kripke $\lambda\Pi$ -Prestructures and -Structures

As mentioned above, our Kripke $\lambda\Pi$ -prestructure is intended to interpret assertions of the form $\Gamma \vdash_{\Sigma} M : A$ at a world W . Our models are based on the contextual categories of Cartmell (1986) and their reformulation by Ritter (1992) and Pitts (2000).

Definition 3.1 (Kripke $\lambda\Pi$ -prestructure)

A *Kripke $\lambda\Pi$ -prestructure* \mathcal{J} is a functor

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

where \mathcal{W} is a small category (“of worlds”), $\mathcal{D}^{op} = \coprod_{W \in |\mathcal{W}|} \mathcal{D}_W^{op}$, where W ranges over each object W in \mathcal{W} and each category \mathcal{D}_W (the base at W) is small and \mathcal{V} , a subcategory of $\mathcal{C}at$, is a category of values, such that:

1. Each \mathcal{D}_W has a terminal object $1_{\mathcal{D}_W}$;
2. Each $\mathcal{J}(W)(D)$, where W is an object of \mathcal{W} and D is an object of \mathcal{D}_W , has a terminal object $1_{\mathcal{J}(W)(D)}$, preserved on the nose by each functor $f^*(= \mathcal{J}(W)(f)) : \mathcal{J}(W)(D) \rightarrow \mathcal{J}(W)(E)$, where $f : E \rightarrow D$ is an arrow in \mathcal{D}_W ;
3. For each object W in \mathcal{W} , D in \mathcal{D}_W , A in $\mathcal{J}(W)(D)$, there is a $D \bullet A$ in \mathcal{D}_W together with canonical projections $D \bullet A \xrightarrow{p_{D,A}} D$ in \mathcal{D}_W and $1_{\mathcal{J}(W)(D \bullet A)} \xrightarrow{q_{D,A}} p_{D,A}^*(A)$ and canonical pullbacks

$$\begin{array}{ccc}
 E \bullet f^* A & \xrightarrow{f \bullet A} & D \bullet A \\
 \downarrow p_{E, f^* A} & \lrcorner & \downarrow p_{D, A} \\
 E & \xrightarrow{f} & D
 \end{array}$$

in \mathcal{D}_W satisfying the strictness conditions

$$\text{id}_D^*(A) = A$$

and

$$\text{id} \bullet A = \text{id}_{D \bullet A},$$

for each A in $\mathcal{J}(W)(D)$, and that

$$g^*(f^*(A)) = (g; f)^*(A)$$

and

$$(g \bullet (f^*(A))); (f \bullet A) = (g; f) \bullet A,$$

for each appropriate A , f and g . Moreover, for each W and D ,

$$D \bullet 1_{\mathcal{J}(W)(D)} = D;$$

4. At each object W in \mathcal{W} , the arrow $p_{D,A}^*(= \mathcal{J}(W)(p_{D,A}))$ has a right adjoint

$$p_{D,A}^* \dashv \Pi_{D,A}: \mathcal{J}(W)(D \bullet A) \rightarrow \mathcal{J}(W)(D)$$

that satisfies the following (strict) Beck-Chevalley condition: for each $E \xrightarrow{f} D$ in \mathcal{D}_W , each A in $\mathcal{J}(W)(D)$ and each B in $\mathcal{J}(W)(D \bullet A)$

$$f^*(\Pi_{D,A} B) = \Pi_{E, f^* A} ((f \bullet A)^* B)$$

and

$$(f \bullet A)^*(\text{app}_W(A, B)) = \text{app}_W(f^* A, (f \bullet A)^* B)$$

where app_W is the co-unit of the adjunction.

5. For each arrow $W \xrightarrow{\alpha} W'$ in \mathcal{W} , then

- (a) there is a functor $\mathcal{K}^\alpha: \mathcal{D}_W \rightarrow \mathcal{D}_{W'}$; and
- (b) $\mathcal{J}(W)(D) = \mathcal{J}(W')(D)$ for all D in both \mathcal{D}_W and $\mathcal{D}_{W'}$; otherwise $\mathcal{J}(W')(D)$ is undefined.

Where no confusion can arise, we shall write just $\mathcal{D}^{(op)}$ instead of $\mathcal{D}_W^{(op)}$. ■

A few remarks concerning this definition are in order.

- We take the co-product of the categories \mathcal{D}_W because it is conceptually natural. It is analogous to the Kripke model for intuitionistic logic where there is a classical model at each world. For similar reasons, we use the category $[\mathcal{W}, [\mathcal{D}^{op}, \mathcal{V}]]$ rather than use the simpler category $[\mathcal{W} \times \mathcal{D}^{op}, \mathcal{V}]$.
- In (Ritter 1992), the context-extension operator \bullet arises from the Grothendieck construction ((Jacobs 1999)). We have avoided using this construction by using the pullback structure found in (Pitts 2000).

- The requirement of (4) amounts to the existence of a natural isomorphism,

$$cur_W : \text{hom}_{\mathcal{J}(W)(D \bullet A)}(p_{D,A}^* C, B) \cong \text{hom}_{\mathcal{J}(W)(D)}(C, \Pi_{D,A}(B)) : cur_W^{-1}, \quad (3.1)$$

where B is an object in $\mathcal{J}(W)(D \bullet A)$ and C is an object in $\mathcal{J}(W)(D)$ with the co-unit of the adjunction, the application map,

$$app_W : p_{D,A}^* \Pi_{D,A} \Rightarrow 1_{\mathcal{J}(W)(D \bullet A)}$$

given by arrows

$$p_{D,A}^* \Pi_{D,A}(B) \xrightarrow{app_W(A,B)} B \quad (3.2)$$

in $\mathcal{J}(W)(D \bullet A)$.

The Kripke $\lambda\Pi$ -prestructure provides enough structure to allow us to interpret assertions of the form $\Gamma \vdash_{\Sigma} M : A$, cf., Definition 2.3. However, we are also interested in realizations and hence assertions of the form $\Gamma \vdash_{\Sigma} \Delta \rightarrow \Theta$. This is mirrored in the move from a Kripke $\lambda\Pi$ -prestructure to a Kripke $\lambda\Pi$ -structure. The fibres $\mathcal{K}_{\mathcal{J}}(W)(D)$ use components of the fibres in the Kripke $\lambda\Pi$ -prestructure: arrows in $\mathcal{J}(W)(D)$ are taken to be objects in $\mathcal{K}_{\mathcal{J}}(W)(D)$.

Definition 3.2 (Kripke $\lambda\Pi$ -structure)

Let \mathcal{J} be a Kripke $\lambda\Pi$ -prestructure, $\mathcal{J} : [\mathcal{W}, [\mathcal{D}^{op}, \mathcal{V}]]$. A *Kripke $\lambda\Pi$ -structure on \mathcal{J}* is a functor

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

such that the category $\bar{\mathcal{V}}$ has the following properties:

Objects: Categories $\bar{V} = \mathcal{K}_{\mathcal{J}}(W)(E)$ built out of $V = \mathcal{J}(W)(E)$ with

Objects: Arrows

$$\bar{A} \xrightarrow{f_{\bar{A}, \bar{B}}} \bar{B}$$

of \mathcal{D}_W , where $\bar{A} = A_1 \bullet \dots \bullet A_m$ and each A_i is an object of $\mathcal{J}(W)(E \bullet A_1 \bullet \dots \bullet A_{i-1})$ and $\bar{B} = B_1 \bullet \dots \bullet B_n$ and each B_i is an object of $\mathcal{J}(W)(E \bullet B_1 \bullet \dots \bullet B_{i-1})$ such that $E \bullet \bar{A} \xrightarrow{\langle 1_E, f_{\bar{A}, \bar{B}} \rangle} E \bullet \bar{B}$ is an arrow of \mathcal{D}_W ;

Arrows: Arrows

$$(\bar{A} \xrightarrow{f_{\bar{A}, \bar{B}}} \bar{B}) \rightarrow (\bar{C} \xrightarrow{f_{\bar{C}, \bar{D}}} \bar{D})$$

are given by $E \bullet \bar{A} \xrightarrow{\langle 1_E, f_{\bar{A}, \bar{C}} \rangle} E \bullet \bar{C}$ in \mathcal{D}_W .

Arrows: Functors $\mathcal{K}_{\mathcal{J}}(W)(f)$, where $E \xrightarrow{f} D$ is an arrow in \mathcal{D}_W , defined as follows. Let $\mathbf{A} = A_1 \bullet \dots \bullet A_m \xrightarrow{f_{\bar{A}, \bar{B}}} B_1 \bullet \dots \bullet B_n$ be an arrow in $\mathcal{K}_{\mathcal{J}}(W)(D)$

and $\mathbf{C} = C_1 \bullet \dots \bullet C_m \xrightarrow{f_{\bar{C}, \bar{D}}} D_1 \bullet \dots \bullet D_n$ be an arrow in $\mathcal{K}_{\mathcal{J}}(W)(E)$. The functor $\mathcal{K}_{\mathcal{J}}(W)(f)$ satisfies the following conditions:

1. $\mathcal{K}_{\mathcal{J}}(W)(f)(\mathbf{A}) = \mathbf{C}$, where $C_1 = f^*(A_1)$ and for each

$$C_i = (\dots ((f \bullet A_1) \bullet A_2) \dots \bullet A_{i-1})^*(C_i);$$

2. If $\bar{A} \xrightarrow{\mu} \bar{A}'$ is an arrow in $\mathcal{K}_{\mathcal{J}}(W)(D)$, then $\mathcal{K}_{\mathcal{J}}(W)(f)(\mu) = \nu$, the unique mediating arrow determined by the pullback

$$\begin{array}{ccc}
 E \bullet f^* \bar{A} & \xrightarrow{f \bullet \bar{A}} & D \bullet \bar{A} \\
 \downarrow \nu (=_{\text{def}} f^* \mu) & & \downarrow \mu \\
 E \bullet f^* \bar{A}' & \xrightarrow{f \bullet \bar{A}'} & D \bullet \bar{A}' \\
 \downarrow \bar{p}_{E, f^* \bar{A}} & & \downarrow \bar{p}_{D, \bar{A}'} \\
 E & \xrightarrow{f} & D
 \end{array}$$

where each $\bar{\quad}$ denotes the obvious composite. ■

We refer to a Kripke $\lambda\Pi$ -structure rather than a Kripke $\lambda\Pi$ -structure on \mathcal{J} when no confusion can arise by our doing so.

3.2 Examples

We provide some suitable examples of Kripke $\lambda\Pi$ -structures. We begin with the simplest example, the term model.

3.2.1 Term Model

We begin by defining a posetal subcategory of $\mathcal{B}(\Sigma)$ defined in Definition 2.2.

Definition 3.3

The category $\mathcal{P}(\Sigma)$ is defined as follows:

Objects: The empty context, $\langle \rangle$, is an object of $\mathcal{P}(\Sigma)$. If Γ is an object of $\mathcal{P}(\Sigma)$ and there exists an arrow $\Gamma \xrightarrow{\sigma} \Gamma, \Gamma'$ in $\mathcal{B}(\Sigma)$, then Γ, Γ' is an object of $\mathcal{P}(\Sigma)$;

Arrows: $\Gamma \rightarrow \Delta$ if and only if there is an arrow $\Gamma \rightarrow \Delta$ in $\mathcal{B}(\Sigma)$ and $\Delta \equiv \Gamma, \Gamma'$ for some (possibly $\langle \rangle$) Γ' . ■

We wish to extend a context Γ by Δ whilst ensuring that variable names are kept consistent and any duplications are removed, we define this operation as follows.

Definition 3.4 (Consistent Merge)

Let Γ and Δ be valid $\lambda\Pi$ -contexts. The *consistent merge* of Γ and Δ , $\Gamma \bowtie \Delta$, is defined as follows:

- If $Dom(\Gamma) \cap Dom(\Delta) = \emptyset$, then $\Gamma \bowtie \Delta \equiv \Gamma, \Delta$;
- If $Dom(\Gamma) \cap Dom(\Delta) \neq \emptyset$ then we rename all occurrences in Δ of each variable in $Dom(\Gamma) \cap Dom(\Delta)$, taking care with dependencies, and define $\Gamma \bowtie \Delta \equiv \Gamma, \Delta'$, where $\Delta' =_{\alpha} \Delta$. ■

We are now in a position to define the base category at each world Γ , \mathcal{C}_{Γ} .

Definition 3.5

The category \mathcal{C}_{Γ} , for each Γ in $\mathcal{P}(\Sigma)$ is defined as follows:

Objects: $\Gamma \bowtie \Delta$, where Δ is an object of $\mathcal{B}(\Sigma)$.

Arrows: $\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma}, M_1, \dots, M_n \rangle} \Gamma \bowtie \Delta'$, where $\Delta \xrightarrow{\langle M_1, \dots, M_n \rangle} \Delta'$ is an arrow in $\mathcal{B}(\Sigma)$.

Given arrows $\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma}, M_1, \dots, M_n \rangle} \Gamma \bowtie \Delta'$ and $\Gamma \bowtie \Delta' \xrightarrow{\langle 1_{\Gamma}, N_1, \dots, N_p \rangle} \Gamma \bowtie \Delta''$, their composition is $\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma}, N_1[M_j/y_j]_{j=1}^n, \dots, N_p[M_j/y_j]_{j=1}^n \rangle} \Gamma \bowtie \Delta''$. ■

We define $\mathcal{A}^{op} = \coprod_{\Gamma \in |\mathcal{P}(\Sigma)|} \mathcal{C}_{\Gamma}$ and we are now able to define the indexed category $\mathcal{T}_{\Sigma}: [\mathcal{P}(\Sigma), [\mathcal{A}(\Sigma)^{op}, \mathcal{V}]]$ which we then show is a Kripke $\lambda\Pi$ -prestructure.

Definition 3.6

Let Σ be a $\lambda\Pi$ -signature. The functor category $\mathcal{T}_{\Sigma}: [\mathcal{P}(\Sigma), [\mathcal{A}(\Sigma)^{op}, \mathcal{V}]]$ is defined as follows:

- if $\Gamma \rightarrow \Gamma'$ is an arrow in $\mathcal{P}(\Sigma)$, then we define a natural transformation $\mathcal{T}_{\Sigma}(\Gamma) \Rightarrow \mathcal{T}_{\Sigma}(\Gamma')$. Since Γ' is an extension of Γ , we have inclusions $\mathcal{I}_{\Delta}: \mathcal{T}_{\Sigma}(\Gamma)(\Gamma \bowtie \Delta) \rightarrow \mathcal{T}_{\Sigma}(\Gamma')(\Gamma' \bowtie \Delta)$ for each Δ . We take these as our components:

$$\begin{array}{ccc}
\Gamma \bowtie \Delta & \mathcal{T}_{\Sigma}(\Gamma)(\Gamma \bowtie \Delta) \xrightarrow{\mathcal{I}_{\Delta}} \mathcal{T}_{\Sigma}(\Gamma')(\Gamma' \bowtie \Delta) & \\
\sigma \downarrow & \downarrow \mathcal{T}_{\Sigma}(\Gamma)(\sigma) & \downarrow \mathcal{T}_{\Sigma}(\Gamma')(\sigma) \\
\Gamma \bowtie \Delta' & \mathcal{T}_{\Sigma}(\Gamma)(\Gamma \bowtie \Delta') \xrightarrow{\mathcal{I}_{\Gamma'}} \mathcal{T}_{\Sigma}(\Gamma')(\Gamma' \bowtie \Delta') & \\
& & \\
& & \downarrow \mathcal{T}_{\Sigma}(\Gamma')(\sigma)
\end{array}$$

- If Γ is an object of $\mathcal{P}(\Sigma)$ and $\Gamma \bowtie \Delta$ is an object in $\mathcal{A}(\Sigma)$. The fibre $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$ is a category defined as follows:

Objects: Types A such that $\Gamma \bowtie \Delta \vdash_\Sigma A : \text{Type}$;

Arrows: $A \xrightarrow{M} B$, where the arrow M is such that $\Gamma \bowtie \Delta, x : A \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, M \rangle} \Gamma \bowtie \Delta, y : B$ is an arrow in \mathcal{C}_Γ and if $x : A$ or $y : B$ are not in $\Gamma \bowtie \Delta$ then $x, y \notin \text{Dom}(\Gamma \bowtie \Delta)$. If $A \xrightarrow{M} B$ and $B \xrightarrow{N} C$, then $A \xrightarrow{M;N} C$ is given by $A \xrightarrow{M[N/x]} C$, where N is such that $\Gamma \bowtie \Delta, y : B \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, N \rangle} \Gamma \bowtie \Delta, z : C$.

- If σ is an arrow $\Gamma \bowtie \Delta' \rightarrow \Gamma \bowtie \Delta$ in \mathcal{C}_Γ , then we have a functor $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)(\sigma) = \sigma^*$ between $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$ and $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta')$. The functor σ^* is defined in terms of substitution, *i.e.*, an arrow $A \xrightarrow{M} B$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$ is sent to $A[\sigma/\vec{x}] \xrightarrow{M[\sigma/\vec{x}]} B[\sigma/\vec{x}]$ such that $\Gamma \bowtie \Delta', x : A[\sigma/\vec{x}] \xrightarrow{\langle 1_{\Gamma \bowtie \Delta'}, M[\sigma/\vec{x}] \rangle} \Gamma \bowtie \Delta', y : B[\sigma/\vec{x}]$. ■

Lemma 3.7

\mathcal{T}_Σ is a Kripke $\lambda\Pi$ -prestructure.

Proof It is straightforward to verify that \mathcal{T}_Σ is a functor. We show that it satisfies the other conditions.

1. Each \mathcal{C}_Γ contains a terminal object: $\Gamma \bowtie \emptyset \equiv \Gamma$.
2. Each $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$ has a terminal object the unit type $\langle \rangle$. Let $\Gamma \bowtie \Delta' \xrightarrow{\sigma} \Gamma \bowtie \Delta$ be an arrow in \mathcal{C}_Γ , then $\sigma^*(\langle \rangle) = \langle \rangle[\sigma/\vec{x}] = \langle \rangle$. Thus the terminal object is preserved on the nose by every appropriate arrow σ in \mathcal{C}_Γ .
3. Let Γ be a object in $\mathcal{P}(\Sigma)$, Δ be an object in \mathcal{C}_Γ and A be an object in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$. $\Gamma \bowtie \Delta \bullet A = \Gamma \bowtie \Delta \bowtie x : A$ is an object in \mathcal{C}_Γ . We have a projection $\Gamma \bowtie \Delta, x : A \xrightarrow{p_{\Gamma \bowtie \Delta, A}} \Gamma$ in \mathcal{C}_Γ and a projection $1 \xrightarrow{q_{\Gamma \bowtie \Delta, A}} p_{\Gamma \bowtie \Delta, A}^*(A)$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$. We observe that $p_{\Gamma \bowtie \Delta, A}^*(A) = A[\vec{x}/\vec{x}] = A$.

Claim 3.8

Let Γ be a object of $\mathcal{P}(\Sigma)$, $\Gamma \bowtie \Delta \equiv \Gamma, x_1 : A_1, \dots, x_m : A_m \xrightarrow{\langle \text{id}_\Gamma, M_1, \dots, M_n \rangle} \Gamma \bowtie \Delta' \equiv \Gamma, y_1 : B_1, \dots, y_n : B_n$ be a morphism in \mathcal{C}_Γ and A be an object in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$. Then

$$\begin{array}{ccc}
 \Gamma \bowtie \Delta, y : A[M_j/y_j]_{j=1}^n & \xrightarrow{\langle \text{id}_\Gamma, M_1, \dots, M_n, y \rangle} & \Gamma \bowtie \Delta', x : A \\
 \downarrow p_{\Gamma \bowtie \Delta, A[M_j/y_j]_{j=1}^n} & & \downarrow p_{\Gamma \bowtie \Delta', A} \\
 \Gamma \bowtie \Delta & \xrightarrow{\langle \text{id}_\Gamma, M_1, \dots, M_n \rangle} & \Gamma \bowtie \Delta'
 \end{array}$$

is a pullback.

A similar proof can be in (Pitts 2000).

Proof Suppose we have arrows $\Gamma \bowtie \Delta'' \xrightarrow{\sigma'} \Gamma \bowtie \Delta$ and $\Gamma \bowtie \Delta'' \xrightarrow{\sigma''} \Gamma \bowtie \Delta', x:A$ in \mathcal{C}_Γ satisfying $\langle M_1, \dots, M_n, \text{id}_\Delta \rangle \circ \sigma' = p_{\Gamma \bowtie \Delta', A} \circ \sigma'' : \Gamma \bowtie \Delta'' \rightarrow \Gamma \bowtie \Delta'$, *i.e.*, the diagram

$$\begin{array}{ccc} \Gamma \bowtie \Delta'' & \xrightarrow{\sigma''} & \Gamma \bowtie \Delta', x:A \\ \sigma' \downarrow & & \downarrow p_{\Gamma \bowtie \Delta', A} \\ \Gamma \bowtie \Delta & \xrightarrow{\langle M_1, \dots, M_n, \text{id}_\Delta \rangle} & \Gamma \bowtie \Delta' \end{array}$$

commutes. We have to show that there is a unique arrow $\delta : \Gamma \bowtie \Delta'' \rightarrow \Gamma \bowtie \Delta, x' : A[M_j/y_j]_{j=1}^n$ satisfying $p_{\Gamma \bowtie \Delta, A[M_j/y_j]_{j=1}^n} \circ \delta = \sigma' : \Gamma \bowtie \Delta'' \rightarrow \Gamma \bowtie \Delta$ and $\langle \text{id}_\Gamma, M_1, \dots, M_n, x' \rangle \circ \delta = \sigma'' : \Gamma \bowtie \Delta'' \rightarrow \Gamma \bowtie \Delta', x:A$, *i.e.* the triangles

$$\begin{array}{ccc} \Gamma \bowtie \Delta'' & \xrightarrow{\delta} & \Gamma \bowtie \Delta, x:A[M_j/y_j]_{j=1}^n \\ & \searrow \sigma' & \downarrow p_{\Gamma \bowtie \Delta, A[M_j/y_j]_{j=1}^n} \\ & & \Gamma \bowtie \Delta \end{array}$$

and

$$\begin{array}{ccc} \Gamma \bowtie \Delta'' & & \\ \delta \downarrow & \searrow \sigma'' & \\ \Gamma \bowtie \Delta, x' : A[M_j/y_j]_{j=1}^n & \xrightarrow{\langle \text{id}_\Gamma, M_1, \dots, M_n, x' \rangle} & \Gamma \bowtie \Delta', x:A \end{array}$$

commute. Since $\langle \text{id}_\Gamma, M_1, \dots, M_n \rangle \circ \sigma' = p_{\Gamma \bowtie \Delta, A} \circ \sigma'' : \Gamma \bowtie \Delta'' \rightarrow \Gamma \bowtie \Delta'$, we must have that σ'' is of the form $\langle \text{id}_\Gamma, M_1[N_j/x_j]_{j=1}^m, \dots, M_n[N_j/x_j]_{j=1}^m, N \rangle$ where $\sigma' = \langle \text{id}_\Gamma, N_1, \dots, N_m \rangle$ and N is a term such that \mathbf{N} proves $\Gamma \bowtie \Delta'' \vdash_\Sigma N : A[M_i[N_j/x_j]_{j=1}^m/y_i]_{i=1}^n$. Now since $A[M_i[N_j/x_j]_{j=1}^m/y_i]_{i=1}^n = A[M_i/y_i]_{i=1}^n[N_j/x_j]_{j=1}^m$, we get a morphism $\delta =_{def} \langle \text{id}_\Gamma, N_1, \dots, N_m, N \rangle :$

$\Gamma \bowtie \Delta \rightarrow \Gamma \bowtie \Delta, x : A[M_j/y_j]_{j=1}^n$ satisfying

$$\begin{aligned} \langle \text{id}_\Gamma, M_1, \dots, M_n, x' \rangle \circ \delta &= \langle \text{id}_\Gamma, M_1, \dots, M_n, x' \rangle \circ \langle \text{id}_\Gamma N_1, \dots, N_m, N \rangle \\ &= \langle \text{id}_\Gamma, M_1[N_j/x_j]_{j=1}^m, \dots, M_n[N_j/x_j]_{j=1}^m, N \rangle \\ &= \sigma'' \end{aligned}$$

and

$$p_{\Gamma \bowtie \Delta, A[M_j/y_j]_{j=1}^n} \circ \delta = \vec{x} \circ \langle \text{id}_\Gamma, N_1, \dots, N_m, N \rangle$$

as required. If $\delta' : \Gamma \bowtie \Delta'' \rightarrow \Gamma \bowtie \Delta, x' : A[M_j/y_j]_{j=1}^n$ were any such morphism, then from the requirement $p_{\Gamma \bowtie \Delta, A[M_j/y_j]_{j=1}^n} \circ \delta' = \sigma'$ we can conclude that δ' is of the form $\langle \text{id}_\Gamma, N_1, \dots, N_m, N' \rangle$. The requirement $\langle \text{id}_\Gamma, M_1, \dots, M_n, x' \rangle \circ \delta = \sigma''$ tells us that \mathbf{N} proves $\Gamma \bowtie \Delta'' \vdash N' : A[M_j/y_j]_{j=1}^n[N_i/x_i]_{i=1}^m$. Hence $\delta' = \delta : \Gamma \bowtie \Delta'' \rightarrow \Gamma \bowtie \Delta, x' : A[M_j/y_j]_{j=1}^n$. ■

It remains to show the strictness conditions. Firstly, $\text{id}_{\Gamma \bowtie \Delta} = \langle x_1, \dots, x_n, y_1, \dots, y_m \rangle$ when $\Gamma \bowtie \Delta = x_1 : A_1, \dots, x_n : A_n, y_1 : B_1, \dots, y_m : B_m$ and $\text{id}_{\Gamma \bowtie \Delta, xA} = \langle x_1, \dots, x_n, y_1 : B_1, \dots, y_m : B_m, x \rangle$. We have that $\langle \text{id}_\Gamma, x_1, \dots, x_n \rangle \bullet A = \langle x_1, \dots, x_n, x \rangle$ as required. Also, $\text{id}_{\Gamma \bowtie \Delta}^*(A) = A[x_i/x_i]_{i=1}^n = A$. The next two conditions just require us to be careful.

Let $f : \Gamma \bowtie \Delta \rightarrow \Gamma \bowtie \Delta'$ and $g : \Gamma \bowtie \Delta' \rightarrow \Gamma \bowtie \Delta''$ where $f = \langle \text{id}_\Gamma, M_1, \dots, M_n \rangle, \Gamma \bowtie \Delta' \equiv \Gamma, y_1 : B_1, \dots, y_n : B_n, g = \langle \text{id}_\Gamma, N_1, \dots, N_p \rangle$ and $\Gamma'' \equiv z_1 : C_1, \dots, z_p : C_p$. Thus

$$\begin{aligned} g^*(f^*(A)) &= g^*(A[M_i/y_i]_{i=1}^n) \\ &= A[M_i/y_i]_{i=1}^n[N_j/z_j]_{j=1}^p \\ &= A[N_j[M_i/y_i]_{i=1}^n/z_j]_{j=1}^p \\ &= \langle \text{id}_\Gamma, N_1[M_i/y_i]_{i=1}^n, \dots, N_p[M_i/y_i]_{i=1}^n \rangle^* A \\ &= (g; f)^*(A) \end{aligned}$$

We show that $(g \bullet (f^*(A))); (f \bullet A) = (g; f) \bullet A$ by calculating the two sides and showing that they are equal. We begin with the left hand side:

$$f \bullet A = \langle \text{id}_\Gamma, M_1, \dots, M_n, x \rangle$$

and

$$\begin{aligned} g \bullet (f^*(A)) &= g \bullet (A[M_i/y_i]_{i=1}^n) \\ &= \langle N_1, \dots, N_p, \text{id}_\Delta, x \rangle \end{aligned}$$

and

$$(g \bullet (f^*(A))); (f \bullet A) = \langle \text{id}_\Gamma N_1[M_i/y_i]_{i=1}^n, \dots, N_p[M_i/y_i]_{i=1}^n, x \rangle$$

The right hand side is

$$(g; f) \bullet A = \langle \text{id}_\Gamma N_1[M_i/y_i]_{i=1}^n, \dots, N_p[M_i/y_i]_{i=1}^n, x \rangle$$

and these are equal.

Finally, $D \bullet \langle \rangle = D, x : \langle \rangle = D$.

4. The functor $p_{\Gamma \bowtie \Delta, A}^* : \mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta) \rightarrow \mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A)$ has a right adjoint if for all objects B in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A)$, there exists an object $\Pi x : A . B$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$ and a morphism $app_B : p_{\Gamma \bowtie \Delta, A}^*(\Pi x : A . B) \rightarrow B$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A)$, such that for all objects C in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$, and for all morphisms $M : p_{\Gamma \bowtie \Delta, A}^*(C) \rightarrow B$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A)$, there exists a unique morphism $N : C \rightarrow \Pi x : A . B$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$ such that

$$\begin{array}{ccccc}
 C & & p_{\Gamma \bowtie \Delta, A}^*(C) & & \\
 \vdots & & \downarrow & \searrow M & \\
 N & & p_{\Gamma \bowtie \Delta, A}^*(N) & & \\
 \vdots & & \downarrow & & \\
 \Pi x : A . B & & p_{\Gamma \bowtie \Delta, A}^*(\Pi x : A . B) & \xrightarrow{app_B} & B
 \end{array}$$

commutes.

Since $p_{\Gamma \bowtie \Delta, A}^*(C) \xrightarrow{M} B$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A)$ we have that $\Gamma \bowtie \Delta, x : A, y : p_{\Gamma \bowtie \Delta, A}^*(C) \vdash_\Sigma M : B$. We can apply λ -abstraction to obtain $\Gamma \bowtie \Delta, y : p_{\Gamma \bowtie \Delta, A}^*(C) \vdash_\Sigma \lambda x : A . M : \Pi x : A . B$. Since $p_{\Gamma \bowtie \Delta, A}^*(C) = C$, we have that $\Gamma \bowtie \Delta, y : C \vdash_\Sigma \lambda x : A . M : \Pi x : A . B$. By the definition of the fibres, we have an arrow $C \xrightarrow{\lambda x : A . M} \Pi x : A . B$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$. We take this arrow to be N and observe that it is unique. We now define $\Pi_{\Gamma \bowtie \Delta, A}$ to be the functor from $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A) \rightarrow \mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$, which sends every B in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A)$ to $\Pi x : A . B$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$.

The morphism app_B is also dependent on the choice of A , so we define $app(A, B) = app_B : \Pi x : A . B \rightarrow B$. We now show that $\Pi_{\Gamma \bowtie \Delta, A}$ satisfies the strict Beck-Chevalley conditions. Let $\Gamma \bowtie \Delta' \xrightarrow{\sigma} \Gamma \bowtie \Delta$ be a morphism in \mathcal{C}_Γ , A an object in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$ and B be an object in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A)$. We have that

$$\begin{aligned}
 \sigma^*(\Pi_{\Gamma \bowtie \Delta, A} B) &= \sigma^*(\Pi x : A . B) \\
 &= \Pi x : A . B[M_i/x_i]_{i=1}^n
 \end{aligned}$$

while

$$\begin{aligned}\Pi_{\Gamma \bowtie \Delta', \sigma^*(A)}((\sigma \bullet A)^* B) &= \Pi_{\Gamma \bowtie \Delta', \sigma^*(A)}(B[\sigma \bullet A/\bar{x}]) \\ &= \Pi y : A[M_i/x_i]_{i=1}^n \cdot B[M_i/x_i]_{i=1}^n[x/y]\end{aligned}$$

which are equal.

We have that $(\sigma \bullet A)^*(app(A, B)) = B[M_i/x_i]_{i=1}^n[x/y]$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A[M_i/x_i]_{i=1}^n)$ and $app(\sigma^* A; (\sigma \bullet A)^* B) = B[M_i/x_i]_{i=1}^n[x/y]$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta, x : A[M_i/x_i]_{i=1}^n)$. Thus $\Pi_{\Gamma, A}$ satisfies the strict Beck-Chevalley conditions.

We can now conclude that \mathcal{T}_Σ is a Kripke $\lambda\Pi$ -prestructure. \blacksquare

We are now able to define a Kripke $\lambda\Pi$ -structure $\mathcal{K}_{\mathcal{T}_\Sigma}$ on \mathcal{T}_Σ . Let $\mathcal{K}_{\mathcal{T}_\Sigma} : [\mathcal{P}(\Sigma), [\mathcal{A}^{op}, \bar{V}]]$. Let $\Gamma \bowtie \Delta \in \mathcal{C}_\Gamma$. We define $\mathcal{K}_{\mathcal{T}_\Sigma}(\Delta)(\Gamma \bowtie \Delta)$ as follows:

Objects: Categories $\bar{V} = \mathcal{K}_{\mathcal{T}_\Sigma}(\Gamma)(\Gamma \bowtie \Delta)$ with

Objects: Sections $\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, N \rangle} \Gamma \bowtie \Delta, z : B$ (arrows in \mathcal{C}_Γ), such that B is an object of $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$;

Arrows: Identities $\Gamma \bowtie \Delta \xrightarrow{\langle id_{\Gamma \bowtie \Delta} \rangle} \Gamma \bowtie \Delta$.

Arrows: Functors $\sigma^* : \mathcal{K}_{\mathcal{T}_\Sigma}(\Delta)(\Gamma \bowtie \Delta) \rightarrow \mathcal{K}_{\mathcal{T}_\Sigma}(\Delta)(\Gamma \bowtie \Delta')$ induced by arrows $\Gamma \bowtie \Delta' \xrightarrow{\sigma} \Gamma \bowtie \Delta$ in \mathcal{C}_Γ . σ^* sends objects $\Gamma \bowtie \Delta \xrightarrow{\langle id_{\Gamma \bowtie \Delta}, N \rangle} \Gamma \bowtie \Delta, z : C$ to objects $\Gamma \bowtie \Delta \xrightarrow{\langle id_{\Gamma \bowtie \Delta}, N[M_j/x_j]_{j=1}^n \rangle} \Gamma \bowtie \Delta, z : C[M_j/x_j]_{j=1}^n$. σ^* sends arrows $\Gamma \bowtie \Delta \xrightarrow{\langle id_{\Gamma \bowtie \Delta} \rangle} \Gamma \bowtie \Delta$ to $\Gamma \bowtie \Delta \xrightarrow{\langle id_{\Gamma \bowtie \Delta} \rangle} \Gamma \bowtie \Delta$.

Lemma 3.9

$\mathcal{K}_{\mathcal{T}_\Sigma}$ is a Kripke $\lambda\Pi$ -structure on \mathcal{T}_Σ .

Proof We begin by showing that arrows of the form $\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, N \rangle} \Gamma \bowtie \Delta, z : C$ are of the right form. We first fix $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$. We need to show that $\Gamma \bowtie \Delta$ is of the form $A_1 \bullet \dots \bullet A_n$, where each A_i is an object of $\mathcal{J}(\Gamma)(\Gamma \bowtie \Delta \bullet A_1 \bullet \dots \bullet A_{i-1})$. We observe that for any A_i which is in $\Gamma \bowtie \Delta \equiv A_1 \bullet \dots \bullet A_n$, $\Gamma \bowtie \Delta \bullet A_1 \bullet \dots \bullet A_{i-1} \equiv \Gamma \bowtie \Delta$ and each A_i is an object of $\mathcal{J}(\Gamma)(\Gamma \bowtie \Delta)$. We only need to show that C is an object of $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$. This holds by definition and further, $\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, N \rangle} \Gamma \bowtie \Delta, z : C$ is an arrow of \mathcal{C}_Γ . The arrows $id_{\Gamma \bowtie \Delta}$ are of the right form since $\Gamma \bowtie \Delta \bullet \Gamma \bowtie \Delta \equiv \Gamma \bowtie \Delta$.

Let $\Gamma \bowtie \Delta' \xrightarrow{\sigma} \Gamma \bowtie \Delta$ be an arrow in \mathcal{C}_Γ . We show that σ^* satisfies conditions (1) and (2) of Definition 3.2.

1. This holds by definition, since σ^* is a substitution which changes Δ to Δ' and leaves Γ unchanged.

2. Let $\Gamma \bowtie \Delta \xrightarrow{\langle \text{id}_{\Gamma \bowtie \Delta} \rangle} \Gamma \bowtie \Delta$ be an arrow in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$. We need to show that $\Gamma \bowtie \Delta' \xrightarrow{\langle \text{id}_{\Gamma \bowtie \Delta'} \rangle} \Gamma \bowtie \Delta'$ is the unique mediating arrow given by the following canonical pullback:

$$\begin{array}{ccc}
\Gamma \bowtie \Delta' \bullet \sigma^*(\Gamma \bowtie \Delta) & \xrightarrow{\sigma \bullet \Gamma \bowtie \Delta} & \Gamma \bowtie \Delta \bullet \Gamma \bullet \Delta \\
\downarrow \sigma^*(\text{id}_{\Gamma \bowtie \Delta}) & & \downarrow \text{id}_{\Gamma \bowtie \Delta} \\
\Gamma \bowtie \Delta' \bullet \sigma^*(\Gamma \bowtie \Delta) & \xrightarrow{\sigma \bullet \Gamma \bowtie \Delta} & \Gamma \bowtie \Delta \bullet \Gamma \bowtie \Delta \\
\downarrow p & & \downarrow p \\
\Gamma \bowtie \Delta' & \xrightarrow{\sigma} & \Gamma \bowtie \Delta
\end{array}$$

We have a lot of redundancies and so we have the diagram

$$\begin{array}{ccc}
\Gamma \bowtie \Delta' & \xrightarrow{\langle \sigma, \text{id} \rangle} & \Gamma \bowtie \Delta \\
\downarrow \text{id} & & \downarrow \text{id} \\
\Gamma \bowtie \Delta' & \xrightarrow{\langle \sigma, \text{id} \rangle} & \Gamma \bowtie \Delta \\
\downarrow p & & \downarrow p \\
\Gamma \bowtie \Delta' & \xrightarrow{\sigma} & \Gamma \bowtie \Delta
\end{array}$$

from which it is clear that the result holds. ■

It is important to keep this example in mind when we define Kripke $\lambda\Pi$ -models in the next section. We also observe that $\Gamma \bowtie \Delta \vdash_\Sigma M : A$ if and only if there exists an arrow $\langle \rangle \xrightarrow{M} A$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$.

3.2.2 The $\{\forall, \supset\}$ -fragment of many-sorted minimal first-order logic

We write $\mathcal{L}_{\{\forall, \supset\}}$ for the $\{\forall, \supset\}$ -fragment of many-sorted first-order logic. This is another important example because $\mathcal{L}_{\{\forall, \supset\}}$ is the internal logic of the $\lambda\Pi$ -calculus; that is, it is in propositions-as-types correspondence with it. We stress that the

Kripke $\lambda\Pi$ -structure we are about to construct is *not* a model of $\mathcal{L}_{\{\forall, \supset\}}$, rather, it is a Kripke $\lambda\Pi$ -structure built out of the syntax of $\mathcal{L}_{\{\forall, \supset\}}$. We discuss $\mathcal{L}_{\{\forall, \supset\}}$ in more detail in § 5 and provide a Kripke model of the logic there. We sketch the definition of $\mathcal{L}_{\{\forall, \supset\}}$ here with the fuller definition being found in § 5.

We have an alphabet A consisting of the following sets:

- A countable set of sorts, including the sort of individuals ι and propositions o ;
- A finite set of constant symbols c_1, \dots, c_p each given a sort T_i ;
- A finite set of function symbols f_1, \dots, f_n each given an sort $S_1^i, \dots, S_{m_i}^i \rightarrow S^i$;
- A finite set of predicate symbols P_1, \dots, P_p with sort $T_1^i, \dots, T_{q_i}^i \rightarrow o$;
- A set of connectives containing \supset and \forall , which have sorts $(o, o) \rightarrow o$ and $(\iota \rightarrow o) \rightarrow o$.

This definition should be compared with the more general definition of an alphabet for a judged object-logic in § 7. We further distinguish the sorts into those sorts which have variables and those which do not. For each sort which has variables, we have assign a countable set of variables.

We generate the expressions of $\mathcal{L}_{\{\forall, \supset\}}$ by the following rules:

- All constant symbols, function symbols, variables and connectives are expressions;
- (Application) Given an expression of sort $(S_1, \dots, S_m) \rightarrow S$ and expressions e_1, \dots, e_m with sorts S_1, \dots, S_m respectively, then $ee_1 \dots e_m$ is an expression of sort S ;
- (Abstraction) Given expressions e_1, \dots, e_n with sorts S_1, \dots, S_m respectively and an expression e then $(e_1, \dots, e_n)e$ is an expression of sort $S_1, \dots, S_n \rightarrow S$.

We call all expressions of sort o the propositions of $\mathcal{L}_{\{\forall, \supset\}}$. We define a consequence relation \vdash on the set of propositions. This consequence relation satisfies the following:

Reflexivity For all propositions ϕ , $\phi \vdash \phi$;

Transitivity (Cut) If $\Delta \vdash \phi$ and $\Delta, \phi, \Delta' \vdash \psi$ then $\Delta, \Delta' \vdash \psi$ for all propositions ϕ and ψ and all sets of propositions Δ and Δ' ;

Weakening If $\Delta \vdash \phi$ then $\Delta, \Delta' \vdash \phi$ for all propositions ϕ and sets of propositions Δ and Δ' ;

Permutation If $\Delta, \phi, \psi, \Delta' \vdash \chi$ then $\Delta, \psi, \phi, \Delta' \vdash \chi$ for all propositions ϕ, ψ and χ and all sets of propositions Δ and Δ' .

We have the following natural deduction rules for the connectives \forall and \supset .

$$\begin{array}{c}
\frac{}{\Delta, \phi \vdash \phi} Ax \\
\frac{\Delta, \phi \vdash \psi}{\Delta \vdash \phi \supset \psi} \supset I \\
\frac{\Delta \vdash \phi \quad \Delta \vdash \phi \supset \psi}{\Delta \vdash \psi} \supset E \\
\frac{\Delta \vdash \phi}{\Delta \vdash \forall x : S. \phi} \forall I \quad (x \text{ not free in any assumption on which } \phi \text{ depends}) \\
\frac{\Delta \vdash \forall x : S. \phi}{\Delta \vdash \phi[t/x]} \forall E
\end{array}$$

The rules are extended to use proof-objects and we use Barendregt's (1992) notation.

$$\begin{array}{c}
\frac{}{\Delta, y : \phi \vdash y : \phi} Ax \\
\frac{\Delta, y : \phi \vdash \delta : \psi}{\Delta \vdash I_\phi \delta : \phi \supset \psi} \supset I \\
\frac{\Delta \vdash \delta_1 : \phi \quad \Delta \vdash \delta_2 : \phi \supset \psi}{\Delta \vdash \delta_1 \delta_2 : \psi} \supset E \\
\frac{X, x : S \quad \Delta \vdash \delta : \phi}{\Delta \vdash Gx : \iota. \delta : \forall x : S. \phi} \forall I \\
\frac{\Delta \vdash \delta : \forall x : S. \phi}{\Delta \vdash \delta t : \phi[t/x]} \forall E
\end{array}$$

We fix the alphabet A of $\mathcal{L}_{\{\forall, \supset\}}$ for the remainder of the chapter.

Definition 3.10

The category $\mathcal{B}(A)$ is defined as follows:

objects: sets of sorts labelled by first-order syntactic variables, $x_1 : S_1, \dots, x_n : S_n$;

Arrows: tuples of terms $x_1 : S_1, \dots, x_n : S_n \xrightarrow{\langle t_1, \dots, t_m \rangle} y_1 : T_1, \dots, y_n : T_n$, where, for each $1 \leq i \leq n$, $x_1 : S_1, \dots, x_n : S_n \vdash_T t_i : T_i$.

Given two arrows $x_1 : S_1, \dots, x_n : S_n \xrightarrow{\langle t_1, \dots, t_m \rangle} y_1 : T_1, \dots, y_m : T_m$ and $y_1 : T_1, \dots, y_m : T_m \xrightarrow{\langle u_1, \dots, u_p \rangle} z_1 : U_1, \dots, z_p : U_p$, we define their composition to be the arrow $x_1 : S_1, \dots, x_n : S_n \xrightarrow{\langle u_1[t_i/y_i]_{i=1}^m, \dots, u_p[t_i/y_i]_{i=1}^m \rangle} z_1 : U_1, \dots, z_p : U_p$. ■

We write $Dom(x_1 : S_1, \dots, x_n : S_n)$ for the set of variables $\{x_1, \dots, x_n\}$. We define a postel category of worlds which is a subcategory of $\mathcal{B}(A)$.

Definition 3.11

We define the category \mathcal{W} of worlds as follows:

Objects: Sets $X = x_1 : S_1, \dots, x_m : S_m$, where $x_1 : S_1, \dots, x_m : S_m$ is an object of $\mathcal{B}(A)$;

Arrows: $X \rightarrow X'$ if and only if $X \subseteq X'$. ■

We take sets of sorts labelled by first-order syntactic variables to be worlds so that at each world, X , we only define those propositions whose free variables are in X . Further, since every future world X' contains X , more propositions can be defined at future worlds. We now define a category of proof-variables.

Definition 3.12

We define the category $\mathcal{P}(A)$ as follows:

Objects: sets of proof-variables $\mathbf{y}_1 : \phi_1, \dots, \mathbf{y}_n : \phi_n$, where ϕ_i is a proposition of $\mathcal{L}_{\{\forall, \supset\}}$;

Arrows: tuples of proofs $\underbrace{\mathbf{y}_1 : \phi_1, \dots, \mathbf{y}_n : \phi_n}_{\Delta} \xrightarrow{\langle \delta_1, \dots, \delta_m \rangle} \mathbf{z}_1 : \psi_1, \dots, \mathbf{z}_m : \psi_m$, where for $1 \leq i \leq n$, $\Delta \vdash_{\mathcal{L}_{\{\forall, \supset\}}} \delta_i : \psi_i[\delta_j/\mathbf{y}_j]_{j=1}^{i-1}$ is provable in $\mathcal{L}_{\{\forall, \supset\}}$.

Given two arrows $\mathbf{y}_1 : \phi_1, \dots, \mathbf{y}_n : \phi_n \xrightarrow{\langle \delta_1, \dots, \delta_m \rangle} \mathbf{z}_1 : \psi_1, \dots, \mathbf{z}_m : \psi_m$ and $\mathbf{z}_1 : \psi_1, \dots, \mathbf{z}_m : \psi_m \xrightarrow{\langle \delta'_1, \dots, \delta'_p \rangle} \mathbf{x}_1 : \chi_1, \dots, \mathbf{x}_p : \chi_p$, we define their composition to be $\mathbf{y}_1 : \phi_1, \dots, \mathbf{y}_n : \phi_n \xrightarrow{\langle \delta'_1[\delta_i/\mathbf{y}_i]_{i=1}^m, \dots, \delta'_p[\delta_i/\mathbf{y}_i]_{i=1}^m \rangle} \mathbf{x}_1 : \chi_1, \dots, \mathbf{x}_p : \chi_p$. ■

We define $Dom(\mathbf{y}_1 : \phi_1, \dots, \mathbf{y}_n : \phi_n)$, for each $\mathbf{y}_1 : \phi_1, \dots, \mathbf{y}_n : \phi_n$ in $\mathcal{P}(A)$ to be $\{\mathbf{y}_1, \dots, \mathbf{y}_n\}$.

We define \mathcal{C}_X , a subcategory of $\mathcal{P}(A)$ to be the base category at each world X as follows:

Definition 3.13

For each world X in \mathcal{W} , the category \mathcal{C}_X is defined as follows:

Objects: Sets Δ , where Δ is an object of $\mathcal{P}(A)$, the free variables of each ϕ_i of Δ are in X and $Dom(\Delta) \cap Dom(X) = \emptyset$;

Arrows: tuples of proofs $\Delta \xrightarrow{\langle \delta_1, \dots, \delta_n \rangle} \Delta'$ such that $\Delta \xrightarrow{\langle \delta_1, \dots, \delta_n \rangle} \Delta'$ is an arrow of $\mathcal{P}(A)$.

Composition is inherited from $\mathcal{P}(A)$. ■

We now define $\mathcal{A}^{op} = \coprod_{X \in |\mathcal{W}|} \mathcal{C}_X^{op}$ and we are now able to define a functor $\mathcal{T}_A : [\mathcal{P}(A), [\mathcal{A}^{op}, \mathcal{V}]]$, which will turn out to be our Kripke $\lambda\Pi$ -prestructure. We define the fibres $\mathcal{T}_A(X)(\Delta)$ as follows:

Objects: Propositions ϕ such that $(X) \Delta \vdash \phi : o$;

Arrows: Proofs $\phi \xrightarrow{\delta} \phi'$ such that $\Delta, y : \phi \xrightarrow{\langle \text{id}, \delta \rangle} \Delta, z : \phi'$ is an arrow in \mathcal{C}_X .

Functors between fibres are induced by a morphism $\Delta' \xrightarrow{\sigma} \Delta$ such that

- for all objects $\phi \in \mathcal{T}_A(X)(\Delta)$ $\sigma^*(\phi) = \phi[M_i/y_i]_{i=1}^n$ where $\sigma = \langle M_1, \dots, M_n \rangle$ and $X' = y_1 : S_1, \dots, y_n : S_n$.
- for all arrows $\phi \xrightarrow{\delta} \psi$, $\sigma^*(\phi \xrightarrow{\delta} \psi) = \phi[M_j/y_j]_{j=1}^n \xrightarrow{\delta[M_j/y_j]_{j=1}^n} \psi[M_j/y_j]_{j=1}^n$ such that $\Delta', x : \phi[M_j/y_j]_{j=1}^n \xrightarrow{\langle \text{id}_{\Delta'}, \delta[M_j/y_j]_{j=1}^n \rangle} \Delta', y : \psi[M_j/y_j]_{j=1}^n$ is an arrow in $\mathcal{C}_{X'}$.

We have that any morphism $X \rightarrow X'$ in $\mathcal{P}(A)$ induces a natural transformation $\mathcal{T}_A(X) \Rightarrow \mathcal{T}_A(X')$. Since $X \subseteq X'$, we have inclusions $\mathcal{I}_\Delta : \mathcal{T}_A(X)(\Delta) \rightarrow \mathcal{T}_A(X')(\Delta)$ for each Δ . We take these as our components:

$$\begin{array}{ccc}
 \Delta & & \mathcal{T}_A(X)(\Delta) \xrightarrow{\mathcal{I}_\Delta} \mathcal{T}_A(X')(\Delta) \\
 \downarrow t & & \downarrow \mathcal{T}_A(X)(t) \quad \downarrow \mathcal{T}_A(X')(t) \\
 \Delta' & & \mathcal{T}_A(X)(\Delta') \xrightarrow[\mathcal{I}_{\Delta'}]{} \mathcal{T}_A(X')(\Delta')
 \end{array}$$

We now show that \mathcal{T}_A satisfies the conditions in the definition of a Kripke $\lambda\Pi$ -structure.

1. Each \mathcal{C}_X has a terminal object: the empty set \emptyset .
2. Each $\mathcal{T}_A(X)(\Delta)$ has a terminal object, the unit proposition. Let $\Delta' \xrightarrow{\sigma} \Delta$ be an arrow in \mathcal{C}_X then $\sigma^*(\langle \rangle) = \langle \rangle[M_j/y_j]_{j=1}^n = \langle \rangle$.
3. For each object X in $\mathcal{P}(A)$, Δ in \mathcal{C}_X and ϕ in $\mathcal{T}_A(X)(\Delta)$, we define $\Delta \bullet \phi$ as follows:
 - If $\mathbf{y} : \phi \in \Delta$, for some \mathbf{y} , then $\Delta \bullet \phi \equiv \Delta$;

- If $\mathbf{y}:\phi \notin \Delta$, for some \mathbf{y} , then $\Delta \bullet \phi \equiv \Delta, \mathbf{z}:\phi$, where $\mathbf{z} \notin \text{Dom}(\Delta)$.

There are projections $\Delta, \mathbf{y}:\phi \xrightarrow{p_{\Delta, \phi}} \Delta$ and $\langle \rangle \xrightarrow{q_{\Delta, \phi}} \phi$.

Claim 3.14

Let X be an object in $\mathcal{P}(A)$, $\Delta \equiv \mathbf{y}_1:\phi_1, \dots, \mathbf{y}_m:\phi_m \xrightarrow{\langle M_1, \dots, M_n \rangle} \Delta' \equiv \mathbf{z}_1:\psi_1, \dots, \mathbf{z}_n:\psi_n$ be a morphism in \mathcal{C}_X and ϕ an object in $\mathcal{T}_A(X)(\Delta)$. Then

$$\begin{array}{ccc}
 \Delta, \mathbf{y}:\phi[M_j/\mathbf{y}_j]_{j=1}^n & \xrightarrow{\langle M_1, \dots, M_n, \mathbf{y} \rangle} & \Delta', \mathbf{x}:\phi \\
 \downarrow p_{\Delta, \phi[M_j/\mathbf{y}_j]_{j=1}^n} & & \downarrow p_{\Delta, \phi} \\
 \Delta & \xrightarrow{\langle M_1, \dots, M_n \rangle} & \Delta
 \end{array}$$

is a pullback.

This proof is similar to a proof found in (Pitts 2000).

Proof Suppose $\sigma':\Delta'' \rightarrow \Delta$ and $\sigma'':\Delta'' \rightarrow \Delta', \mathbf{x}:\phi$ are morphisms of \mathcal{C}_X satisfying $\langle M_1, \dots, M_n \rangle \circ \sigma' = p_{\Delta, \phi} \circ \sigma''$, i.e., the diagram

$$\begin{array}{ccc}
 \Delta'' & \xrightarrow{\sigma} & \Delta', \mathbf{x}:\phi \\
 \downarrow \sigma' & & \downarrow p_{\Delta', \phi} \\
 \Delta'' & \xrightarrow{\langle M_1, \dots, M_n \rangle} & \Delta'
 \end{array}$$

commutes. We have to show that there is a unique arrow $\delta:\Delta'' \rightarrow \Delta, \mathbf{x}':\phi[M_j/\mathbf{y}_j]_{j=1}^n$ satisfying $p_{\Delta, \phi[M_j/\mathbf{y}_j]_{j=1}^n} \circ \delta = \sigma':\Delta'' \rightarrow \Delta$ and $\langle M_1, \dots, M_n, \mathbf{x}' \rangle \circ \delta = \sigma'':\Delta'' \rightarrow \Delta', \mathbf{x}:\phi$, i.e., the triangles

$$\begin{array}{ccc}
 \Delta'' & & \\
 \downarrow \sigma & \searrow \sigma & \\
 \Delta & & \Delta, \mathbf{x}:\phi[M_j/\mathbf{y}_j]_{j=1}^n \\
 & \swarrow p_{\Delta, \mathbf{x}:\phi[M_j/\mathbf{y}_j]_{j=1}^n} & \\
 & \Delta &
 \end{array}$$

and

$$\begin{array}{ccc}
& \Delta'' & \\
& \searrow \sigma'' & \\
& \Delta', \mathbf{x} : \phi & \\
& \swarrow \langle M_1, \dots, M_n \rangle & \\
\Delta, \mathbf{x}' : \phi[M_j/\mathbf{y}_j]_{j=1}^n & &
\end{array}$$

δ (vertical arrow from Δ'' to $\Delta, \mathbf{x}' : \phi[M_j/\mathbf{y}_j]_{j=1}^n$)

commute.

Since $\langle M_1, \dots, M_n \rangle \circ \sigma' = p_{\Delta, \phi} \circ \sigma'' : \Delta'' \rightarrow \Delta'$. σ' must be of the form $\langle M_1[N_j/\mathbf{x}_j]_{j=1}^m, \dots, M_n[N_j/\mathbf{x}_j]_{j=1}^m, N \rangle$, where $\sigma' = \langle N_1, \dots, N_m \rangle$ and N is such that $\Delta'' \vdash N : \phi[M_i[N_j/\mathbf{x}_j]_{j=1}^m/\mathbf{y}_i]_{i=1}^n$ holds. Now since $\phi[M_i[N_j/\mathbf{x}_j]_{j=1}^m/\mathbf{y}_i]_{i=1}^n = \phi[M_i/\mathbf{y}_i]_{i=1}^n[M_j/\mathbf{x}_j]_{j=1}^m$, we get a morphism

$$\delta =_{def} \langle N_1, \dots, N_n, N \rangle : \Delta'' \rightarrow \Delta, \mathbf{x}' : \phi[M_j/\mathbf{y}_j]_{j=1}^n$$

satisfying

$$\begin{aligned}
\langle M_1, \dots, M_n \rangle \circ \delta &= \langle M_1, \dots, M_n, x' \rangle \circ \langle N_1, \dots, N_m, N \rangle \\
&= \langle M_1[N_j/\mathbf{x}_j]_{j=1}^m, \dots, M_n[N_j/\mathbf{x}_j]_{j=1}^m, N \rangle \\
&= \sigma''
\end{aligned}$$

and $p_{\Delta, \phi[M_j/\mathbf{y}_j]_{j=1}^n} \circ \delta = \vec{\mathbf{x}} \circ \langle N_1, \dots, N_m, N \rangle$ as required. If $\delta' : \Delta'' \rightarrow \Delta, \mathbf{x}' : \phi[M_j/\mathbf{y}_j]_{j=1}^n$ were any such morphism, then from the requirement that $p_{\Delta, \phi[M_j/\mathbf{y}_j]_{j=1}^n} \circ \delta' = \sigma'$ we conclude that $\Delta' \vdash N' : \phi[M_j/\mathbf{y}_j]_{j=1}^n[N_i/\mathbf{x}_i]_{i=1}^m$. Hence $\delta' = \delta : \Delta'' \rightarrow \Delta, \mathbf{x}' : \phi[M_j/\mathbf{x}_j]_{j=1}^n$. \blacksquare

It remains to show the strictness conditions:

$$\text{id}_{\Delta}^*(\phi) = \phi[\mathbf{x}_i/\mathbf{x}_i]_{i=1}^n = \phi$$

$$\text{id}_{\Delta} \bullet \phi = \langle \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x} \rangle$$

which is the identity $\text{id}_{\Delta, \mathbf{y} : \phi}$.

For the remaining two conditions, we have to be careful. Let $f : \Delta \rightarrow \Delta'$, where $f = \langle M_1, \dots, M_n \rangle$, and $g : \Delta' \rightarrow \Delta''$ and $g = \langle N_1, \dots, N_p \rangle$. We need

to show that $g^*(f^*(\phi)) = (f; g)^*(\phi)$, we have

$$\begin{aligned}
g^*(f^*(\phi)) &= g^*(\phi[M_j/\mathbf{x}_j]_{j=1}^n) \\
&= \phi[M_j/\mathbf{x}_j]_{j=1}^n [N_i/\mathbf{y}_i]_{i=1}^m \\
&= \phi[M_j[N_i/\mathbf{y}_i]_{i=1}^m/\mathbf{x}_j]_{j=1}^n \\
&= (f; g)^*(\phi)
\end{aligned}$$

We now show that $(g \bullet (f^*(\phi)); (f \bullet \phi)) = (g; f) \bullet \phi$. We have that

$$\begin{aligned}
f^*(\phi) &= \phi[M_j/\mathbf{x}_j]_{j=1}^n \\
g \bullet \phi[M_j/\mathbf{x}_j]_{j=1}^n &= \langle N_1, \dots, N_m, \mathbf{x} \rangle \\
f \bullet \phi &= \langle M_1, \dots, M_m, \mathbf{y} \rangle
\end{aligned}$$

which we compose to obtain $\langle N_1[M_i/\mathbf{x}_i]_{i=1}^m, \dots, N_m[M_i/\mathbf{x}_i]_{i=1}^m, \mathbf{y} \rangle$ which is equal to $(g; f) \bullet \phi$.

4. $p_{\Delta, \phi}^* : \mathcal{T}_A(X)(\Delta) \rightarrow \mathcal{T}_A(X)(\Delta, \mathbf{x} : \phi)$ has a right adjoint if for all objects ψ in $\mathcal{T}_A(X)(\Delta, \mathbf{x} : \phi)$, there exists an object $\phi \supset \psi$ in $\mathcal{T}_A(X)(\Delta)$ and a morphism $app_\psi : p_{\Delta, \phi}^*(\phi \supset \psi) \rightarrow \psi$ in $\mathcal{T}_A(X)(\Delta, \mathbf{x} : \phi)$, such that for all objects χ in $\mathcal{T}_A(X)(\Delta)$, and for all morphisms $\delta : p_{\Delta, \phi}^*(\chi) \rightarrow \chi$ in $\mathcal{T}_A(X)(\Delta, \mathbf{x} : \phi)$ there exists a unique morphism $\delta' : \chi \rightarrow \phi \supset \psi$, such that the diagram

$$\begin{array}{ccc}
\chi & & p_{\Delta, \phi}^*(\chi) \\
\delta' \downarrow \text{---} & & \downarrow p_{\Delta, \phi}^*(\delta) \\
\phi \supset \psi & & p^*(\phi \supset \psi) \xrightarrow{app_\psi} \psi
\end{array}$$

commutes.

Since $p_{\Delta, \phi}^*(\chi) \xrightarrow{\delta} \psi$ in $\mathcal{T}_A(X)(\Delta, \mathbf{x} : \phi)$ we have that $(X) \Delta, \mathbf{x} : \phi, \mathbf{y} : p_{\Delta, \phi}^*(\chi) \vdash_\Sigma \delta : \psi$. We can deduce that $(X) \Delta, \mathbf{y} : p_{\Delta, \phi}^*(\chi) \vdash_\Sigma I_\phi \delta : \phi \supset \psi$. Since $p_{\Delta, \phi}^*(\chi) = \chi$, we have that $\Delta, \mathbf{y} : \chi \vdash_\Sigma I_\phi \delta : \phi \supset \psi$. By the definition of the fibres, we have an arrow $\chi \xrightarrow{I_\delta} \phi \supset \psi$ in $\mathcal{T}_A(X)(\Delta)$. We take this arrow to be δ' and observe that it is unique. We now define $\Pi_{\Delta, \phi}$ to be the functor from $\mathcal{T}_A(X)(\Delta, \mathbf{x} : \phi) \rightarrow \mathcal{T}_A(X)(\Delta)$, which sends every ψ in $\mathcal{T}_\Sigma(\Gamma)(\Delta, \mathbf{x} : \phi)$ to $\phi \supset \psi$ in $\mathcal{T}_\Sigma(\Gamma)(\Gamma \bowtie \Delta)$.

We now show that $\Pi_{\Delta, \phi}$ satisfies the strict Beck-Chevalley conditions. Let $\Delta' \xrightarrow{\sigma} \Delta$ be a morphism in \mathcal{C}_X and ϕ be an object in $\mathcal{T}_A(X)(\Delta, x : \phi)$. We

have that

$$\begin{aligned}\sigma^*(\Pi_{\Delta,\phi}(\psi)) &= \sigma^*(\phi \supset \psi) \\ &= \phi[M_i/\mathbf{x}_i]_{i=1}^n \supset \psi[M_i/\mathbf{x}_i]_{i=1}^n\end{aligned}$$

while

$$\begin{aligned}\Pi_{\Delta',\phi[M_i/\mathbf{x}_i]_{i=1}^n}((\sigma \bullet \phi)^*\psi) &= \Pi_{\Delta',\phi[M_i/\mathbf{x}_i]_{i=1}^n}(\psi[M_i/\mathbf{x}_i]_{i=1}^n[\mathbf{x}/\mathbf{y}]) \\ &= \phi[M_i/\mathbf{x}_i]_{i=1}^n \supset \psi[M_i/\mathbf{x}_i]_{i=1}^n[\mathbf{x}/\mathbf{y}]\end{aligned}$$

which are equal.

Finally, we have that $(\sigma \bullet \phi)^*(app *_{\mathcal{W}}(\phi, \psi)) = \psi[M_i/\mathbf{x}_i]_{i=1}^n[\mathbf{x}/\mathbf{y}]$ in $\mathcal{T}_A(X)$ $(\Delta, \mathbf{x} : \phi[M_i/\mathbf{x}_i]_{i=1}^n)$ and $app_{\mathcal{W}}(\sigma^*(\phi), (\sigma \bullet \phi)^*\psi) = \psi[M_i/\mathbf{x}_i]_{i=1}^n[\mathbf{x}/\mathbf{y}]$ in $\mathcal{T}_A(X)$ $(\Delta, \mathbf{x} : \phi[M_i/\mathbf{x}_i]_{i=1}^n)$. Thus \mathcal{T}_A satisfies all the conditions in the definition of a Kripke $\lambda\Pi$ -prestructure.

We are now able to define a Kripke $\lambda\Pi$ -structure $\mathcal{K}_{\mathcal{T}_A}$ on \mathcal{T}_A . Let $\mathcal{K}_{\mathcal{T}_A} : [\mathcal{P}(A), [\mathcal{A}^{op}, \bar{\mathcal{V}}]]$ and X an object in $\mathcal{B}(A)$ and Δ be object in \mathcal{C}_X . We define $\mathcal{K}_{\mathcal{T}_A}(X)(\Delta)$ as follows:

Objects: Categories $\bar{\mathcal{V}} = \mathcal{K}_{\mathcal{T}_A}(X)(\Delta)$ with

Objects: Sections $\Delta \xrightarrow{\langle \text{id}, \delta \rangle} \Delta, \mathbf{z} : \phi$ (arrows in \mathcal{C}_X) such that ϕ is an object of $\mathcal{T}_A(X)(\Delta)$.

Arrows: Identities $\Delta \xrightarrow{\langle \text{id} \rangle} \Delta$.

Arrows: Functors $\sigma^* : \mathcal{K}_{\mathcal{T}_A}(X)(\Delta) \rightarrow \mathcal{K}_{\mathcal{T}_A}(X)(\Delta')$ induced by arrows $\Delta' \xrightarrow{\sigma} \Delta$ in \mathcal{C}_X . σ^* sends an object $\Delta \xrightarrow{\langle \text{id}, \delta \rangle} \Delta, \mathbf{z} : \phi$ to $\Delta' \xrightarrow{\langle \text{id}, \delta[\delta'_j/\mathbf{x}_j]_{j=1}^n \rangle} \Delta', \mathbf{z} : \phi[\delta'_j/\mathbf{x}_j]_{j=1}^n$. σ^* sends an arrow $\Delta \xrightarrow{\langle \text{id}_{\Delta} \rangle} \Delta$ to the arrow $\Delta' \xrightarrow{\langle \text{id}_{\Delta'} \rangle} \Delta'$.

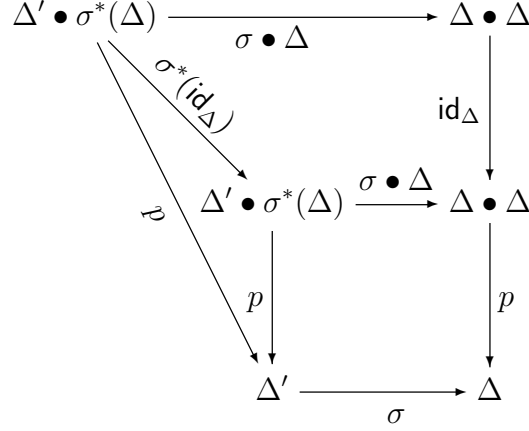
Lemma 3.15

$\mathcal{K}_{\mathcal{T}_A}$ is a Kripke $\lambda\Pi$ -structure on \mathcal{T}_A .

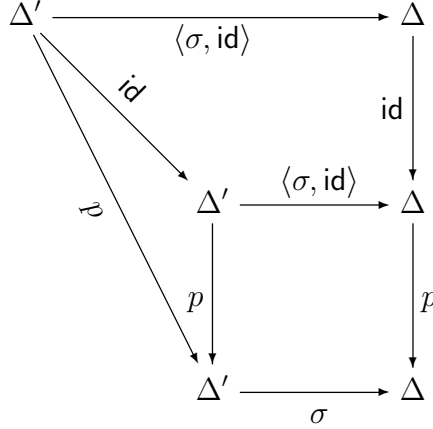
Proof We begin by showing that arrows of the form $\Delta \xrightarrow{\langle \text{id}_{\Delta}, \delta \rangle} \Delta, \mathbf{y} : \phi$ are of the right form. We first fix $\mathcal{T}_A(X)(\Delta)$. We need to show that Δ is of the form $\phi_1 \bullet \dots \bullet \phi_n$, where each ϕ_i is an object of $\mathcal{J}(X)(\Delta \bullet \phi_1 \dots \bullet \phi_{i-1})$. We observe that any ϕ_i in $\Delta \equiv \phi_1 \bullet \dots \bullet \phi_n$, $\Delta \bullet \phi_1 \bullet \dots \bullet \phi_{i-1} \equiv \Delta$ and that ϕ_i is an object of $\mathcal{T}_A(X)(\Delta)$. We only need to show that ϕ is an object of $\mathcal{T}_A(X)(\Delta)$. This holds by definition and further, $\Delta \xrightarrow{\langle \text{id}_{\Delta}, \delta \rangle} \Delta, \mathbf{y} : \phi$ is an arrow of \mathcal{C}_X . The arrows id_{Δ} are of the right form since $\Delta \bullet \Delta \equiv \Delta$.

Let $\Delta''' \xrightarrow{\sigma} \Delta'$ be an arrow in $\mathcal{B}(A)_\Delta$. We show that σ^* satisfies conditions (1) and (2) of Definition 3.2.

1. This holds by definition, since σ^* is a substitution which changes Δ to Δ' .
2. Let $\Delta \xrightarrow{\langle \text{id}_\Delta \rangle} \Delta$ be an arrow in $\mathcal{K}_{\mathcal{T}_A}(X)(\Delta)$. We need to show that $\Delta' \xrightarrow{\langle \text{id}_{\Delta'} \rangle} \Delta'$ is the unique mediating arrow given by the following canonical pullback:



We have a lot of redundancies and so we have the diagram



from which it is clear that the result holds. ■

Looking ahead to the next section, where we define a Kripke $\lambda\Pi$ -model, we remark that the interpretation function which interprets the syntax of the $\lambda\Pi$ -calculus in this Kripke $\lambda\Pi$ -structure, is closely related to the proposition-as-types correspondence.

3.3 Kripke Σ - $\lambda\Pi$ -models

Having define a Kripke $\lambda\Pi$ -structure, we are in a position to provide an interpretation of the $\lambda\Pi$ -calculus in a Kripke $\lambda\Pi$ -structure. The interpretation itself is long and complex because we are interpreting a dependent type theory. The components in the model are interdependent, thus requiring that they all be defined simultaneously by induction on the raw syntax apart from the structure. Ignoring these interdependencies for a moment, we explain the purpose of each component in the model. It is helpful to have the term model in § 3.2.1 in mind at this point. First, the Kripke $\lambda\Pi$ -structure provides the abstract domain where the type theory may be interpreted. We require the Kripke $\lambda\Pi$ -structure to have additional structure, namely Σ -operations, to ensure it has enough structure to interpret the signature Σ . Second, the interpretation $\llbracket - \rrbracket$ is a partial function. It maps raw (that is not necessarily well-typed) contexts Γ to objects of \mathcal{D} . Types over raw contexts A_Γ are sent to objects in the category indexed by (the interpretation of) Γ . It sends terms over raw contexts M_Γ to arrows in the category indexed by (the interpretation of) Γ . Types and terms are interpreted up to $\beta\eta$ -equivalence. The model also needs to be constrained so that multiple occurrences of variables in a context get the same interpretation. Finally, satisfaction is defined to be a relationship on worlds and sequents axiomatizing the desired properties of the model. The abstract definition of the model is sufficient to derive Van Dalen’s (1994) description of a Kripke model for intuitionistic logic.

We remark that we restrict our discussion of semantics to the $\Gamma \vdash_\Sigma M : A$: **Type**-fragment of the $\lambda\Pi$ -calculus. The treatment of $\Gamma \vdash_\Sigma A : -$ — K -fragment is dealt with analogously — in a sense, the $A : K$ -fragment has the same logical structure as the $M : A$ -fragment. To interpret the kind **Type**, we must require the existence of a chosen object, call it Ω , in each fibre. The object Ω must obey several equations: it must be preserved on the nose by any f^* and must behave well under quantification. Details of the $A : K$ -fragment in the case of contextual categories can be found in Streicher’s (1989) thesis. The analogous development in our setting is similar and we omit the details. The logical motivation for this restriction is that we intend to use the $\lambda\Pi$ -calculus as the language of a logical framework and then we will be interested in terms $M : A$, since these will represent the proof of a judgement.

There are several important notions of partiality in the model. The interpretation function is a partial one because it is defined for raw objects of the syntax and we are only interested in interpreting well-typed terms. Since we are dealing with a dependent type theory, there is no guarantee that an object can be defined until everything it depends on can be defined. The partiality allows us to assume objects are interpreted together with their dependent objects, inductively, so that we can “bootstrap” the definition. There is also a Kripke semantic partiality of information. The further up the world structure one goes, the more objects have

defined interpretations. We refer the interested reader to (Streicher 1989) and (Mitchell & Moggi 1991) for further comments on these matters.

Note that since models are $\beta\eta$ -extensional, *q.v.* Theorem 3.35, the interpretation of X does not differ from that of $X[M/x]$, *etc.* We subscript an object by its context Γ to indicate that it is being interpreted in the fibre over (the interpretation of) Γ . We first define Σ -operations. These provide enough structure to ensure that we can interpret every constant in a given signature.

Definition 3.16 (Σ -operations)

Let Σ be a $\lambda\Pi$ -signature. A Kripke $\lambda\Pi$ -structure, $\mathcal{K}_{\mathcal{J}}$ has Σ -operations if for all objects W in \mathcal{W} corresponding to each constant:

1. $c : \Pi x_1 : A_1 \dots \Pi x_m : A_m \cdot \text{Type} \in \Sigma$, there is in each fibre $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$, an operation op_c such that $op_c(\llbracket (M_1)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \dots, \llbracket (M_m)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ is an object of $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$, where each $\llbracket M_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an object in $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$;
2. $c : \Pi x_1 : A_1 \dots \Pi x_m : A_m \cdot A \in \Sigma$, there is in each fibre $\mathcal{J}(W)(D)$ an arrow $1_{\mathcal{K}_{\mathcal{J}}(W)(D)} \xrightarrow{op_c} \llbracket A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, where $D = \llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \bullet \llbracket A_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \bullet \dots \bullet \llbracket A_m \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$

where $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^-$ is a partial function from the (raw) syntax of the $\lambda\Pi$ -calculus to (the components of) $\mathcal{K}_{\mathcal{J}}$ defined in the following definition. ■

The Σ -operations guarantee that a Kripke $\lambda\Pi$ -structure has enough structure to be able to interpret all constants in the signature, Σ .

Definition 3.17 (Kripke Σ - $\lambda\Pi$ -Model)

Let Σ be a $\lambda\Pi$ -signature. A *Kripke Σ - $\lambda\Pi$ -model* is an ordered pair $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \overline{\mathcal{V}}]]$ is a Kripke $\lambda\Pi$ -structure that has Σ -operations and $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^-$ is a partial function from the (raw) syntax of the $\lambda\Pi$ -calculus to (the components of) $\mathcal{K}_{\mathcal{J}}$, defined simultaneously on the structure of the (raw) syntax of the $\lambda\Pi$ -calculus as follows:

1. $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = 1_{\mathcal{D}_W}$;
2. $\llbracket \Gamma, x : A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \bullet \llbracket A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$;
3. $\llbracket \Gamma \xrightarrow{\langle (M_1)_{\Gamma}, \dots, (M_n)_{\Gamma} \rangle} \Delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\langle \llbracket (M_1)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \dots, \llbracket (M_n)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rangle} \llbracket \Delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$;
4. $\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = 1_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$ in $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$;
5. $\llbracket (cM_1 \dots M_m)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = op_c(\llbracket (M_1)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \dots, \llbracket (M_m)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ in $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$, where $c : \Pi x_1 : A_1 \dots \Pi x_m : A_m \cdot \text{Type} \in \Sigma$, such that if $cM_1 \dots M_m \equiv_{\beta\eta} cM'_1 \dots M'_m$, then $\llbracket (cM_1 \dots M_m)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket (cM'_1 \dots M'_m)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$;
6. $\llbracket (\Pi x : B \cdot C)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \Pi_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} (\llbracket C_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$;

7. $\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = 1_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$ (*i.e.*, the identity map on the terminal object);
8. $\llbracket c_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \text{cur}_W^m(\text{op}_c)$ in $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$, where $c: \Pi x_1: A_1 \dots \Pi x_m: A_m \cdot A$;
9. $\llbracket x_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = p_{\llbracket \Delta, x: A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket \Delta' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^*$ ($q_{\llbracket \Delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$), where $\Gamma = \Delta, x: A, \Delta'$;
10. $\llbracket (\lambda x: A. M)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \text{cur}_W(\llbracket M_{\Gamma, x: A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$;
11. $\llbracket (MN)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = (\llbracket \langle x_1, \dots, x_m, N \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)^*(p_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^* \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$; $\text{app}_W(\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$), where $\Gamma \equiv x_1: A_1, \dots, x_m: A_m$.

We require that the following conditions are satisfied:

1. Syntactic monotonicity: if $\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined, then so is $\llbracket X' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, for every subterm X' of X , where X ranges over all the raw terms of the $\lambda\Pi$ -calculus with signature Σ ;
2. Accessibility: if there is an arrow $W \xrightarrow{\alpha} W'$ in \mathcal{W} , then (*i*) there is a functor $\mathcal{K}^{\alpha}: \mathcal{D}_W \rightarrow \mathcal{D}_{W'}$; and, (*ii*) $\mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) = \mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})$ and $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) = \mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})$ for each Γ ; otherwise, $\mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ is undefined.

The second accessibility condition (*ii*) is a simple instance of a more general condition. We can require that there exists functors $\tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^{\alpha}$ and $\tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}^{\alpha}$ such that the following diagram, in which $\eta_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^{\alpha}$ and $\eta_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}^{\alpha}$ are components of the natural transformation $\mathcal{J}(\alpha)$,

$$\begin{array}{ccc}
\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) & \xrightarrow{\eta_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^{\alpha}} & \mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) \\
\tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^{\alpha} \downarrow & & \downarrow \tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}^{\alpha} \\
\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}) & \xrightarrow{\eta_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}^{\alpha}} & \mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})
\end{array}$$

commutes. We also require the following coherence condition:

$$\mathcal{J}(W')(\mathcal{K}^{\alpha}(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)) = (\tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^{\alpha}; \eta_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}^{\alpha})(\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)).$$

In this definition, we have the simple case in which both $\tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^{\alpha}$ and $\tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}^{\alpha}$ are identities in \mathcal{V} . In this simple setting, we shall refer to $\tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^{\alpha}; \eta_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}^{\alpha}$ ($= \eta_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^{\alpha}; \tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}^{\alpha}$) as \mathcal{N}^{α} . In the case where \mathcal{W} is posetal, *i.e.*, $\alpha: W \leq W'$, we write $\mathcal{N}^{W, W'}$. \blacksquare

There is substantial scope, beyond the reach of this thesis, for investigating different choices of $\tau_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}}^{\alpha}$ in the above definition. We have taken the simplest accessibility condition with regards to relativization: the idea of interpreting constructs at one world, and reasoning about them from another.

A syntactic term can be seen, in a certain sense, as a “rigid designator”. That is one whose interpretation is the same over different worlds, for a semantic object. For example, suppose \mathbf{N} proves $\Gamma \vdash_{\Sigma} M : A$. If $\llbracket M \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined (given soundness, this will be the case), then, for all objects $W \leq W'$ in \mathcal{W} , $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ is defined and equal to $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. In a sense, the syntactic term M designates all objects $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ at all worlds W where they are defined.

The next three Lemmas, 3.18, 3.19 and 3.20, are consequences of the previous definition. We include them here in order to emphasize the organisation of the models we have defined. However, their proofs, which are by induction on the structure of proofs in \mathbf{N} , must be performed simultaneously with the proof of Theorem 3.35, the soundness theorem for the $M : A : \mathbf{Type}$ -fragment of the $\lambda\Pi$ -calculus. To see why this must be so, consider that the well-foundedness of types, and so of contexts, depends in general on the well-formedness of objects. Moreover, the definedness of an interpretation of an object in a model depends upon the definedness of its type and the context in which its variables are declared.

Lemma 3.18 (Context Interpretation)

Let Σ be a valid $\lambda\Pi$ -signature and Γ be a valid $\lambda\Pi$ -context. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke Σ - $\lambda\Pi$ -model. If \mathbf{N} proves $\vdash_{\Sigma} \Gamma$ context, then, for each object W at which it is defined, $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an object of \mathcal{D}_W . ■

Lemma 3.19 (Type Interpretation)

Let Σ be a valid $\lambda\Pi$ -signature and Γ be a valid $\lambda\Pi$ -context. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke Σ - $\lambda\Pi$ -model. If \mathbf{N} proves $\Gamma \vdash_{\Sigma} A : \mathbf{Type}$, then, for each W at which it is defined, $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an object of $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. ■

Lemma 3.20 (Term Interpretation)

Let Σ be a valid $\lambda\Pi$ -signature and Γ be a valid $\lambda\Pi$ -signature and M be a valid $\lambda\Pi$ -object. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke Σ - $\lambda\Pi$ -model. If \mathbf{N} proves $\Gamma \vdash_{\Sigma} M : A$, then for each W at which it is defined $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an arrow of $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. ■

The astute reader will have noticed that the results above do not make use of the Kripke $\lambda\Pi$ -structure. This is because we interpret terms, types and assertions of the form $\Gamma \vdash_{\Sigma} M : A$ in the Kripke $\lambda\Pi$ -prestructure. The Kripke $\lambda\Pi$ -structure is defined in such a way that we are able to interpret realizations $\vdash_{\Sigma} \Delta \xrightarrow{\langle M_1, \dots, M_n \rangle} \Theta$ in each of its fibres. This will be developed in detail in § 3.8.

3.4 Examples

We return to our previous examples and provide an interpretation.

3.4.1 Term Model

We take the interpretation function to be the identity function. It is clear that syntactic monotonicity and accessibility hold.

3.4.2 The $\{\forall, \supset\}$ -fragment of many-sorted first-order logic

The interpretation function is close to being given by the propositions-as-types correspondence. The main difference is that $\Pi x : A. B$ is interpreted as $\phi \supset \psi$, rather than $\forall x : S. \psi$. Other than this important point, the interpretation is straightforward and it is clear that syntactic monotonicity and accessibility hold.

3.5 Adding Definitional Equality: $\lambda\Pi_{=}$

So far we have considered the basic $\lambda\Pi$ -calculus, which comes equipped with the intensional $\alpha\beta\eta$ -equality. However, from the point of view of the $\lambda\Pi$ -calculus as a theory of functions, it is both important and interesting to consider also definitional equality. We extend our signatures to include declarations of the form $M = N : A$ by taking the following rule of signature (see also Table 2):

$$\frac{\vdash \Sigma \text{ sig} \quad \vdash_{\Sigma} M : A \quad \vdash_{\Sigma} N : A}{\vdash \Sigma, M = N : A \text{ sig}}$$

which generates these declarations.

Roughly, equational declarations of this form correspond, under the propositions-as-types correspondence, to theories in the internal logic. We spell this out in detail in § 5, but for now consider the following: if the term language of the internal logic were extended with arithmetic terms, one might wish to assert that

$$(0 + 0) = 0 : \mathbb{N}$$

which makes use of equational declarations. We might also add declarations of the form $A = B : K$, or indeed of the form $K = L \text{ Kind}$, to signatures but these extensions are beyond our present requirements.

Our presentation of definitional equality is similar to that of (Harper 1988). In addition to the assertions in the $\lambda\Pi$ -calculus, we consider also the following

equality assertions:

$$\begin{aligned}
& \Gamma \vdash_{\Sigma} K = L \text{ Kind } K \text{ and } L \text{ are equal kinds} \\
& \Gamma \vdash_{\Sigma} A = B : K \text{ } A \text{ and } B \text{ are equal types of kind } K \\
& \Gamma \vdash_{\Sigma} M = N : A \text{ } M \text{ and } N \text{ are equal objects of type } A
\end{aligned}$$

The additional rules required to support these assertions are given in Table 2; the system \mathbf{N} extended with these rules is called $\mathbf{N}_{=}$. When $c \in \text{Dom}(\Sigma)$ we write $\Sigma(c)$ for the unique K or A such that $c : K$ or $c : A \in \Sigma$. When $x \in \text{Dom}(\Gamma)$, we write that $x : A \in \Gamma$ and Γ_x for the prefix of Γ up to, but not including, the declaration of x .

Equations in Signatures

$$\frac{\vdash_{\Sigma} \text{sig} \quad \vdash_{\Sigma} M : A \quad \vdash_{\Sigma} N : A}{\vdash_{\Sigma}, M = N : A \text{ sig}} \quad (3.3)$$

Equivalence Relation

$$\frac{\Gamma \vdash_{\Sigma} K \text{ Kind}}{\Gamma \vdash_{\Sigma} K = K \text{ Kind}} \quad (3.4)$$

$$\frac{\Gamma \vdash_{\Sigma} A : K}{\Gamma \vdash_{\Sigma} A = A : K} \quad (3.5)$$

$$\frac{\Gamma \vdash_{\Sigma} M : A}{\Gamma \vdash_{\Sigma} M = M : A} \quad (3.6)$$

$$\frac{\Gamma \vdash_{\Sigma} K = L \text{ Kind}}{\Gamma \vdash_{\Sigma} L = K \text{ Kind}} \quad (3.7)$$

$$\frac{\Gamma \vdash_{\Sigma} A = B : K}{\Gamma \vdash_{\Sigma} B = A : K} \quad (3.8)$$

$$\frac{\Gamma \vdash_{\Sigma} M = N : A}{\Gamma \vdash_{\Sigma} N = M : A} \quad (3.9)$$

$$\frac{\Gamma \vdash_{\Sigma} J = K \text{ Kind} \quad \Gamma \vdash_{\Sigma} K = L \text{ Kind}}{\Gamma \vdash_{\Sigma} J = L \text{ Kind}} \quad (3.10)$$

Table 2: Rules for Definitional Equality (continued on the next page)

$\frac{\Gamma \vdash A = B : K \quad \Gamma \vdash_{\Sigma} B = C : K}{\Gamma \vdash_{\Sigma} A = C : K}$	(3.11)
$\frac{\Gamma \vdash_{\Sigma} M = N : A \quad \Gamma \vdash_{\Sigma} N = P : A}{\Gamma \vdash_{\Sigma} M = P : A}$	(3.12)
Structural Equality Rules	
$\frac{\vdash_{\Sigma} \Gamma \text{ context} \quad M = N : A \in \Sigma}{\Gamma \vdash_{\Sigma} M = N : A}$	(3.13)
$\frac{\Gamma \vdash_{\Sigma} A = B : K \quad \Gamma \vdash_{\Sigma} K = L}{\Gamma \vdash_{\Sigma} A = B : L}$	(3.14)
$\frac{\Gamma \vdash_{\Sigma} M = N : A \quad \Gamma \vdash_{\Sigma} A = B : \text{Type}}{\Gamma \vdash_{\Sigma} M = N : B}$	(3.15)
$\frac{\Gamma \vdash_{\Sigma} A = B : K \quad \vdash_{\Sigma} \Gamma' \text{ context} \quad \Gamma_x \vdash_{\Sigma} \Gamma(x) = \Gamma'(x) : \text{Type} \quad x \notin FV(A, B, K)}{\Gamma' \vdash_{\Sigma} A = B : K}$	(3.16)
$\frac{\Gamma \vdash_{\Sigma} M = N : A \quad \vdash_{\Sigma} \Gamma' \text{ context} \quad \Gamma_x \vdash_{\Sigma} \Gamma(x) = \Gamma'(x) : \text{Type} \quad x \notin FV(M, N, A)}{\Gamma' \vdash_{\Sigma} M = N : A}$	(3.17)
Kind Equality	
$\frac{\Gamma \vdash_{\Sigma} A = B : \text{Type} \quad \Gamma, x : A \vdash_{\Sigma} K = L : \text{Kind}}{\Gamma \vdash_{\Sigma} \Pi x : A . K = \Pi x : B . L : \text{Kind}}$	(3.18)
Types and Families Equality	
$\frac{\Gamma \vdash_{\Sigma} A = B : \text{Type} \quad \Gamma, x : A \vdash_{\Sigma} C = D : \text{Type}}{\Gamma \vdash_{\Sigma} \Pi x : A . C = \Pi x : B . D : \text{Type}}$	(3.19)
$\frac{\Gamma \vdash_{\Sigma} A = B : \text{Type} \quad \Gamma, x : A \vdash_{\Sigma} C = D : K}{\Gamma \vdash_{\Sigma} \lambda x : A . C = \lambda x : B . D : \text{Type}}$	(3.20)
$\frac{\Gamma \vdash_{\Sigma} B = C : \Pi x : A . L \quad \Gamma \vdash_{\Sigma} M = N : A}{\Gamma \vdash_{\Sigma} BM = CN : K[M/x]}$	(3.21)
$\frac{\Gamma, x : A \vdash_{\Sigma} B : K \quad \Gamma \vdash_{\Sigma} M : A}{\Gamma \vdash_{\Sigma} (\lambda x : A . B)M = B[M/x] : K[M/x]}$	(3.22)
$\frac{\Gamma \vdash_{\Sigma} B : \Pi x : A . K \quad x \notin FV(B)}{\Gamma \vdash_{\Sigma} \lambda x : A . K}$	(3.23)
$\frac{\Gamma \vdash_{\Sigma} A : K \quad \Gamma \vdash_{\Sigma} K = L \text{ Kind}}{\Gamma \vdash_{\Sigma} A : L}$	(3.24)

Table 2: Rules for Definitional Equality (continued on the next page)

Object Equality

$$\frac{\Gamma \vdash_{\Sigma} A = B : \mathbf{Type} \quad \Gamma, x : A \vdash_{\Sigma} M = N : C}{\Gamma \vdash_{\Sigma} \lambda x : A. M = \lambda x : B. N : \Pi x : A. C} \quad (3.25)$$

$$\frac{\Gamma \vdash_{\Sigma} M = N : \Pi x : A. B \quad \Gamma \vdash_{\Sigma} P = Q : A}{\Gamma \vdash_{\Sigma} MP = NQ : B[P/x]} \quad (3.26)$$

$$\frac{\Gamma, x : A \vdash_{\Sigma} M : B \quad \Gamma \vdash_{\Sigma} P = Q : A}{\Gamma \vdash_{\Sigma} (\lambda x : A. M)N = NQ : B[P/x]} \quad (3.27)$$

$$\frac{\Gamma \vdash_{\Sigma} M : \Pi x : A. B \quad x \notin FV(M)}{\Gamma \vdash_{\Sigma} \lambda x : A. Mx = M : \Pi x : A. B} \quad (3.28)$$

$$\frac{\Gamma \vdash_{\Sigma} \lambda x : A. Mx = M : \Pi x : A. B \quad \Gamma \vdash_{\Sigma} M : A \quad \Gamma \vdash_{\Sigma} A = B : \mathbf{Type}}{\Gamma \vdash_{\Sigma} M : B} \quad (3.29)$$

Table 2: Rules for Definitional Equality

The main syntactic metatheoretic properties of $\lambda\Pi_{=}$ are summarized in Propositions 3.21 and 3.22. These are minor variations on (Harper 1988) and extend Theorem 2.1. For clarity, they are stated separately. They must, however, be proved simultaneously with Theorem 2.1, by induction on the structure of proofs in \mathbf{N} .

Proposition 3.21 ((Harper 1988))

Let X range over the basic assertions of the form $K \mathbf{Kind}$, $A : K$ and $M : A$.

1. If $\Gamma \vdash_{\Sigma} X$, then $FV(X) \subseteq \text{Dom}(\Gamma)$.
2. If $\Gamma \vdash_{\Sigma} X$, $\vdash_{\Sigma} \Gamma'$ context, and for all $x \in FV(X)$, $\Gamma_x \vdash_{\Sigma} \Gamma(x) = \Gamma'(x) : \mathbf{Type}$, then $\Gamma' \vdash_{\Sigma} X$.
3. Unicity of classifier:
 - (a) If $\Gamma \vdash_{\Sigma} A : K$ and $\Gamma \vdash_{\Sigma} A : L$, then $\Gamma \vdash_{\Sigma} K = L : \mathbf{Kind}$;
 - (b) If $\Gamma \vdash_{\Sigma} M : A$ and if $\Gamma \vdash_{\Sigma} M : B$, then $\Gamma \vdash_{\Sigma} M = N : A$. ■

Proposition 3.22 ((Harper 1988))

The following hold in the $\lambda\Pi_{=}$ -calculus.

1. If $\Gamma \vdash_{\Sigma} K = L \mathbf{Kind}$, $\vdash_{\Sigma} \Gamma'$ context, and for all $x \in FV(K) \cup FV(L)$, $\Gamma_x \vdash_{\Sigma} \Gamma(x) = \Gamma'(x) : \mathbf{Type}$, then $\Gamma' \vdash_{\Sigma} K = L \mathbf{Kind}$.
2. Well-formedness of equands:
 - (a) If $\Gamma \vdash_{\Sigma} K = L \mathbf{Kind}$, then $\Gamma \vdash_{\Sigma} K \mathbf{Kind}$ and $\Gamma \vdash_{\Sigma} L \mathbf{Kind}$.

- (b) If $\Gamma \vdash_{\Sigma} A = B : K$, then $\Gamma \vdash_{\Sigma} A : K$ and $\Gamma \vdash_{\Sigma} B : K$.
(c) If $\Gamma \vdash_{\Sigma} M = N : A$, then $\Gamma \vdash_{\Sigma} M : A$ and $\Gamma \vdash_{\Sigma} N : A$.

3. *Invertibility of type and kind equations:*

- (a) If $\Gamma \vdash_{\Sigma} \Pi c : A . K = \Pi x : B . L$, then $\Gamma \vdash_{\Sigma} A = B : \mathbf{Type}$ and $\Gamma \vdash_{\Sigma} K = L \mathbf{Kind}$.
(b) If $\Gamma \vdash_{\Sigma} \Pi x : A . C = \Pi x : B . D : \mathbf{Type}$, then $\Gamma \vdash_{\Sigma} A = B : \mathbf{Type}$ and $\Gamma, x : A \vdash_{\Sigma} C = D : \mathbf{Type}$.

4. *Substitution (Cut):* If $\Gamma, x : A \vdash_{\Sigma} \epsilon$ and $\Gamma \vdash_{\Sigma} M : A$, then $\Gamma \vdash_{\Sigma} [M/x]\epsilon$, where ϵ ranges over assertions of the form $K = L \mathbf{Kind}$, $A = B : K$ and $M = N : A$. ■

Turning to the algebraic formulation of the syntax. We construct a syntactic category of contexts and realizations, $\mathcal{B}(\Sigma)_{=}$. We observe that the equality defined in Table 2 is an equivalence relation. We extend the equality on types to contexts by induction on the length of contexts and equality on objects to realizations by induction on the length of realizations. We now have a notion of equality on contexts and realizations which is an equivalence relation. We thus construct $\mathcal{B}(\Sigma)_{=}$ by quotienting $\mathcal{B}(\Sigma)$ by this equivalence relation, *i.e.*, provable equality. Similarly, we can obtain the categories $\mathcal{E}(\Sigma)_{=}$, $\mathcal{Con}(\Sigma)_{=}$ and $\mathcal{Fam}(\Sigma)_{=}$ by quotienting. We extend our definition of Kripke $\lambda\Pi$ -model to an equational Kripke Σ - $\lambda\Pi_{=}$ -model.

Definition 3.23 (Equational Kripke Σ - $\lambda\Pi_{=}$ -model)

Let Σ be a $\lambda\Pi_{=}$ -signature. An *equational Kripke Σ - $\lambda\Pi_{=}$ -model* is a Kripke Σ - $\lambda\Pi$ -model which also satisfies the following condition:

- If $M = N : A \in \Sigma$, then $\llbracket M \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket N \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined at each world W and $\llbracket M \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. ■

3.6 Satisfaction

We define two notions of satisfaction both of which are based on Kripke forcing, (Kripke 1963). The first notion of satisfaction is intended to be the semantic counterpart of the assertion $\Gamma \vdash_{\Sigma} M : A$. We have

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : A)[\Gamma]$$

which is read as in the Kripke Σ - $\lambda\Pi$ -model $\mathcal{K}_{\mathcal{J}}$, W forces $(M : A)$ with respect to Γ . The second notion is intended to be the semantic counterpart to the assertion

$\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M_n \rangle} \Theta$. We discuss this assertion in more detail in § 3.8. We have

$$W \Vdash_W^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\langle M_1, \dots, M_n \rangle} \Theta)[\Gamma]$$

which is read as in the Kripke Σ - $\lambda\Pi$ -model, $\mathcal{K}_{\mathcal{J}}$, W forces $(\Gamma \xrightarrow{\langle M_1, \dots, M_n \rangle} \Theta)$ with respect to Γ . Clearly, the former can be considered a special case of the latter by setting $\Delta \equiv \langle \rangle$ and $\Theta \equiv x : A$. This is based on the fact that the interpretation equates $\langle \rangle \xrightarrow{\langle M \rangle} x : A$ with $M : A$. It is not the case that we can always construct \Vdash from \models . We are always able, however, to do so in the term model. This special case will be particularly important when we construct Herbrand models in § 11. Both of these notions of satisfaction are easily extended to the $\lambda\Pi_{=}$ -calculus,

The first notion of satisfaction will mainly be used in § 6 - § 9 of this thesis. There we will be considering the $\lambda\Pi$ -calculus as the language of a logical framework. \models will allow us to force the representation of a proof-term, *i.e.*, $M_{\delta} : A_{j(\phi)}$. The second notion of satisfaction will mainly be used in § 11. There we will be considering the $\lambda\Pi$ -calculus as a logic programming language. We then use \Vdash to allow us to force a realization and describe resolution.

Definition 3.24 (\models -satisfaction for $\lambda\Pi$)

Let Σ be a valid $\lambda\Pi$ -signature, Γ be a valid $\lambda\Pi$ -context and M be a valid $\lambda\Pi$ -object. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke Σ - $\lambda\Pi$ -model, Γ be a context, A be a type and M be an object. In the Kripke Σ - $\lambda\Pi$ -model, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, the world W satisfies the inhabitation of A by M with respect to Γ , *i.e.*,

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : A)[\Gamma],$$

if and only if $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are all defined and, for all arrows $W \xrightarrow{\alpha} W'$ in \mathcal{W} , $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ is defined such that

$$\begin{array}{ccc} \llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} & \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \\ \downarrow \mathcal{N}^{\alpha} & & \downarrow \mathcal{N}^{\alpha} \\ \llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} & \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} & \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \end{array}$$

holds. ■

We extend the definition of satisfaction to an equational Kripke Σ - $\lambda\Pi_{=}$ -model.

Definition 3.25 (\models -satisfaction for $\lambda\Pi_{=}$)

Let Σ be a $\lambda\Pi_{=}$ -signature, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be an equational

Kripke Σ - $\lambda\Pi$ -model, Γ be a context, A be a type and M and N be objects. In the equational Kripke Σ - $\lambda\Pi$ -model, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, the world W satisfies (i) the equation $A = B : \text{Type}$ with respect to Γ , *i.e.*,

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (A = B : \text{Type})[\Gamma],$$

if and only if $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are all defined and $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, and (ii) the equation $M = N : A$ with respect to Γ , *i.e.*,

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M = N : A)[\Gamma],$$

if and only if $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : A)[\Gamma]$, $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (N : A)[\Gamma]$ and $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. ■

The raw syntax of signatures, contexts, kinds, types and objects can be extended to a raw syntax of contexts and realizations as follows: if Γ and Δ , where $\Delta \equiv y_1 : B_1, \dots, y_n : B_n$, are raw contexts. Then a raw realization from Γ to Δ is a n -tuple $\langle M_1, \dots, M_n \rangle$ of raw objects, extending the interpretation $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^-$ to raw realizations over an object in the base in the obvious way. If $\sigma = \langle M_1, \dots, M_n \rangle$, then

$$\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \langle \llbracket (M_1)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \dots, \llbracket (M_n)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rangle$$

is defined over $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. We are now able to define our second notion of satisfaction.

Definition 3.26 (\Rightarrow -satisfaction for $\lambda\Pi$)

Let Σ be a valid $\lambda\Pi$ -signature, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke Σ - $\lambda\Pi$ -model, Γ , Δ and Θ be valid $\lambda\Pi$ -contexts, and σ be a valid $\lambda\Pi$ -realization. In the Kripke Σ - $\lambda\Pi$ -model $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, the world W satisfies the realization $\Delta \xrightarrow{\sigma} \Theta$ with respect to Γ , *i.e.*,

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma],$$

if and only if $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket \sigma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are all defined and, for each $W \xrightarrow{\alpha} W'$,

$$\llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \xrightarrow{\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} \llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$$

is defined such that

$$\begin{array}{ccc} \llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} & \llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \\ \downarrow \mathcal{N}^{\alpha} & & \downarrow \mathcal{N}^{\alpha} \\ \llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} & \xrightarrow{\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} & \llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \end{array}$$

holds. ■

Again, we extend the definition to an equational Kripke Σ - $\lambda\Pi$ -model in the natural way.

Definition 3.27 (\models -satisfaction for $\lambda\Pi$)

Let Σ be a Kripke $\lambda\Pi$ -signature, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be an equational Kripke Σ - $\lambda\Pi$ -model, $\Gamma, \Delta \equiv y_1 : B_1, \dots, y_m : B_m, \Delta' \equiv y'_1 : B'_1, \dots, y'_m : B'_m, \Theta \equiv z_1 : C_1, \dots, z_p : C_p$ and $\Theta \equiv z'_1 : C'_1, \dots, z'_p : C'_p$ be contexts and $\sigma = \langle M_1, \dots, M_n \rangle$ and $\sigma' = \langle M'_1, \dots, M'_n \rangle$ be realizations. In the equational Kripke Σ - $\lambda\Pi$ -model, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, the world W satisfies the realization $\Delta = \Delta' \xrightarrow{\sigma=\sigma'} \Theta = \Theta'$ with respect to Γ , *i.e.*,

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta = \Delta' \xrightarrow{\sigma=\sigma'} \Theta = \Theta') [\Gamma],$$

if and only if $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta) [\Gamma]$, $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta' \xrightarrow{\sigma'} \Theta) [\Gamma]$, $\llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket \Delta'_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket \Theta'_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket \sigma'_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined and $\llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket \Delta'_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket \Theta'_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket \sigma'_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. ■

An important special case of this satisfaction relation occurs when $\Delta = \Delta'$ and $\Theta = \Theta'$, *i.e.*, $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta = \Delta \xrightarrow{\sigma=\sigma'} \Theta = \Theta) [\Gamma]$, which amounts to $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma=\sigma'} \Theta) [\Gamma]$.

All the remaining proofs in this section are sketches. Each proof requires an induction over the interpretation which we have left out and instead concentrated on the main argument required for each inductive step. Turning these sketch proofs into formal proofs is straightforward, but tedious.

The following lemma follows immediately from Definitions 3.24 and 3.26. It formalises our earlier comment about the relationship between the two notions of satisfaction.

Lemma 3.28

The satisfaction relation $\models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}}$ is a special case of the satisfaction relation $\models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}}$ as follows: if $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\langle \rangle \xrightarrow{\langle M \rangle} x : A) [\Gamma]$, then $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : A) [\Gamma]$.

Proof (Sketch) Since $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\langle \rangle \xrightarrow{\langle M \rangle} x : A) [\Gamma]$, we know that $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket x : A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined and that $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined and is an object of $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$.

We also have that $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an object in $\mathcal{K}_{\mathcal{J}}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. If $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\langle id, \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rangle} \llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \bullet \llbracket A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an arrow in \mathcal{D}_W . The definition of the interpretation tells us that $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is going to be defined as an arrow in $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. Thus

$\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined. Since $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket x : A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined for

all W' , such that $W \xrightarrow{\alpha} W'$, it follows that $\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ is also defined for all W' . Since

$$\begin{array}{ccc} \llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{\langle \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rangle} & \llbracket x : A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \\ \downarrow \mathcal{N}^{\alpha} & & \downarrow \mathcal{N}^{\alpha} \\ \llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} & \xrightarrow{\langle \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \rangle} & \llbracket x : A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \end{array}$$

holds and we know that \mathcal{N}^{α} sends $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ to $\mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})$. We infer that

$$\begin{array}{ccc} \llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{\langle \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rangle} & \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \\ \downarrow \mathcal{N}^{\alpha} & & \downarrow \mathcal{N}^{\alpha} \\ \llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} & \xrightarrow{\langle \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \rangle} & \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \end{array}$$

holds and we conclude that $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : A)[\Gamma]$. ■

If $W \xrightarrow{\alpha} W'$, then we write $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{[\alpha]}$ to denote $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ after α , *e.g.*, if $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an object of $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ and $W \xrightarrow{\alpha} W'$, then we write $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{[\alpha]}$ for the object $\mathcal{N}^{\alpha}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ in $\mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})$, *etc.*, *...* We also write $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{[\alpha]}$ for $\kappa^{\alpha}(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. Similarly, if $E \xrightarrow{f} D$ in \mathcal{D}_W and if $(-)$ is interpreted over D , then we write $(-)^{[f]}$ to denote $(-)$ interpreted over E , *e.g.*, if $(\Delta \xrightarrow{\sigma} \Theta)$ is interpreted as $\llbracket \Delta \xrightarrow{\sigma} \Theta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ over D , then $(\Delta \xrightarrow{\sigma} \Theta)^{[f]}$ is interpreted as $\mathcal{K}_{\mathcal{J}}(W)(f)(\llbracket \Delta \xrightarrow{\sigma} \Theta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ over E .

The remaining lemmata in this section give the basic logical properties of the satisfaction relations \models and $\models\Rightarrow$. The previous lemma tells us that if we prove the result for $\models\Rightarrow$ then it holds for \models .

Lemma 3.29 (Monotonicity of Satisfaction)

Let Σ be a valid $\lambda\Pi$ -signature, Γ , Δ and Θ be valid $\lambda\Pi$ -contexts and σ be a valid $\lambda\Pi$ -realization. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \overline{\mathcal{V}}]]$, be a Kripke $\lambda\Pi$ -model. If $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma]$ and $W \xrightarrow{\alpha} W'$, then $W' \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma]$.

Proof (Sketch) Since $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma]$, we know that $\llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$

$\llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^W$ is defined and that

$$\begin{aligned} \mathcal{N}^\alpha(\llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^W \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^W} \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^W) &= (\llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^W \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^W} \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^W)^{[\alpha]} \\ &= \llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^{[\alpha]} \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^{[\alpha]}} \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^{[\alpha]} \end{aligned}$$

is defined. It remains to show that for all $W' \xrightarrow{\alpha'} W''$ $\llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^{W''} \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^{W''}} \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^{W''}$ is defined and

$$\begin{array}{ccc} \llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^{W'} & \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^{W'}} & \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^{W'} \\ \downarrow \mathcal{N}^{\alpha'} & & \downarrow \mathcal{N}^{\alpha'} \\ \llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^{W''} & \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^{W''}} & \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^{W''} \end{array}$$

holds. Since \mathcal{W} is a category, given $W \xrightarrow{\alpha} W'$ and $W' \xrightarrow{\alpha'} W''$, there exists an arrow $W \xrightarrow{\alpha; \alpha'} W''$. Thus we have that $\llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^{W''} \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^{W''}} \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^{W''}$ is defined and that

$$\begin{array}{ccc} \llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^W & \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^W} & \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^W \\ \downarrow \mathcal{N}^{\alpha; \alpha'} & & \downarrow \mathcal{N}^{\alpha; \alpha'} \\ \llbracket \Delta_\Gamma \rrbracket_{\mathcal{K}_J}^{W''} & \xrightarrow{\llbracket \sigma_\Gamma \rrbracket_{\mathcal{K}_J}^{W''}} & \llbracket \Theta_\Gamma \rrbracket_{\mathcal{K}_J}^{W''} \end{array}$$

holds. $\mathcal{N}^{\alpha; \alpha'} = \mathcal{N}^\alpha; \mathcal{N}^{\alpha'}$, so we get the necessary conditions for $W' \models_{\Sigma}^{\mathcal{K}_J} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma]$ \blacksquare

Lemma 3.30 (Weakening of Satisfaction)

Let Σ be a valid $\lambda\Pi$ -signature, $\Gamma, \Gamma', \Delta, \Xi$ and Ξ' be valid $\lambda\Pi$ -contexts and σ be a valid $\lambda\Pi$ -realization. Let $\langle \mathcal{K}_J, \llbracket - \rrbracket_{\mathcal{K}_J} \rangle$, where $\mathcal{K}_J : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke $\lambda\Pi$ -model. If $W \models_{\Sigma}^{\mathcal{K}_J} (\Xi \xrightarrow{\sigma} \Xi')[\Gamma, \Gamma']$ and if \mathbf{N} proves $\vdash_{\Sigma} \Gamma, \Delta, \Gamma'$ context, then $W \models_{\Sigma}^{\mathcal{K}_J} (\Xi \xrightarrow{\sigma} \Xi')^{[p]}[\Gamma, \Delta, \Gamma']$, where $\llbracket \Gamma, \Delta, \Gamma' \rrbracket_{\mathcal{K}_J}^W \xrightarrow{p} \llbracket \Gamma, \Gamma' \rrbracket_{\mathcal{K}_J}^W$ is the obvious projection.

Proof (Sketch) Since \mathbf{N} proves $\vdash_{\Sigma} \Gamma, \Delta, \Gamma'$ context, we know that we have a context Γ, Δ, Γ' , which we can interpret in our Kripke Σ - $\lambda\Pi$ -model and a projection

$[[\Gamma, \Delta, \Gamma']]_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{p} [[\Gamma, \Gamma']]_{\mathcal{K}_{\mathcal{J}}}^W$. p induces a functor p^* which sends the realization $[[\Xi_{\Gamma, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{[[\sigma_{\Gamma, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W} [[\Xi'_{\Gamma, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W$ to $[[\Xi_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{[[\sigma_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W} [[\Xi'_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W = [[(\Xi \xrightarrow{\sigma} \Xi')^{[p]}]]_{\mathcal{K}_{\mathcal{J}}}^W$. We also have that for all W' , such that $W \xrightarrow{\alpha} W'$, $[[\Xi_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^{W'} \xrightarrow{[[\sigma_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^{W'}} [[\Xi'_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^{W'}$ is defined, and

$$\begin{array}{ccc} [[\Xi_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{[[\sigma_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W} & [[\Xi'_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W \\ \downarrow \mathcal{N}^\alpha & & \downarrow \mathcal{N}^\alpha \\ [[\Xi_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^{W'} & \xrightarrow{[[\sigma_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^{W'}} & [[\Xi'_{\Gamma, \Delta, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^{W'} \end{array}$$

holds because we can apply \mathcal{N}^α to the realization in Γ, Γ' and then apply p^* . This is equivalent to applying \mathcal{N}^α after p^* because p^* only sends the realization to another fibre. Thus we can conclude $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Xi \xrightarrow{\sigma} \Xi')^{[p]}[\Gamma, \Delta, \Gamma']$. \blacksquare

Lemma 3.31 (Substitutivity of Satisfaction)

Let Σ be a valid $\lambda\Pi$ -signature, Γ, Γ', Θ and Δ be valid $\lambda\Pi$ -contexts and σ be a realization. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}}: [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke $\lambda\Pi$ -model. If $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma, x: A, \Gamma']$, \mathbf{N} proves $\Gamma \vdash_{\Sigma} N: C$ and $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (N: C)[\Gamma]$, then $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta[N/x] \xrightarrow{\sigma[N/x]} \Theta[N/x])[\Gamma, \Gamma'[N/x]]$.

Proof (Sketch) Since \mathbf{N} proves $\Gamma \vdash_{\Sigma} N: C$ we know that N is a term of type C .

Thus we have arrows $[[\langle \rangle_{\Gamma}]]_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{[[N_{\Gamma}]]_{\mathcal{K}_{\mathcal{J}}}^W} [[C_{\Gamma}]]_{\mathcal{K}_{\mathcal{J}}}^W$ and $[[\Gamma]]_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\langle id, [N_{\Gamma}]_{\mathcal{K}_{\mathcal{J}}}^W \rangle} [[\Gamma, x: C]]_{\mathcal{K}_{\mathcal{J}}}^W$ in $\mathcal{J}(W)(\Gamma)$ and \mathcal{D}_W respectively. We now have the pullback

$$\begin{array}{ccc} [[\Gamma, \Gamma'[N/x]]]_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{[[\langle 1_{\Gamma}, N, 1_{\Gamma'[N/x]} \rangle]]_{\mathcal{K}_{\mathcal{J}}}^W} & [[\Gamma, x: C, \Gamma']]_{\mathcal{K}_{\mathcal{J}}}^W \\ \downarrow p & & \downarrow p \\ [[\Gamma]]_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{[[\langle 1_{\Gamma}, N \rangle]]_{\mathcal{K}_{\mathcal{J}}}^W} & [[\Gamma, x: C]]_{\mathcal{K}_{\mathcal{J}}}^W \end{array}$$

where each p denotes the composition of projections and $[[\langle 1_{\Gamma}, N \rangle]]_{\mathcal{K}_{\mathcal{J}}}^W \bullet [[\Gamma']]_{\mathcal{K}_{\mathcal{J}}}^W = [[\langle 1_{\Gamma}, N, 1_{\Gamma'[N/x]} \rangle]]_{\mathcal{K}_{\mathcal{J}}}^W$. We can now apply $([[\langle 1_{\Gamma}, N, 1_{\Gamma'[N/x]} \rangle]]_{\mathcal{K}_{\mathcal{J}}}^W)^*$ to $\mathcal{K}_{\mathcal{J}}(W)([[\Gamma, x: C, \Gamma']]_{\mathcal{K}_{\mathcal{J}}}^W)$. Thus the realization $[[\Delta_{\Gamma, x: C, \Gamma'}]]_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\sigma_{\Gamma, x: C, \Gamma'}} \Theta_{\Gamma, x: C, \Gamma'}^W$ is sent to $[[\Delta[N/x]_{\Gamma, \Gamma'[N/x]}]]_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\sigma[N/x]_{\Gamma, \Gamma'[N/x]}} \Theta[N/x]_{\Gamma, \Gamma'[N/x]}^W$ in $\mathcal{K}_{\mathcal{J}}(W)([[\Gamma, \Gamma'[N/x]]]_{\mathcal{K}_{\mathcal{J}}}^W)$.

We have that $\llbracket \Delta[N/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$, $\llbracket \Theta[N/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ and $\llbracket \sigma[N/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ are defined for all W' , where $W \xrightarrow{\alpha} W'$, because $\llbracket \Delta_{\Gamma, x:C, \Gamma'} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$, $\llbracket \Theta_{\Gamma, x:C, \Gamma'} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ and $\llbracket \sigma_{\Gamma, x:C, \Gamma'} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ are defined for all W' , where $W \xrightarrow{\alpha} W'$. We apply $(\llbracket \langle 1_{\Gamma}, N, 1_{\Gamma'[N/x]} \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)^*$ to these and obtain the required results. Similarly, we see that

$$\begin{array}{ccc} \llbracket \Delta[N/x]_{\Gamma, \Gamma'[N/x]} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{\llbracket \sigma[N/x]_{\Gamma, \Gamma'[N/x]} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} & \llbracket \Theta[N/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \\ \downarrow \mathcal{N}^{\alpha} & & \downarrow \mathcal{N}^{\alpha} \\ \llbracket \Delta[N/x]_{\Gamma, \Gamma'[N/x]} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} & \xrightarrow{\llbracket \sigma[N/x]_{\Gamma, \Gamma'[N/x]} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} & \llbracket \Theta[N/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \end{array}$$

holds since applying $(\llbracket \langle 1_{\Gamma}, N, 1_{\Gamma'[N/x]} \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)^*$ followed by \mathcal{N}^{α} gives the same result as applying \mathcal{N}^{α} followed by $(\llbracket \langle 1_{\Gamma}, N, 1_{\Gamma'[N/x]} \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)^*$. Thus $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta[N/x] \xrightarrow{\sigma[N/x]} \Theta[N/x])[\Gamma, \Gamma'[N/x]]$. \blacksquare

Lemma 3.32 (Strengthening of Satisfaction)

Let Σ be a valid $\lambda\Pi$ -signature, Γ , Δ and Θ be valid $\lambda\Pi$ -contexts and σ be a realization. Let $(\mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}})$, where $\mathcal{K}_{\mathcal{J}}: [W, [D^{op}, \bar{V}]]$, be a Kripke Σ - $\lambda\Pi$ -model. If $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma, x:C]$, \mathbf{N} proves $\vdash_{\Sigma} \Gamma$ context and $x \notin FV(\Delta, \sigma, \Theta)$, then $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)^{[q]}[\Gamma]$, where $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{q} \llbracket \Gamma, x:C \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$.

Proof (Sketch) Since \mathbf{N} proves $\vdash_{\Sigma} \Gamma$ context, there exists an arrow $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{q} \llbracket \Gamma, x:C \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, where $q = \langle id, \llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rangle$, for some $\llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. q induces a functor $q^*: \mathcal{K}_{\mathcal{J}}(W)(\llbracket \Gamma, x:C \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) \rightarrow \mathcal{K}_{\mathcal{J}}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. q^* takes $\llbracket \Delta \xrightarrow{\sigma} \Theta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ to $\llbracket (\Delta \xrightarrow{\sigma} \Theta)[N/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ in $\mathcal{K}_{\mathcal{J}}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. Since $x \notin FV(\Delta, \sigma, \Theta)$, $(\Delta \xrightarrow{\sigma} \Theta)[N/x] = (\Delta \xrightarrow{\sigma} \Theta)$. $\llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \xrightarrow{\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} \llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ is defined for all W' , such that $W \xrightarrow{\alpha} W'$, because we apply q^* to all future interpretations. Similarly,

$$\begin{array}{ccc} \llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W & \xrightarrow{\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} & \llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \\ \downarrow \mathcal{N}^{\alpha} & & \downarrow \mathcal{N}^{\alpha} \\ \llbracket \Delta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} & \xrightarrow{\llbracket \sigma_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} & \llbracket \Theta_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \end{array}$$

holds and thus $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)^{[q]}[\Gamma]$. \blacksquare

Lemma 3.33 (Exchange)

Let Σ be a valid $\lambda\Pi$ -signature, Γ and Δ be valid $\lambda\Pi$ -contexts and let X range over all appropriate expressions of the form $M : A$. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\bar{\cdot}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke Σ - $\lambda\Pi$ -model. If $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (X)[\Gamma, \Delta]$ and $\llbracket \Delta, \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket X_{\Delta, \Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined, then $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (X)[\Delta, \Gamma]$.

Proof (Sketch) We have an arrow $\llbracket \Delta, \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{f} \llbracket \Gamma, \Delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ since $\llbracket \Delta, \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined. This arrow induces a functor $f^* : \mathcal{J}(W)(\llbracket \Gamma, \Delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) \rightarrow \mathcal{J}(W)(\llbracket \Delta, \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ such that $\llbracket X_{\Gamma, \Delta} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is sent to $\llbracket X_{\Delta, \Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. For every choice of X , we have an arrow

$\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \xrightarrow{\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}} \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ which is defined for all W' such that $W \xrightarrow{\alpha} W'$ because we can apply f^* to all future interpretations. Similarly, we get the coherence condition for \mathcal{N}^{α} . Thus $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (X)[\Delta, \Gamma]$. \blacksquare

Lemma 3.34 (Π -forcing)

Let Σ be a $\lambda\Pi$ -signature and $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\bar{\cdot}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke $\lambda\Pi$ -model. $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : \Pi x : A . B)[\Gamma]$ if and only if, for all $W \xrightarrow{\alpha} W'$ and for all N such that $W' \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (N : A)^{[\alpha]}[\Gamma]$, there is a P such that $W' \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (P : B[N/x])^{[\alpha]}[\Gamma]$ and $P =_{\beta\eta} MN$. Similarly, for the non-dependent function space, \rightarrow .

Proof (Sketch) If $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : \Pi x : A . B)[\Gamma]$, then $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket \Pi x : A . B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined over $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. If $W \xrightarrow{\alpha} W'$, then by monotonicity, we have that $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{[\alpha]} \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{[\alpha]}} \llbracket \Pi x : A . B \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{[\alpha]}$ is defined over $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{[\alpha]}$. Moreover, we must have that the adjunction defining the function space at W' is in the image of $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$, i.e.,

$$p_{[\Gamma]_{\mathcal{K}_{\mathcal{J}}}^{W'}, [A_{\Gamma}]_{\mathcal{K}_{\mathcal{J}}}^{W'}}^* \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} ; app_{W'}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}, \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})$$

is an arrow $\llbracket \langle \rangle_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \rightarrow \llbracket B_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$. By Definition 3.17, $\llbracket MN \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ is defined and equal to $(\llbracket \langle x_1, \dots, x_m \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})^* (p_{[\Gamma]_{\mathcal{K}_{\mathcal{J}}}^{W'}, [A_{\Gamma}]_{\mathcal{K}_{\mathcal{J}}}^{W'}}^* \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} ; app_{W'}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}, \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}))$,

where as usual, $\Gamma \equiv x_1 : A_1, \dots, x_m : A_m$). As we noted above, $p_{[\Gamma]_{\mathcal{K}_{\mathcal{J}}}^{W'}, [A_{\Gamma}]_{\mathcal{K}_{\mathcal{J}}}^{W'}}^*$

$\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} ; app_{W'}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}, \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})$ is an arrow $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \rightarrow \llbracket B_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$ over $\llbracket \Gamma, x :$

$A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$. The functor $(\llbracket \langle x_1, \dots, x_n, N \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'})^*$ sends this arrow to $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'} \xrightarrow{\llbracket MN_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}}$

$\llbracket B[N/x]_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W'}$. It remains to show that this is defined for all future worlds W'' such that $W' \xrightarrow{\alpha} W''$ and that it satisfies the coherence conditions. We observe that the argument holds for all accessible worlds and thus $W' \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (MN : B[N/x])[\Gamma]$.

For the converse, we observe that the argument can be reversed. \blacksquare

3.7 Soundness and Completeness of $\lambda\Pi$ and $\lambda\Pi_{=}$

We now prove soundness and completeness for our Kripke Σ - $\lambda\Pi$ -models. We need to provide proofs for both notions of satisfaction. We begin with \models because we need to introduce a calculus of realizers before we can deal with $\models_{=}$. Soundness is proven by an induction over the structure of \mathbf{N} and \mathbf{C} , the calculus of realizers and is thus rather lengthy. Completeness is obtained via a constructed term model. This model will turn out to be the Kripke Σ - $\lambda\Pi$ -model we gave in § 3.4.1.

Soundness and Completeness of $\lambda\Pi$ and $\lambda\Pi_{=}$ for \models

The proof of soundness is by induction over the structure of proofs in the system \mathbf{N} and must, formally, be performed simultaneously with the proofs of Lemmas 3.18, 3.19 and 3.20. To see why this must be so, consider that in order for the interpretation of a type to be well-defined, the interpretation of the context in which its variables are declared must be well-defined. Similarly, if the interpretation of an object as an arrow is to be well-defined, then the interpretation of the context in which its variables are declared must be well-defined. Provided we are mindful of these induction dependencies, we can proceed without undue concern.

Theorem 3.35 (Soundness of $\lambda\Pi$ for \models)

Let Σ be a valid $\lambda\Pi$ -signature, Γ be a valid $\lambda\Pi$ -context and M be a valid $\lambda\Pi$ -object. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}}: [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke Σ - $\lambda\Pi$ -model. Let W be a world, i.e., an object of \mathcal{W} . If \mathbf{N} proves $\Gamma \vdash_{\Sigma} M:A$ and if $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined then $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M:A)[\Gamma]$ (and so $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is a section of $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ over $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$). Moreover, if $U =_{\beta\eta} V$, then $\llbracket U \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket V \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$.

Proof (Sketch) With the above remarks and the statements of Lemmas 3.18, 3.19 and 3.20 in mind, we proceed to give the main cases of the argument. We give the main steps in each of the cases, leaving the reader to perform the extra calculations should they so desire. We show that the rules (2.13) - (2.17) hold.

Suppose $\Gamma \vdash_{\Sigma} c: \underbrace{\Pi x_1:A_1 \dots \Pi x_m:A_m}_{\mathcal{C}} \cdot A$ is an axiom sequent of \mathbf{N} (2.13).

By Definition 3.3, $\mathcal{K}_{\mathcal{J}}$ has Σ -operations. $c:C \in \Sigma$ means we have $\llbracket \langle \rangle_{\Gamma \bullet \bar{A}} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{op_c} \llbracket A_{\Gamma \bullet \bar{A}} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, where $\Gamma \bullet \bar{A} = \Gamma, x_1:A_1, \dots, x_m:A_m$. We thus interpret c_{Γ} as $cur_W^m(op_c)$. In order to show that $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (c:C)[\Gamma]$, for W such that $\llbracket C_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined, we need to show that

$$\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket c_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket C_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

is an arrow in $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. By the induction hypothesis, *pace* Lemma 3.18, we have that $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is well-defined. According to Definition 3.3, $\llbracket c_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W =$

$cur_W^m(op_c)$ holds. The satisfaction condition follows immediately from (3.1).

Suppose $(\Gamma \equiv) \Delta, x : A, \Delta' \vdash_{\Sigma} x : A$ is an axiom sequent (2.14). In order to show that $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (x : A)[\Gamma]$, for W such that $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined. We need to prove that

$$\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket x_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

is an arrow in $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. By the induction hypothesis, *pace* Lemmas 3.18, 3.19 and 3.20, we have that $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket x_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are well-defined. According to Definition 3.3,

$$\llbracket x_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = p_{\llbracket \Delta, x : A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket \Delta' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^* (q_{\llbracket \Delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}).$$

The satisfaction condition is obtained by noting that, according to Definition 3.2, $q_{\llbracket \Delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket A_{\Delta} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$ has the appropriate domain and co-domain and that the pullback functors preserve the terminal object on the nose.

Suppose that the last rule of **N** applied is ΠI (2.15),

$$\frac{\Gamma, x : A \vdash_{\Sigma} M : B}{\Gamma \vdash_{\Sigma} \lambda x : A . M : \Pi x : A . B}$$

By the induction hypothesis, *pace* Lemmas 3.18, 3.19 and 3.20. We have, for W such that $\llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined, that

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : B)[\Gamma, x : A]$$

i.e., that $\llbracket \langle \rangle_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket M_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket B_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. According to Definition 3.3, $\llbracket (\lambda x : A)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined and equal to $cur_W(\llbracket M_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$, from which it follows, via (3.1) that

$$\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket (\lambda x : A . M)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket (\Pi x : A . B)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

is an arrow in $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$.

Suppose that the last rule of **N** applied is ΠE (2.16),

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

then by the induction hypothesis, *pace* Lemmas 3.18, 3.19 and 3.20. We have that

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : \Pi x : A . B)[\Gamma]$$

i.e., that $\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket (\Pi x : A . B)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. Also

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (N : A)[\Gamma],$$

i.e., that $\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. According to Definition 3.3, the arrow $\llbracket (MN)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is defined and equal to

$$(\llbracket \langle x_1, \dots, x_m, N \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)^* (p_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}^* \llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W; \text{app}_W(\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket B_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)),$$

where $\Gamma \equiv x_1 : A_1, \dots, x_m : A_m$. A brief inspection of this expression reveals that it has the correct type: from (3.2), and by the induction hypothesis, $\text{app}_W(\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket B_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$ is an arrow

$$\llbracket \langle \rangle_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rightarrow \llbracket (\Pi x : A . B)_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

over $\llbracket \Gamma, x : A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. Finally, $\llbracket (MN)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is the image of this arrow over $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, under the pullback functor $(\llbracket \langle x_1, \dots, x_m, N \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)^*$, which performs the required substitution.

Suppose that the last rule of **N** applied is (2.17),

$$\frac{\Gamma \vdash_{\Sigma} M : A \quad \Gamma \vdash_{\Sigma} A' : \text{Type} \quad A = \beta\eta A'}{\Gamma \vdash_{\Sigma} M : A'}$$

i.e., $\beta\eta$ -equality. This case follows in a standard way (*cf.* (Pitts 2000), (Ritter 1992), (Streicher 1989) and (Jacobs 1991)) from the interpretation of the dependent function space via the right adjoint to substitution. The only novelty here is that the focus of our attention must be the rightmost premiss $A =_{\beta\eta} A'$. It is convenient to observe that since we are restricting our semantics to the $M : A : \text{Type}$ -fragment of the $\lambda\Pi$ -calculus, β -equalities are generated by the rule

$$\frac{\Gamma, x : A \vdash_{\Sigma} M : B \quad \Gamma \vdash_{\Sigma} N : A}{\Gamma \vdash_{\Sigma} (\lambda x : A . M)N =_{\beta} M[N/x] : B[N/x]}$$

for some appropriate Γ . Similarly, η -equalities are generated by the rule

$$\frac{\Gamma \vdash_{\Sigma} M : \Pi x : A . B \quad y \notin FV(\Gamma, x : A)}{\Gamma \vdash_{\Sigma} \lambda y : A . My =_{\eta} M : B}$$

for some appropriate Γ . In our fragment, the type A is a $\beta\eta$ -redex just in case there is an object M , on which A depends, is a $\beta\eta$ -redex. The argument exploits the natural isomorphism (2.17) and Lemma 3.31 to get β -equality and Lemmas 3.30 and 3.31 to get η -equality. So if $M =_{\beta\eta} N$, then $\llbracket M \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$.

A similar argument is presented in full in (Streicher 1989); we leave the detailed calculation in our setting to the reader. \blacksquare

In the proof, the dependent function space was modelled by the right adjunction to substitution. If we did not have η -equality, then a semi-adjunction would be sufficient, *cf.* (Hayashi 1985) and (Jacobs 1991).

Corollary 3.36 (Definitional Equality)

Let Σ be a valid $\lambda\Pi_{=}$ -signature, Γ be a valid $\lambda\Pi_{=}$ -context and M and N be valid $\lambda\Pi_{=}$ -objects. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be an equational Kripke Σ - $\lambda\Pi_{=}$ -model.

1. If $\Gamma \vdash_{\Sigma} A : \mathbf{Type}$ and $\Gamma \vdash_{\Sigma} B : \mathbf{Type}$ are provable in $\mathbf{N}_{=}$, then $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$.
2. If $\Gamma \vdash_{\Sigma} M : A$, $\Gamma \vdash_{\Sigma} N : A$ and $\Gamma \vdash_{\Sigma} M = N : A$ are provable in $\mathbf{N}_{=}$, then $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$.

Proof (Sketch) By induction on the structure of proofs in $\mathbf{N}_{=}$, making essential use of the requirement in the definition of a Kripke Σ - $\lambda\Pi_{=}$ -model that if $M = N : A \in \Sigma$, then $\llbracket M_{\langle \rangle} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\langle \rangle} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$.

We prove transitivity (3.11) as an example. By the induction hypothesis, $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket C_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined and $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket C_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. Since $=$ is an equivalence relation, we get $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket C_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$.

Suppose that the last rule used was (3.13). By Definition 3.23, $\llbracket M_{\langle \rangle} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\langle \rangle} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. The induction hypothesis tells us that $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is well-defined and that there is an arrow $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{p} \llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. We apply p^* and observe that $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$.

If the last rule used was (3.15), then we can apply the induction hypothesis to obtain $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an arrow

$$\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rightarrow \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

but since $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, we also have an arrow

$$\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rightarrow \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

as required. $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : A)[\Gamma]$ and $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (N : B)[\Gamma]$ by Theorem 3.35.

Suppose that the last rule applied was (3.16). We apply the induction hypothesis to obtain $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are well-defined. The assertion $\Gamma_x \vdash_{\Sigma} \Gamma(x) = \Gamma'(x) : \mathbf{Type}$ tells us that Γ and Γ' are equal up to the choice of a free variable. This means that there is an arrow

$$\llbracket \Gamma' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\langle \text{id}, \llbracket x_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rangle} \llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

which induces a functor $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rightarrow \mathcal{J}(W)(\llbracket \Gamma' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W)$. This takes $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ to $\llbracket A_{\Gamma'} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma'} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. Since $x \notin FV(A, B, K)$. We have that $\llbracket A_{\Gamma'} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma'} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. This argument is also used for (3.17). We observe that $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : A)[\Gamma]$ and $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (N : B)[\Gamma]$, by Theorem 3.35.

We assume that the last rule used was (3.19). By the induction hypothesis we have that $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket C_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket D_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. We can apply $\Pi_{\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W, \llbracket A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$ to $\llbracket C_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ to obtain

$$\llbracket (\Pi x : B . C)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket (\Pi x : B . D)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W.$$

Since $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket C_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket D_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket C_{\Gamma, x : B} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket D_{\Gamma, x : B} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ we have that

$$\llbracket (\Pi x : A . C)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket (\Pi x : B . D)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W.$$

If the last rule used was (3.25) then by the induction hypothesis, we have that $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket M_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. Since $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, we also have

$$\llbracket M_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma, x : A} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket M_{\Gamma, x : B} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma, x : B} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W.$$

We can now apply cur_W to these arrows to obtain

$$\llbracket (\lambda x : A . M)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket (\lambda x : A . N)_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

as required.

If (3.27) was the last rule used then we apply the induction hypothesis to obtain $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket P_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket Q_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. From Theorem 3.35, we observe that $\llbracket MP_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is well-defined and gives the correct arrow. It then follows from the above equalities that

$$\llbracket MN_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket NQ_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$$

as required.

Finally, if the last rule was (3.29), then applying the induction hypothesis gives us that $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are well-defined. $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is an arrow $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rightarrow \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. Since $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ is also an arrow $\llbracket \langle \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \rightarrow \llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. \blacksquare

Having proven soundness, we now turn to completeness. The proof of completeness is obtained via a term model construction. We show that there exists a (equational) Kripke Σ - $\lambda\Pi$ ($=$)-model where the object cannot be forced at a given world. Usually, a term model is obtained via the prime extension of a theory, *cf.*,

the proof of the completeness theorem for the intuitionistic predicate calculus in (Van Dalen 1994). In the absence of any positive connectives, we are able to do without such a construction. A consequence of a direct construction is that we construct long $\beta\eta$ -normal forms when dealing with occurrences of function types of the form

$$\Gamma, x:\Pi y:A. B \vdash_{\Sigma} X$$

where X ranges over the assertions of the $\lambda\Pi$ -calculus. Before we can prove model existence results, we need to define validity for \models .

Definition 3.37 (\models -validity for $\lambda\Pi$)

Let Σ be a valid $\lambda\Pi$ -signature, Γ be a valid $\lambda\Pi$ -context and M be a valid $\lambda\Pi$ -object. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}}: [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke Σ - $\lambda\Pi$ -model. We say that $M:A$ is *valid* with respect to Γ , *i.e.*,

$$\Gamma \models_{\Sigma} M:A$$

if and only if, for all Kripke Σ - $\lambda\Pi$ -models and all worlds W where $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined, $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M:A)[\Gamma]$. \blacksquare

We extend this definition to equational Kripke Σ - $\lambda\Pi_{=}$ -models.

Definition 3.38 (\models -validity for $\lambda\Pi_{=}$)

Let Σ be a valid $\lambda\Pi_{=}$ -signature, Γ be a valid $\lambda\Pi_{=}$ -context and M be a valid $\lambda\Pi_{=}$ -object. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}}: [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be an equational Kripke Σ - $\lambda\Pi_{=}$ -model.

1. $A = B:\text{Type}$ is *valid* with respect to Γ , *i.e.*,

$$\Gamma \models_{\Sigma} A = B:\text{Type}$$

if and only if, for all equational Kripke Σ - $\lambda\Pi_{=}$ -models and all worlds W where $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket B_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined, $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (A = B:\text{Type})[\Gamma]$.

2. $M = N:A$ is *valid* with respect to Γ , *i.e.*,

$$\Gamma \models_{\Sigma} M = N:A$$

if and only if, for all equational Kripke Σ - $\lambda\Pi_{=}$ -models and all worlds W where $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket N_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M = N:A)[\Gamma]$. \blacksquare

The following lemma is key to our completeness proof.

Lemma 3.39 (Model Existence)

Let Σ be a valid $\lambda\Pi$ -signature, Δ be a valid $\lambda\Pi$ -context and M be a valid $\lambda\Pi$ -object. There is a Kripke Σ - $\lambda\Pi$ -model $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \overline{\mathcal{V}}]]$, and a world W_0 such that if $\Delta \not\vdash_{\Sigma} M : A$, then $W_0 \not\models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : A)[\Delta]$.

Proof Recall the Kripke Σ - $\lambda\Pi$ -model we constructed in § 3.2.1 and 3.4.1. We have that $\Gamma \models_{\Sigma}^{\mathcal{K}_{\mathcal{T}_{\Sigma}}} (M : A)[\Gamma \bowtie \Delta]$ if and only if $\Gamma \bowtie \Delta \vdash_{\Sigma} M : A$. We take $W_0 = \Gamma = \langle \rangle$ and then we have $W_0 \models_{\Sigma}^{\mathcal{K}_{\mathcal{T}_{\Sigma}}} (M : A)[\Delta]$ if and only if $\Delta \vdash_{\Sigma} M : A$. This is the required model. \blacksquare

The appropriate equational Kripke Σ - $\lambda\Pi_{=}$ -model is obtained by quotienting the term model.

Corollary 3.40 (Equational Model Existence)

Let Σ be a valid $\lambda\Pi_{=}$ -signature, Δ be a valid $\lambda\Pi$ -context, M and N be valid $\lambda\Pi$ -objects. There is an equational Kripke Σ - $\lambda\Pi_{=}$ -model $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \overline{\mathcal{V}}]]$, and a world W_0 such that

1. if $\Delta \vdash_{\Sigma} A = B : \text{Type}$, then $W_0 \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (A = B : \text{Type})[\Delta]$;
2. if $\Delta \vdash_{\Sigma} M = N : A$, then $W_0 \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M = N : A)[\Delta]$. \blacksquare

Proof We take the model $\mathcal{K}_{\mathcal{T}_{\Sigma}}$ used in Lemma 3.39 and form a model $\mathcal{K}_{\mathcal{T}_{\Sigma}}^{\equiv}$. We firstly quotient $\mathcal{B}(\Sigma)$ to obtain $\mathcal{B}(\Sigma)_{=}$. We replace $\mathcal{B}(\Sigma)$ by $\mathcal{B}(\Sigma)_{=}$ in the definitions of $\mathcal{P}(\Sigma)$ and \mathcal{C}_{Γ} to obtain $\mathcal{P}(\Sigma)_{=}$ and $\mathcal{C}_{\Gamma}^{\equiv}$. We now define $\mathcal{T}_{\Sigma}^{\equiv}$ according to the definition of \mathcal{T}_{Σ} but using $\mathcal{P}(\Sigma)_{=}$ and $\mathcal{C}_{\Gamma}^{\equiv}$ instead of $\mathcal{P}(\Sigma)$ and $\mathcal{C}_{\Gamma}^{\equiv}$, and quotienting each fibre $\mathcal{T}_{\Sigma}(\Gamma)(\Gamma \bowtie \Delta)$ by $=$. The category $\mathcal{K}_{\mathcal{T}_{\Sigma}}^{\equiv}$ is then defined according to the definition of $\mathcal{K}_{\mathcal{T}_{\Sigma}}$ except that $\mathcal{T}_{\Sigma}^{\equiv}$ is used instead of \mathcal{T}_{Σ} .

We observe that we still have $\Gamma \bowtie \Delta \vdash_{\Sigma} A : \text{Type}$ if and only if, A is an object of $\mathcal{T}_{\Sigma}^{\equiv}(\Gamma)(\Gamma \bowtie \Delta)$. Since we have quotiented by provable equality, we get that $\llbracket A_{\Gamma \bowtie \Delta} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\Gamma} = \llbracket B_{\Gamma \bowtie \Delta} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\Gamma}$ and $\llbracket M_{\Gamma \bowtie \Delta} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\Gamma} = \llbracket N_{\Gamma \bowtie \Delta} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\Gamma}$ for all worlds Γ . Thus at world $W_0 = \langle \rangle = \Gamma$, $\Delta \vdash_{\Sigma} A = B : \text{Type}$ if and only if $W_0 \models_{\Sigma}^{\mathcal{K}_{\mathcal{T}_{\Sigma}}} (A = B : \text{Type})[\Delta]$, and $\Gamma \vdash_{\Sigma} M = N : A$ if and only if, $W_0 \models_{\Sigma}^{\mathcal{K}_{\mathcal{T}_{\Sigma}}} (M = N : A)[\Delta]$. \blacksquare

We have the following results.

Theorem 3.41 (Completeness)

Let Σ be a valid $\lambda\Pi$ -signature, Γ be a valid $\lambda\Pi$ -context and M be a valid $\lambda\Pi$ -object. $\Gamma \vdash_{\Sigma} M : A$ if and only if $\Gamma \models_{\Sigma} M : A$.

Proof

(Only If) By soundness, Theorem 3.35.

(If) Suppose that $\Gamma \not\vdash_{\Sigma} M : A$, then Lemma 3.39 yields a contradiction. \blacksquare

Corollary 3.42 (Equational Completeness)

Let Σ be a valid $\lambda\Pi_{=}$ -signature, Γ be a valid $\lambda\Pi_{=}$ -context and M and N be valid $\lambda\Pi_{=}$ -objects. We have

1. $\Gamma \vdash_{\Sigma} A = B : \mathbf{Type}$ if and only if $\Gamma \models_{\Sigma} A = B : \mathbf{Type}$;
2. $\Gamma \vdash_{\Sigma} M = N : A$ if and only if $\Gamma \models_{\Sigma} M = N : A$.

Proof

(Only If) By soundness, Corollary 3.36.

(If) Suppose that $\Gamma \not\vdash_{\Sigma} A = B : \mathbf{Type}$, then Lemma 3.40 yields a contradiction. Similarly, if $\Gamma \not\vdash_{\Sigma} M = N : A$, then Lemma 3.40 yields a contradiction. ■

3.8 Soundness and Completeness of $\lambda\Pi$ and $\lambda\Pi_{=}$ for \models

We now turn to our second notion of satisfaction. The calculi $\lambda\Pi$ and $\lambda\Pi_{=}$ do not deal with assertions of the form $\vdash_{\Sigma} \Delta \xrightarrow{\sigma} \Theta$, which \models forces. We introduce a calculus of realizers, \mathbf{C} , which handles assertions of this form. The calculus we use was first introduced by Galmiche & Pym (2000).

Definition 3.43 (The system \mathbf{C})

Let Σ be a $\lambda\Pi$ -signature and let $\Gamma \equiv x_1 : A_1, \dots, x_m : A_m$ and $\Theta = y_1 : D_1, \dots, y_n : D_n$ be valid $\lambda\Pi$ -contexts. The system \mathbf{C} is given by the following rules:

$$\frac{}{\vdash_{\Sigma} \Gamma \xrightarrow{\langle @_1, \dots, @_n \rangle} \Theta} \text{Axiom} \quad (3.30)$$

where each $@_i \in \Sigma \cup \Gamma$ and \mathbf{N} proves $\Gamma \vdash_{\Sigma} @_i : D_i[\@_j/y_j]_{j=1}^{i-1}$ for $1 \leq i \leq n$;

$$\frac{\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M_n \rangle} \Theta \quad \vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M'_i \rangle} \Gamma, D_i[M_j/y_j]_{j=1}^{i-1}}{\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_n \rangle} \Theta} \text{Application} \quad (3.31)$$

where $M_i : \Pi x : B. C \in \Sigma \cup \Gamma$, \mathbf{N} proves $\Gamma \vdash_{\Sigma} P : B$ and $M_i P =_{\beta\eta} M'_i$, for some $1 \leq i \leq n$;

$$\frac{\vdash_{\Sigma} \Gamma, x : A \xrightarrow{\langle 1_{\Gamma}, M \rangle} \Gamma, x : A, y : B}{\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, \lambda x : A. M \rangle} \Gamma, y : \Pi x : A. B} \text{Introduction} \quad (3.32)$$

$$\frac{\vdash_{\Sigma} \Gamma \xrightarrow{\sigma} \Theta \quad \Gamma =_{\beta\eta} \Gamma' \quad \sigma =_{\beta\eta} \sigma' \quad \Theta =_{\beta\eta} \Theta'}{\vdash_{\Sigma} \Gamma' \xrightarrow{\sigma'} \Theta'} \text{Equality} \quad (3.33)$$

where each equality is defined componentwise. \blacksquare

We extend this system by adding all the definitional equalities induced by the rules of Table 2. Equality is defined componentwise and $\Gamma = \Gamma'$ means $x_i : A_i = A'_i$, for $1 \leq i \leq m$.

Definition 3.44 (The System $\mathbf{C}_=$)

Let Σ be a valid $\lambda\Pi_-$ -signature, Γ, Γ', Θ and Θ' be valid $\lambda\Pi_-$ -contexts and σ and σ' be valid $\lambda\Pi_-$ -realizations. We add the following rules to \mathbf{C} to obtain $\mathbf{C}_=$.

$$\frac{\vdash_{\Sigma} \Gamma \xrightarrow{\sigma} \Theta}{\vdash_{\Sigma} \Gamma = \Gamma \xrightarrow{\sigma=\sigma} \Theta = \Theta} \quad (3.34)$$

$$\frac{\vdash_{\Sigma} \Gamma = \Gamma' \xrightarrow{\sigma=\sigma'} \Theta = \Theta'}{\vdash_{\Sigma} \Gamma' = \Gamma \xrightarrow{\sigma'=\sigma} \Theta' = \Theta} \quad (3.35)$$

$$\frac{\vdash_{\Sigma} \Gamma = \Gamma' \xrightarrow{\sigma=\sigma'} \Theta = \Theta' \quad \vdash_{\Sigma} \Gamma' = \Gamma'' \xrightarrow{\sigma'=\sigma''} \Theta' = \Theta''}{\vdash_{\Sigma} \Gamma = \Gamma'' \xrightarrow{\sigma=\sigma''} \Theta = \Theta''} \quad (3.36)$$

$$\frac{}{\vdash_{\Sigma} \Gamma \xrightarrow{\sigma=\sigma'} \Theta} \quad (3.37)$$

where $M_i = N_i : D_i[M_j/y_j]_{j=1}^{i-1} \in \Sigma$ and $\mathbf{N}_=$ proves $\Gamma \vdash_{\Sigma} M_i = N_i : D_i[M_j/y_j]_{j=1}^{i-1}$.

$$\frac{\vdash_{\Sigma} \Gamma = \Gamma' \xrightarrow{\sigma} \Theta = \Theta' \quad \vdash_{\Sigma} \Gamma \xrightarrow{\sigma=\sigma'} \Theta}{\vdash_{\Sigma} \Gamma = \Gamma' \xrightarrow{\sigma=\sigma'} \Theta = \Theta'} \quad (3.38)$$

$$\frac{\vdash_{\Sigma} \Gamma \xrightarrow{\langle \text{id}, M=N \rangle} \Gamma, x : A = B \quad \vdash_{\Sigma} \Gamma_{x_m} \xrightarrow{\langle 1_{\Gamma_{x_m}} \rangle} x_1 : A_1 = A'_1, \dots, x_{m-1} : A_{m-1} = A'_{m-1}}{\vdash_{\Sigma} \Gamma' \xrightarrow{\langle 1_{\Gamma'}, M=N \rangle} \Gamma', x : A = B} \quad (3.39)$$

where $\mathbf{N}_=$ proves $\vdash_{\Sigma} \Gamma'$ context, for all m .

$$\frac{\vdash_{\Sigma} (\Gamma, x : A) = (\Gamma', x : B) \xrightarrow{\langle 1_{\Gamma, x}, M=N \rangle} (\Gamma, x : A, y : C) = (\Gamma', x : B, y : D)}{\vdash_{\Sigma} \Gamma = \Gamma' \xrightarrow{\langle 1_{\Gamma}, \lambda x : A. M = \lambda x : A. N \rangle} \Gamma, \Pi x : A. C = \Gamma', \Pi x : B. D} \quad (3.40)$$

$$\frac{\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M_i = N_i, \dots, M_n \rangle} \Theta \quad (\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M'_i = N'_i \rangle} \Gamma, D_i[M_j/y_j]_{j=1}^{i-1})}{\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M'_i = N'_i, \dots, M_n \rangle} \Theta} \quad (3.41)$$

where $M_i = N_i : \Pi x : B. C \in \Sigma \cup \Gamma$, $\mathbf{N}_=$ proves $\Gamma \vdash_{\Sigma} P = Q : B$, $M_i P =_{\beta\eta} M'_i$ and

$N_i P = N'_i$, for some $1 \leq i \leq n$.

$$\frac{\vdash_{\Sigma} \Gamma \xrightarrow{\sigma} \Theta \quad \vdash_{\Sigma} \Gamma = \Gamma' \xrightarrow{\langle 1_{\Gamma} \rangle} \Theta = \Theta'}{\vdash_{\Sigma} \Gamma' \xrightarrow{\sigma} \Theta'} \quad (3.42)$$

■

Having defined the systems \mathbf{C} and $\mathbf{C}_=$, we examine the relationship between them and \mathbf{N} and $\mathbf{N}_=$. The relationship is similar to the relationship between \Vdash and \models .

Lemma 3.45 (Soundness of \mathbf{C} for \mathbf{N})

Let Σ be a valid $\lambda\Pi$ -signature and let $\Gamma \equiv x_1 : A_1, \dots, x_m : A_m$ and $\Theta \equiv y_1 : D_1, \dots, y_n : D_n$ be valid $\lambda\Pi$ -contexts. If \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M_n \rangle} \Theta$ then \mathbf{N} proves $\Gamma \vdash_{\Sigma} M_i : D_i[M_j/y_j]_{j=1}^{i-1}$ for $1 \leq i \leq n$.

Proof The proof is by induction on \mathbf{C} proofs and the length of terms. We begin with the axiom rule. \mathbf{N} proves $\Gamma \vdash_{\Sigma} M_i : D_i[M_j/y_j]_{j=1}^{i-1}$ is a side condition of the rule and this is all we need to prove, so we are done.

We assume that application was the last rule used. We first fix i . Since $M_i : \Pi x : B . C \in \Gamma \cup \Sigma$, we conclude that \mathbf{N} proves $\Gamma \vdash_{\Sigma} M_i : \Pi x : B . C$. We also observe that \mathbf{N} proves $\Gamma \vdash_{\Sigma} M_i P : C[P/x]$ by an instance of the application rule in \mathbf{N} with $\Gamma \vdash_{\Sigma} M_i : \Pi x : B . C$ and $\Gamma \vdash_{\Sigma} P : B$ as premisses. Since $M_i P =_{\beta\eta} M'_i$ and \mathbf{N} proves $\Gamma \vdash_{\Sigma} M'_i : C[P/x]$, we can apply the induction hypothesis to obtain $\Gamma \vdash_{\Sigma} M'_i : D_i[M_j/y_j]_{j=1}^{i-1}$ and conclude that $C[P/x] =_{\beta\eta} D_i[M_j/y_j]_{j=1}^{i-1}$. We replace M_i by M'_i . We have that for $1 \leq k \leq i-1$, $\Gamma \vdash_{\Sigma} M_k : D_k[M_j/y_j]_{j=1}^{k-1}$ and $\Gamma \vdash_{\Sigma} M'_i : D_i[M_j/y_j]_{j=1}^{i-1}$. We then have that $\Gamma \vdash_{\Sigma} M_k : D_k[M_j/y_j]_{j=1}^{i-1}[M'_i/y_i][M_p/y_p]_{p=i+1}^k$, for $i+1 \leq k \leq n$. If the last rule applied was the introduction rule then by the induction hypothesis we have $\Gamma, x : A \vdash_{\Sigma} M : D$. We apply (2.15) to obtain $\Gamma \vdash_{\Sigma} \lambda x : A . M : \Pi x : A . D$ as required.

Finally, we consider the case where the last rule used was equality. We have $\beta\eta$ -equality on all levels. We apply the induction hypothesis and the various $\beta\eta$ -rules to obtain $\Gamma' \vdash_{\Sigma} M'_i : D'_i[M_j/y_j]_{j=1}^{i-1}$, for $1 \leq i \leq n$. ■

Corollary 3.46 (Soundness of $\mathbf{C}_=$ for $\mathbf{N}_=$)

Let Σ be a $\lambda\Pi_=$ -signature and let $\Gamma \equiv x_1 : A_1, \dots, x_m : A_m$ and $\Theta = y_1 : D_1, \dots, y_n : D_n$ be valid $\lambda\Pi_=$ - contexts. If $\mathbf{C}_= \vdash_{\Sigma} \Gamma = \Gamma' \xrightarrow{\sigma=\sigma'} \Theta = \Theta'$, then \mathbf{N} proves $\vdash_{\Sigma} \Gamma = \Gamma' \xrightarrow{\sigma=\sigma'} \Theta = \Theta'$. $\mathbf{N}_=$ proves $\Gamma = \Gamma' \vdash_{\Sigma} M_i = M'_i : (D_i = D'_i)[M_j/y_j]_{j=1}^{i-1}$, for $1 \leq i \leq n$.

Proof By induction on the structure of $\mathbf{C}_=$ proofs. The equivalence relation rules are straightforward. We only show transitivity. The induction hypothesis yields $\Gamma = \Gamma' \vdash_{\Sigma} M_i = M'_i : ((D_i = D'_i)[M_j/y_j]_{j=1}^{i-1})$ and $\Gamma' = \Gamma'' \vdash_{\Sigma} M'_i = M''_i :$

$((D'_i = D''_i)[M_j/y_j]_{j=1}^{i-1})$. By the transitivity rules in $\mathbf{N}_=$, we can conclude that $\Gamma = \Gamma'$ and $\Gamma' = \Gamma''$ imply $\Gamma = \Gamma''$. $(D_i = D'_i)[M_j/y_j]_{j=1}^{i-1}$ and $(D'_i = D''_i)[M_j/y_j]_{j=1}^{i-1}$ imply $(D_i = D''_i)[M_j/y_j]_{j=1}^{i-1}$. $M_i = M'_i$ and $M'_i = M''_i$ imply $M_i = M''_i$. Putting all these together gives us the required result.

If the last applied rule was (3.37), we apply (3.13) which gives $\Gamma \vdash_{\Sigma} D_i : ((M_i = N_i)[M_j/y_j]_{j=1}^{i-1})$ and we are done.

Suppose that the last rule applied was (3.38). We apply the induction hypothesis to obtain $\Gamma \vdash_{\Sigma} M_i = M'_i : ((D_i = D'_i)[M_j/y_j]_{j=1}^{i-1})$. The assertions are well-typed so we can apply (3.14) to obtain $\Gamma = \Gamma' \vdash_{\Sigma} M_i = M'_i : ((D_i = D'_i)[M_j/y_j]_{j=1}^{i-1})$ as required.

If the last rule applied was (3.39), we can apply the induction hypothesis to the first premise to obtain $\Gamma \vdash_{\Sigma} M : A = B$. This term is well-typed, so we have $\Gamma \vdash_{\Sigma} A = B : \mathbf{Type}$. We apply the induction hypothesis to the second premise to obtain $\Gamma_{x_m} \vdash_{\Sigma} M_i : A_i = A'_i$. This term is also well-typed, so that $\Gamma_{x_m} \vdash_{\Sigma} \Gamma(x) = \Gamma'(x) : \mathbf{Type}$. We now apply (3.16) to obtain $\Gamma' \vdash_{\Sigma} M : A = B$.

We assume the last rule applied was (3.40). The induction hypothesis yields $(\Gamma, x : A) = (\Gamma', x : B) \vdash_{\Sigma} M = N : (C = D)$. We apply (3.25) to obtain $\Gamma = \Gamma' \vdash_{\Sigma} (\lambda x : A. M) = (\lambda x : B. N) : (\Pi x : A. C) = (\Pi x : B. D)$ as required.

Suppose the last rule was (3.41), we apply the induction hypothesis to obtain $\Gamma \vdash_{\Sigma} M_i = N_i : D_i[M_j/y_j]_{j=1}^{i-1}$ and $\Gamma \vdash_{\Sigma} M'_i = N'_i : D_i[M_j/y_j]_{j=1}^{i-1}$. Since $M_i = N_i : \Pi x : B. C$ and $\Gamma \vdash_{\Sigma} P : B$, we can apply (3.26). $\mathbf{N}_=$ proves $\Gamma \vdash_{\Sigma} M_i P = N_i P : B[P/x]$. Since $M_i =_{\beta\eta} M'_i$ and $N_i P =_{\beta\eta} N'_i$ we conclude that $\mathbf{N}_=$ proves $\Gamma \vdash_{\Sigma} M'_i = N'_i : B[P/x]$, hence $B[P/x] =_{\beta\eta} D_i[M_j/y_j]_{j=1}^{i-1}$. Using the rules for $\beta\eta$ equality we get $\Gamma \vdash_{\Sigma} M'_i = N'_i : D_i[M_j/y_j]_{j=1}^{i-1}$, $\Gamma \vdash_{\Sigma} M_k : D_k[M_j/y_j]_{j=1}^{k-1}$, $1 \leq k \leq i-1$ and $\Gamma \vdash_{\Sigma} M_p : D_p[M_j/y_j]_{j=1}^{i-1}[M'_i/y_i][M_j/y_j]_{j=i+1}^p$, for $i+1 \leq p \leq n$.

Finally, suppose that the last rule applied was (3.42). We apply the induction hypothesis to obtain $\Gamma \vdash_{\Sigma} D_i[M_j/y_j]_{j=1}^{i-1}$ and $\Theta \vdash_{\Sigma} x : D_i = D'_i$. Since this is well-typed we have $\Theta \vdash_{\Sigma} D_i = D'_i : \mathbf{Type}$. We now apply (3.29) to conclude \mathbf{N} proves $\Gamma \vdash_{\Sigma} M_i : D'_i[M_j/y_j]_{j=1}^{i-1}$, as required. \blacksquare

It is important to understand that Lemma 3.45 and Corollary 3.46 *do not* imply that \mathbf{N} or $\mathbf{N}_=$ is an adequate approximation of \models . It is possible for arrows of the form $\llbracket \Delta \xrightarrow{\sigma} \Theta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ to exist in the fibre over $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ in the absence of putatively corresponding arrows of the form $\llbracket \langle \rangle_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{\llbracket M_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W} \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. The satisfaction relation \models is defined on the raw syntax independently of \models and gives the semantic counterpart to the assertion $\vdash_{\Sigma} - \xrightarrow{-} -$ of \mathbf{C} . Just as \models gives the semantic counterpart of the assertion $- \vdash_{\Sigma} - : -$ of \mathbf{N} .

Lemma 3.47 (Completeness of \mathbf{C} for \mathbf{N})

Let Σ be a valid $\lambda\Pi$ -signature, let $\Gamma \equiv x_1 : A_1, \dots, x_m : A_m$ be a valid $\lambda\Pi$ -context and M_i , for $1 \leq i \leq m$, be valid $\lambda\Pi$ -objects. If \mathbf{N} proves $\Gamma \vdash_{\Sigma} M_i : D_i[M_j/y_j]_{j=1}^{i-1}$, for each $1 \leq i \leq n$, then \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M_i \rangle} \Gamma, x : D_i[M_j/y_j]_{j=1}^{i-1}$ for $1 \leq i \leq n$.

Proof The proof is by induction on i and proofs in \mathbf{N} . We first fix i and proceed by induction on the structure of \mathbf{N} proofs. Suppose we have the axiom sequent $\Gamma \vdash_{\Sigma} c : \Pi x_1 : A_1 \dots \Pi x_m : A_m . A[M_j/y_j]_{j=1}^{i-1}$, we apply (3.30). There is a problem since we do not know whether or not M_j for $1 \leq j \leq i-1$ is in $\Sigma \cup \Gamma$. We also know, however, that \mathbf{N} proves $\Gamma \vdash_{\Sigma} c : C[M_j/y_j]_{j=1}^{i-1}[x_k/x_k]_{k=1}^{i-1}$ and $x_k \in \Sigma \cup \Gamma$ for $1 \leq k \leq i-1$, which means we can apply (3.30) to get \mathbf{C} proves $\Gamma \vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, c \rangle} \Gamma, x : C[M_j/y_j]_{j=1}^{i-1}$.

Similarly, if we have the axiom sequent $\Gamma \vdash_{\Sigma} y : A[M_j/y_j]_{j=1}^{i-1}$, then \mathbf{N} proves $\Gamma \vdash_{\Sigma} x : A[M_j/y_j]_{j=1}^{i-1}$ and hence, by (3.30), we have \mathbf{C} proves $\Gamma \vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, y \rangle} \Gamma, x : A[M_j/y_j]_{j=1}^{i-1}$.

Suppose that the last rule used was (2.15), which in the current setting is

$$\frac{\Gamma, x : A \vdash_{\Sigma} M : B[M_j/y_j]_{j=1}^{i-1}}{\Gamma \vdash_{\Sigma} \lambda x : A . M : \Pi x : A . B[M_j/y_j]_{j=1}^{i-1}}$$

where no y_j occurs free in A and each y_j is distinct from x , we apply the induction hypothesis to obtain \mathbf{C} proves $\Gamma \vdash_{\Sigma} \Gamma, x : A \xrightarrow{\langle 1_{\Gamma}, x : A, M \rangle} \Gamma, x : A, y : B[M_j/y_j]_{j=1}^{i-1}$. We apply (3.32) to obtain \mathbf{C} proves $\Gamma \vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, \lambda c A . M \rangle} \Gamma, y : \Pi x : A . B[M_j/y_j]_{j=1}^{i-1}$ as required.

Suppose the last rule applied was (2.16), which in our situation is

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

where no y_i occurs free in A or N and each y_i is distinct from x . Applying the induction hypothesis gives us that \mathbf{C} proves $\Gamma \vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M \rangle} \Gamma, y : \Pi x : A . B[M_j/y_j]_{j=1}^{i-1}$. The side condition \mathbf{N} proves $\Gamma \vdash_{\Sigma} N : A$ is satisfied so we can apply (3.32) and (3.33) to get \mathbf{C} proves $\Gamma \vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, MN \rangle} \Gamma, y : B[N/x][M_j/y_j]_{j=1}^{i-1}$ as required.

Finally, the last rule applied was (2.17), which in our situation is

$$\frac{\Gamma \vdash M : A[M_j/y_j]_{j=1}^{i-1} \quad \Gamma \vdash_{\Sigma} A'[M_j/y_j]_{j=1}^{i-1} : \mathbf{Type} \quad A =_{\beta\eta} A'}{\Gamma \vdash_{\Sigma} M : A'[M_j/y_j]_{j=1}^{i-1}}$$

We apply the induction hypothesis to get \mathbf{C} proves $\Gamma \vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M \rangle} \Gamma, y : A[M_j/y_j]_{j=1}^{i-1}$. Since $A =_{\beta\eta} A'$, we apply (3.33) to get \mathbf{C} proves $\Gamma \vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M \rangle} \Gamma, y : A'[M_j/y_j]_{j=1}^{i-1}$. \blacksquare

Corollary 3.48 (Completeness of $\mathbf{C}_=$ for $\mathbf{N}_=$)

Let Σ be a valid $\lambda\Pi_=$ -signature, let $\Gamma \equiv x_1 : A_1, \dots, x_m : A_m$ be a valid $\lambda\Pi_=$ -context and M_i and N_i , for $1 \leq i \leq n$, be valid $\lambda\Pi_=$ -objects. If $\mathbf{N}_=$ proves

$\Gamma \vdash_{\Sigma} (M_i = N_i) : (A_i = B_i)[M_j/y_j]_{j=1}^{i-1}$ for $1 \leq i \leq n$, then $\mathbf{C}_=$ proves \vdash_{Σ}
 $\Gamma \xrightarrow{\langle 1_{\Gamma}, M_i = N_i \rangle} \Gamma, y : (A_i = B_i)[M_j/y_j]_{j=1}^{i-1}$ for $1 \leq i \leq n$.

Proof The proof is by induction over i and $\mathbf{N}_=$ proofs. We begin by fixing i and then proceed by induction on the structure of $\mathbf{N}_=$ proofs. The equivalence relations are straightforward. We show transitivity as an example. We apply the induction hypothesis to get $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M_i = M'_i \rangle} \Gamma, y : A_i = A'_i[M_j/y_j]_{j=1}^{i-1}$ and $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M'_i = M''_i \rangle} \Gamma, y : A'_i = A''_i[M_j/y_j]_{j=1}^{i-1}$. (3.36) gives $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M_i = M''_i \rangle} \Gamma, y : A_i = A''_i[M_j/y_j]_{j=1}^{i-1}$ as required.

Suppose that we have the axiom sequent $\mathbf{N}_=$ proves $\Gamma \vdash_{\Sigma} M = N : A[M_j/y_j]_{j=1}^{i-1}$. We apply (3.37) and thus $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M = N \rangle} \Gamma, y : A[M_j/y_j]_{j=1}^{i-1}$.

Suppose that we have rules (3.16) and (3.17). We apply the induction hypothesis to obtain that $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma, x : A = B \xrightarrow{\langle 1_{\Gamma}, x : A = B, M = N \rangle} \Gamma, x : A = B, y : C = D[M_j/y_j]_{j=1}^{i-1}$. We apply rule (3.39) to get $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, \lambda x : A. M = \lambda x : B. N \rangle} \Gamma, y : (\Pi x : A. C = \Pi x : B. D)[M_j/y_j]_{j=1}^{i-1}$.

Suppose the last rule applied was (3.26). We apply the induction hypothesis to get $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M = N \rangle} \Gamma, y : \Pi x : A. B[M_j/y_j]_{j=1}^{i-1}$. We apply (3.41) and (3.42) to see that $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M^P = N^Q \rangle} \Gamma, y : B[N/x][M_j/y_j]_{j=1}^{i-1}$.

Finally, assume that the last rule applied was (3.29). We apply the induction hypothesis to obtain $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M \rangle} \Gamma, x : A$ and $\mathbf{C}_=$ proves $\vdash_{\Sigma} x : A \xrightarrow{\langle 1_{x : A} \rangle} x : A = B$. We apply (3.42) to get $\mathbf{C}_=$ proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle 1_{\Gamma}, M \rangle} \Gamma, x : B$. ■

Having explored the relationship between $\mathbf{N}(=)$ and $\mathbf{C}(=)$, we now turn to the relationship between \Vdash and $\vdash_{\Sigma} - \xrightarrow{-} -$. We restrict our attention to realizations of the form $\Delta \xrightarrow{\langle 1_{\Gamma}, N \rangle} \Delta, z : A$ for two reasons. They correspond to the provable assertions of the form $\Gamma \vdash_{\Sigma} N : A$ in \mathbf{N} and in § 12 and § 13, when we consider $\lambda\Pi$ as the language of a logical framework, these will be the realizations we are interested in when we carry out proof search. With this restriction, it is clear that soundness for \mathbf{C} with respect to Kripke Σ - $\lambda\Pi$ -models follows from Lemma 3.45 and Theorem 3.35. Soundness for $\mathbf{C}_=$ follows from Corollary 3.46 and Corollary 3.36. We move straight to completeness.

Definition 3.49 (\Vdash -validity for $\lambda\Pi$)

Let Σ be a $\lambda\Pi$ -signature. Let Γ be a valid $\lambda\Pi$ -context and let $\Delta \xrightarrow{\sigma} \Theta$ be a realization. We say that $\Delta \xrightarrow{\sigma} \Theta$ is *valid* with respect to Γ , *i.e.*,

$$\Gamma \Vdash_{\Sigma} (\Delta \xrightarrow{\sigma} \Theta)$$

if and only if, for all Kripke Σ - $\lambda\Pi$ -models, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$,

and worlds W such that $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket \Delta \xrightarrow{\sigma} \Theta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined and $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma]$. ■

We extend this definition to $\lambda\Pi_{=}$ in the natural way.

Definition 3.50 (\Vdash -validity for $\lambda\Pi_{=}$)

Let Σ be a $\lambda\Pi_{=}$ -signature, Γ be a valid $\lambda\Pi_{=}$ -context and $\Delta \xrightarrow{\sigma} \Theta$ and $\Delta' \xrightarrow{\sigma'} \Theta'$ be valid $\lambda\Pi_{=}$ -realizations. We say that $\Delta = \Delta' \xrightarrow{\sigma=\sigma'} \Theta = \Theta'$ is *valid* with respect to Γ , *i.e.*

$$\Gamma \Vdash_{\Sigma} (\Delta = \Delta' \xrightarrow{\sigma=\sigma'} \Theta = \Theta')$$

if and only if, for all Kripke Σ - $\lambda\Pi_{=}$ -models, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, \mathcal{D}^{op}, \bar{\mathcal{V}}]$, and worlds W such that $\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, $\llbracket \Delta \xrightarrow{\sigma} \Theta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $\llbracket \Delta' \xrightarrow{\sigma'} \Theta' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ are defined $\llbracket \Delta \xrightarrow{\sigma} \Theta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W = \llbracket \Delta' \xrightarrow{\sigma'} \Theta' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ we have that $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\sigma} \Theta)[\Gamma]$ and $W \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta' \xrightarrow{\sigma'} \Theta')[\Gamma]$. ■

Lemma 3.51 (Model Existence)

Let Σ be a $\lambda\Pi$ -signature, let Γ be a valid $\lambda\Pi$ -context and let $\Delta \xrightarrow{\langle 1_{\Gamma}, N \rangle} \Delta, z : A$ be a realization. If $\not\vdash_{\Sigma} \Delta \xrightarrow{\langle 1_{\Delta}, N \rangle} \Delta, z : A[M_j/y_j]_{j=1}^n$ (in \mathbf{C}) then there exists a Kripke Σ - $\lambda\Pi$ -model $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, and a world W_0 such that $W_0 \not\Vdash_{\Sigma} (\Delta \xrightarrow{\langle 1_{\Gamma}, N \rangle} \Delta, z : A)[\Delta]$.

Proof In the term-model, $\mathcal{K}_{\mathcal{T}_{\Sigma}}$, in § 3.2.1, $\Gamma \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, N \rangle} \Gamma \bowtie \Delta, z : A)[\Gamma \bowtie \Delta]$ if and only if \mathbf{N} proves $\Gamma \bowtie \Delta \vdash_{\Sigma} N : A[M_j/y_j]_{j=1}^n$. By Lemma 3.47, $\Gamma \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, N \rangle} \Gamma \bowtie \Delta, z : A)[\Gamma \bowtie \Delta]$ if and only if \mathbf{C} proves $\vdash_{\Sigma} \Gamma \bowtie \Delta \rightarrow \Gamma \bowtie \Delta, z : B[M_j/y_j]_{j=1}^n$. Thus we take $W_0 = \langle \rangle$. ■

Corollary 3.52 (Equational Model Existence)

Let Σ be a $\lambda\Pi_{=}$ -model, let Γ be a valid $\lambda\Pi_{=}$ -context and let $\Delta \xrightarrow{\langle 1_{\Delta}, M=N \rangle} \Delta, z : A = B$ be a realization. If $\not\vdash_{\Sigma} \Delta \xrightarrow{\langle 1_{\Delta}, M=N \rangle} \Delta, z : A = B[M_j/y_j]_{j=1}^n$ then there exists an equational Kripke Σ - $\lambda\Pi_{=}$ -model $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$ and a world W_0 such that $W_0 \not\Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\langle 1_{\Delta}, M=N \rangle} \Delta, z : A = B)[\Delta]$.

Proof In the equational Kripke Σ - $\lambda\Pi_{=}$ -model $\mathcal{K}_{\mathcal{T}_{\Sigma}}$, $\Gamma \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, M=N \rangle} \Gamma \bowtie \Delta, z : A = B)[\Gamma]$ if and only if $\mathbf{N}_{=}$ proves $\Gamma \bowtie \Delta \vdash_{\Sigma} M = N : A = B[M_j/y_j]_{j=1}^n$. By Corollary 3.48, we have that $\Gamma \Vdash_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Gamma \bowtie \Delta \xrightarrow{\langle 1_{\Gamma \bowtie \Delta}, M=N \rangle} \Gamma \bowtie \Delta, z : A = B)[\Gamma]$ if and only if \mathbf{C} proves $\vdash_{\Sigma} \Gamma \bowtie \Delta \rightarrow \Gamma \bowtie \Delta, z : A = B[M_j/y_j]_{j=1}^n$. Thus we take $W_0 = \langle \rangle$. ■

Theorem 3.53 (Completeness)

Let Σ be a $\lambda\Pi$ -signature, let Γ be a valid $\lambda\Pi$ -context and let $\Delta \xrightarrow{\langle 1_\Delta, N \rangle} \Delta, z : A[M_j/y_j]_{j=1}^n$ be a realization. \mathbf{C} proves $\Delta \xrightarrow{\langle 1_\Delta, N \rangle} \Delta, z : A[M_j/y_j]_{j=1}^n$ if and only if $\Delta \Vdash_\Sigma (\Delta \xrightarrow{\langle 1_\Delta, N \rangle} \Delta, z : A[M_j/y_j]_{j=1}^n)$.

Proof

Only If By Lemma 3.45 and Theorem 3.35.

If Suppose $\not\vdash_\Sigma \Delta \xrightarrow{\langle 1_\Delta, N \rangle} \Gamma \Delta, z : A[M_j/y_j]_{j=1}^n$, then Lemma 3.51 yields a contradiction. \blacksquare

Corollary 3.54 (Equational Completeness)

Let Σ be a $\lambda\Pi_-$ -signature, let Γ be a valid $\lambda\Pi_-$ -context and let $\Delta \xrightarrow{\langle 1_\Delta, N \rangle} \Delta, z : A[M_j/y_j]_{j=1}^n$ be a realization. $\mathbf{C}_=$ proves $\vdash_\Sigma \Delta \xrightarrow{\langle 1_\Delta, M=N \rangle} \Delta, z : A = B[M_j/y_j]_{j=1}^n$ if and only if $\Delta \Vdash_\Sigma \Delta \xrightarrow{\langle 1_\Delta, M=N \rangle} \Delta, z : A = B[M_j/y_j]_{j=1}^n$.

Proof

Only If By Corollary 3.46 and Corollary 3.36.

If Suppose $\not\vdash_\Sigma \Delta \xrightarrow{\langle 1_\Delta, M=N \rangle} \Delta, z : A = B[M_j/y_j]_{j=1}^n$, then Lemma 3.52 yields a contradiction. \blacksquare

Chapter 4

Applicative Structures and Kripke Logical Relations

Having defined Kripke Σ - $\lambda\Pi$ -models and proven the relevant soundness and completeness results, we now define a set-theoretic class of Kripke Σ - $\lambda\Pi$ -models. The starting point for these models is the work of (Dybjer 1995) and (Hofmann 1996). The motivation for this chapter is § 6 of (Mitchell & Moggi 1991). As we have mentioned previously, this part of the thesis adapts their work to the $\lambda\Pi$ -calculus. We show that their work on logical relations can be generalized to our setting. We begin by defining a class of Kripke Σ - $\lambda\Pi$ -models with families. These are then restricted to a set-theoretic class of Kripke Σ - $\lambda\Pi$ -models which we call Σ - $\lambda\Pi$ -applicative structures. We are then able to define Kripke Σ - $\lambda\Pi$ -logical relations on these, which are the generalization of Mitchell and Moggi’s Kripke logical relations to our setting. Finally, we examine classical Σ - $\lambda\Pi$ -applicative structures and show how they can arise from Σ - $\lambda\Pi$ -applicative structures.

The presentation of the material in this chapter is new. The general theory which has been adapted to this setting is, however, well-known. The majority of the proofs are similar to those found in (Mitchell & Moggi 1991).

4.1 Applicative Structures

4.1.1 Kripke Σ - $\lambda\Pi$ -models with Families

(Dybjer 1995) and (Hofmann 1996) have defined set-theoretic models of dependent types using the category of families of sets, \mathcal{Fam} . The idea is simple, yet provides a different class of models from those described in § 3, they are distinguished here by their interpretation of the syntax. Starting with a category \mathcal{C} of contexts and realizations, one constructs a functor $\mathcal{F}:\mathcal{C}^{op} \rightarrow \mathcal{Fam}$ as follows:

- Take a category \mathcal{C} of semantic contexts and semantic context morphisms;

- Take the object-part to be

$$\mathcal{F}(\Gamma) = (\text{Ty}(\Gamma), \text{Tm}(\Gamma)) = (\text{Ty}(\Gamma), \text{Tm}(\Gamma, A)_{A \in \text{Ty}(\Gamma)})$$

where $\text{Ty}(\Gamma)$ is a set of semantic types A such that $\Gamma \vdash_{\Sigma} A : \text{Type}$ and $\text{Tm}(\Gamma, A)$ is a set of semantic terms;

- The arrow-part of \mathcal{F} is given by semantic substitutions via inverse images.

The familial fibre given in Definition 2.6 is an instance of this definition. The following example can be found in (Dybjer 1995) and (Hofmann 1996). We take \mathcal{C} to be the category of all small sets and functions, *Set*. Contexts, Γ , are interpreted as sets whose cardinality is equal to the number of variables in Γ . Realizations are interpreted as functions between sets. We then define the set $\text{Ty}(\Gamma)$ to be the set of Γ -indexed small sets, $(\sigma_{\gamma})_{\gamma \in \Gamma}$ and each element of $\text{Tm}(\Gamma, \sigma)$ is the assignment of an element $M(\gamma)$ of σ_{γ} for each $\gamma \in \Gamma$.

It is possible to define a whole class of Kripke $\lambda\Pi$ -prestructures with families. Our attention turns, however, to Kripke $\lambda\Pi$ -structures with families built on a Kripke $\lambda\Pi$ -structure. We extend the definition of families of sets, Definition 2.5, to the following category.

Definition 4.1

Let \mathcal{V} be a category. We define the *category*, $\mathcal{Fam}(\mathcal{V})$, as follows:

Objects: Families, $\{V_i\}_{i \in I}$, of objects of \mathcal{V} , which can be described as an ordered pair (I, V) , in which V is indexed over I ;

Arrows: An arrow $(f, \{f_i\}_{i \in I}) : (I, V) \rightarrow (J, V')$ is given by a function $f : I \rightarrow J$ such that for each $i \in I$, $f_i : V_i \rightarrow V'_{f(i)}$.

Given arrows $(f, \{f_i\}_{i \in I}) : (I, V) \rightarrow (J, V')$ and $(g, \{g_j\}_{j \in J}) : (J, V') \rightarrow (K, V'')$ their composition is a function $g \circ f$ and a family of functions $\{g_{f(i)} \circ f_i\}_{i \in I}$ determined by function composition for each $i \in I$. ■

Having generalized families of sets to the above category, it seems natural to also generalize the index to allow indexing by objects. For our purposes, it is sufficient to keep the index as a set. To define Kripke Σ - $\lambda\Pi$ -models with families, we begin with a Kripke $\lambda\Pi$ -prestructure

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

as in Definition 3.1. For Kripke $\lambda\Pi$ -structures with families, we must have not only the construction $\bar{-}$, but also $\mathcal{Fam}(-)$:

$$\mathcal{F}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathcal{Fam}(\bar{\mathcal{V}})]]$$

and we recall that for each \mathcal{V} , there is a choice of $\bar{\mathcal{V}}$ such that $\mathcal{V} \cong \bar{\mathcal{V}}$. An interpretation with families, $\llbracket - \rrbracket_{\mathcal{F}_{\mathcal{J}}}^-$, is following the usual pattern, a partial function from the raw syntax of the $\lambda\Pi$ -calculus to the prescription below.

- Contexts Γ are mapped to objects $\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W$ of \mathcal{D} . Realizations $\sigma = \langle M_1, \dots, M_n \rangle$ are mapped to arrows $\llbracket \sigma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W$ of \mathcal{D} .

For example, take each \mathcal{D} to be *Set*. We then interpret Γ as a set whose cardinality is equal to the number of variables in Γ and a realization σ as a function between sets.

- It follows that for each world W , $\mathcal{F}(W)(\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W)$ is an object of $\mathcal{F}(\bar{\mathcal{V}})$, *i.e.*, a family (I, \bar{V}) , where each \bar{V}_i has as objects arrows $\Delta \xrightarrow{f} \Theta$ over Γ in \mathcal{D} and as arrows, arrows $\Delta \rightarrow \Delta'$ in \mathcal{D}

Types A , in context Γ , are mapped to elements of the indexing sets I in the pairs (I, \bar{V}) . Corresponding to each $i (= \llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) \in I$ is a category \bar{V}_i , chosen as in Definition 3.2 as a choice of \mathcal{D} -arrows over $\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W$.

For example, take I to be the set of all $\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W$ -indexed small sets, $(\rho_{\gamma})_{\gamma \in \llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W}$. Each $\bar{V}_{\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$ has as objects the functions $\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W \rightarrow \llbracket \Gamma, x : A \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W$ and arrows the identity function $\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W \rightarrow \llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W$.

- Objects M in context Γ are mapped to objects of the category $\bar{V}_{\llbracket A_{\Gamma} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$, for some A , and so is a \mathcal{D} -arrow $\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W \xrightarrow{f} \llbracket \Gamma, x : A \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W$ such that $f; p_{\llbracket \Gamma, x : A \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W} = 1_{\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W}$. For this purpose, it is sufficient that each \bar{V}_i , for $i \in I$, be discrete.

For example, assign M to the function $\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W \xrightarrow{f} \llbracket \Gamma, x : A \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W$ in $\bar{V}_{\llbracket A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W}$. It is clear that we $f; p_{\llbracket \Gamma, x : A \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W} = 1_{\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W}$.

We are now in a position to define a Kripke Σ - $\lambda\Pi$ -model with families. While we could restrict our attention to the $\lambda\Pi$ -calculus, we instead go straight to the $\lambda\Pi_{=}$ -calculus. For the remainder of this chapter we shall be concerned primarily with equational theories. We thus sketch the formal definition of an equational Kripke Σ - $\lambda\Pi_{=}$ -model with families, eliding repetitive details, as follows:

Definition 4.2 (Equational Kripke Σ - $\lambda\Pi_{=}$ -models with families)

Let Σ be a $\lambda\Pi_{=}$ -signature. An *equational Kripke Σ - $\lambda\Pi_{=}$ -model with families* is an ordered pair, $\langle \mathcal{F}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{F}_{\mathcal{J}}}^- \rangle$, where $\mathcal{F}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathcal{Fam}(\bar{\mathcal{V}})]]$, is a Kripke Σ - $\lambda\Pi$ -structure with families. $\llbracket - \rrbracket_{\mathcal{F}_{\mathcal{J}}}^-$ is an interpretation with families, defined simultaneously by induction on the (raw) syntax of the $\lambda\Pi_{=}$ -calculus according to the prescription above and following the cases of Definitions 3.17 and 3.23. ■

It should be clear that we obtain soundness and completeness of the system $\mathbf{N}_=$ for equational Kripke Σ - $\lambda\Pi$ -models with families. If we were to permit in the third point above, the interpretations of realizations of Γ , as in Definition 3.26. Then we would be able to obtain soundness and completeness of $\mathbf{C}_=$ for equational Kripke Σ - $\lambda\Pi$ -models with families.

The presence of families allows a type and its inhabitants to be interpreted as a single object (I, \bar{V}) . It is this feature which makes Kripke Σ - $\lambda\Pi$ -models with families the appropriate basis for an account of logical relations (*q.v.* § 4.2) for dependent types. To this end, we now define set-theoretic equational Kripke Σ - $\lambda\Pi$ -applicative structures.

4.1.2 Equational Kripke Σ - $\lambda\Pi$ -Applicative Structures

We modify the Kripke Σ - $\lambda\Pi$ -model with families to a set-theoretic structure. We produce a structure similar to Cartmell's category \mathcal{Fam} .

Definition 4.3 (Equational Kripke Σ - $\lambda\Pi$ -applicative structures)

Let Σ be a $\lambda\Pi$ -signature. An *equational Kripke Σ - $\lambda\Pi$ -applicative structure* is an equational Kripke Σ - $\lambda\Pi$ -model with families $\mathcal{F}_{\mathcal{J}}$ in which

- \mathcal{W} is assigned to any poset, regarded as a category;
- \mathcal{D} is assigned to \mathcal{Set} ;
- $\mathcal{Fam}(\bar{\mathcal{V}})$ is assigned to \mathcal{Fam} .

The interpretation $\llbracket - \rrbracket_{\mathcal{F}_{\mathcal{J}}}^-$ is the one sketched above. ■

It should now be clear that equational Kripke Σ - $\lambda\Pi$ -applicative structures can be written in the form

$$\mathcal{F}_{\mathcal{J}}(W)(\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W) = (\text{Ty}(\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W), \text{Trm}(\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W, \llbracket A_{\Gamma} \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W)_{\llbracket A_{\Gamma} \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W \in \text{Ty}(\llbracket \Gamma \rrbracket_{\mathcal{F}_{\mathcal{J}}}^W)})$$

We drop the \mathcal{J} from $\mathcal{F}_{\mathcal{J}}$ when no confusion can arise.

4.2 Kripke Logical Relations

In the classical model theory of the λ -calculus, logical relations ((Plotkin n.d.), (Plotkin 1980), (Statman 1985) and (Friedman 1975)) are families of relations indexed by types which indicate a condition implying closure under application and λ -abstraction.

Our notion of Kripke Σ - $\lambda\Pi$ -logical relation is the generalization of Mitchell & Moggi's (1991) Kripke logical relations to dependent types, *i.e.*, our logical relations indicate closure under application and Π -abstraction. Mitchell & Moggi's (1991) work is in turn a generalization of Plotkin's (1980) I -relations.

The worlds in Kripke Σ - $\lambda\Pi_{=}$ -logical relations are intended to be type assignments; *cf.* (Mitchell & Moggi 1991). In § 4.2.3 we use the worlds to provide a counter-model to the following: let $f, g \in \llbracket (\Pi x : A . B)_{\Gamma} \rrbracket_{=}^W$. For all $W \leq W'$ and all $a \in \llbracket A_{\Gamma} \rrbracket_{=}^{W'}$

$$(\mathcal{N}^{W,W'} f)a = (\mathcal{N}^{W,W'} g)a \text{ implies } f = g.$$

Definition 4.4 (Equational Kripke Σ - $\lambda\Pi_{=}$ -logical relations)

Let Σ be a $\lambda\Pi_{=}$ -signature and let $\mathcal{F}_1, \mathcal{F}_2$ be equational Kripke Σ - $\lambda\Pi_{=}$ -applicative structures over the same poset of worlds, \mathcal{W} . An *equational Kripke Σ - $\lambda\Pi_{=}$ -logical relation over \mathcal{F}_1 and \mathcal{F}_2* is a triple of families of relations. Indexed respectively by worlds; worlds and contexts; and, worlds, contexts and types.

$$\mathcal{R} = (\mathcal{R}^{Con}, \mathcal{R}^{Ty}, \mathcal{R}^{Ob})$$

where

$$\text{(Con)} \quad \mathcal{R}_W^{Con} \subseteq (\text{Set} \upharpoonright \llbracket - \rrbracket_{\mathcal{F}_2}^W) \times (\text{Set} \upharpoonright \llbracket - \rrbracket_{\mathcal{F}_2}^W),$$

$$\text{(Ty)} \quad \mathcal{R}_{W,\Gamma}^{Ty} \subseteq \text{Ty}_{\llbracket \Gamma \rrbracket_{\mathcal{F}_1}^W} \times \text{Ty}_{\llbracket \Gamma \rrbracket_{\mathcal{F}_2}^W}, \text{ and}$$

$$\text{(Tm)} \quad \mathcal{R}_{W,\Gamma,A}^{Ob} \subseteq \text{Tm}_{\llbracket \Gamma \rrbracket_{\mathcal{F}_1}^W}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{F}_1}^W) \times \text{Tm}_{\llbracket \Gamma \rrbracket_{\mathcal{F}_2}^W}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{F}_2}^W),$$

subject to the following conditions:

$$\text{(Mon)} \quad \bullet \text{ If } \mathcal{R}_W^{Con}(\llbracket \Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket \Gamma \rrbracket_{\mathcal{F}_2}^W), \text{ then, for all } W \leq W',$$

$$\mathcal{R}_{W'}^{Con}(\kappa^{W,W'}(\llbracket \Gamma \rrbracket_{\mathcal{F}_1}^W), \kappa^{W,W'}(\llbracket \Gamma \rrbracket_{\mathcal{F}_2}^W));$$

$$\bullet \text{ If } \mathcal{R}_{W,\Gamma}^{Ty}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{F}_1}^W, \llbracket A_{\Gamma} \rrbracket_{\mathcal{F}_2}^W), \text{ then, for all } W \leq W',$$

$$\mathcal{R}_{W',\Gamma}^{Ty}(\mathcal{N}^{W,W'}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{F}_1}^W), \mathcal{N}^{W,W'}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{F}_2}^W));$$

$$\bullet \text{ If } \mathcal{R}_{W,\Gamma,A}^{Ob}(\llbracket M_{\Gamma} \rrbracket_{\mathcal{F}_1}^W, \llbracket M_{\Gamma} \rrbracket_{\mathcal{F}_2}^W), \text{ then, for all } W \leq W',$$

$$\mathcal{R}_{W,\Gamma,A}^{Ob}(\mathcal{N}^{W,W'}(\llbracket M_{\Gamma} \rrbracket_{\mathcal{F}_1}^W), \mathcal{N}^{W,W'}(\llbracket M_{\Gamma} \rrbracket_{\mathcal{F}_2}^W));$$

(Kconst) For each $c : \Pi x_1 : A_1 . \dots . \Pi x_m : A_m . \text{Type} \in \Sigma$ and for each M_1, \dots, M_m such that, for all $W \leq W'$, and each $1 \leq i \leq m$,

$$\mathcal{R}_{W,\Gamma,A_i[M_j/x_j]_{j=1}^{i-1}}^{Ob}(\llbracket M_i \rrbracket_{\mathcal{F}_1}^W, \llbracket M_i \rrbracket_{\mathcal{F}_2}^W),$$

we have,

$$\mathcal{R}_{W',\Gamma}^{Ty}(op_c^1(\mathcal{N}^{W,W'}(\llbracket M_1 \rrbracket_{\mathcal{F}_1}^W), \dots, \mathcal{N}^{W,W'}(\llbracket M_m \rrbracket_{\mathcal{F}_1}^W)), op_c^2(\mathcal{N}^{W,W'}(\llbracket M_1 \rrbracket_{\mathcal{F}_2}^W), \dots, \dots,$$

$$\mathcal{N}^{W,W'}(\llbracket M_m \rrbracket_{\mathcal{F}_2}^W)$$

where op_c^1 and op_c^2 are the operations corresponding to c in \mathcal{F}_1 and \mathcal{F}_2 respectively;

(Tconst) For each $c: \Pi x_1: A_1 \dots \Pi x_m: A_m \cdot A \in \Sigma$ and for each M_1, \dots, M_m such that, for all $W \leq W'$, and each $1 \leq i \leq m$,

$$\mathcal{R}_{W', \Gamma, A[M_j/x_j]_{j=1}^{i-1}}^{Ob}(\mathcal{N}^{W,W'}(\llbracket M_i \rrbracket_{\mathcal{F}_1}^W), \mathcal{N}^{W,W'}(\llbracket M_i \rrbracket_{\mathcal{F}_2}^W))$$

we have

$$\mathcal{R}_{W, \Gamma, A[M_j/x_j]_{j=1}^m}^{Ob}(op_c^1(\llbracket M_1 \rrbracket_{\mathcal{F}_1}^W, \dots, \llbracket M_m \rrbracket_{\mathcal{F}_1}^W), op_c^2(\llbracket M_1 \rrbracket_{\mathcal{F}_2}^W, \dots, \llbracket M_m \rrbracket_{\mathcal{F}_2}^W))$$

where op_c^1 and op_c^2 are the operations corresponding to c in \mathcal{F}_1 and \mathcal{F}_2 , respectively;

(Connex) $\mathcal{R}_{W, \Gamma}^{Con}(\llbracket \Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket \Gamma \rrbracket_{\mathcal{F}_2}^W)$ and $\mathcal{R}_{W, \Gamma}^{Ty}(\llbracket A_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket A_\Gamma \rrbracket_{\mathcal{F}_2}^W)$, if and only if,

$$\mathcal{R}^{Con}(\llbracket \Gamma, x: A \rrbracket_{\mathcal{F}_1}^W, \llbracket \Gamma, x: A \rrbracket_{\mathcal{F}_2}^W)$$

(Compre) $\mathcal{R}_{W, \Gamma, \Pi x: A \cdot B}^{Ob}(\llbracket M_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket M_\Gamma \rrbracket_{\mathcal{F}_2}^W)$ if and only if, for all $W \leq W'$,

$$\mathcal{R}_{W, \Gamma, A}^{Ob}(\llbracket N_\Gamma \rrbracket_{\mathcal{F}_1}^{W'}, \llbracket N_\Gamma \rrbracket_{\mathcal{F}_2}^{W'})$$

implies

$$\mathcal{R}_{W', \Gamma, B[N/x]}^{Ob}(\mathcal{N}^{W,W'}(\llbracket M_\Gamma \rrbracket_{\mathcal{F}_1}^W) \llbracket N_\Gamma \rrbracket_{\mathcal{F}_2}^{W'}, \mathcal{N}^{W,W'}(\llbracket M_\Gamma \rrbracket_{\mathcal{F}_1}^W) \llbracket N_\Gamma \rrbracket_{\mathcal{F}_2}^{W'}). \quad \blacksquare$$

Having defined equational Kripke Σ - $\lambda\Pi$ -logical relations, we now prove the fundamental lemma.

Lemma 4.5 (Fundamental Lemma)

Let Σ be a $\lambda\Pi$ -signature, let \mathcal{F}_1 and \mathcal{F}_2 be equational Kripke Σ - $\lambda\Pi$ -applicative structures and let \mathcal{R} be an equational Kripke Σ - $\lambda\Pi$ -logical relation on them. If $\mathbf{N}_=$ proves $\Gamma \vdash_\Sigma M: A$ and, at each world W , $\mathcal{R}^{Con}(\llbracket \Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket \Gamma \rrbracket_{\mathcal{F}_2}^W)$, then, at each world W ,

1. $\mathcal{R}_{W, \Gamma}^{Ty}(\llbracket A_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket A_\Gamma \rrbracket_{\mathcal{F}_2}^W)$, and
2. $\mathcal{R}_{W, \Gamma, A}^{Ob}(\llbracket M_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket M_\Gamma \rrbracket_{\mathcal{F}_2}^W)$.

Proof By induction on the structure of proofs in $\mathbf{N}_=$, as usual, both parts must be proven simultaneously because of dependencies.

We begin with the case where $\Gamma \vdash_\Sigma c: \Pi x_1: A_1 \dots \Pi x_m: A_m \cdot A$ is an axiom sequent. *(Tconst)* satisfies the condition of the second part of *(Compre)* m -times.

Thus we obtain that

$$\mathcal{R}_{W,\Gamma}^{\mathcal{T}y}(\llbracket \Pi x_1 : A_1 \dots \Pi x_m : A_m \cdot A \rrbracket_{\mathcal{F}_1}^W, \llbracket \Pi x_1 : A_1 \dots \Pi x_m : A_m \cdot A \rrbracket_{\mathcal{F}_2}^W)$$

and

$$\mathcal{R}_{W,\Gamma,\Pi x_1 : A_1 \dots \Pi x_m : A_m \cdot A}^{\mathcal{O}b}(\llbracket c_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket c_\Gamma \rrbracket_{\mathcal{F}_2}^W).$$

Suppose $(\Gamma \equiv) \Delta, x : A, \Delta' \vdash_\Sigma x : A$ is an axiom sequent. We repeatedly apply (*Connex*) to obtain

$$\mathcal{R}_{W,\Delta}^{\mathcal{T}y}(\llbracket A_\Delta \rrbracket_{\mathcal{F}_1}^W, \llbracket A_\Delta \rrbracket_{\mathcal{F}_2}^W)$$

which also holds at Γ . We can thus conclude

$$\mathcal{R}_{W,\Gamma,A}^{\mathcal{O}b}(\llbracket x_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket x_\Gamma \rrbracket_{\mathcal{F}_2}^W)$$

and

$$\mathcal{R}_{W,\Gamma}^{\mathcal{T}y}(\llbracket A_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket A_\Gamma \rrbracket_{\mathcal{F}_2}^W).$$

Suppose the last rule used was ΠI , *i.e.*,

$$\frac{\Gamma, x : A \vdash_\Sigma M : B}{\Gamma \vdash_\Sigma \lambda x : A. M : \Pi x : A. B}$$

To show $\mathcal{R}_{W,\Gamma,\Pi x : A. B}^{\mathcal{O}b}(\llbracket (\lambda x : A. M)_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket (\lambda x : A. M)_\Gamma \rrbracket_{\mathcal{F}_2}^W)$ we must show

$$\mathcal{R}_{W,\Gamma,A}^{\mathcal{O}b}(\llbracket N_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket N_\Gamma \rrbracket_{\mathcal{F}_2}^W)$$

implies

$$\mathcal{R}_{W,\Gamma,B[N/x]}^{\mathcal{O}b}(\mathcal{N}^{W,W'}(\llbracket (\lambda x : A. M)_\Gamma \rrbracket_{\mathcal{F}_1}^W) \llbracket N_\Gamma \rrbracket_{\mathcal{F}_1}^{W'}, \mathcal{N}^{W,W'}(\llbracket (\lambda x : A. M)_\Gamma \rrbracket_{\mathcal{F}_2}^W) \llbracket N_\Gamma \rrbracket_{\mathcal{F}_2}^{W'})$$

So we must assume $\mathcal{R}_{W,\Gamma,A}^{\mathcal{O}b}(\llbracket N_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket N_\Gamma \rrbracket_{\mathcal{F}_2}^W)$. We apply the induction hypothesis and apply terms to obtain $\mathcal{R}_{W',\Gamma,B[N/x]}^{\mathcal{O}b}(\mathcal{N}^{W,W'}(\llbracket (\lambda x : A. M)_\Gamma \rrbracket_{\mathcal{F}_1}^W), \mathcal{N}^{W,W'}(\llbracket (\lambda x : A. M)_\Gamma \rrbracket_{\mathcal{F}_2}^W))$ and $\mathcal{R}_{W,\Gamma}^{\mathcal{T}y}(\llbracket \Pi x : A. B \rrbracket_{\mathcal{F}_1}^W, \llbracket \Pi x : A. B \rrbracket_{\mathcal{F}_2}^W)$ as required.

If the last rule applied was ΠE , *i.e.*,

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

then applying the induction hypothesis yields $\mathcal{R}_{W,\Gamma,\Pi x : A. B}^{\mathcal{O}b}(\llbracket M_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket M_\Gamma \rrbracket_{\mathcal{F}_2}^W)$ and $\mathcal{R}_{W,\Gamma,A}^{\mathcal{O}b}(\llbracket N_\Gamma \rrbracket_{\mathcal{F}_1}^W, \llbracket N_\Gamma \rrbracket_{\mathcal{F}_2}^W)$. We apply (*Connex*) and (*Mon*) to obtain

$$\mathcal{R}_{W',\Gamma,B[N/x]}^{\mathcal{O}b}(\mathcal{N}^{W,W'}(\llbracket M_\Gamma \rrbracket_{\mathcal{F}_1}^W) \llbracket N_\Gamma \rrbracket_{\mathcal{F}_1}^{W'}, \mathcal{N}^{W,W'}(\llbracket M_\Gamma \rrbracket_{\mathcal{F}_2}^W) \llbracket N_\Gamma \rrbracket_{\mathcal{F}_2}^{W'})$$

and we are done.

Suppose the last rule applied was

$$\frac{\Gamma \vdash_{\Sigma} M : A \quad \Gamma \vdash_{\Sigma} A' : \text{Type } A =_{\beta\eta} A'}{\Gamma \vdash_{\Sigma} M : A'}$$

We recall from soundness that since we are restricting to the $M : A : \text{Type}$ -fragment of the $\lambda\Pi$ -calculus, the type A is a $\beta\eta$ -redex just in case where an object M on which A depends is a $\beta\eta$ -redex. We can then use (*Connex*) to obtain the result. ■

4.2.1 Partial Logical Equivalence Relations

Mitchell & Moggi (1991) show that partial equivalence relations can be used to explain the interplay between the classical meta-theory of the simply typed λ -calculus and Kripke λ -models. Here, we show that a similar technique works for models of the $\lambda\Pi$ -calculus. In the next section we will define a classical Σ - $\lambda\Pi_{=}$ -applicative structure (essentially a Kripke Σ - $\lambda\Pi_{=}$ -applicative structure at a single world) and the Kripke Σ - $\lambda\Pi_{=}$ -logical partial equivalence relations allow us to quotient the classical Σ - $\lambda\Pi_{=}$ -applicative structure to obtain an equational Kripke Σ - $\lambda\Pi_{=}$ -applicative structure.

A Kripke Σ - $\lambda\Pi_{=}$ -logical partial equivalence relation is, as its names suggests, a Kripke Σ - $\lambda\Pi_{=}$ -logical relation which is symmetric and transitive. The following lemma shows that Kripke Σ - $\lambda\Pi_{=}$ -logical partial equivalence relations can be constructed by a partial equivalence relation on any level of the $\lambda\Pi$ -calculus.

Lemma 4.6 (Partial Equivalence)

Let Σ be a $\lambda\Pi_{=}$ -signature. Let \mathcal{F} be an equational Kripke Σ - $\lambda\Pi_{=}$ -applicative structure and let $\mathcal{R} = (\mathcal{R}^{Con}, \mathcal{R}^{Ty}, \mathcal{R}^{Ob})$ be an equational Kripke Σ - $\lambda\Pi_{=}$ -logical relation on it. Then the following are equivalent:

1. \mathcal{R}^{Con} is a partial equivalence;
2. \mathcal{R}^{Ty} is a partial equivalence;
3. \mathcal{R}^{Ob} is a partial equivalence.

Proof (Sketch) The proof is by induction on the structure of contexts, types and objects. The proof involves showing that the property of a relation on any level defines the same property for relations on other levels. ■

4.2.2 Kripke Logical Relations on Classical Applicative Structures

We take a classical Σ - $\lambda\Pi_{=}$ -applicative structure to be a pair $\mathcal{U} = \langle \mathcal{U}, \llbracket - \rrbracket_{\mathcal{U}} \rangle$, in which $\mathcal{U} : [\mathcal{D}^{op}, \mathcal{F}am]$ carries the structure carried by an equational Kripke Σ - $\lambda\Pi_{=}$ -applicative structure at a fixed world and $\llbracket X \rrbracket_{\mathcal{U}}$ is defined for all X derivable

in $\mathbf{N}_=$. Alternatively, we could have taken a Kripke Σ - $\lambda\Pi$ -applicative structure in which \mathcal{W} is taken to be the category with one object and one arrow.

Given a classical applicative structure $\mathcal{U} = \langle \mathcal{U}, \llbracket - \rrbracket_{\mathcal{U}} \rangle$, we define a Kripke applicative structure $[\mathcal{W}, \mathcal{U}]$ as follows (we sketch just a few key points):

- Take \mathcal{W} to be a poset;
- At each world W we have the functor \mathcal{U} ;
- The natural transformation induced by arrows between worlds is taken to be the identity.

Let \mathcal{U} and \mathcal{V} be classical Σ - $\lambda\Pi$ -applicative structures. We say that \mathcal{R} is a Kripke logical relation on the classical Σ - $\lambda\Pi$ -applicative structures \mathcal{U} and \mathcal{V} if \mathcal{R} is a Kripke logical relation on $[\mathcal{W}, \mathcal{U}]$ and $[\mathcal{W}, \mathcal{V}]$.

In the case of a partial equivalence relation on \mathcal{U} , *i.e.*, $\mathcal{V} = \mathcal{U}$, we can form the quotient \mathcal{U}/\mathcal{R} .

Lemma 4.7 (quotients yield Kripke applicative structures)

In the notion of the discussion above, \mathcal{U}/\mathcal{R} is an equational Kripke Σ - $\lambda\Pi$ -applicative structure.

Proof (Sketch) We need to show that $\llbracket X \rrbracket_{\mathcal{U}/\mathcal{R}}^W = \llbracket \llbracket X \rrbracket_{\mathcal{U}}^W \rrbracket_{\mathcal{R}}$ for all derivable X in $\mathbf{N}_=$. This is done by an induction over the structure of $\mathbf{N}_=$. ■

Lemma 4.8

Let \mathcal{U}/\mathcal{R} be as above and let $f, g \in \llbracket (\Pi x : A . B)_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^W$. Then, for all $W \leq W'$ and for all $a \in \llbracket A_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^{W'}$, \mathcal{U}/\mathcal{R} satisfies the following extensionality condition:

$$(\mathcal{N}^{W, W'} f)a = (\mathcal{N}^{W, W'} g)a \quad \text{implies} \quad f = g \quad (4.1)$$

Proof (Sketch) $\llbracket A_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^{W'}$ is a set and so it is either empty or non-empty. If it is empty, then $f = g$. If $\llbracket A_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^{W'}$ is non-empty, then $fa = ga$ for all appropriate worlds. Thus f and g are equal on all values of f and g , hence $f = g$. ■

Theorem 4.9 (Classical Equational Completeness)

Let Σ be a $\lambda\Pi$ -signature. Let $\Gamma \vdash_{\Sigma} M = N : A$ and $\Gamma \vdash_{\Sigma} A = B : \mathbf{Type}$ be provable assertions in $\mathbf{N}_=$. There exists a classical Σ - $\lambda\Pi$ -applicative structure \mathcal{U} and a Kripke partial logical equivalence relation on \mathcal{U} such that $\Gamma \Vdash_{\Sigma}^{\mathcal{U}/\mathcal{R}} M = N : A$ and $\Gamma \Vdash_{\Sigma}^{\mathcal{U}/\mathcal{R}} A = B : \mathbf{Type}$.

Proof (Sketch) We recall the Kripke $\lambda\Pi$ -structure $\mathcal{K}_{\mathcal{T}_{\Sigma}^=}$, we defined in § 3.7. The idea is to carry out the above construction on this Kripke $\lambda\Pi$ -structure. We first have to modify our model slightly so that it is a Kripke Σ - $\lambda\Pi$ -applicative structure. We define the fibre $\mathcal{K}_{\mathcal{T}_{\Sigma}^=}(\Delta)(\Delta \bowtie \Gamma)$ to be the ordered pair (I, \bar{V}) , where I is the set of types A , such that $\Delta \bowtie \Gamma \vdash_{\Sigma} A : \mathbf{Type}$ and \bar{V} is the set

of sections $\Delta \bowtie \Gamma \xrightarrow{\langle id, M \rangle} \Delta \bowtie \Gamma, x : A$. We then take the classical applicative structure \mathcal{U} to be the fibre at the world $\Delta = \langle \rangle$. We take the Kripke logical relation \mathcal{R} to be the one which identifies equations $\Gamma \vdash_{\Sigma} M = N : A$ and $\Gamma \vdash_{\Sigma} A = B : \text{Type}$. It remains to show that this is a partial equivalence relation. This is straightforward. \blacksquare

The above proof illustrates the difficulties involved in proving completeness in the absence of worlds.

4.2.3 A Counter-model to Semantic Implication

As an application of Kripke quotients, we provide a counter-model to the extensionality condition shown in Lemma 4.8. We construct a classical Σ - $\lambda\Pi_{=}$ -applicative structure together with a Kripke logical relation. We let $f, g \in \llbracket (\Pi x : A. B)_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^W$. We then show that for all $W \leq W'$ and for all $a \in \llbracket A_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^{W'}$

$$(\mathcal{N}^{W, W'} f)a = (\mathcal{N}^{W, W'} g)a \quad \text{does not imply} \quad f = g \quad (4.2)$$

We take the classical Σ - $\lambda\Pi_{=}$ -applicative structure we described in the proof of Theorem 4.9. We take \mathcal{W} to be the poset containing two worlds $\{0, 1\}$. The proof of Lemma 4.8 required that $\llbracket A_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^W$ be either globally empty or non-empty. We take $\mathcal{R}_{\Gamma}^{\mathcal{T}y}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{U}}^0) = \emptyset$ and $\mathcal{R}_{\Gamma}^{\mathcal{T}y}(\llbracket A_{\Gamma} \rrbracket_{\mathcal{U}}^1) = id$. Clearly, when we take the quotient, $\llbracket A_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^W$ will be empty at world 0 and non-empty at world 1, thus breaking the argument of Lemma 4.8. We now need to show that $(\mathcal{N}^{W, W'} f)a = (\mathcal{N}^{W, W'} g)a$. This holds trivially at world 0. At world 1, we must make $fa = ga$, an easy way to do this is to chose the relation $\mathcal{R}_{\Gamma}^{\mathcal{T}y}$ so that $\llbracket A_{\Gamma} \rrbracket_{\mathcal{U}/\mathcal{R}}^1$ has only one element. We have constructed our counter-model and it is straightforward to verify that equation 4.2 holds.

Chapter 5

The Internal Logic and its Models

In this chapter, we depart from the work of (Mitchell & Moggi 1991) and provide a preview of the material to come in § 8. The internal logic of a type theory is the logic obtained using the propositions-as-types ((Curry 1934), (Curry & Feys 1958), (Howard 1980)) correspondence. For the $\lambda\Pi$ -calculus, this is the $\{\forall, \supset\}$ -fragment of many sorted minimal first-order logic. We begin this chapter with a syntactic account of this logic. We present the logic as a natural deduction system with proof-objects. Following this presentation, we provide an algebraic account of this logic. We provide models in the same spirit as Kripke $\lambda\Pi$ -models. These models, which we call Kripke \mathcal{L}_T -models, are closely related to the hyperdoctrines of Lawvere (1970) and Seely (1983). We begin with Kripke \mathcal{L}_T -prestructures, with the move to Kripke \mathcal{L}_T -structures corresponding to the move from studying proofs to logical consequence. The final section of this chapter provides the pattern and motivation for § 8. We show that the propositions-as-types correspondence induces an (indexed) isomorphism between Kripke \mathcal{L}_T -models and Kripke Σ - $\lambda\Pi$ -models.

The material in this chapter up to § 5.3 is well-known. The Kripke \mathcal{L}_T -models are new, although the principles behind them are well-understood. The work on the propositions-as-types isomorphism is original research.

5.1 The Propositions-as-types Correspondence

Within this section, technical details have been toned down to allow us to present, clearly, the essential points needed to motivate and explain the propositions-as-types isomorphism. The material in this section is already well-explained within the literature. The reader is advised to consult the appropriate references for a fuller account; for example, (Barendregt 1991) and (Howard 1980).

The $\lambda\Pi$ -calculus is in propositions-as-types (Curry-Howard-de Bruijn-

Barendregt) correspondence with the $\{\forall, \supset\}$ -fragment of many sorted minimal first-order logic. The correspondence can be formulated in the style of Barendregt (1991).

Let \mathcal{L}_T denote the $\{\forall, \supset\}$ -fragment of many sorted minimal first-order logic with theory T consisting of a finite set of constants of basic sorts, function symbols of finite arity (0-ary functions are constants) and atomic predicate letters of finite arity. The alphabet A of \mathcal{L}_T to consist the following sets:

- A countable set of basic sorts including ι and o ;
- The set $\{\forall, \supset, c_1, \dots, c_n, f_1, \dots, f_m, P_1, \dots, P_p\}$.

We assume a countably infinite stock of variables of each basic sort which has variables. We define \mathbf{Term}_S , the collection of terms with sort S , as follows:

- If x has sort S , then $x \in \mathbf{Term}_S$;
- If c has sort S , then $c \in \mathbf{Term}_S$;
- If f has sort $S_1, \dots, S_n \rightarrow S$ and if for $1 \leq i \leq n$, $t_i \in \mathbf{Term}_{S_i}$, then $f(t_1, \dots, t_n) \in \mathbf{Term}_S$.

We define \mathbf{Form} , the collection of formulæ of \mathcal{L}_T , as follows:

- If $P \subseteq T_1 \times \dots \times T_m$ is a predicate letter and $t_i \in \mathbf{Term}_{T_i}$ for $1 \leq i \leq m$. then $P(t_1, \dots, t_m) \in \mathbf{Form}$;
- If $\phi \in \mathbf{Form}$ and $\psi \in \mathbf{Form}$, then $\phi \supset \psi \in \mathbf{Form}$;
- If $\phi \in \mathbf{Form}$, then $\forall x:S. \phi \in \mathbf{Form}$, where x is free in ϕ .

A natural deduction system for \mathcal{L}_T is defined by the following rules.

$$\begin{array}{c}
 \frac{[\phi] \quad \vdots \quad \psi}{\phi \supset \psi} \supset E \\
 \\
 \frac{\vdots \quad \phi}{\forall x:S. \phi} \forall I \quad \begin{array}{l} (x \text{ not free in any} \\ \text{assumption upon} \\ \text{which } \phi \text{ depends}) \end{array} \\
 \\
 \frac{\vdots \quad \phi \quad \phi \supset \psi}{\psi} \supset E \\
 \\
 \frac{\vdots \quad \forall x:S. \phi}{\phi[t/x]} \forall E \quad (t \in \mathbf{Term}_S)
 \end{array}$$

We extend our view of \mathcal{L}_T to include proof-objects. To this end, we use Barendregt's (1991) notation for natural deductions, $\delta: (\Delta \vdash_T \phi)$, as follows:

$$\begin{aligned}
& \phi \in \Delta \Rightarrow \alpha_\phi: (\Delta \vdash_T \phi) \\
& \delta_1: (\Delta \vdash_T \phi) \quad \delta_2: (\Delta \vdash_T \phi \supset \psi) \Rightarrow \delta_1 \delta_2: (\Delta \vdash_T \psi) \\
& \delta: (\Delta, \phi \vdash_T \psi) \Rightarrow I_\phi \delta: (\Delta \vdash_T \phi \supset \psi) \\
& \delta: (\Delta \vdash_T \forall x: S. \phi), \quad t \in \mathbf{Term}_S \Rightarrow \delta t: (\Delta \vdash_T \phi[t/x]) \\
& \delta: (\Delta \vdash_T \phi), \quad x: S \notin FV(\Delta) \Rightarrow Gx: S. \delta: (\Delta \vdash_T \forall x: S. \phi)
\end{aligned}$$

The theory T defined over \mathcal{L} plays a rôle similar to that of the signature in the $\lambda\Pi$ -calculus. We are now able to define the (proof-theoretic) propositions-as-types correspondence, following (Barendregt 1991).

Definition 5.1 (Propositions-as-types Translation)

We define a translation $\{\!\{-}\!\}$ from \mathcal{L}_T to the $\Sigma_{\mathcal{L}_T}$ - $\lambda\Pi$ -calculus as a triple $(\{\!\{-}\!\}_T, \{\!\{-}\!\}_F, \{\!\{-}\!\}_P)$, defined below.

We define the translation, $\{\!\{-}\!\}_T$, from \mathbf{Term}_S to objects of the $\lambda\Pi$ -calculus. The sort S is translated to the type S .

- $\{\!\{x}\!\}_T = x: S$
- $\{\!\{c}\!\}_T = c: S$
- $\{\!\{f(t_1, \dots, t_n)\}\!\}_T = f \{\!\{t_1}\!\}_T \dots \{\!\{t_n}\!\}_T$

We define the translation $\{\!\{-}\!\}_F$ from \mathbf{Form} to types of the $\lambda\Pi$ -calculus.

- $\{\!\{\phi}\!\}_F = A_\phi$
- $\{\!\{\phi \supset \psi}\!\}_T = \Pi x: \{\!\{\phi}\!\}_F. \{\!\{\psi}\!\}_F \quad (\{\!\{\phi}\!\}_F \rightarrow \{\!\{\psi}\!\}_F)$ where x does not occur free in $\{\!\{\psi}\!\}_F$
- $\{\!\{\forall x: S. \phi}\!\}_F = \Pi x: S. \{\!\{\phi}\!\}_F$

The translations $\{\!\{-}\!\}_T$ and $\{\!\{-}\!\}_F$ also sends the theory T to the signature Σ_T . Σ_T contains $\{\!\{c}\!\}_T$, $\{\!\{f}\!\}_T$ and $\{\!\{P}\!\}_F$.

We define the translation $\{\!\{-}\!\}_P$ from the proof-objects of \mathcal{L}_T to objects of the $\lambda\Pi$ -calculus.

- $\{\!\{\alpha_\phi}\!\}_P = x: \{\!\{\phi}\!\}_P$
- $\{\!\{I_\phi}\!\}_P = \lambda x: \{\!\{\phi}\!\}_P$
- $\{\!\{G}\!\}_P = \lambda$ ■

While $\delta: (\Delta \vdash_T \phi)$ is a natural deduction proof, from now on we consider it as a realizer.

Proposition 5.2 (Propositions-as-types Correspondence)

The sequent $X \delta : (\Delta \vdash_T \phi)$ is provable if and only if \mathbf{N} proves $\{\{X\}\}, \{\{\Delta\}\} \vdash_{\Sigma_T} \{\{\delta\}\} : \{\{\phi\}\}$, where Σ_T is the set which contains $\{\{c\}\}$, $\{\{f\}\}$ and $\{\{P\}\}$ for all the constants, function symbols and predicate letters in T and X is the set of syntactic variables in δ .

Proof (Sketch) Both directions of this proof require a straightforward induction. The if direction requires an induction over the structure of proof-objects of \mathcal{L}_T , while the only if direction requires an induction over \mathbf{N} . \blacksquare

A few comments are needed about the relationship between the $\lambda\Pi_-$ -calculus and \mathcal{L}_T . We need to extend \mathcal{L}_T so that it includes equality. This is done by adding a predicate $=$ and adding axioms about the properties of this predicate, *e.g.*, it should be an equivalence relation. Thus we define a theory $T_=$. We then identify the predicate $=$ with equality in the $\lambda\Pi_-$ -calculus under the propositions-as-types correspondence. The equality predicate can, however, be treated as just another predicate, rather than a special one which corresponds to equality in the $\lambda\Pi_-$ -calculus. So from now on, we just consider the $\lambda\Pi$ -calculus, since it can adequately represent (*q.v.* § 6) \mathcal{L}_T for any T , including ones with equality.

5.2 The Semantics of the Internal Logic \mathcal{L}_T

The semantics of the logic \mathcal{L}_T can be given in many ways. Perhaps, the most familiar is Kripke's (1965) approach in which many propositions can be interpreted in a structure at a world. Informally, let \mathcal{M} be a Kripke model of \mathcal{L}_T , consisting of a preordered set of worlds, with enough structure to interpret the constants, function symbols and predicate letters of T , and an assignment ρ of the variables of \mathcal{L}_T in a structure at a world. The satisfaction relation $w, \rho \Vdash_T^{\mathcal{M}} \phi$, read as, "the world w forces proposition ϕ in Kripke model \mathcal{M} with respect to assignment ρ ". It is defined by induction on the structure of propositions as follows:

- $w, \rho \Vdash_T^{\mathcal{M}} p(\bar{t})$ if and only if $\llbracket \bar{t} \rrbracket_{\mathcal{M}}^w$ is defined and is in $\llbracket p \rrbracket_{\mathcal{M}}^w$;
- $w, \rho \Vdash_T^{\mathcal{M}} \phi \supset \psi$ if and only if, for all $w \xrightarrow{f} w'$, $(w', \rho[f] \Vdash_T^{\mathcal{M}} \phi$ implies $w', \rho[f] \Vdash_T^{\mathcal{M}} \psi)$;
- $w, \rho \Vdash_T^{\mathcal{M}} \forall x : S . \phi$ if and only if, for all $w \xrightarrow{f} w'$, all $a : \llbracket S \rrbracket_{\mathcal{M}}^w \rightarrow \llbracket S \rrbracket_{\mathcal{M}}^w$ and terms t such that $\llbracket t \rrbracket_{\mathcal{M}}^w = a$, $(w', \rho[f][x := a] \Vdash_T^{\mathcal{M}} \phi[t/x])$.

We are concerned with the extension of forcing to consequences labelled with proof-objects. We consider a version of Kripke semantics for the $\{\forall, \supset\}$ -fragment of many sorted minimal first-order logic with proof-objects. Our formulation will be appropriate for considering a semantic account of the propositions-as-types correspondence. Specifically, we formulate Kripke models of \mathcal{L}_T within the same categorical framework as our Kripke Σ - $\lambda\Pi$ -models.

5.2.1 Kripke Prestructures and Structures for \mathcal{L}_T

We use indexed categories to define a Kripke prestructure for \mathcal{L}_T . Indexed categories, or doctrines, are used to model algebraic theories, *cf.* (Kock & Reyes 1977). The Kripke prestructure for \mathcal{L}_T , we will presently define, is at each world a special case of a hyperdoctrine.

The definition of a hyperdoctrine in Lawvere (1969) provides a model for intuitionistic logic. The relationship between this definition and ours is as follows:

- We only require the existence of a right adjoint to functors induced by projections in the base category. Lawvere requires right and left adjoints to all functors between fibres. We only need right adjoints because we are only interested in the $\{\forall, \supset\}$ -fragment and so do not require that there is enough structure to interpret the existential. Intuitionistic logic coincides with minimal logic for the fragment we are interested in.
- The Beck-Chevalley condition we use is a special case of the condition given by Lawvere. Our Beck-Chevalley condition is sufficient for our prestructure, *cf.* Seely (1983), § 8.

Definition 5.3 (Kripke Prestructures for \mathcal{L}_T)

A *Kripke prestructure for the logic \mathcal{L}_T* is a functor $\mathcal{J}: [\mathcal{W}, [\mathcal{B}^{op}, \mathcal{V}]]$, such that (i) \mathcal{W} is a small category of worlds; (ii) $\mathcal{B}^{op} = \coprod_{W \in \mathcal{W}} \mathcal{B}_W^{op}$, where each \mathcal{B}_W is a small cartesian closed category; and, (iii) \mathcal{V} is a (sub)category (of $\mathcal{C}at$) of values such that

- For all worlds W in \mathcal{W} and objects U in \mathcal{B}_W , $\mathcal{J}(W)(U)$ is cartesian closed;
- For all worlds W in \mathcal{W} and arrows $f: R \rightarrow U$ in \mathcal{B}_W , there is a functor $f^*: \mathcal{J}(W)(U) \rightarrow \mathcal{J}(W)(R)$. This functor preserves on the nose the terminal object $1_{\mathcal{J}(W)(U)}$ in $\mathcal{J}(W)(U)$ and the cartesian closed structure of $\mathcal{J}(W)(U)$;
- For all worlds W in \mathcal{W} and projections $p_{U,V}: U \times V \rightarrow U$, each functor $p_{U,V}^*: \mathcal{J}(W)(U) \rightarrow \mathcal{J}(W)(U \times V)$ has a right adjoint

$$p_{U,V}^* \dashv \forall_{U,V}: \mathcal{J}(W)(U \times V) \rightarrow \mathcal{J}(W)(U)$$

that satisfies the following (strict) Beck-Chevalley conditions: for each $f: R \rightarrow U$ in \mathcal{B}_W , each L in \mathcal{B}_W and each V in $\mathcal{J}(W)(U \times L)$ we have

$$f^*(\forall_{U,L} V) = \forall_{R,L} ((f \times L)^* V)$$

and

$$(f \times L)^*(app_W(L, V)) = app_W(L, (f \times L)^* V)$$

where app_W is the counit to the adjunction. ■

We take base categories at each world and then define fibres over their co-product to follow the structure of a Kripke $\lambda\Pi$ -prestructure. We recall that the reason for this is that we want our Kripke models to be analogous to Kripke models of intuitionistic logic where there is a model of classical logic at each world. We will find when we construct the term model of \mathcal{L}_T that we need to take the same category at each world. This is because we need to define constants and functions at each world.

We now have a Kripke prestructure for \mathcal{L}_T in which we can interpret its formulæ and proofs. We wish, however, to be able to interpret logical consequence as well. We achieve this by moving to a Kripke structure for \mathcal{L}_T . The objects in each fibre of the Kripke structure for \mathcal{L}_T are tuples of arrows in the fibres of the Kripke prestructure for \mathcal{L}_T . The objects in the fibres of the Kripke structure for \mathcal{L}_T are used to interpret proofs, so that the arrows interpret proofs. Then the arrows between these proofs interpret proof transformations, *i.e.*, logical consequence.

Definition 5.4 (Kripke Structures for \mathcal{L}_T)

Let \mathcal{J} be a Kripke prestructure for \mathcal{L}_T , $\mathcal{J}: [\mathcal{W}, [\mathcal{B}^{op}, \mathcal{V}]]$. A *Kripke structure for \mathcal{L}_T* on \mathcal{J} is a functor

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

such that the category $\bar{\mathcal{V}}$ has the following properties:

Objects: Categories $\bar{\mathcal{V}}$ built out of $V = \mathcal{J}(W)(U)$ with:

Objects: Arrows

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

in \mathcal{V} , where $\bar{A} = A_1 \times \dots \times A_m$;

Arrows: Arrows

$$(\bar{A} \xrightarrow{f_{\bar{A},A}} A) \rightarrow (\bar{B} \xrightarrow{f_{\bar{B},B}} B)$$

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

Arrows: Functors $f^*: \mathcal{K}_{\mathcal{J}}(W)(U) \rightarrow \mathcal{K}_{\mathcal{J}}(W)(R)$, where $U \xrightarrow{f} R$ is an arrow in \mathcal{B}_W , defined as follows:

1. The functor $\mathcal{K}_{\mathcal{J}}(W)(f)$ takes an object of $\mathcal{K}_{\mathcal{J}}(W)(U)$, the arrow $f_{\bar{A},A}$, and returns an object in $\mathcal{K}_{\mathcal{J}}(W)(R)$, which is the arrow:

$$\mathcal{K}_{\mathcal{J}}(W)(f)(f_{\bar{C},C}) = \prod_{i=1}^n \mathcal{J}(W)(f)(C_i) \xrightarrow{\mathcal{J}(W)(f)(f_{\bar{C},C})} \mathcal{J}(W)(f)(C).$$

2. The functor $\mathcal{K}_{\mathcal{J}}(W)(f)$ takes an arrow of $\mathcal{K}_{\mathcal{J}}(W)(U)$, $\bar{A} \xrightarrow{\mu} \bar{B}$, and returns the arrow $\nu = \mathcal{J}(W)(f)(U)$, where $C_1 \times \dots \times C_m \xrightarrow{\nu} D_1 \times \dots \times$

D_n is such that $\mathcal{J}(W)(f)(A_i) = C_i$ for $1 \leq i \leq m$ and $\mathcal{J}(W)(f)(B_j) = D_j$ for $1 \leq j \leq n$. ■

5.2.2 Kripke Models of \mathcal{L}_T

We need to ensure that a Kripke model for \mathcal{L}_T has enough structure to interpret all the constants and function symbols in T . This is similar to the requirement that Kripke Σ - $\lambda\Pi$ -models have Σ -operations.

Definition 5.5 (Enough Structure)

Let $\mathcal{K}_{\mathcal{J}}$ be a Kripke structure for \mathcal{L}_T . We say that $\mathcal{K}_{\mathcal{J}}$ has *enough structure to interpret T* , if for all worlds W in \mathcal{W} , the following conditions are satisfied:

1. There are at least as many objects in \mathcal{B}_W as there are sorts in L , the language underlying \mathcal{L} ;
2. For all worlds W and all function symbols $f : S_1, \dots, S_n \rightarrow S$, there exists a morphism $\llbracket S_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \dots \times \llbracket S_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ in \mathcal{B}_W ;
3. For all worlds W and all predicate symbols P with arity S_1, \dots, S_n , there exists an object $P(x_1, \dots, x_n)$, where $x_1 : S_1, \dots, x_n : S_n$, in $\mathcal{J}(W)(X)$ and $X = \llbracket S_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \dots \times \llbracket S_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$. ■

Since constants are 0-ary function symbols, we have a morphism $\llbracket c \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} : 1 \rightarrow \llbracket C \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ in \mathcal{B}_W for each constant $c : C$.

Definition 5.6 (Kripke Models of \mathcal{L}_T)

A *Kripke model of \mathcal{L}_T* consists of a pair $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-,\rho} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{B}^{op}, \bar{\mathcal{V}}]]$ is a Kripke structure for \mathcal{L}_T and the partial function $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-,\rho}$ is an interpretation of \mathcal{L}_T in $\mathcal{J}_{\mathcal{J}}$. The interpretation is defined by induction on the structure of (i) sorts which are interpreted as objects in \mathcal{B} ; (ii) terms, which are interpreted as arrows in \mathcal{B} ; and, (iii) propositions, with variables in the set $X = \{x_1 : S_1, \dots, x_m : S_m\}$, which are interpreted in the fibre over $\llbracket S_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \dots \times \llbracket S_m \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$. If $X = \emptyset$, then $\llbracket \phi(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ is an object of $\mathcal{K}_{\mathcal{J}}(W)(1)$.

The sorts, terms and functions are interpreted as follows:

1. For each sort S , $\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ is an object of \mathcal{B}_W defined by induction on sorts;
 - For each sort S , $\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ is (a choice of) an object in \mathcal{B}_W ;
 - For each function sort, *i.e.*, $S = S_1, \dots, S_n \rightarrow T$,

$$\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = \llbracket T \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} (\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}),$$

the internal hom in \mathcal{B}_W ;

2. For each variable $x : S$, $\llbracket x \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ is an arrow $\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \xrightarrow{\rho(x)} \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ in \mathcal{B}_W , not dependent on W ;
3. For each function $f : S_1, \dots, S_n \rightarrow S$, we interpret it as the arrow given by Definition 5.5. We have that:

- Constants c of sort S are interpreted as an arrow $\llbracket c \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} : 1 \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ in \mathcal{B}_W ;
- Functions $f : S_1, \dots, S_n \rightarrow S$ are interpreted as an arrow

$$\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$$

in \mathcal{B}_W , and given $t_i : S_i$ for $1 \leq i \leq n$, then

$$\llbracket f t_1 \dots t_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = \llbracket f \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \llbracket t_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \dots \llbracket t_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho},$$

using the cartesian category structure of \mathcal{B}_W in the usual way (Lambek & Scott 1986);

- Tuples of terms are interpreted as arrows in \mathcal{B}_W :

$$\langle \llbracket t_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}, \dots, \llbracket t_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \rangle : \llbracket A_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \dots \times \llbracket A_m \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \rightarrow \llbracket B_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \dots \times \llbracket B_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho},$$

where, for each $1 \leq i \leq n$, $x_1 : A_1, \dots, x_m : A_m \vdash_T t_i : B_i$;

- Term-formation by application is interpreted by function space application in \mathcal{B}_W .

The connectives are interpreted as objects in $\mathcal{J}(W)(X)$ by induction over the structure of formulæ, exploiting the cartesian closed structure of the fibres, as follows:

Atomic: For each predicate letter P with arity S_1, \dots, S_n , $\llbracket P(x_1, \dots, x_n) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ is an object of $\mathcal{J}(W)(\llbracket S_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \dots \times \llbracket S_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})$, given by Definition 5.5 together with an arrow $1 \rightarrow \llbracket P(x_1, \dots, x_n) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$;

Implication: if $\llbracket p_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = A_1$ in $\mathcal{K}_{\mathcal{J}}(W)(X)$ and $\llbracket p_2 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = A_2$, then $\llbracket p_1 \supset p_2 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = A_2^{A_1}$ in $\mathcal{J}(W)(X)$;

Universal: if $\llbracket \rho(x) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ in $\mathcal{J}(W)(X, x : A)$, then $\llbracket \forall x : A. p \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = \forall_{X,A} \llbracket p \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho/x}$ in $\mathcal{J}(W)(X)$.

We interpret the proof-objects as arrows in $\mathcal{J}(W)(X)$ by induction over the structure of proof-objects, exploiting the cartesian closed structure of the fibres, as follows:

- $\llbracket \alpha_\phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = p_\phi$, the projection $\prod_{i=1}^n \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \rightarrow \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$;
- $\llbracket \delta_1 \delta_2 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = eval \circ (\llbracket \delta_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \llbracket \delta_2 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})$, where *eval* is the application map;
- $\llbracket I_\phi \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = \lambda \llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$, where λ is the unique morphism guaranteed by exponentiation;
- $\llbracket \delta t \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = t^* \llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$;
- $\llbracket Gx : S . \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = \forall_{X,S} \llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$.

We require, in order for the definition of interpretations by induction on the syntactic structure of \mathcal{L}_T to work, the following syntactic monotonicity condition: if $\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ is defined, then so is $\llbracket X' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$, for every subterm or subformula X' of X , where X ranges over the whole syntax of \mathcal{L}_T sequents. We also require the following accessibility condition: if there is an arrow $W \xrightarrow{\alpha} W'$ in \mathcal{W} , then $\mathcal{J}(W')(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \cong \mathcal{J}(W')(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W',\rho})$ and $\mathcal{J}(W)(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \cong \mathcal{J}(W)(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W',\rho})$. ■

We give the term model as an example of a Kripke model for \mathcal{L}_T .

5.2.3 Term Model

We sketch the construction of a term model $\langle \mathcal{K}_T, \llbracket - \rrbracket_{\mathcal{K}_T}^{-,\rho} \rangle$. For a fixed alphabet A the category $\mathcal{B}(A)$ is defined as follows:

Objects: Contexts of the form $x_1 : S_1, \dots, x_m : S_m$, for $m \geq 0$ ($m = 0$ gives the unique empty context, $\langle \rangle$, the terminal object of $\mathcal{B}(A)$);

Arrows: Tuples of the form

$$x_1 : S_1, \dots, x_m : S_m \xrightarrow{\langle t_1, \dots, t_n \rangle} y_1 : T_1, \dots, y_n : T_n$$

such that, for each $1 \leq i \leq n$, $x_1 : S_1, \dots, x_m : S_m \vdash_T t_i : T_i$. (Terms t_i will be of the form $f_i s_1 \dots s_{m_i}$. In particular, a variable x of sort S arises as an arrow $x : S \xrightarrow{\langle x \rangle} x : S$.)

The posetal category of worlds, a subcategory of $\mathcal{B}(A)$ is defined just as in Definition 3.3.

Objects: The empty context, $\langle \rangle$, is an object of \mathcal{W} . If X is an object of \mathcal{W} and there exists an arrow $X \xrightarrow{t} X, X'$ in $\mathcal{B}(A)$, then X, X' is an object of \mathcal{W} ;

Arrows: There is an arrow $X \rightarrow X'$ if and only if $X \subseteq X'$.

At each world X , we take \mathcal{C}_X to be $\mathcal{B}(A)$. We then define $\mathcal{B}^{op} = \prod_{X \in |\mathcal{W}|} \mathcal{C}_X^{op}$.

We show that \mathcal{C}_X has a product. We claim that given two objects, $x_1 : S_1, \dots, x_m : S_m$ and $y_1 : T_1, \dots, y_n : T_n$, their product is $x_1 : S_1, \dots, x_m : S_m, y_1 : T_1, \dots, y_n : T_n$ with projections $x_1 : S_1, \dots, x_m : S_m, y_1 : T_1, \dots, y_n : T_n \xrightarrow{\langle x_1, \dots, x_m \rangle} x_1 : S_1, \dots, x_m : S_m$ and $x_1 : S_1, \dots, x_m : S_m, y_1 : T_1, \dots, y_n : T_n \xrightarrow{\langle y_1, \dots, y_n \rangle} y_1 : T_1, \dots, y_n : T_n$.

Suppose that $z_1 : U_1, \dots, z_p : U_p$ is another object in \mathcal{C}_X together with two arrows $z_1 : U_1, \dots, z_p : U_p \xrightarrow{\langle f_1, \dots, f_m \rangle} x_1 : S_1, \dots, x_m : S_m$ and $z_1 : U_1, \dots, z_p : U_p \xrightarrow{\langle g_1, \dots, g_n \rangle} y_1 : T_1, \dots, y_n : T_n$. We show that there exists a unique arrow $z_1 : U_1, \dots, z_p : U_p \xrightarrow{\langle h_1, \dots, h_{n+m} \rangle} x_1 : S_1, \dots, x_m : S_m, y_1 : T_1, \dots, y_n : T_n$ such that the diagram

$$\begin{array}{ccc}
 & z_1 : U_1, \dots, z_p : U_p & \\
 \swarrow f & \vdots \exists! h & \searrow g \\
 x_1 : S_1, \dots, x_m : S_m & \xleftarrow{\langle x_1, \dots, x_m \rangle} x_1 : S_1, \dots, y_n : T_n \xrightarrow{\langle y_1, \dots, y_n \rangle} & y_1 : T_1, \dots, y_n : T_n
 \end{array}$$

commutes. We begin by showing the existence of h . The arrow f is such that for $1 \leq i \leq m$, $f_i : S_i$ is a term in \mathcal{L}_T . Similarly, for $1 \leq i \leq n$, $g_i : T_i$ is a term in \mathcal{L}_T . For the diagram to commute, we need $x_i \circ h_i = f_i$, for $1 \leq i \leq m$ but since $x_i \circ h_i = h_i$, we have that $h_i = f_i$ for $1 \leq i \leq m$. Similarly, $h_{j+m} = g_j$ for $1 \leq j \leq n$. Thus we can define h to be $\langle f_1, \dots, f_m, g_1, \dots, g_n \rangle$. Uniqueness follows, since if we were to chose another arrow $k : z_1 : U_1, \dots, z_p : U_p \rightarrow x_1 : S_1, \dots, x_m : S_m, y_1 : T_1, \dots, y_n : T_n$, then the same equalities hold and so $k = h$. Hence \mathcal{B} has a product.

We now show that \mathcal{C}_X has an exponential. We take $X^Y = (x_1 : S_1, \dots, x_m : S_m)^{(y_1 : T_1, \dots, y_n : T_n)}$ to be a candidate for the exponential together with the evaluation arrow $(x_1 : S_1, \dots, x_m : S_m)^{(y_1 : T_1, \dots, y_n : T_n)} \times (y_1 : T_1, \dots, y_n : T_n) \xrightarrow{\langle f_1 x_1 \dots x_n, \dots, f_m x_1 \dots x_n \rangle} (x_1 : S_1, \dots, x_m : S_m)$, where each f_i has sort $T_1, \dots, T_n \rightarrow S_i$. Let $z_1 : U_1, \dots, z_p : U_p$ be an object in \mathcal{B} and let $(z_1 : U_1, \dots, z_p : U_p) \times (y_1 : T_1, \dots, y_n : T_n) \xrightarrow{\langle g_1, \dots, g_m \rangle} x_1 : S_1, \dots, x_m : S_m$ be an arrow in \mathcal{B} . We need to show that there exists a unique arrow $h : z_1 : U_1, \dots, z_p : U_p \rightarrow (x_1 : S_1, \dots, x_m : S_m)^{(y_1 : T_1, \dots, y_n : T_n)}$ such that the diagram

$$\begin{array}{ccc}
z_1:U_1, \dots, z_p:U_p & (z_1:U_1, \dots, z_p:U_p) \times (y_1:T_1, \dots, y_n:T_n) & \\
\downarrow \exists! h & \downarrow h \times \langle y_1, \dots, y_n \rangle & \searrow \mathcal{G} \\
X^Y & X^Y \times (y_1:T_1, \dots, y_n:T_n) & x_1:S_1, \dots, x_m:S_m \\
& & \nearrow eval
\end{array}$$

commutes. We take h to be the map $z_1:U_1, \dots, z_p:U_p \xrightarrow{\langle (y_1, \dots, y_n)g_1, \dots, (y_1, \dots, y_n)g_m \rangle} X^Y$. So that the diagram

$$\begin{array}{ccc}
z_1:U_1, \dots, z_p:U_p, y_1:T_1, \dots, y_n:T_n & & \\
\downarrow \langle (y_1, \dots, y_n)\bar{g} \rangle \times \langle \bar{y} \rangle & \searrow \langle g_1, \dots, g_m \rangle & \\
X^Y \times (y_1:T_1, \dots, y_n:T_n) & \xrightarrow{eval} & x_1:S_1, \dots, x_m:S_m
\end{array}$$

commutes.

It should be clear that the above arrow is unique. Hence \mathcal{C}_X has an exponential and is cartesian closed.

We define a functor $\mathcal{T} : [\mathcal{W}, [\mathcal{B}^{op}, \mathcal{V}]]$ as follows: at each object $X = x_1 : S_1, \dots, x_m : S_m$ of \mathcal{W} and each object $\Delta = y_1 : T_1, \dots, y_n : T_n$ of \mathcal{C}_X , we define a category $\mathcal{T}(X)(\Delta)$ as follows:

$$\mathcal{T}(X)(\Delta) = \begin{cases} \text{Objects: Propositions } \phi \text{ such that } Fv(\phi) \subseteq Dom(X \bowtie \Delta), \\ \text{where } \bowtie \text{ is defined analogously to syntactic merge in } \S 3; \\ \text{Arrows: Proofs } \Phi \text{ such that } \phi \xrightarrow{\Phi} \psi \text{ if and only if} \\ (X \bowtie \Delta) \phi \vdash_T \Phi : \psi. \end{cases}$$

At each object Y of \mathcal{W} and each arrow $X' \xrightarrow{t} X$, we define the functor $t^*(= \mathcal{T}(W)(t)) : \mathcal{T}(Y)(X) \rightarrow \mathcal{T}(Y)(X')$, as usual ((Lawvere 1970) and (Seely 1983)) this is given by substitution.

At each arrow $X \rightarrow X'$ of \mathcal{W} , we must define a natural transformation $\mathcal{T}(X) \Rightarrow \mathcal{T}(X')$. As in the example constructed in § 3.2.1, inclusions will do:

$$\begin{array}{ccc}
 \Delta & & \mathcal{T}(X)(\Delta) \xrightarrow{\mathcal{I}_\Delta} \mathcal{T}(X')(\Delta) \\
 \downarrow t & & \downarrow \mathcal{T}(Y)(t) \quad \downarrow \mathcal{T}(X')(t) \\
 \Delta' & & \mathcal{T}(X)(\Delta') \xrightarrow[\mathcal{I}_{\Delta'}]{} \mathcal{T}(X')(\Delta')
 \end{array}$$

We show that every fibre $\mathcal{T}(X)(\Delta)$ has a product. Let ϕ_1 and ϕ_2 be objects in $\mathcal{T}(X)(\Delta)$, together with arrows $\phi_1, \phi_2 \rightarrow \phi_1$ and $\phi_1, \phi_2 \rightarrow \phi_2$. We show that for all objects C and arrows $C \xrightarrow{f} \phi_1$ and $C \xrightarrow{g} \phi_2$ that there exists a unique arrow $h: C \rightarrow \phi_1, \phi_2$ such that the diagram

$$\begin{array}{ccccc}
 & & C & & \\
 & \swarrow f & \vdots \exists! h & \searrow g & \\
 \phi_1 & \xleftarrow{\Phi} & \phi_1, \phi_2 & \xrightarrow{\Psi} & \phi_2
 \end{array}$$

commutes. Given proofs $C \vdash_T \phi_1$ and $C \vdash_T \phi_2$, it is possible to combine them using meta-theoretic and to obtain a proof $C \vdash_T \phi_1, \phi_2$. We now show uniqueness. Suppose we have another proof $C \vdash_T \phi_1, \phi_2$, then it is either equal to the above proof or involves a detour, which we can resolve since all the inference rules in \mathcal{L}_T satisfy the local reduction property, *cf.* S 7.

We show that every fibre $\mathcal{T}(X)(\Delta)$ has an exponential. We take $\phi \supset \psi$ as the candidate for the exponential together with the evaluation arrow $\phi \supset \psi, \phi \rightarrow \psi$. Let ϕ be an object in $\mathcal{T}(X)(\Delta)$ and $\tau, \phi \xrightarrow{\Psi} \psi$ be an arrow of $\mathcal{T}(X)(\Delta)$. We need to show that there exists a unique arrow δ from τ to $\phi \supset \psi$ such that the diagram

$$\begin{array}{ccc}
 \tau & & \tau, \phi \\
 \vdots \exists! \delta & & \downarrow \delta, \delta' \\
 \phi \supset \psi & & \phi \supset \psi, \phi \\
 & & \downarrow \Psi \quad \swarrow \text{eval} \\
 & & \psi
 \end{array}$$

Since $\tau, \phi \vdash_T \psi$, we can conclude that $\tau \vdash_T \phi \supset \psi$ which gives us the arrow δ . This arrow is unique since any other proof $\tau \vdash_T \phi \supset \psi$ will either be identical or involve a detour. We can remove the detour because \mathcal{L}_T satisfies the local reduction property.

We are now able to define a Kripke structure for $\mathcal{L}_T, \mathcal{K}_T : [\mathcal{W}, [\mathcal{B}, \bar{\mathcal{V}}]]$. We define the category $\bar{\mathcal{V}}$ as follows:

Objects: Categories $\bar{\mathcal{V}}$ built out of $V = \mathcal{J}(X)(\Delta)$, which contain

Objects: Arrows $\phi_1 \times \dots \times \phi_n \rightarrow \psi$ in $\mathcal{J}(X)(\Delta)$;

Arrows: Arrows $\Gamma \rightarrow \psi$ to $\Gamma' \rightarrow \tau$ are given by arrows $\Gamma \rightarrow \Gamma'$ in $\mathcal{J}(X)(\Delta)$.

Arrows: Functors $f^* : \mathcal{K}_T(X)(\Delta) \rightarrow \mathcal{K}_T(X)(\Delta')$, where $f : \Delta' \rightarrow \Delta$ is an arrow in $\mathcal{T}(X)$ are defined to be the usual substitution.

It is straightforward to check that the functors f^* satisfy the definition. The interpretation is taken to be the identity function in the fibres, in the base categories \mathcal{C}_X , we interpret a sort S by $x : S$.

5.2.4 Satisfaction

Satisfaction in Kripke models of \mathcal{L}_T follows a pattern similar to that for Kripke Σ - $\lambda\Pi$ -models. Since the base category is concerned only with terms, rather than propositions as well, we can begin with the satisfaction of propositions, rather than of consequences.

Definition 5.7 (\Vdash -satisfaction)

Let $\mathcal{K}_{\mathcal{J}}, \Vdash_{\mathcal{K}_{\mathcal{J}}}$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [[\mathcal{B}^{op}, \bar{\mathcal{V}}]]]$, be a Kripke model of \mathcal{L}_T . The *satisfaction (forcing) relation* $W, \rho \Vdash_{\phi}^{\mathcal{K}_{\mathcal{J}}}$ is defined by induction on the structure of formulæ, as follows:

- $W, \rho \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{T}} p(X)$ if and only if there exists an arrow $1 \xrightarrow{f} \llbracket p(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ in $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho})$, where $X = \{x_1 : S_1, \dots, x_m : S_m\}$ and $S = S_1 \times \dots \times S_m$;
- $W, \rho \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \phi \supset \psi$ if and only if, for all $W \xrightarrow{f} W'$, $(W', \rho[f] \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \phi$ implies $W', \rho[f] \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \psi$);
- $W, \rho \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \forall x : S. \phi$ if and only if, for all $W \xrightarrow{f} W'$ if, for all $a \in \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ and terms t such that $\llbracket t \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho} = a$, $W', \rho[x := a] \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{T}} \phi[t/x]$.

If $\Gamma = \phi_1, \dots, \phi_m$, then we write $W, \rho \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \Gamma$ if, for each $1 \leq i \leq m$, $W, \rho \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \phi_i$. We write $W, \rho \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} (\Gamma \vdash_T \phi)$, or more commonly, $W, \rho, \Gamma \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \phi$, if $W, \rho \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \Gamma$ implies $W, \rho \Vdash_{\mathcal{K}_{\mathcal{J}}}^{\mathcal{K}} \phi$. ■

The notion of satisfaction given above is a straightforward generalization of the informal one discussed earlier. Moreover, it is monotone: if $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$, if $W \xrightarrow{f} W'$ and if $\llbracket \phi \rrbracket_{\rho[f]}^{W'} \mathcal{K}_{\mathcal{J}}$ is defined, then $W', \rho[f] \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$. More economically, we have the following characterization of satisfaction.

Lemma 5.8 (\Vdash -forcing via global sections)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, be a Kripke model of \mathcal{L}_T . Let ϕ be a proposition with variables in X of sorts S_1, \dots, S_m . Let $S = S_1 \times \dots \times S_m$. Then $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$ if and only if there is an arrow $1 \xrightarrow{m} \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ in $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho})$.

Proof (Sketch) By induction on the structure of propositions. For example, suppose $\phi = \psi_1 \supset \psi_2$. The induction hypothesis gives arrows $1 \xrightarrow{m_1} \llbracket \psi_1(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ and $1 \xrightarrow{m_2} \llbracket \psi_2(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ in $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho})$. Since $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho})$ is cartesian closed, it follows that there is an arrow $1 \times \llbracket \psi_1(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho} \rightarrow \llbracket \psi_2(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ in $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho})$. Consequently, there is an arrow $1 \rightarrow (\llbracket \psi_2(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho})^{\llbracket \psi_1(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}}$.

Conversely, given an arrow $1 \rightarrow \llbracket \psi_1(X) \supset \psi_2(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$, it follows immediately that the existence of an arrow $1 \rightarrow \llbracket \psi_1(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ implies the existence of an arrow $1 \rightarrow \llbracket \psi_2(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$.

The other cases are similar. ■

5.2.5 Soundness and Completeness for \Vdash

We readily obtain the following by induction on the structure of proofs:

Proposition 5.9 (Soundness for \Vdash)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{B}^{op}, \overline{\mathcal{V}}]]$, be any Kripke model of \mathcal{L}_T . If $\Gamma \vdash_T \phi$ has a natural deduction proof, then, at every world W , we have $W, \rho, \Gamma \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$.

Proof We proceed by induction on the structure of proofs in \mathcal{L}_T . We begin with the axiom rule, i.e., $\phi \in \Gamma$. We assume that $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \Gamma$, i.e., $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi_i$ for all $\phi_i \in \Gamma$, therefore, $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$ and we are done.

We now assume that we have a proof of $\phi \supset \psi$, i.e.,

$$\frac{\Gamma, \phi \vdash_T \psi}{\Gamma \vdash_T \phi \supset \psi} \supset I$$

We apply the induction hypothesis to obtain $W, \rho, \Gamma, \phi \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \psi$. This means that we have $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$ and $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \psi$, i.e., there are arrows $1 \xrightarrow{m_1} \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ and $1 \xrightarrow{m_2} \llbracket \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ in $\mathcal{J}(W)(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho})$. It follows that $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi \supset \psi$ and hence $W, \rho, \Gamma \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi \supset \psi$.

We assume that the last rule used was

$$\frac{\Gamma \vdash_T \phi \quad \Gamma \vdash_T \phi \supset \psi}{\Gamma \vdash_T \psi} \supset E$$

We apply the induction hypothesis to the premisses to obtain $W, \rho, \Gamma \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \phi$ and $W, \rho, \Gamma \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \phi \supset \psi$. Hence we have arrows $1 \rightarrow \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ and $1 \rightarrow \llbracket \phi \supset \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$. It follows that we have an arrow $1 \rightarrow \llbracket \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ and so we have $W, \rho, \Gamma \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \psi$ as required.

Suppose that the last rule used was

$$\frac{\Gamma \vdash_T \phi}{\Gamma \vdash_T \forall x : S . \phi} \forall I$$

We apply the induction hypothesis to obtain $W, \rho, \Gamma \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \phi$ and thus there is an arrow $1 \rightarrow \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ in $\mathcal{J}(W)(X, x : S)$. We can apply $\forall_{X, S}$ to obtain an arrow $1 \rightarrow \llbracket \forall x : S . \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho/x}$ in $\mathcal{J}(W)(X)$. Thus $W, \rho, \Gamma \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \forall x : S . \phi$.

Finally, we assume that the last rule used was

$$\frac{\Gamma \vdash_T \forall x : S . \phi}{\Gamma \vdash_T \phi[t/x]} \forall E$$

We apply the induction hypothesis to obtain $W, \rho, \Gamma \Vdash_{-\forall x : S . \phi}^{\mathcal{K}_{\mathcal{J}}}$. Hence $W, \rho \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $W, \rho \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \forall x : S . \phi$. By Definition 5.7, we have that $W, \rho[x := a] \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \forall x : S . \phi$; we take f to be the identity. Hence $W, \rho, \Gamma \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \phi[t/x]$. \blacksquare

We now prove a model existence lemma. It is worthwhile to contrast our proof with that of Van Dalen (1994). We call a set Δ of propositions prime if Δ is closed under \vdash_T . This is all we require of the usual definition of a prime theory, which also requires disjunction and existence properties, since we are only dealing with the $\{\forall, \supset\}$ -fragment. By a standard result ((Van Dalen 1994), Lemma 5.3.8, pp. 168-169), we can extend Γ to a prime Γ' such that $\Gamma' \not\vdash_T \phi$. For this lemma, just as in the corresponding lemma for Kripke $\lambda\Pi$ -models (Lemma 3.39), we do not require such a construction. This is because we will construct our model not out of propositional consequence, but out of the proof-objects which realize consequences. We consider, for each Γ , all propositions ϕ and all proofs Φ such that Φ realizes, *i.e.*, is a proof of ϕ from Γ .

Lemma 5.10 (Model Existence)

There exists a Kripke model of \mathcal{L}_T , $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-, \rho} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{B}^{op}, \overline{\mathcal{V}}]]$, with a world W_0 such that if $\Gamma \not\vdash_T \phi$ then $W_0, \rho \Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ and $W_0, \rho \not\Vdash_{-T}^{\mathcal{K}_{\mathcal{J}}} \phi$.

Proof The term model we constructed in § 5.2.3 is the required Kripke model of \mathcal{L}_T . We take $W_0 = \langle \rangle$. \blacksquare

Analogously to the first-order situation, *cf.* (Van Dalen 1994), or § 3.7, we define $\Gamma \Vdash_T \phi$ as follows $\Gamma \Vdash_T \phi$, where $\Gamma = \phi_1, \dots, \phi_m$, if, for all Kripke models of \mathcal{L}_T and all worlds W of $\mathcal{K}_{\mathcal{J}}$, $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \Gamma$ (*i.e.*, $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi_i$, for each $1 \leq i \leq m$) implies $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$.

Theorem 5.11 (Completeness for \Vdash)

$\Gamma \Vdash_T \phi$ if and only if $\Gamma \vdash_T \phi$ has a natural deduction proof.

Proof

Only If This is soundness, Lemma 5.9.

If Suppose $\Gamma \not\vdash_T \phi$, then Lemma 5.10 yields a contradiction. ■

5.2.6 Soundness and Completeness for \Vdash

The astute reader may have noticed that because we have a product in $\mathcal{J}(W)(U)$, we can interpret conjunction. We extend our interpretation to include $\llbracket \phi \wedge \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho} = \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho} \times \llbracket \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ and add the following condition to the satisfaction relation:

Conjunction: $w, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi \wedge \psi$ if and only if $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$ and $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \psi$.

We can also extend the satisfaction of propositions to consequences as follows: if

$\Gamma = \phi_1, \dots, \phi_m$, then $W, \rho, \Gamma \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$ if and only if, for all $W \xrightarrow{f} W'$, $(W, \rho[f] \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \bigwedge \Gamma$ implies $W, \rho[f] \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$), where $\bigwedge \Gamma = \phi_1 \wedge \dots \wedge \phi_m$.

Lemma 5.12 (Satisfaction of Consequences)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho} \rangle$ be any Kripke model of \mathcal{L}_T . If $\Gamma = \phi_1, \dots, \phi_m$, then $W, \rho, \bigwedge \Gamma \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$ if and only if $W, \rho, \Gamma \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$.

Proof Let $\Gamma = \phi_1, \dots, \phi_m$. Since $\Gamma \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$, we know that for all worlds W , we have that $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi_i$ for $1 \leq i \leq m$, implies $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi$. By the definition of conjunction, we are able to rewrite $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi_i$, for $1 \leq i \leq m$, as $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \bigwedge \Gamma$. For the converse, observe that this argument can be reversed. ■

Now we see that our models have enough structure to interpret not only the consequences but also the proofs, or realizers of consequences of \mathcal{L}_T (see also (Seely 1983)). Let $x_1 : S_1, \dots, x_m : S_m$ and let X denote the set of variables $\{x_1, \dots, x_m\}$. Let $\delta : (\phi_1(X), \dots, \phi_m(X) \vdash_T \phi(X))$ be a natural deduction proof.

Let, for each $1 \leq i \leq m$, $\llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ and $\llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ be defined. If $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$, the interpretation of δ , is defined, then it is an object

$$\left(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho} \right) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}} \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$$

in $\mathcal{K}_{\mathcal{J}}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})$, defined by induction on the structure of natural deduction proofs. We write $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_T \phi)$ if and only if

$$\left(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \right) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}} \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$$

is defined in $\mathcal{K}_{\mathcal{J}}$.

We have the following relationship between \Vdash and \Vdash . \Vdash is a special case of \Vdash and, in the term model, we can obtain \Vdash from \Vdash . We now prove soundness for \Vdash .

Proposition 5.13 (Soundness for \Vdash)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{B}^{op}, \overline{\mathcal{V}}]]$ be any Kripke model for \mathcal{L}_T . If $\delta : (\Gamma \vdash_T \phi)$ is a natural deduction proof and $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ is defined, then $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_T \phi)$.

Proof We proceed by induction on the structure of proof-objects. Firstly, we consider the case where $\delta = \alpha_{\phi}$. Thus we have $\alpha_{\phi} : (\Gamma \vdash_T \phi)$ and $\phi \in \Gamma$. Since

$\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})$ is cartesian closed, there exists a projection $(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{\llbracket \alpha_{\phi} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}} \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$, thus $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \phi \alpha_{\phi} : (\Gamma \vdash_T \phi)$.

We assume that $\delta = \delta_1 \delta_2$. We apply the induction hypothesis and obtain

$(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{\llbracket \delta_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}} \llbracket \phi \supset \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ and $(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{\llbracket \delta_2 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}} \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$. Since these are both arrows in $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})$ and $\llbracket \phi \supset \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} = (\llbracket \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})^{\llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}}$, we exploit evaluation in a cartesian closed category to obtain $(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{eval \circ (\llbracket \phi_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \llbracket \delta_2 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})} \llbracket \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$. Hence $W, \rho \Vdash_{\delta_1 \delta_2 (\Gamma \vdash_T \psi)}^{\mathcal{K}_{\mathcal{J}}} \delta$.

Suppose that $\delta = I_{\phi} \delta$. We apply the induction hypothesis to obtain $(\prod_{i=1}^m$

$\llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \times \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}} \llbracket \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$. By definition, this is an arrow of $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})$.

We exploit the cartesian closed structure of $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})$ to obtain an arrow

$(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{\lambda \llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}} \llbracket \phi \supset \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$. Hence $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} I_{\phi} \delta : (\Gamma \vdash_T \phi \supset \psi)$.

We consider the case where $\delta = \delta t$. Applying the induction hypothesis

yields $(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}} \llbracket \forall x : T . \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$. By definition, this is an arrow in $\mathcal{J}(W)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho})$. We have a morphism $\llbracket t \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} : \llbracket T \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho} \rightarrow \llbracket T \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$, which induces t^* ,

which sends $(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}} \llbracket \forall x : T . \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$ to $(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) \xrightarrow{t^* \llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}}$

$\llbracket \phi[t/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}$. We use the first of the Beck-Chevalley conditions to obtain this

arrow. We now have $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \delta t : (\Gamma \vdash_T \phi[t/x])$.

Finally, we have the proof-object $Gx : A . \delta$. We apply the induction hypothesis to obtain $(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho} \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}} \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$. By definition, this is an arrow in $\mathcal{J}(W)(\llbracket S \times A \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho})$. We apply the functor $\forall_{S, A}$ to obtain an arrow $(\prod_{i=1}^m \llbracket \phi_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}) \xrightarrow{\forall_{S, A} \llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}} \llbracket \forall x : A . \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ since $x : A \notin FV(A)$. Hence $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} Gx : A . \delta : (\Gamma \vdash_T \forall x : A . \phi)$. ■

Lemma 5.14 (Model Existence for \Vdash)

There exists a Kripke model of \mathcal{L}_T , $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{B}^{op}, \bar{\mathcal{V}}]]$, and a world W_0 such that if there does not exist a natural deduction proof $\delta : (\Gamma \vdash_T \phi)$, then $W_0, \rho \not\Vdash_T^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_T \phi)$.

Proof Again we use the term model we constructed in § 5.2.3. World $W_0 = \langle \rangle$ gives the required condition. ■

We write $(\Vdash)_T \delta : (\Gamma \vdash_T \phi)$ if and only if for all Kripke models $\mathcal{K}_{\mathcal{J}}$ and all worlds W , we have that $W, \rho \Vdash_T^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_T \phi)$.

Theorem 5.15 (Completeness for \Vdash)

$\Vdash_T \vdash \delta : (\Gamma \vdash_T \phi)$ if and only if $\delta : (\Gamma \vdash_T \phi)$ is a natural deduction proof and $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$ is defined.

Proof

Only If By Soundness, Proposition 5.13.

If Suppose that $\delta : (\Gamma \vdash_T \phi)$ is not a natural deduction proof, then Lemma 5.14 yields a contradiction. ■

5.3 Propositions-as-types Isomorphism

We are now able to set up the propositions-as-types isomorphism, an indexed isomorphism between suitable Kripke models, induced by the propositions-as-types correspondence. Gardner (1992a) provides similar models and morphisms between models to ours; she does not have the worlds structure.

We begin with the definition of an indexed functor, between indexed categories.

Definition 5.16 (Indexed Functors)

Let $\mathcal{F} : [\mathcal{W}, [\mathcal{A}^{op}, \mathcal{C}]]$ and $\mathcal{G} : [\mathcal{X}, [\mathcal{B}^{op}, \mathcal{C}]]$ be strict indexed categories. An *indexed functor from \mathcal{F} to \mathcal{G}* consists of a triple

$$\tau = (\alpha, \beta, (\epsilon_W)_{W \in |\mathcal{W}|}),$$

where $\alpha : \mathcal{W} \rightarrow \mathcal{X}$, $\beta : \mathcal{A} \rightarrow \mathcal{B}$ are functors and, for each object W in \mathcal{W} , $\epsilon_W : \mathcal{F}(W) \Rightarrow \beta^{op}; \mathcal{G}(W)$ is a natural transformation such that for each $f : v \rightarrow w$ in \mathcal{W} , the diagram

$$\begin{array}{ccc}
 v & & \mathcal{F}(v) \xrightarrow{\epsilon_W} \beta^{op}; \mathcal{G}(\alpha(w)) \\
 \downarrow f & & \Downarrow \mathcal{F}(f) \qquad \qquad \Downarrow \mathcal{G}(\alpha)(f) \\
 w & & \mathcal{W}(w) \xrightarrow{\epsilon_W} \beta^{op}; \mathcal{G}(\alpha(w))
 \end{array}$$

commutes ■

Our definition of an indexed functor is similar to that found in (Gardner 1992a). Our definition has a triple $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$ whereas Gardner only has a pair (σ, σ_{base}) , where σ is a natural transformation. This is because we have two levels of indexing in our Kripke $\lambda\Pi$ -model and so require an extra functor at the second indexing category. We also have a coherence condition on the natural transformations, unlike Gardner. This ensures that a transition between worlds has the same effect as applying the corresponding transition after the appropriate natural transformation.

Definition 5.17 (Indexed Isomorphism)

An indexed functor $\tau = (\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$ is an *indexed isomorphism* if α and β are isomorphisms and each ϵ_w is a natural isomorphism. ■

To avoid confusion, from now on we use $\mathcal{R}_{\mathcal{S}} : [\mathcal{X}, [\mathcal{E}^{op}, \bar{\mathcal{U}}]]$ to refer to Kripke structures for \mathcal{L}_T .

Definition 5.18 (Category of Models)

We define a category \mathcal{M} of *models* as follows:

Objects: Equational Kripke Σ - $\lambda\Pi$ -models, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, and Kripke models of \mathcal{L}_T , $\langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-, \rho} \rangle$;

Arrows: There are four cases:

1. An arrow

$$\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \xrightarrow{h} \langle \mathcal{K}'_{\mathcal{J}'}, \llbracket - \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^- \rangle$$

is given by an indexed functor $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|}) : \mathcal{K}_{\mathcal{J}} \rightarrow \mathcal{K}'_{\mathcal{J}'}$ such that if $\alpha W = W'$, then $h(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) = \llbracket X \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^{W'}$;

2. An arrow

$$\langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-, \rho} \rangle \xrightarrow{h} \langle \mathcal{R}'_{\mathcal{S}'}, \llbracket - \rrbracket_{\mathcal{R}'_{\mathcal{S}'}}^{-, \rho'} \rangle$$

is given by an indexed functor $(\alpha, \beta, (\epsilon_x)_{x \in |\mathcal{X}|}) : \mathcal{R}_S \rightarrow \mathcal{R}'_{S'}$ such that if $\alpha x = x'$, then $h(\llbracket X \rrbracket_{\mathcal{R}_S}^{x, \rho}) = \llbracket X \rrbracket_{\mathcal{R}'_{S'}}^{x', \rho'}$;

3. An arrow

$$\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \xrightarrow{h} \langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-, \rho} \rangle$$

is given by an indexed functor $(\alpha, \beta, (\epsilon_W)_{W \in |\mathcal{W}|}) : \mathcal{K}_{\mathcal{J}} \rightarrow \mathcal{R}_S$ such that if $\alpha W = x$, then $h(\llbracket \{X\} \rrbracket_{\mathcal{R}_S}^W) = \llbracket X \rrbracket_{\mathcal{R}_S}^{x, \rho}$;

4. An arrow

$$\langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-, \rho} \rangle \xrightarrow{h} \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$$

is given by an indexed functor $(\alpha, \beta, (\epsilon_x)_{x \in |\mathcal{X}|}) : \mathcal{R}_S \rightarrow \mathcal{K}_{\mathcal{J}}$ such that if $\alpha x = W$, then $h(\llbracket X \rrbracket_{\mathcal{R}_S}^{x, \rho}) = \llbracket \{X\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. \blacksquare

Proposition 5.19 (\mathcal{M} is Well-defined)

The category \mathcal{M} defined in Definition 5.18 is well-defined.

Proof We have to show the following: there exists an identity; for all arrows f and g , with appropriate domains and co-domains, $f \circ g$ is also an arrow in \mathcal{M} ; composition is associative.

For the identity, we take the indexed functor which consists of the identity functors and natural transformations.

We have to show that given arrows $h_1 : \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \rightarrow \langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-, \rho} \rangle$ and $h_2 : \langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-, \rho} \rangle \rightarrow \langle \mathcal{K}'_{\mathcal{J}'}, \llbracket - \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^- \rangle$, which consist of indexed functors $(\alpha_1, \beta_1, (\epsilon_w^1)_{w \in |\mathcal{W}|})$ and $(\alpha_2, \beta_2, (\epsilon_x^2)_{x \in |\mathcal{X}|})$ respectively, the composition $h_2 \circ h_1 : \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \rightarrow \langle \mathcal{K}'_{\mathcal{J}'}, \llbracket - \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^- \rangle$ is an indexed functor and if $\alpha_2 \circ \alpha_1(W) = W'$, then $h_2 \circ h_1(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) = \llbracket X \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^{W'}$. $h_2 \circ h_1$ is the indexed functor $(\alpha_2 \circ \alpha_1, \beta_2 \circ \beta_1, (\epsilon_{\alpha_1(w)}^2 \circ \epsilon_w^1)_{w \in |\mathcal{W}|})$. If $\alpha_2 \circ \alpha_1(W) = W'$, then $\alpha_1(W) = x$ and $\alpha_2(x) = W'$. So we have that $h_1(\llbracket \{X\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) = \llbracket X \rrbracket_{\mathcal{R}_S}^{x, \rho}$ and $h_2(\llbracket X \rrbracket_{\mathcal{R}_S}^{x, \rho}) = \llbracket \{X\} \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^{W'}$, hence $h_2 \circ h_1(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) = \llbracket X \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^{W'}$.

We now show that given arrows $h_1 : \langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-, \rho} \rangle \rightarrow \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$ and $h_2 : \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \rightarrow \langle \mathcal{R}'_{S'}, \llbracket - \rrbracket_{\mathcal{R}'_{S'}}^{-, \rho'} \rangle$, which consist of indexed functors $(\alpha_1, \beta_1, (\epsilon_x^1)_{x \in |\mathcal{X}|})$ and $(\alpha_2, \beta_2, (\epsilon_w^2)_{w \in |\mathcal{W}|})$ respectively, the composition $h_2 \circ h_1 : \langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-, \rho} \rangle \rightarrow \langle \mathcal{R}'_{S'}, \llbracket - \rrbracket_{\mathcal{R}'_{S'}}^{-, \rho'} \rangle$ is an indexed functor and if $\alpha_2 \circ \alpha_1(x) = x'$, then $h_2 \circ h_1(\llbracket X \rrbracket_{\mathcal{R}_S}^{w, \rho}) = \llbracket X \rrbracket_{\mathcal{R}'_{S'}}^{w', \rho'}$. $h_2 \circ h_1$ is the indexed functor $(\alpha_2 \circ \alpha_1, \beta_2 \circ \beta_1, (\epsilon_{\alpha_1(x)}^2 \circ \epsilon_x^1)_{x \in |\mathcal{X}|})$. Let $\alpha_1(x) = W$ and $\alpha_2(W) = x'$, then $\alpha_2 \circ \alpha_1(x) = x'$. Also $h_1(\llbracket X \rrbracket_{\mathcal{R}_S}^{x, \rho}) = \llbracket \{X\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and $h_2(\llbracket \{X\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) = \llbracket X \rrbracket_{\mathcal{R}'_{S'}}^{x', \rho'}$. We thus have $h_2 \circ h_1(\llbracket X \rrbracket_{\mathcal{R}_S}^{x, \rho}) = \llbracket X \rrbracket_{\mathcal{R}'_{S'}}^{x', \rho'}$.

The other cases are straightforward. Associativity follows from the fact that the composition of two indexed functors $(\alpha_1, \beta_1, (\epsilon_x^1)_{x \in X})$ and $(\alpha_2, \beta_2, (\epsilon_w^2)_{w \in W})$ is defined to be $(\alpha_2 \circ \alpha_1, \beta_2 \circ \beta_1, (\epsilon_{\alpha_1(x)}^2 \circ \epsilon_x^1)_{x \in X})$. \blacksquare

The definition of Kripke prestructures and prestructures, for both the λ -calculus and the internal logic, involve the categories satisfying certain properties.

The parts of the categories which satisfy these properties are those which interpret the syntax of either the $\lambda\Pi$ -calculus or the internal logic. We restrict our attention to morphisms between these parts of the Kripke models.

Before we are able to show the existence of an isomorphism of models between a Kripke $\lambda\Pi$ -model and a Kripke model of \mathcal{L}_T , we need one further restriction. The Kripke $\lambda\Pi$ -models have more structure in the Kripke $\lambda\Pi$ -prestructure than the Kripke-prestructure for \mathcal{L}_T . The extra structure allows us to interpret realizations of the form $\Delta \xrightarrow{\sigma} \Theta$ in the Kripke $\lambda\Pi$ -model, while we are only able to interpret realizations of the form $\Gamma \xrightarrow{\delta} \phi$ in a Kripke model of \mathcal{L}_T . We thus restrict to the arrows in the Kripke $\lambda\Pi$ -models which interpret realizations of the form $\Delta \xrightarrow{\sigma} x : A$, which correspond to realizations $\Gamma \xrightarrow{\delta} \phi$ under the propositions-as-types correspondence.

We are now in a position to define an isomorphism of models.

Definition 5.20 (Isomorphism of Models)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$ and $\langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle$ be objects of \mathcal{M} . Let $h : \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \rightarrow \langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle$ be a morphism of models. We say that h is an *isomorphism of models* if the indexed functor $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|}) : \mathcal{K}_{\mathcal{J}} \rightarrow \mathcal{R}_{\mathcal{S}}$ (corresponding to h) is an indexed isomorphism when its domain is restricted to those objects and arrows in $\mathcal{K}_{\mathcal{J}}$ which interpret the syntax of the $\lambda\Pi$ -calculus excluding realizations of the form $\Delta \rightarrow z : A$ and its range is restricted to those objects and arrows in $\mathcal{R}_{\mathcal{S}}$ which interpret the syntax of \mathcal{L}_T . \blacksquare

Proposition 5.21 (Propositions-as-types Isomorphism)

Let T be a theory of the $\{\forall, \supset\}$ -fragment of many sorted minimal first-order logic and let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, be a Kripke $\Sigma_{\mathcal{L}_T}$ - $\lambda\Pi$ -model, where $\Sigma_{\mathcal{L}_T}$ is the $\lambda\Pi$ -signature in propositions-as-types correspondence with \mathcal{L}_T . Then there is a Kripke model for \mathcal{L}_T , $\langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle$, where $\mathcal{R}_{\mathcal{S}} : [\mathcal{X}, [\mathcal{E}^{op}, \bar{\mathcal{U}}]]$, together with an isomorphism of models

$$h : \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \rightarrow \langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle$$

induced by the propositions-as-types correspondence. Specifically, abusing notation by allowing X to range over all the syntax of \mathcal{L}_T and suppressing information above worlds, if $\llbracket X \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{x,\rho}$ and $\llbracket \{X\} \rrbracket_{\mathcal{R}_{\mathcal{S}}}^W$ are defined, then

$$h(\llbracket \{X\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^-) = (\llbracket X \rrbracket_{\mathcal{R}_{\mathcal{S}}}^-).$$

Proof (Sketch) Given $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, we sketch the construction of $\mathcal{R}_{\mathcal{S}}$, together with an indexed isomorphism $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|}) : \mathcal{K}_{\mathcal{J}} \rightarrow \mathcal{R}_{\mathcal{S}}$.

- We take $\mathcal{X} = \mathcal{W}$ with $\alpha = 1_{\mathcal{W}}$. It should be clear that α is an isomorphism.
- We take \mathcal{E} to be the subcategory of \mathcal{D} defined as follows:

Objects: objects D of \mathcal{D} such that $D = \llbracket \{\{T\}\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, where T is a sort of \mathcal{L}_T ;

Arrows: all arrows in \mathcal{D} whose domains and co-domains are objects of \mathcal{E} .

We define the functor $\beta : \mathcal{D} \rightarrow \mathcal{E}$ to be the functor which is the identity functor on all objects and arrows in \mathcal{D} which are also in \mathcal{E} and sends any other objects in \mathcal{D} to the terminal object in \mathcal{E} and any other arrows in \mathcal{D} to the identity arrow on the terminal object in \mathcal{E} . It should be clear that β is an isomorphism.

- We take \mathcal{U} and $\bar{\mathcal{U}}$ to be the subcategories of \mathcal{V} and $\bar{\mathcal{V}}$ defined as follows:

Objects of \mathcal{U} : objects $\mathcal{J}(W)(D)$ in \mathcal{V} such that for each object A in $\mathcal{J}(W)(D)$, where $A = \llbracket \{\{\phi\}\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$ and ϕ is a formula in \mathcal{L}_T , and for each arrow $A \xrightarrow{m} B$ in $\mathcal{J}(W)(D)$, $m = \llbracket \{\{\Phi\}\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, where Φ is a proof in \mathcal{L}_T ;

Arrows of \mathcal{U} : arrows in \mathcal{V} whose domain and co-domain are objects in \mathcal{U} .

We could now define an (indexed) isomorphism of Kripke $\lambda\Pi$ -prestructures, with the obvious family of natural transformations, however, we instead turn to the category $\bar{\mathcal{U}}$.

Objects of $\bar{\mathcal{U}}$: objects $\mathcal{K}_{\mathcal{J}}(W)(D)$ of $\bar{\mathcal{V}}$ where for each object $\bar{A} \rightarrow A$ in $\mathcal{K}_{\mathcal{J}}(W)(D)$, $\bar{A} \rightarrow A = \llbracket \{\{\delta\}\} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$, where δ is a natural deduction proof in \mathcal{L}_T ;

Arrows of $\bar{\mathcal{U}}$: arrows of $\bar{\mathcal{V}}$ whose domains and co-domains are objects of $\bar{\mathcal{U}}$.

This completes our construction of $\mathcal{R}_{\mathcal{S}}$ and it is straightforward to show that $\mathcal{R}_{\mathcal{S}}$ is a Kripke structure for \mathcal{L}_T . We continue with the construction of an indexed isomorphism $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$. We now define a family of natural transformations $(\epsilon_w)_{w \in |\mathcal{W}|} : \mathcal{K}_{\mathcal{J}}(w) \Rightarrow \beta^{op}; \mathcal{R}_{\mathcal{S}}(\alpha(w))$. We fix w and define each component of ϵ_w , $\eta_a^w : \mathcal{K}_{\mathcal{J}}(w)(a) \rightarrow (\beta^{op}; \mathcal{R}_{\mathcal{S}}(\alpha(w)))a$, where $a \in |\mathcal{D}|$, to be the functor which is the identity functor on objects and arrows in $\mathcal{K}_{\mathcal{J}}(w)(a)$ which are also in $\mathcal{R}_{\mathcal{S}}(\alpha(w))(\beta^{op}(a))$ and sends objects in $\mathcal{K}_{\mathcal{J}}(w)(a)$ which are not in $\mathcal{R}_{\mathcal{S}}(\alpha(w))(\beta^{op}(a))$ to the terminal object in $\mathcal{R}_{\mathcal{S}}(\alpha(w))(\beta^{op}(a))$ and arrows in $\mathcal{K}_{\mathcal{J}}(w)(a)$ which are not in $\mathcal{R}_{\mathcal{S}}(\alpha(w))(\beta^{op}(a))$ to the identity arrow on

$\mathcal{R}_S(\alpha(w))(\beta^{op}(a))$. We need to show that the diagram

$$\begin{array}{ccccc}
 a & & \mathcal{K}_{\mathcal{J}}(w)(a) & \xrightarrow{\eta_a^w} & \mathcal{R}_S(\alpha(w))(\beta^{op}(a)) \\
 \downarrow f & & \downarrow \mathcal{K}_{\mathcal{J}}(w)(f) & & \downarrow \mathcal{R}_S(\alpha(w))(\beta^{op}(f)) \\
 b & & \mathcal{K}_{\mathcal{J}}(w)(b) & \xrightarrow{\eta_b^w} & \mathcal{R}_S(\alpha(w))(\beta^{op}(b))
 \end{array}$$

commutes. This follows from the definition of η_-^w . It should be clear that each ϵ_w is a natural isomorphism. Once we have shown that the diagram

$$\begin{array}{ccc}
 v & \mathcal{K}_{\mathcal{J}}(v) \xrightarrow{\epsilon_v} \beta^{op}; \mathcal{R}_S(\alpha(v)) & \\
 \downarrow f & \Downarrow \mathcal{K}_{\mathcal{J}}(f) & \Downarrow \mathcal{R}_S(\alpha(f)) \\
 w & \mathcal{K}_{\mathcal{J}}(w) \xrightarrow{\epsilon_w} \beta^{op}; \mathcal{R}_S(\alpha(w)) &
 \end{array}$$

commutes, we have shown that $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$ is an indexed isomorphism. The commutativity of the diagram follows from the definition of each natural transformation ϵ_w .

It remains to show that there is a Kripke model of \mathcal{L}_T which uses \mathcal{R}_S and that there is an isomorphism of models h . We use the propositions-as-types correspondence and the interpretation function $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^-$ to define the interpretation function $\llbracket - \rrbracket_{\mathcal{R}_S}^{-;\rho}$. Letting X range over the syntax of \mathcal{L}_T , we define $\llbracket X \rrbracket_{\mathcal{R}_S}^{x;\rho} = \llbracket \epsilon(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w$, where $\alpha(w) = x$. Showing that $\langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-;\rho} \rangle$ is a Kripke model of \mathcal{L}_T is straightforward. h is then defined to be the morphism of models which sends $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$ to $\langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-;\rho} \rangle$ using the indexed isomorphism $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$. We observe that the required condition on the interpretation function holds for h to be a morphism of models. \blacksquare

5.4 Kripke Models vs. Classical Models

We conclude this chapter with a reconstruction, in the dependently typed setting, of a simple but pleasing result formulated for models of the simply typed λ -calculus by Mitchell & Moggi (1991).

We need to extend the internal logic to include negation. We do this in a standard, semantic, way by introducing a proposition \perp such that, for every

Kripke model $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-\rho} \rangle$ of \mathcal{L}_T and every world W ,

$$W, \rho \not\Vdash_T^{\mathcal{K}_{\mathcal{J}}} \perp,$$

i.e., there is no model $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-\rho} \rangle$ with any world W in which there is an arrow $1 \rightarrow \llbracket \perp \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W, \rho}$, for any set of variables. We then define the usual intuitionistic negation, $\neg\phi =_{def} \phi \supset \perp$.

In order to establish that there are Kripke models of \mathcal{L}_T which do not arise as a Kripke quotient of a classical model, we shall not need to go beyond models that are based on our (set-theoretic) applicative structures. Moreover, we shall restrict our attention to applicative structures with families, *q.v.* § 4.1.2. The consequence of this restriction, from the point of view of \mathcal{L}_T , is that we are able to interpret not consequences but just propositions. However, such models admit the usual (typed) first-order existential quantifier, *i.e.*, for \mathcal{L}_T , the existential quantifier can be interpreted, at each variable $x : S$, as a functor which is left adjoint to the inverse image of the projection $X, x : S \xrightarrow{p_{x,S}} X$, and for $\lambda\Pi$, the existential quantifier can be interpreted as a functor, that is left adjoint to the inverse image of the projection $\llbracket \Gamma, x : S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W \xrightarrow{p} \llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W$. Elementarily equivalent means that any sentence satisfied in one model is satisfied in the other.

Theorem 5.22 (Kripke Models are Non-classical)

There is an equational Kripke Σ - $\lambda\Pi_{=}$ -applicative structure \mathcal{U} which is not elementary equivalent to any \mathcal{V}/\mathcal{R} , where \mathcal{V} is any classical Σ - $\lambda\Pi_{=}$ -applicative structure and \mathcal{R} is any Kripke Σ - $\lambda\Pi_{=}$ -partial logical equivalence relation on \mathcal{V} .

Proof (Sketch) We give a proposition ϕ in \mathcal{L}_T , extended with \perp as described above, which is valid in all quotients \mathcal{V}/\mathcal{R} but which is not valid in every \mathcal{U} .

The idea is for ϕ to be a predicate which characterizes the inhabitation of a type, *i.e.*,

- if *empty*(p) and $\neg\neg$ *inhabited*($p \supset q$), then *inhabited*($p \supset q$),

where *inhabited*(τ) $\equiv (\exists x : \tau . x = x)$ and *empty*(τ) $\equiv \neg$ *inhabited*(τ). This holds in a classical Σ - $\lambda\Pi_{=}$ -applicative structure, but is not intuitionistically valid. ■

Chapter 6

Introduction to Logical Frameworks

In this chapter, we provide an introduction to one of the key ideas in this thesis; that of a logical framework. Logically, logical frameworks can be seen as arising from Martin-Löf’s intuitionistic theory of iterated inductive definitions ((Martin-Löf 1971) and (Martin-Löf 1975)) in which *form* and *inductive definitional status* in the natural deductive rules are considered. In other words, Martin-Löf provides a formal meta-theory of inference rules. This theory is further developed by extending Kant’s (1800) notion of a judgement in (Martin-Löf 1982).

Computationally, the need for a formal account of the relationship between a logic and its meta-theory arises from the desire, in computer science, to manipulate representations of logics and other formal systems. Here we are mainly concerned with *logics*. Our conception of logic here is a broad one, it is possible to consider the linear λ -calculus with equality judgements as a logic; for example. In order to represent a logic in a machine, the logic must be described in a *programming language* or *metallogic*. Moreover, if we are to understand the resulting program, we must have a *fixed* metalogic.

We develop our account of logical frameworks from their philosophical basis. Our starting point is Kant’s notion of a judgement, with Martin-Löf’s extension to higher-order judgements providing the motivation for the notion of a logical framework. We then proceed to discuss the LF logical framework. All the material in this chapter can be found in the relevant literature. Our method of presentation is slightly unusual in that we start our presentation from the philosophical viewpoint.

6.1 Kant’s Notion of a Judgement

The term *judgement*, in the sense in which we shall use it, was first used by Kant (1800) in his lecture notes on logic. According to analysis by Martin-Löf

(1982), Kant uses the term *urteil* (judgement) instead of proposition. Proposition here meaning the thing that we prove. The origin of the word proposition, used in this sense, comes from the Greek *προτασις* first used by Aristotle in the *Prior Analytics*, the third part of the *Organon*. The result of Kant using the word ‘judgement’ means that in the German philosophical tradition, a *proof* (*beweis*) is always a proof of a judgement. So given the natural deduction inference rule for modus ponens:

$$\frac{\phi \quad \phi \supset \psi}{\psi} MP$$

we can think of the premisses ϕ and $\phi \supset \psi$ as being judgements. The actual judgements, however, are hidden within this presentation, since ϕ , $\phi \supset \psi$ and ψ are not themselves judgements. The judgements are that ϕ , $\phi \supset \psi$ and ψ are true. By true, we mean that we have a proof of each proposition. The proof of ψ is obtained or constructed from the proofs of ϕ and $\phi \supset \psi$. Viewing inference rules in this way leads to the Brouwer- Heyting-Kolmogorov (BHK) interpretation (*cf.* (Brouwer 1924a), (Brouwer 1924b), (Heyting 1934) and (Kolmogorov 1932)). We will write $\text{true}(\phi)$ for the judgement ‘ ϕ is true’. Here we are not concerned which logical system we are working with; for any logical system, a proposition is true whenever there exists a proof of that proposition.

When we explicitly write out the judgements used in a logical system, we call the system a *judged proof system*. In judged proof systems, we will write inference rules with explicit judgements. The natural deduction inference rule for modus ponens thus becomes

$$\frac{\text{true}(\phi) \quad \text{true}(\phi \supset \psi)}{\text{true}(\psi)} MP$$

in a judged proof system. We allow multiple judgements in our systems because some systems have more than one notion of truth. Modal logics; for example, have true and valid formulæ. For more examples of systems with more than one consequence relation, see (Avron 1991).

As we mentioned above, the inference rules are viewed as being constructive. An inference rule takes proofs of its premisses and, from them, constructs a proof of its conclusion. We stress that even though we are thinking of inference rules in a constructive sense, the proofs themselves are not restricted, there is no reason proofs cannot be classical; for example.

6.2 Martin-Löf’s Higher-Order Judgements

In a lecture series in Sienna, Martin-Löf (1982) introduced two higher-order judgements. These higher-order judgements will allow us to describe inference rules in terms of judgements. This presentation is a development of his earlier work on

iterated inductive definitions, (Martin-Löf 1971). The first of these two judgements is the *hypothetical judgement*. This judgement corresponds to logical consequence. A hypothetical judgement is a statement of the form: given a collection of judgements, we can infer another judgement, *i.e.*,

$$J_1, \dots, J_n \vdash K.$$

The link to an inference rule is clear, we can take the premisses to be the judgements on the left hand side of the turnstile and the conclusion to be the judgement on the right hand side. We have, however, only shown an instance of an inference rule is a hypothetical judgement. An inference rule is actually a schema since it holds for all suitable propositions. The second of Martin-Löf's higher-order judgements deals with this quantification.

The second higher-order judgement introduced by Martin-Löf is the *general judgement*, which corresponds to universality. A general judgement is a statement of the form: for all elements of a syntactic category, contained within a judgement the judgement holds; that is, we quantify over all elements within a judgement to obtain a general judgement. Stating this syntactically, we have

$$\Lambda x \in C . J(x)$$

where the Λ is quantification and C is a syntactic category. Clearly, the general judgement will allow us to capture the universal nature of the judgements involved in an inference rule; that is, they hold for all propositions.

We combine these two judgements to obtain the *hypothetico-general judgement* which corresponds directly to an inference rule. A hypothetico-general judgement has the following form:

$$\Lambda x \in C . J_1(x), \dots, J_n(x) \vdash J(x)$$

which can be viewed as a schema. In fact, hypothetico-general judgements are sufficient to derive any inference rule. Returning to the example of modus ponens, we see that

$$\frac{\phi \quad \phi \supset \psi}{\psi} MP$$

corresponds to the hypothetico-general judgement

$$\Lambda \phi, \psi \in \mathbf{Prop} . \mathbf{true}(\phi), \mathbf{true}(\phi \supset \psi) \vdash \mathbf{true}(\psi)$$

where \mathbf{Prop} is the syntactic category of propositions. Modus ponens does not involve discharge, and there is an important point we need to make about rules which involve a discharge of an assumption. A natural deduction proof is a tree with nodes labelled with inference rules and the values of their parame-

ters together with a discharge function. We are not concerned in describing the discharge function but we are interested in how rules involving discharge are described in terms of judgements. When we have a rule involving discharge, the proof involving the assumption is a hypothetical judgement. The following example illustrates this: we take the natural deduction rule for \supset I and describe it in terms of judgement, where the discharged assumption is a nested hypothetical judgement.

$$\frac{\begin{array}{c} [\phi] \\ \vdots \\ \psi \end{array}}{\phi \supset \psi} \supset I$$

corresponds to the hypothetico-general judgement

$$\Lambda \phi, \psi \in \mathbf{Prop}. (\mathbf{true}(\phi) \vdash \mathbf{true}(\psi)) \vdash \mathbf{true}(\phi \supset \psi)$$

where we see that the discharge of ϕ is handled by the nested hypothetical judgement. This example also makes clear that the judgements used in hypothetico-general judgements can themselves be higher-order. We refer to the judgements described by Kant as *basic judgements* to distinguish them from the higher-order judgements of Martin-Löf.

We claimed that hypothetico-general judgements are sufficient to describe any natural deduction inference rule. We justify this claim by showing that hypothetico-general judgements can describe the general natural deduction introduction and elimination rules of (Prawitz 1978). The rule for modus ponens is not in this general form and in fact is a special case which we do not discuss here. A general natural deduction introduction rule is given by schemata of the form

$$\frac{\begin{array}{c} [\psi_{j,1}^i] \cdots [\psi_{j,h_j^i}] \\ \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ \psi_1^i \quad \cdots \quad \psi_j^i \quad \cdots \quad \psi_{p_i}^i \end{array}}{\#(\phi_1, \dots, \phi_n)} \# I$$

for $1 \leq i \leq s$. The general elimination rule is given by the schema

$$\frac{\begin{array}{c} [\Gamma_1] \qquad [\Gamma_s] \\ \vdots \qquad \qquad \qquad \vdots \\ \#(\phi_1, \dots, \phi_n) \quad \psi \quad \cdots \quad \psi \end{array}}{\psi} \# E$$

where each Γ_i is of the form $(\psi_{1,1}^i, \dots, \psi_{i,h_1^i}^i \vdash \psi_1^i), \dots, (\psi_{p_i,1}^i, \dots, \psi_{p_i,h_{p_i}^i}^i \vdash \psi_{p_i}^i)$. The introduction rule is described by the following hypothetico-general judge-

ment:

$$\Lambda \psi_1^i, \dots, \psi_{p_i}^i, \psi_{1,1}^i, \dots, \psi_{p_i, h_{p_i}^i}, \phi_1, \dots, \phi_n \in \mathbf{Prop} . (\mathbf{true}(\psi_{1,1}^i), \dots, \mathbf{true}(\psi_{1, h_i^i}^i))$$

$$\vdash \mathbf{true}(\psi_1^i), \dots, (\mathbf{true}(\psi_{p_i, 1}^i), \dots, \mathbf{true}(\psi_{p_i, h_{p_i}^i}^i)) \vdash \mathbf{true}(\psi_{p_i}^i)) \vdash \#(\phi_1, \dots, \phi_n),$$

while the elimination rule is given by the hypothetico-general judgement:

$$\Lambda \psi, \psi_{1,1}^1, \dots, \psi_{p_s, h_{p_s}^s}, \psi_1, \dots, \psi_n \in \mathbf{Prop} . \mathbf{true}(\#(\phi_1, \dots, \phi_n)), ((\mathbf{true}(\psi_{1,1}^1),$$

$$\dots, \mathbf{true}(\psi_{1, h_1^1}^1)) \vdash \mathbf{true}(\psi_1^1)), \dots, (\mathbf{true}(\psi_{p_1, 1}^1), \dots, \mathbf{true}(\psi_{p_1, h_{p_1}^1}^1)) \vdash \mathbf{true}(\psi_{p_1}^1))$$

$$\vdash \mathbf{true}(\psi)), \dots, ((\mathbf{true}(\psi_{1,1}^s), \dots, \mathbf{true}(\psi_{1, h_1^s}^s)) \vdash \mathbf{true}(\psi_1^s)), \dots, (\mathbf{true}(\psi_{p_s, 1}^s), \dots,$$

$$\mathbf{true}(\psi_{p_s, h_{p_s}^s}^s)) \vdash \mathbf{true}(\psi_{p_s}^s)) \vdash \mathbf{true}(\psi)) \vdash \mathbf{true}(\psi)$$

The rules of Hilbert-type systems can also be expressed in terms of judgements. The axioms are general judgements and the rules are hypothetico-general judgements.

6.3 The Notion of a Logical Framework

A logical framework formalizes the informal discussion above. In order to describe a *logical framework*, we must (Ishtiaq & Pym 2002) have methods of

1. Characterizing the class of (object-)logics to be represented;
2. Describing a meta-logic or language, together with its meta-logical 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:

$$\textit{Framework} = \textit{Language} + \textit{Representation}$$

Our starting point for understanding the representation mechanism is that logical inference rules can be expressed in terms of judgements as we described in the previous section. The *judgements-as-types correspondence* is the formal way of expressing this (informal) observation. We take the judgements-as-types correspondence as our representation mechanism. There are other representation mechanisms: *worlds-as-parameters*, which we discuss in detail in § 9, and *no assumptions* (Avron et al. 1997). For now, we concern ourselves with the judgements-as-types correspondence. To be able to formally express an inference

rule in terms of judgements, we need a language in which to express them. This language will be a typed λ -calculus. To be able to represent hypothetico-general judgements as types, we need a dependently typed λ -calculus. One can use any dependently typed λ -calculus from the λ -cube (Barendregt 1992) as a language provided it is given a signature.

6.4 The LF Logical Framework

In this section, we provide an overview of the LF logical framework. This logical framework is the one presented in (Harper et al. 1987) and (Harper et al. 1993). The language is the dependently typed λ -calculus, $\lambda\Pi$ (*cf.* § 2). In the literature, we find that any logical framework which has the same proof-terms as LF is called LF, regardless of the language and the representation. This can be quite confusing since they are technically different logical frameworks. We will follow the literature in using LF as the name of a family of logical framework, although we will make it clear which representation mechanism and language we are using at any given point.

The main representation mechanism we will use is that of *judgements-as-types*. The two higher-order judgements: the hypothetical, $J_1, \dots, J_n \vdash K$ and the general $\Lambda x \in C. J(x)$, correspond to the ordinary and dependent function spaces respectively. The methodology of judgements-as-types is that judgements are represented by the type of their proofs. A logical system \mathcal{L} is represented by a signature in the type theory which assigns kinds and types to a finite set of constants that represent its syntax, judgements and rule schema. An object-logics rules and proofs can be seen as primitive proofs of hypothetico-general judgements. Hypothetical judgements $J_1, \dots, J_n \vdash K$ are represented by the type $(J_1 \rightarrow \dots \rightarrow J_n) \rightarrow K$ and general judgements $\Lambda x \in C. J(x)$ are represented by the type $\Pi x : C. J$; hypothetico-general judgements are represented by the combination of these two types.

We contend that it is important to formulate the judgements-as-types correspondence in two steps, based on the work in (Avron et al. 1997), – identifiable formally for LF in (Harper et al. 1993) only for particular cases of (classical) first- and higher-order natural deductions – as follows:

1. Consider object-logics as systems for deriving not propositions but rather *judged propositions*, *cf.* § 7;
2. Consider a correspondence between the judged propositions and types in the language of the framework constructed over a signature containing type-constructors corresponding to each judgement of the object-logic, *cf.* § 8.

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

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

$$X, \underbrace{y_1:j_1(\phi_1), \dots, y_n:j_n(\phi_n)}_{\Delta} \vdash_{\mathcal{L}} \delta:j(\phi)$$

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

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

where Γ_X corresponds to the set X of syntactic variables, Γ_{Δ} corresponds to the set Δ of proof-variables, M_{δ} is a $\lambda\Pi$ -term corresponding to the proof-object δ , which we sometimes call a proof-term. We deliberately write the context as $\Gamma_X, \Gamma_{\Delta}$ to emphasize the fact that the two parts have come from the set of syntactic variables and the set of proof-variables.

The propositions-as-types correspondence for the $\{\forall, \supset\}$ -fragment of many sorted minimal first-order logic can be seen as a special case of the judgements-as-types correspondence, where each $j_i(= j) = \mathbf{proof}$. This point of view will become apparent when we look at morphisms between Kripke models in § 9. We will often refer to the propositions-as-types correspondence as a different representation mechanism.

Roughly speaking, LF is concerned with those Hilbert and natural deduction systems for which the correspondence is *uniform*. The basic idea is that an encoding of a logic \mathcal{L} is uniform if there is a surjection from consequences

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

in \mathcal{L} to consequences

$$\Gamma \vdash_{\Sigma_{\mathcal{L}}} M:A$$

in $\Sigma_{\mathcal{L}}$. The term uniform comes from (Harper, Sannella & Tarlecki 1994) but their notion is stronger than ours, requiring quantification over all possible signatures.

One property 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 logical framework. For example, suppose that $\Sigma_{\mathcal{L}}$ is a uniform encoding of \mathcal{L} , and that

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

is the image of the object-consequence

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

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

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

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

is provable, then so is

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

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

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

is then the image of the object-consequence

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

Consequently, LF is unable to uniformly encode relevant or substructural ((Schröder-Heister & Dösen 1993) and (Read 1988)) logics such as intuitionistic linear logic (Girard 1987). A logical framework, also based on the judgements-as-types notion of representation, which is able to uniformly encode intuitionistic linear logic has been presented in (Ishtiaq & Pym 1998). A linear logical framework has also been presented in (Pfenning 2002).

Given a representation in a logical framework, we are interested in the relationship between object-consequence and the encoded-consequence. To ensure that we have encoded the object-consequence, we show that the encoding is adequate. An *adequate encoding* is one in which the encoding does not introduce any additional entities, *i.e.*, *full*, and encodes all entities uniquely, *i.e.*, *faithful*. Issues surrounding representation will be discussed in more detail in § 8.

Chapter 7

LF's Object-Logics

Following on from the previous chapter, we present a characterization of object-logics that are suitable for representation in LF using the judgements-as-types method of representation. We begin this chapter with an extended example which will provide the intuition for our characterization. Object-logics are abstractly characterized by their correct consequences. In our case, this is done via a judged consequence relation. We access the correct consequences of a logic through proof systems and classes of models and satisfaction. Proof systems and classes of models are developed separately, with soundness and completeness results bringing together the two presentations.

The extended example is taken from (Avron et al. 1997). The presentation of a judged proof system builds on known results about proof systems but our use of judgements is original. There is a presentation of a similar system in (Gardner 1992*b*), (Aczel 1980) and (Martin-Löf 1971). The characterization of an object-logic in terms of Kripke models builds on known results in categorical logic and correspondence theory.

7.1 Background

As we have seen, the LF logical framework is intended to provide a formal metatheory for Hilbert-type and natural deduction presentations of logical systems. Indeed, it seems that, (Pfenning 2000), notwithstanding, LF does not provide a suitable metatheory for logical systems based on sequent calculi (Gentzen 1934). We describe representations of sequent calculi in LF in § 12.

Let us suppose, that for some language L , we have a Hilbert-type or natural deduction system \mathcal{L} . The basic idea of a proof in such a system is that of a labelled tree. The labels of the tree are formulæ of \mathcal{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 formulæ which label its successors by one of the inference rules of \mathcal{L} .

Such systems are said to be pure ((Avron 1991) and (Avron et al. 1997)) 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 \Box -rule of the standard Hilbert presentation of $S4$, which requires the global condition that the premiss be a theorem, and \Box -I in Prawitz's (1965) natural deduction system for $S4$, which requires the global condition that all hypotheses have \Box -outermost. In a Hilbert-type system, any rule of proof, *i.e.*, a rule with no side-formulæ, is also an example. Rules with side-formulæ are called rules of derivation. Systems which require such global conditions in order to determine correctness are said to be impure. From the point of view of logical frameworks, impure systems are problematic for at least the following two reasons: firstly, from a structural point of view, the formal description of global conditions may require *ad hoc* additions to either the language or the representation mechanism, or to both; secondly, checking such global conditions can be computationally expensive.

A formula ϕ in \mathcal{L} follows from a set of formulæ Δ (written $\Delta \vdash_{\mathcal{L}} \phi$) if and only if there is a proof-tree Π in which every leaf is labelled by an axiom of \mathcal{L} or by an element of Δ , and the root is labelled by ϕ . Clearly, Δ may contain formulæ which do not label any leaves in Π . A proof Π is a valid proof in \mathcal{L} of ϕ with respect to (X, Δ) , where X is the set of syntactic variables, if $\Delta \vdash_{\mathcal{L}} \Pi : \phi$ and all the free variables in Π are contained in X . We write $(X) \Delta \vdash_{\mathcal{L}} \Pi : \phi$ for Π being a valid proof of ϕ .

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

$$\forall \Gamma_1, \dots, \Gamma_m \frac{\Delta_1(\Gamma_1) \vdash^{i_1} \phi_1 \dots \Delta_m(\Gamma_m) \vdash^{i_m} \phi_m}{(\oplus_{i=1}^m \Gamma_i) \vdash^i \phi} C$$

where each $\Delta_i(\Gamma_i)$ denotes a context, *i.e.*, a collection of bare propositions which includes the components of Γ_i , $(\oplus_{i=1}^m \Gamma_i)$ denotes the combination of the Γ_i s, and C is a possible side-condition, concerning variables or occurrences of modalities. Typically, these side-conditions are global conditions which must be checked at the application of the rule. The $i, i_1, \dots, i_m \in \{1, \dots, n\}$, where $\vdash^1, \dots, \vdash^n$ are the n consequence relations of \mathcal{L} . Viewing a Hilbert-type and natural deduction system in this way means that they are characterized by the focus on assumption-conclusion dependencies rather than theorems. Weakening is also assumed in this

presentation. For further justification of why this presentation of Hilbert-type and natural deduction systems is an appropriate choice see (Avron et al. 1997).

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

Our subsequent informal presentation applies to Hilbert-type presentations of minimal, intuitionistic and higher-order predicate logics and minor variations thereon, as well as to the family K , KT , $KT4$ (or $S4$), $KT45$ (or $S5$) and KL of modal logics as discussed in (Avron et al. 1997). It also applies to the following systems from (Avron, Honsell, Mason & Pollack 1992) 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. An abstract definition of the class of systems we consider is provided in § 7.2 and § 7.3.

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

$$\frac{\phi}{\Box\phi} \Box \phi \text{ depends on no assumptions}$$

Here the condition that ϕ depends on no assumptions is a global one on the Hilbert-type proof of ϕ , thus rendering the rule impure. To check that the rule \Box has been used correctly, we must check that all the formulæ 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 and thus remove the impurity. The presence of two judgements allows us to separate the rules of \mathcal{L} into two groups. The first group contains the usual rules of classical logic and allows the inference just of propositions judged **true**. The second system consists of the rules for \Box which can be used to derive valid propositions only in valid contexts. All axioms are judged **valid**.

true and **valid** are symbols with a propositional arity, so that the pairs $\langle\phi, \mathbf{true}\rangle$ and $\langle\phi, \mathbf{valid}\rangle$ are formulæ of judged \mathcal{L} . The nodes of a proof in judged \mathcal{L} are thus labelled with such pairs. Following (Avron et al. 1997), we say that such a tree is a judged \mathcal{L} -proof if the following conditions hold:

- The tree is a legitimate proof-tree in the system \mathcal{L}' , which is obtained from \mathcal{L} by transforming all rules of proof into rules of derivation (by adding side-formulæ in the obvious way);
- A node which is not a leaf is labelled **valid** if and only if all 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 second group of rules includes the following judged version of the \Box -rule:

$$\frac{\langle \phi, \text{valid} \rangle}{\langle \Box \phi, \text{valid} \rangle} \Box$$

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

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

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 7.1 (Judged Systems (Avron et al. 1997))

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 the leaves which are not axioms are labelled **true**. ■

The formulation of judged systems is such that their metatheory is rather simple and elegant. Specifically, we ensure the purity of proof systems by allowing, in a judged consequence relation, several consequence relations to be treated simultaneously. A proof system with multiple consequence relations is non-uniform according to Avron (1991). Judged systems are uniform because they only have one (judged) consequence relation. 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 $2j(\mathcal{H})$ stand for the family $K, KT, K4, KT4$ (or $S4$), $KT45$ (or $S5$) and KL , formulated as judged Hilbert-type systems.

Theorem 7.2 (Encoding Judged \mathcal{L} (Avron et al. 1997))

For \mathcal{L} one of the family $2j(\mathcal{H})$, there is a compositional bijection between \mathcal{L} -proofs of

$$(X) \langle \phi_1, l_1 \rangle, \dots, \langle \phi_m, l_m \rangle \vdash_{\mathcal{L}} \Pi : \langle \phi, l \rangle$$

and $\lambda\Pi$ -terms M in long $\beta\eta$ -normal form over the signature $\Sigma_{\mathcal{L}}$ such that

$$\Gamma_X, \gamma_V(\Delta), \gamma_t(\Xi) \vdash_{\mathcal{L}} M : j(\epsilon(\phi))$$

where

- Γ_X is the $\lambda\Pi$ -context which corresponds to X under judgements-as-types;
- $\gamma_v(\Delta)$ is the $\lambda\Pi$ -context which corresponds to $\Delta = \{\phi_i | i = \text{valid}\}$ under judgements-as-types;
- $\gamma_t(\Xi)$ is the $\lambda\Pi$ -context which corresponds to $\Xi = \{\phi_i | i = \text{true}\}$ under judgements-as-types;
- j is valid if $l = \text{valid}$, true otherwise;
- $\epsilon(\phi)$ is the encoding of ϕ under judgements-as-types. ■

Corollary 7.3 (Encoding \mathcal{L} and \mathcal{L}' (Avron et al. 1997))

Suppose $\phi_1, \dots, \phi_m, \phi$ are well-formed formulæ of \mathcal{L} with free variables in X .

1. There is a compositional bijection between proofs in \mathcal{L}' of $\phi_1, \dots, \phi_m \vdash \phi$ and $\lambda\Pi$ -terms M in long $\beta\eta$ -normal form such that

$$\Gamma_X, \Gamma_v(\{\phi_1, \dots, \phi_m\}) \vdash_{\Sigma_{\mathcal{L}}} M : \text{valid}(\epsilon(\phi)).$$

2. There is a compositional bijection between proofs in \mathcal{L} of $\phi_1, \dots, \phi_m \vdash \phi$ and $\lambda\Pi$ -terms in long $\beta\eta$ -normal form such that

$$\Gamma_X, \Gamma_t(\{\phi_1, \dots, \phi_m\}) \vdash_{\Sigma_{\mathcal{L}}} M : j(\epsilon(\phi))$$

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

The pattern of encoding judged systems demonstrated informally in this section can be developed to give a formal theory of representation in the LF logical framework. In the sequel, we provide such a theory, beginning, in the next section with the theory of judged object-logics.

Before embarking upon our technical development of judged logics, it will be useful to consider, informally, a range of examples of normative formulation 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 (Avron et al. 1997) for background.

1. Proof-trees labelled with multiple judgements, encoded using the judgements-as-types representation mechanism. For example:
 - Hilbert-type formulations, for both truth and validity, of the modal systems, such as KT , $K4$, $KT4$, $KT5$, 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.
2. Proof-trees labelled (degenerately) with a single 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 representation of global side-conditions found in rules of proof, such as “no assumptions”, by transferring them to meta-logical conditions such as “no free variables”. The details can be found in (Avron et al. 1997).
 - Natural deduction formulations, for truth, of K , $K4$, KT , $S4$, *etc.*
 3. Proof-trees labelled degenerately with a single logical judgement but with additional, syntactic judgements, such as “closed assumptions”, “boxed assumptions” or “boxed fringe” (Avron et al. 1997), encoded using a representation mechanism that is a variant on judgements-as-types in which types which encode a proposition correspond not to propositions that have first been judged logically and then judged structurally.

7.2 The Theory of Judged Object-logics

In this section, we provide a proof-theoretic account of an object-logic. Following on from the previous section, and our general purpose, we provide an account of judged object-logics.

We begin by defining the language L , consisting of an alphabet, syntactic categories, expressions and judgements. We will interleave the formal account with a presentation of classical logic as a judged proof system. We will provide further examples at the end of this section.

Definition 7.4 (Alphabet)

An *alphabet* is a quintuple $A = (S, V, E, C, J)$ of symbols defined 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;
- $C \subset E$ is a distinguished subset of E which contains *connectives*;
- J is a finite set of *judgements* symbols. ■

We remark that the term “connective” in this definition should be interpreted broadly; for example, the assignment operator $:=$ of Hoare’s logic (Avron et al. 1992) should be considered a member of this class of symbols. It is also worth noting that the (natural number) arities assigned to each symbol are different from the arities we shall shortly assign.

The alphabet for classical logic is the following:

- $S = \{\iota, o\}$, where ι and o have 0 arity;
- $V = \{\iota\}$;
- $E = \{f_1, \dots, f_m, P_1, \dots, P_m, c_1, \dots, c_q, \forall, \exists, \neg, \supset, \wedge, \vee, \perp\}$;
- $C = \{\forall, \exists, \neg, \supset, \wedge, \vee\}$;
- $J = \{\text{true}\}$.

f_i are function symbols, P_j are predicate symbols and c_r are constant symbols.

Definition 7.5 (Syntactic Categories)

Let $A = (S, V, E, C, J)$ be an alphabet. The *syntactic generated by A* are inductively defined as follows:

- The nullary symbols are syntactic categories;
- Let c_1, \dots, c_m be syntactic categories and $s \in S$ be any m -ary symbol, then $sc_1 \dots c_m$ 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 $\{o_1, \dots, o_m\}$ as the *syntactic categories of propositions*. ■

The syntactic categories of classical logic are ι and o , where ι is a syntactic category containing variables and o is a syntactic category of propositions.

Following (Martin-Löf 1971), (Aczel 1980) and (Gardner 1992a), we define the expressions of our logical syntax via a notion of arity.

Definition 7.6 (Arities and Levels)

An *arity* a is of the form $(a_0, \dots, a_m) \rightarrow s$, where, for $0 \leq i \leq m$, a_i 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, \dots, a_m as the *domain arities* of a . We refer to a as the *range* arity. ■

Each n -ary connective $\# \in C$ has an arity

$$(a_1, \dots, a_n) \rightarrow o,$$

where o is one of the distinguished categories of propositions. We call any expression $e \in E/C$ whose range arity is one of the syntactic categories of propositions a *predicate letter* and any expression $e \in E/C$, whose range arity is not one of the syntactic categories of propositions a *function symbol*. Function symbols which do not have domain arities are called *constant symbols*.

In classical logic, the connectives \supset, \wedge and \vee all have arity $(o, o) \rightarrow o$ and level 1, \neg has arity $o \rightarrow o$ and level 1, \forall and \exists have arity $(\iota \rightarrow o) \rightarrow o$ and level 2.

The function symbols f_i have arity $\underbrace{(\iota, \dots, \iota)}_{\iota_i\text{-times}} \rightarrow \iota$ and level 1, the predicate symbols P_j have arity $\underbrace{(\iota, \dots, \iota)}_{k_j\text{-times}} \rightarrow o$ and level 1, the constant symbols c_r have arity ι and level 1 and \perp has arity o and level 1.

We now define the set of expressions generated by an alphabet. We assume that each syntactic category s containing variables has an associated countable set of variables V_s .

Definition 7.7 (Expressions)

Let $A = (S, V, E, C, J)$ be an alphabet. The set of *expressions generated by A* is defined as follows:

variables: If $x \in V_s$, then x is an expression with arity s ;

applications: If $e \in E$ with arity $(a_1, \dots, a_n) \rightarrow s$ of level $l \leq 2$ and if e_1, \dots, e_m are expressions of arity a_1, \dots, a_m respectively, then $ee_1 \dots e_m$ is an expression with arity s ;

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

The *logical expressions* are those expressions with level 0 arity. 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 distinguished syntactic categories of propositions. ■

We illustrate the use of application with an example from classical logic. Take the connectives \wedge with arity $(o, o) \rightarrow o$ and two expressions ϕ and ψ both with arity o . Application gives the proposition expression $\wedge\phi\psi$, usually written $\phi \wedge \psi$.

Abstraction allows us to handle quantifiers correctly and any other connectives whose level is 2. Given a proposition ϕ with arity o and a variable x with arity ι ,

we can abstract x to obtain $(x)\phi$ with arity $\iota \rightarrow o$. Applying \forall yields the logical expression $\forall(x)\phi$, usually written $\forall x . \phi$.

Substitution can be defined in terms of abstraction. Given a logical expression ϕ , we can abstract all the free variables we wish to substitute, *i.e.*, we have $(x_1, \dots, x_n)\phi$ and then apply variables y_1, \dots, y_n to obtain $\phi[x_i/y_i]_{i=1}^m$.

In classical logic the term expressions are the terms of classical logic and the proposition expressions are the propositions of classical logic.

An alphabet includes a set of judgement symbols J . We take each judgement symbol $j \in J$ to be equipped with an arity of the form (o_1, \dots, o_m) , where each o_i is a distinguished syntactic category of propositions. This allows us to form first-order or basic judgements.

Definition 7.8 (Basic Judgements)

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

$$\{j(e_1, \dots, e_m) \mid j \in J \text{ has arity } (o_1, \dots, o_m) \text{ and, for } 1 \leq i \leq m, \text{ each } e_i \text{ is a} \\ \text{proposition expression inhabiting } o_i\}$$

For each distinguished syntactic category of propositions o , we assume the existence of a nullary judgement **null** of arity o . ■

In classical logic, we have the judgement symbol **true** of arity o . Thus the set of basic judgements generated by A is just the set of proposition expressions of A labelled by **true**.

The term basic judgement derives from the work of Martin-Löf (1982) on the meanings of logical constants and rules of inference, which in turn derives from Kant (1800). We let J, K and L , possibly subscripted, range over basic judgements. Martin-Löf further constructs the general judgements, of the form $\Lambda x : C . K$, where C is a syntactic category, and the hypothetical judgements, of the form $J \vdash K$. These higher-order judgements such be read, respectively, as the universal and implication formulæ of a metalogic which has basic judgements as its atomic formulæ. This metalogic is non-other than the internal logic of the language (the $\lambda\Pi$ -calculus) of the LF logical framework. In this meta-logic, we can follow Martin-Löf and define the hypothetico-general judgements, of the Horn form $\Lambda x_1 : C_1 . \dots . \Lambda x_m : C_m . J_1, \dots, J_n \vdash K$, which can be read as meta-logical definitions of (Hilbert-type and natural deduction) inference rules.

Definitions 7.4 to 7.8 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 7.9 (Judged Consequence Relation)

Let $A = (S, V, E, C, J)$ be an alphabet. A *judged consequence relation over A* is

a pair (J, \vdash) , where J is a set of basic judgements over A and $\vdash \subset \wp_f(J) \times J$ is a binary relation such that:

Reflexivity: $J \vdash J$, for every every basic judgement $J \in J$;

Transitivity (cut): If $\Delta \vdash J$ and $\Delta, J, \Delta' \vdash K$ then, $\Delta, \Delta' \vdash K$, for each $\Delta, \Delta' \in \wp_f(J)$ and $J, K \in J$;

Weakening: If $\Delta \vdash J$, then $\Delta, \Delta' \vdash J$, for each $\Delta, \Delta' \in \wp_f(J)$ and each $J \in J$.

A judged consequence relation (J, \vdash) is said to be *permutating* if $\Delta, J, K, \Delta' \vdash L$ implies $\Delta, K, J, \Delta' \vdash L$, for each $\Delta, \Delta' \in \wp_f(J)$ and each $J, K, L \in J$. ■

The judged consequence relation for classical logic is the consequence relation determined by either the Hilbert-type rules or natural deduction rules for classical logic where each proposition is labelled with the judgement **true**.

Judged consequence relations provide an abstract characterization of the correct consequences of a logical system. Access to consequence relations is provided either through a class of models and satisfaction relations, the topic of § 7.3, 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. LF is not a suitable meta-theory for sequent systems because there is not a natural relationship between them and lambda-calculi, unlike natural deduction systems where the propositions-as-types correspondence holds. Before we turn to the definition of the inference rules for these systems, a more detailed analysis of purity is required. We recall that an inference rule is pure when the check required by $(*)$ is local. Following (Avron 1991), we distinguish three different levels of impurity:

Level 1: The side-conditions are related to the structure of the side-formulæ.

For example, the condition that there are no side-formulæ; that is, rules of proof in Hilbert-type systems;

Level 2: The applicability of the rule depends also on the structure of side-formulæ. For example, the introduction rule for \Box in Prawitz's (1965) natural deduction system for $S4$. The rule says that we can deduce $\Box\phi$ only if all the assumptions of ϕ are boxed;

Level 3: The applicability of the rule depends not only on its potential premisses, but also on their proofs.

The first two levels of impurity can be dealt with by introducing multiple judgements, the third, however, cannot. The reason for this is that the third level breaks uniformity. A system is uniform when it treats exactly one consequence relation, and once a judged proposition has been derived we do not need to know

how it was derived to use it in other inference rules. The third level of impurity breaks this condition because we need to keep track of proofs and are only able to apply certain rules if their proofs satisfy certain conditions. Uniformity is a much more general condition than purity and since we can handle impure rules if the system is uniform; only uniform systems are suitable for representation in LF.

Definition 7.10 (Hilbert-type Systems)

Let $A = (S, V, E, C, J)$ be an alphabet and \vdash be a judged consequence relation over A . A *Hilbert-type system* for \vdash is given by the following:

- A set of *axioms* $A \subset J$;
- A set of *rules* of the form

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

where, for $1 \leq i \leq n$, $J_i \in J$ and $J \in J$.

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

Classical logic can be presented as a Hilbert-type system with axioms:

- $\text{true}(\phi \supset (\psi \supset \phi))$
- $\text{true}((\phi \supset (\psi \supset \tau)) \supset ((\phi \supset \psi) \supset (\phi \supset \tau)))$
- $\text{true}(\phi \supset (\phi \vee \psi))$
- $\text{true}(\psi \supset (\phi \vee \psi))$
- $\text{true}((\phi \supset \tau) \supset ((\psi \supset \tau) \supset (\phi \vee \psi) \supset \tau))$
- $\text{true}((\phi \wedge \psi) \supset \phi)$
- $\text{true}((\phi \wedge \psi) \supset \psi)$
- $\text{true}(\phi \supset (\psi \supset (\phi \wedge \psi)))$
- $\text{true}(\forall x \phi \supset \phi[x/t])$
- $\text{true}(\phi[x/t] \supset \exists x \phi)$
- $\text{true}(\forall x(\phi \supset \psi) \supset (\psi \supset \forall y \phi[x/y]))$
- $\text{true}(\forall x(\phi \supset \psi) \supset (\exists y \phi[x/y] \supset \psi))$
- $\text{true}(\perp \supset \psi)$
- $\text{true}(\neg\neg\phi \supset \psi)$

and rules:

$$\frac{\text{true}(\phi) \quad \text{true}(\phi \supset \psi)}{\text{true}(\psi)} MP$$

$$\frac{\text{true}(\phi)}{\text{true}(\forall y \phi[x/y])}$$

Definition 7.11 (Natural Deduction Systems)

Let $A = (S, V, E, C, J)$ be an alphabet and \vdash be a judged consequence relation over A . A *natural deduction system* for \vdash is given by the following:

- A set of *axioms* $A \subset J$;
- For each connective $\# \in C$, an *introduction rule* given by schemata of the form

$$\frac{\begin{array}{ccc} [K_{j,1}^i] \cdots [K_{j,h_j^i}] \\ \vdots & \vdots & \vdots \\ J_1^i & J_j^i & J_{p_i}^i \end{array}}{J} \# I$$

for $i = 1, \dots, s$, where $K_{j,1}^i, \dots, K_{j,h_j^i}^i, J_j^i$, for $1 \leq i \leq p_i$, and $J = j(\#(e_1, \dots, e_n))$ are all basic judgements and e_i is of the form $e_i(\phi_1, \dots, \phi_m)$. The inference infers a basic judgement $j(e_1, \dots, e_n)$ from p_i premisses J_1, \dots, J_{p_i} and can bind assumptions of the form $K_{j,1}^i, \dots, K_{j,h_j^i}^i$ that occur above the j^{th} premiss;

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

$$\frac{\begin{array}{ccc} [\Gamma_1] & \cdots & [\Gamma_s] \\ \vdots & & \vdots \\ k(\#(e_1, \dots, e_n)) & J & J \end{array}}{J} \# E$$

with s minor premisses of the form J , a basic judgement, and

$$\Gamma_i = \bigwedge_{k=1}^{h_1^i} H_{1,k}^i \supset G_1^i, \dots, \bigwedge_{k=1}^{h_{p_i}^i} H_{p_i,k}^i \supset G_{p_i}^i$$

where \bigwedge stands for iterated conjunction;

- If $\# \in C$ and $\# = \supset$, then we have an *elimination rule* given by the schema

$$\frac{\begin{array}{ccc} [J(\phi \supset \psi)] & & [J(\psi)] \\ \vdots & \vdots & \vdots \\ & J(\phi) & J(\tau) \end{array}}{J(\tau)} \supset E$$

- *Judgement rules* of the form

$$\frac{J(\phi)}{K(\phi)}$$

where $J(\phi)$ and $K(\phi)$ are basic judgements.

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

The general non-judged introduction and elimination rules can be found in (Prawitz 1978), however, his schema for $\supset E$,

$$\frac{\begin{array}{c} [\mathbf{J}(\phi) \supset \mathbf{J}(\psi)] \\ \vdots \\ \mathbf{J}(\phi \supset \psi) \end{array} \quad \begin{array}{c} \mathbf{J}(\tau) \\ \supset E \end{array}}{\mathbf{J}(\tau)} \supset E$$

has no deductive force as pointed out in (Schroeder-Heister 1982). There is also a typo in the formulation of the general schema in (Prawitz 1978). This is corrected in (Prawitz 1982). I am indebted to Prof. Prawitz for drawing my attention to this, and the lack of deductive force in $\supset E$, (Prawitz 2008). The general elimination rule forces us to have implication in the system whenever there is a rule involving discharge and conjunction whenever we have a rule with more than one premiss.

The natural deduction system for classical logic has the following axiom:

$$\mathbf{true}(\phi \vee \neg\phi)$$

and rules:

$$\frac{\mathbf{true}(\phi) \quad \mathbf{true}(\psi)}{\mathbf{true}(\phi \wedge \psi)} \wedge I$$

$$\frac{\begin{array}{c} [\mathbf{true}(\phi \wedge \psi)] \\ \vdots \\ \mathbf{true}(\phi \wedge \psi) \end{array} \quad \mathbf{true}(\tau)}{\mathbf{true}(\tau)} \wedge E$$

$$\frac{\mathbf{true}(\phi)}{\mathbf{true}(\phi \vee \psi)} \vee I_1 \quad \frac{\mathbf{true}(\psi)}{\mathbf{true}(\phi \vee \psi)} \vee I_2$$

$$\frac{\begin{array}{c} [\mathbf{true}(\phi)] \\ \vdots \\ \mathbf{true}(\tau) \end{array} \quad \begin{array}{c} [\mathbf{true}(\psi)] \\ \vdots \\ \mathbf{true}(\tau) \end{array}}{\mathbf{true}(\tau)} \vee E$$

$$\frac{\begin{array}{c} [\mathbf{true}(\phi)] \\ \vdots \\ \mathbf{true}(\psi) \end{array}}{\mathbf{true}(\phi \supset \psi)} \supset I$$

$$\begin{array}{c}
\text{true}(\phi \supset \psi) \quad \text{true}(\phi) \quad \begin{array}{c} [\text{true}(\psi)] \\ \vdots \\ \text{true}(\tau) \end{array} \\
\hline
\text{true}(\tau) \supset E \\
\begin{array}{c} [\text{true}(\phi)] \\ \vdots \\ \text{true}(\perp) \end{array} \\
\hline
\text{true}(\neg\phi) \neg I \\
\begin{array}{c} [\text{true}(\phi) \supset \text{true}(\perp)] \\ \vdots \\ \text{true}(\tau) \end{array} \\
\hline
\text{true}(\neg\phi) \quad \text{true}(\tau) \neg E \\
\hline
\text{true}(\tau) \\
\text{true}(\phi) \\
\hline
\text{true}(\forall x\phi[x/a]) \forall I
\end{array}$$

where a must not occur in any assumption on which ϕ depends.

$$\begin{array}{c}
\begin{array}{c} [\text{true}(\phi)] \\ \vdots \\ \text{true}(\tau) \end{array} \\
\hline
\text{true}(\forall x\phi) \quad \text{true}(\tau) \forall E \\
\hline
\text{true}(\tau) \\
\text{true}(\phi[t/x]) \\
\hline
\text{true}(\exists x\phi) \exists I \\
\begin{array}{c} [\text{true}(\phi[a/x])] \\ \vdots \\ \text{true}(\tau) \end{array} \\
\hline
\text{true}(\exists x\phi) \quad \text{true}(\tau) \exists E \\
\hline
\text{true}(\tau)
\end{array}$$

where a must not occur in $\exists x\phi$, in τ , or in any assumption on which the upper occurrence of ϕ depends other than $\phi[a/x]$.

In our definition of a natural deduction system, the elimination rule for each connective “comes for free”, in the sense that an introduction rule provides all the information needed to define the corresponding elimination rule. This corresponds to the philosophical understanding of logical connectives; the meaning of a logical connective is given by its use, *cf.* (Sundholm 2001). For example, Gentzen (1934) suggested that “it should be possible to display the elimination rules as unique functions of the corresponding introduction rules on the basis of requirements.” Prior (1960) showed that if the meaning of a logical connective

depended on both its introduction and elimination rules then we obtain examples of the “*tonk*” form, which causes the system to become inconsistent.

We require that the introduction and elimination rules be, in a suitable sense, inverses of one another (*cf.* (Prawitz 1965) and (Lorenzen 1955)). We need to show that we cannot obtain examples of the “*tonk*” form. Given the introduction and elimination rules for *tonk*:

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

it follows that, for all ϕ and ψ , $\phi \vdash \psi$. To avoid, *tonk*, we require that our system has the local reduction property (Restall 2008).

Definition 7.12 (Local Reduction Property)

Let A be an alphabet, \vdash be a judged consequence relation over A and let \mathcal{L} be a natural deduction system \mathcal{L} for \vdash over A . Given a proof Π in \mathcal{L} 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 premiss, then the introduction and elimination rule can be eliminated, leading to a more direct (*i.e.*, shorter) proof of the conclusion. ■

This corresponds to the philosophical idea found in (Hodges 2001); avoiding *tonk* means that we can infer a formula from what we necessarily had to know to infer the formula.

We show why a system with *tonk* fails to have the local reduction property. Assume we have a proof containing an application of *tonk I* followed immediately by *tonk E*.

$$\frac{\begin{array}{c} \vdots \\ \phi \end{array}}{\phi \text{ tonk } \psi} \text{ tonk } I \\ \frac{\phi \text{ tonk } \psi}{\psi} \text{ tonk } E \\ \vdots$$

It is impossible to eliminate this step because the only way we can prove ϕ from ψ is to use the *tonk* introduction and elimination rules.

The introduction and elimination rules for $\#$ can be reduced in the following

way:

$$\begin{array}{c}
 [\mathbf{K}_{j,1}^i] \cdots [\mathbf{K}_{j,h_j^i}^i] \\
 \vdots \Pi_1^i \quad \quad \quad \vdots \Pi_j^i \quad \quad \quad \vdots \Pi_{p_i}^i \quad [\Gamma_1] \quad \quad [\Gamma_r] \\
 \mathbf{J}_1^i \quad \quad \quad \mathbf{J}_j^i \quad \quad \quad \mathbf{J}_{p_i}^i \# I \quad \quad \quad \vdots \delta_1 \quad \quad \quad \vdots \delta_r \\
 \hline
 \mathbf{j}(\#(e_1, \dots, e_n)) \quad \quad \quad \mathbf{J} \quad \quad \quad \dots \quad \quad \quad \mathbf{J} \# E \\
 \hline
 \mathbf{J}
 \end{array}$$

can be reduced to

$$\begin{array}{c}
 \Pi_1^i \cdot \dots \cdot \Pi_{p_i}^i \\
 \vdots \delta_i \\
 \mathbf{J}
 \end{array}$$

where we remove the cancellations of Γ_i in δ_i and all the assumptions are no longer discharged. This reduction means that our rules satisfy the local reduction property and so, provided $\mathbf{j} = \mathbf{k}$ for a given pair of introduction and elimination rules, we have a consistent system.

Definition 7.13 (Judged Proof Systems)

Given an alphabet A and a judged consequence relation \vdash , the *judged proof system* \mathcal{L} is a Hilbert-type or natural deduction system for \vdash over A . ■

We need to provide a rigorous account of how proofs behave in a natural deduction system for \vdash over A . We do this by introducing proof-objects which realize consequence, *cf.* (Harper et al. 1993). Natural deduction proofs are trees with nodes labelled by inference rules and the value of their parameters, together with a discharge function. The discharge function assigns rule occurrences to hypothesis occurrences and specifies which hypothesis occurrences are discharged by which rule occurrence. The usual way of including the information contained in the discharge function is to number each application of a rule which involves discharge and then number (with the same number) the hypotheses discharged by that rule. A *valid proof* is a proof in which every rule occurrence is a correct instance of the rule scheme that labels it. In particular, if the side-conditions on the applicability of the rule are satisfied.

In our presentation, we have employed the parenthesis convention, where discharged hypotheses are written in square brackets. The square brackets around a judged proposition indicate that zero or more occurrences of that hypothesis are discharged by an application of the rule. The failing of this presentation, is that we do not know what, if any, hypotheses are discharged by the rule, this information is only present in the proof by means of a discharge function.

To be able to represent a natural deduction proof in a logical framework, we need to know which hypotheses and how many are discharged by a given rule, so we need to include this information in a proof. A discharge function is

necessitated by the fact that it is hypothesis occurrences rather than a hypothesis which is discharged. We can have a valid proof which contains an inference rule which discharges $j(\phi)$ but still leaves certain occurrences of $j(\phi)$ undischarged. To this end, we provide a formal account of valid proof expressions, which we often call proof-objects or realizers of consequence.

We introduce *proof-variables* $y : j(\phi)$, where y is the proof-variable for $j(\phi)$. We allow countably many proof-variables which are distinct from the set of first-order variables. These proof-variables will play the role of *occurrence markers*. When a hypothesis is discharged, a certain set of its occurrences is discharged and these are all marked by a particular y . We now define *proof-expressions* Π by the grammar:

$$\Pi ::= HYP_{\phi}(y) \mid \#-I(\Pi_1, \dots, \Pi_n) \mid \#-E(\Pi_1, \dots, \Pi_n)$$

where $HYP_{\phi}(y)$ indicates the use of a hypothesis $y : j(\phi)$ and there are constants corresponding to introduction and elimination rules for each connective $\# \in C$. We use $HYP_{\phi}(y)$ rather than y to ensure consistency between our general definition of a proof-expression and that given in (Harper et al. 1993). We subscript $\#-I$ and $\#-E$ by the judged formulæ and variables used in the corresponding introduction and elimination rules.

The proof-expressions for classical logic are given by the grammar:

$$\begin{aligned} \Pi ::= & HYP_{\phi}(y) \mid AND-I_{\phi,\psi}(\Pi_1, \Pi_2) \mid AND-E_{\phi,\psi,\tau}(\Pi_1, (y_1, y_2) : \Pi) \mid \\ & OR-I_{\phi,\psi}(\Pi_i) \mid OR-E_{\phi,\psi,\tau}(\Pi, y_1 : \Pi_1, y_2 : \Pi_2) \mid IMP-I_{\phi,\psi}(y : \Pi) \mid \\ & IMP-E_{\phi,\psi,\tau}(\Pi, y : \Pi') \mid NEG-I_{\phi}(y : \Pi) \mid NEG-E_{\phi,\psi}(\Pi, y : \Pi') \mid ALL-I_{x,\phi}(\Pi) \mid \\ & ALL-E_{x,\phi,\psi}(\Pi, y : \Pi') \mid SOME-I_{x,t,\phi}(\Pi) \mid SOME-E_{x,\phi,\psi}(\Pi, y : \Pi') \end{aligned}$$

where $(y_1, y_2) : \Pi$ stands for the discharge of the hypotheses whose occurrences are marked by y_1, y_2 .

Not all proof-expressions are valid, so we need rules to generate them. A *proof-context* is a pair (X, Δ) , where X is a finite set of variables and Δ is a set of proof-variables, $y : j_1(\phi_1), \dots, y_n : j_n(\phi_n)$, where each y_i is distinct. We write $FV(\Delta)$ for the set of free-variables occurring in Δ and $dom(\Delta)$ for the set of y_i 's occurring in Δ . We give the rules for proving assertions of the form $(X) \Delta \vdash \Pi : j(\phi)$, which is read as ‘ Π is a valid proof of $j(\phi)$ with respect to the proof-context (X, Δ) .’ The derivability of such an assertion means that a number of general rules are obeyed (*e.g.*, no two occurrences of a y_i mark different judged formulæ) and that the restrictions in the rules for valid proof-expressions are obeyed. We say that Π is a valid proof with respect to the proof-context (X, Δ) if and only if $(X) \Delta \vdash \Pi : j(\phi)$ holds for some $j(\phi)$. There is a minimal such proof-context, if it exists at all: take X to be the set of all free variables in Π , take Δ

to be a list of the set of all the $y:j(\phi)$ such that $HYP_\phi(y)$ is a subexpression of Π .

Validity is preserved under substitution, in the sense that, if $\Pi[x_1, \dots, x_m, y_1:j_1(\phi_1), \dots, y_n:j_n(\phi_n)]$ is a valid proof of $j(\phi[x_1, \dots, x_m])$ with respect to $(\{x_1, \dots, x_m\}, \{y_1:j_1(\phi_1), \dots, y_n:j_n(\phi_n)\})$, if t_1, \dots, t_m are terms whose free variables are all in X' and if $\Delta' = \{y_i:j_i(\phi_i) \mid HYP_{\phi_i}(y_i) \text{ is a subexpression of } \Pi_i, 1 \leq i \leq n\}$ and if Π_1, \dots, Π_n are valid proofs of $j_1(\phi_1), \dots, j_n(\phi_n)$ with respect to (X', Δ') , then $\Pi[t_1, \dots, t_m, \Pi_1, \dots, \Pi_n]$ is a valid proof of $j(\phi[t_1, \dots, t_m])$ with respect to (X', Δ') .

The rules for generating valid proof-expressions for classical logic are the following:

$$\frac{y:\text{true}(\phi) \in \Delta}{(X) \Delta \vdash HYP_\phi(y):\text{true}(\phi)} v\text{-HYP}$$

$$\frac{(X) \Delta \vdash \Pi:\text{true}(\phi) \quad (X) \Delta \vdash \Pi':\text{true}(\psi)}{(X) \Delta \vdash AND\text{-}I_{\phi,\psi}(\Pi, \Pi'):\text{true}(\phi \wedge \psi)} v\text{-AND}\text{-}I$$

$$\frac{(X) \Delta \vdash \Pi:\text{true}(\phi \wedge \psi) \quad (X) \Delta, y_1:\text{true}(\phi), y_2:\text{true}(\psi) \vdash \Pi':\text{true}(\tau)}{(X) \Delta \vdash AND\text{-}E_{\phi,\psi,\tau}(\Pi, (y_1, y_2):\Pi'):\text{true}(\tau)} v\text{-AND}\text{-}E$$

where $y_1, y_2 \notin \text{dom}(\Delta)$.

$$\frac{(X) \Delta \vdash \Pi_i:\text{true}(\phi_1)}{(X) \Delta \vdash OR\text{-}I_{\phi_1,\phi_2}(\Pi_i):\text{true}(\phi_i)} v\text{-OR}\text{-}I_i$$

for $i = 1, 2$.

$$\frac{(X) \Delta \vdash \Pi:\text{true}(\phi_1 \vee \phi_2) \quad ((X) \Delta, y_i:\text{true}(\phi_i) \vdash \Pi_i:\text{true}(\tau))_{i=1,2}}{(X) \Delta \vdash OR\text{-}E_{\phi,\psi,\tau}(\Pi, y_1:\Pi_1, y_2:\Pi_2):\text{true}(\tau)} v\text{-OR}\text{-}E$$

where $y_1, y_2 \notin \text{dom}(\Delta)$.

$$\frac{(X) \Delta, y:\text{true}(\phi) \vdash \Pi:\text{true}(\psi)}{(X) \Delta \vdash IMP\text{-}I_{\phi,\psi}(y:\Pi):\text{true}(\phi \supset \psi)} v\text{-IMP}\text{-}I$$

$$\frac{(X) \Delta \vdash \Pi:\text{true}(\phi \supset \psi) \quad (X) \Delta \vdash \Pi':\text{true}(\phi) \quad (X) \Delta, y:\text{true}(\psi) \vdash \Pi'':\text{true}(\tau)}{(X) \Delta \vdash IMP\text{-}E(\Pi, \Pi', y:\Pi''):\text{true}(\tau)}$$

where $y, y_1 \notin \text{dom}(\Delta)$.

$$\frac{(X) \Delta, y:\text{true}(\phi) \vdash \Pi:\text{true}(\perp)}{(X) \Delta \vdash NEG\text{-}I_\phi(y:\Pi):\text{true}(\neg\phi)} v\text{-NEG}\text{-}I$$

$$\frac{(X) \Delta \vdash \Pi : \text{true}(\neg\phi) \quad (X) \Delta, y : ((X) \Delta, y_1 : \text{true}(\phi) \vdash \Pi_1 : \text{true}(\perp)) \vdash \Pi' : \text{true}(\tau)}{(X) \Delta \vdash \text{NEG-}E_{\phi, \tau}(\Pi, y : \Pi') : \text{true}(\tau)}$$

$$\frac{(X, x) \Delta \vdash \Pi : \text{true}(\phi)}{(X) \Delta \vdash \text{ALL-}E_{x, \phi}(\Pi) : \text{true}(\forall x \phi)} \quad v\text{-ALL-I}$$

where $x \notin X$.

$$\frac{(X) \Delta \vdash \Pi : \text{true}(\forall x \phi) \quad (X) \Delta, y : \text{true}(\phi) \vdash \Pi' : \text{true}(\psi)}{(X) \Delta \vdash \text{ALL-}E_{\phi, \psi}(\Pi, y : \Pi') : \text{true}(\psi)} \quad v\text{-ALL-E}$$

$$\frac{(X) \Delta \vdash \Pi : \text{true}(\phi[t/x])}{(X) \Delta \vdash \text{SOME-}I_{x, t, \phi}(\Pi) : \text{true}(\exists x \phi)} \quad v\text{-SOME-I}$$

$$\frac{(X) \Delta \vdash \Pi : \text{true}(\exists \phi) \quad (X) \Delta, y : \text{true}(\phi) \vdash \Pi : \text{true}(\psi)}{(X) \Delta \vdash \text{SOME-}E_{\phi, \psi}(\Pi, y : \Pi) : \text{true}(\psi)} \quad v\text{-SOME-E}$$

We omit the valid proof rules for the general introduction and elimination rules, with the above example providing enough intuition that they should be clear. For higher-order logics, the set X is replaced by an assignment \mathcal{A} governing free-variables of the proof. An assignment $\mathcal{A} = \{x_1 : \sigma_1, \dots, x_n : \sigma_n\}$. We overload notation by writing X for assignments.

As well as providing an account of judged proof systems, it is important that our account of logics includes 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 first-order predicate calculus:

Expression Symbols

o	arity ι
succ	arity $\iota \rightarrow \iota$
$+$	arity $(\iota, \iota) \rightarrow \iota$
$=$	arity $(\iota, \iota) \rightarrow \iota$

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

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

the transitivity rule for $=$,

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

and the rule of induction,

$$\frac{\begin{array}{c} [\text{true}(\phi(x))] \\ \vdots \\ \text{true}(\phi(0)) \quad \text{true}(\phi(\text{succ}(x))) \end{array}}{\text{true}(\forall x: \iota\phi(x))} \text{ (ind)}$$

can all be added to the judged proof system.

Definition 7.14 (Theories)

Let \mathcal{L} be a judged proof system. A theory T over \mathcal{L} is a set of expression symbols, a collection of natural deduction rules, Hilbert-type rules and axioms and a set of proof-expressions. \blacksquare

We now provide two examples of judged proof systems which illustrate how general our definition of a judged proof system is. Well the examples are well known, their presentation as judged proof systems is original.

7.2.1 Classical λ -calculus

Our first example is the classical λ -calculus. We define the judged proof-system of the classical λ -calculus defined in (Avron et al. 1992). We have the alphabet:

- $S = \{o\}$;
- $V = \{o\}$;
- $E = \{\Lambda, App\}$;
- $C = \{\Lambda, App\}$;
- $J = \{\bowtie, =\}$;

where o has zero arity, Λ has arity $(o \rightarrow o) \rightarrow o$ and level 2,0 App has arity $(o, o) \rightarrow o$ and level 1 and \bowtie and $=$ are both 2-ary. o is the only syntactic category. It is a syntactic category of expressions and contains variables. The expressions of the classical λ -calculus are generated according to Definition 7.7. For example, take the variable y , we can abstract x to obtain $(y)x$ with arity $o \rightarrow o$. We can now apply Λ to obtain the term $\Lambda(y)x$, usually written $\Lambda y . x$. We can now apply App then z to obtain the term $App(\Lambda(y)x, z)$, usually written $(\lambda y . x)z$.

We have axioms:

$$\begin{aligned} E_0 &: x \bowtie x \\ E_4 &: App(\Lambda(x), y) \bowtie xy \\ \beta_{\bowtie} &: App(\Lambda(x), y) = xy \\ \beta_{=} &: App(\Lambda(x), y) = xy \end{aligned}$$

together with the rules

$$\begin{array}{c}
\frac{x \bowtie y}{y \bowtie x} E_1 \\
\frac{x \bowtie y}{y \bowtie z} E_2 \\
\frac{x \bowtie y \quad x' \bowtie y'}{App(x, x') \bowtie App(y, y')} E_3 \\
\frac{xz \bowtie yz}{\Lambda(x) \bowtie \Lambda(y)} \zeta \\
\frac{x \bowtie y}{x = y} Con \\
\frac{x = y}{y = x} E_5 \\
\frac{x = y \quad y = z}{x = z} E_6 \\
\frac{x = y \quad x' = y'}{App(x, x') = App(y, y')} E_7
\end{array}$$

7.2.2 Hoare's Logic

This example defines a fragment of Hoare's logic which can be adequately represented in LF, *cf.* (Mason 1903). We have the alphabet:

- $S = \{l, i, o, b, w, h\}$;
- $V = \{l, i, o, b\}$;
- $E = \{\alpha, !, c_1, \dots, c_n, f_1, \dots, f_m, P_o^1, \dots, P_o^q, =_o, \Rightarrow_o, \neg_o, \forall, \wedge_o, \vee_o, R_n^1, \dots, R_b^r, =_b, \Rightarrow_b, \neg_b, \wedge_b, \vee_b, :=, ;, if, while, triple\}$;
- $C = \{!, \Rightarrow_o, \neg_o, \forall, \wedge_o, \vee_o, \exists, =_b, \Rightarrow_b, \wedge_b, \vee_b, :=, ;, if, while\}$;
- $j = \{\vdash_h, \vdash_b, \vdash_o, \neq, \#i, \#o\}$,

where

Expression	Arity	Level
α	$b \rightarrow o$	1
$!$	$l \rightarrow i$	1
c_i	i	1
f_i	$\underbrace{(i, \dots, i)}_{p_i} \rightarrow i$	1
R_o^j	$\underbrace{(i, \dots, i)}_{o_j} \rightarrow o$	1
$=_0$	$(i, i) \rightarrow o$	1
\Rightarrow_o	$(o, o) \rightarrow o$	1
\neg_o	$o \rightarrow o$	0
\forall	$(i \rightarrow o) \rightarrow o$	2
\wedge_o	$(o, o) \rightarrow o$	1
\vee_0	$(o, o) \rightarrow o$	1
\exists	$(i \rightarrow o) \rightarrow o$	2
R_b^m	$\underbrace{(i, \dots, i)}_{l_m} \rightarrow b$	1
$=_b$	$(i, i) \rightarrow b$	1
\Rightarrow_b	$(i, i) \rightarrow b$	1
\neg_b	$b \rightarrow b$	1
\wedge_b	$(b, b) \rightarrow b$	1
\vee_b	$(b, b) \rightarrow b$	1
$:=$	$(l, l) \rightarrow w$	1
$;$	$(w, w) \rightarrow w$	1
<i>if</i>	$(b, w, w) \rightarrow w$	1
<i>while</i>	$(b, w) \rightarrow w$	1
<i>triple</i>	$(o, w, o) \rightarrow h$	1
\vdash_h	h	
\vdash_b	b	
\neq	(l, l)	
$\#_i$	(l, i)	
$\#_o$	(l, o)	

The consequence relation \vdash is the one determined by all the inference rules. We omit the rules for connectives found in classical logic, *i.e.*, $\wedge_o, \vee_o, \Rightarrow_o, \forall, \exists$, *etc.* because they are essentially the same as the ones we gave above.

$$\frac{\vdash_b x}{\vdash_o \alpha(x)} \alpha_1$$

$$\frac{\vdash_o \alpha(x)}{\vdash_b x} \alpha_2$$

$$\begin{array}{c}
\frac{\vdash_o \alpha(R_b(x_1, \dots, x_n))}{\vdash_b R_b(x_1, \dots, x_n)} \alpha_{3R} \\
\frac{\vdash_b R_b(x_1, \dots, x_n)}{\vdash_o \alpha(R_b(x_1, \dots, x_n))} \alpha_{4R} \\
\vdash_o \alpha(x_1 =_b x_2) \Leftrightarrow x_1 =_o x_2 \\
\vdash_i \alpha(\neg_b x) \Leftrightarrow \neg_o \alpha(x) \\
\vdash_o \alpha(b_1 \chi_b b_2) \Leftrightarrow \alpha(b_1) \chi_o \alpha(b_2)
\end{array}$$

for $\chi \in \{\wedge, \vee, \Rightarrow\}$

$x \#_i c_i$ for all constant c_i

$$\begin{array}{c}
\frac{x \neq y}{x \#_i y!} \\
\frac{x \#_i t_1 \cdots x \#_i t_n}{x \#_i f_i(t_1, \dots, t_n)} \\
\frac{x \#_i t + 1 \cdots x \#_i t_n}{x \#_o R_o(t_1, \dots, t_n)} \\
\frac{x \#_o e_1 \quad x \#_o e_2}{x \#_o e_1 \chi_o e_2}
\end{array}$$

for $\chi \in \{\wedge, \vee, \Rightarrow\}$

$$\begin{array}{c}
\frac{x \neq y \quad x \#_i f(y)}{x \#_o \forall f} \\
\{p[t/x]\} x := t \{p\} \\
\frac{\{p\} S_1 \{r\} \quad \{r\} S_2 \{q\}}{\{p\} S_1; S_2 \{q\}} \\
\frac{\{p \wedge e\} S \{q\} \quad \{\wedge \neg e\} S_2 \{q\}}{\{p\} \text{if}(e, S_1, S_2) \{q\}} \\
\frac{\{p \wedge e\} S \{p\}}{\{p\} \text{while}(e, S) \{p \wedge \neg e\}} \\
\frac{p \Rightarrow p_1 \quad \{p_1\} S \{q_1\} \quad q_1 \Rightarrow q}{\{p\} S \{q\}}
\end{array}$$

7.3 Models of Judged Object-logics

This section provides a model-theoretic characterization of consequence relations. We define a class of models of judged object-logics together with a satisfaction relation. We begin with an informal overview of the desired satisfaction relation.

Our satisfaction relation is to be the one of Kripke forcing (Kripke 1963). Let W be a set of worlds and let R be an $(m + 1)$ -ary relation over W . We write $Rww_1 \dots w_m$ for the value of R at worlds w, w_1, \dots, w_m . Let $\phi = \#(\phi_1, \dots, \phi_m)$ be a formula with outermost connective $\#$ of arity m . We distinguish two general classes (*cf.* (Basin, Matthews & Viganò 1997)) of connectives, together with judgements:

1. $\langle \#, j \rangle$ is *local* if the meaning of $\langle \#(\phi_1, \dots, \phi_m), j \rangle$ at world w depends only on the meaning of each $\langle \phi_i, j_i \rangle$ at world w ;
2. $\langle \#, j \rangle$ is *non-local* if the meaning of $\langle \#(\phi_1, \dots, \phi_m), j \rangle$ at world w depends on the meaning of each $\langle \phi_i, j_i \rangle$ at worlds w_i , where $R_{\#}ww_1 \dots w_m$ holds for some chosen relation. We will consider both *universal* non-local connectives, in which we permit universal quantification over the worlds occurring in the definiens, and *existential* non-local connectives, in which we permit existential quantification over the worlds occurring in the definiens.

The classes we have defined are quite broad and clearly more complex classes are possible. The definition of local does not appear to include connectives like \vee , which one would expect to define as

$$\langle \phi \vee \psi, \text{true} \rangle \text{ if and only if } \langle \phi, \text{true} \rangle \text{ or } \langle \psi, \text{true} \rangle$$

but we observe that the meaning of $\langle \phi \vee \psi, \text{true} \rangle$ does depend on $\langle \phi, \text{true} \rangle$ and $\langle \psi, \text{true} \rangle$, but they are not required to both be satisfied at the same time. This is the analogue of having multiple introduction schema for the same connective, they all need to be formally defined but only one introduces the connective in a given proof.

Examples of local connectives include intuitionistic and classical disjunction and conjunction and classical implication. Examples of universal non-local connectives include intuitionistic implication, for which W must be preordered, relevant implication and $S4 \Box$. The \Diamond of $S4$ is an example of an existential non-local connective.

Following this idea, we define Kripke models of object-logics in our usual indexed setting, following the pattern established for the $\lambda\Pi$ -calculus, 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 logic.

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 $\llbracket - \rrbracket^{-\rho}$ of the syntax in a model. We can require, however, at the level of prestructures, that there be enough structure to interpret the function symbols of the term language in the base category.

Thus, in order to define object-logic models, we take an object-logic \mathcal{O}_T to be given by the following:

- A *first-order language* L over an alphabet $A = (S, V, E, C, J)$;
- A collection of *local connectives* $C_L \subset C$, each defined by a satisfaction relation;
- A collection of *non-local or global connectives* $G_G \subset C$ defined by a satisfaction relation and a world relation;
- A *theory* consisting of a set of expressions symbols and constants used in the grammar for valid proof-expressions.

Such a logic \mathcal{O}_T may or may not have a complete axiomatization as a Hilbert-type or natural deduction type system. Drawing heavily on the work of (Basin et al. 1997), (Basin et al. 1998) and (Viganò 2000), we present classes of logics which have axiomatizations as Hilbert-type or natural deduction systems. We begin by describing a satisfaction relation $\Vdash_{\mathcal{O}_T}$ for local and non-local connectives. The generalized introduction rule of Prawitz (1978) provides the intuition for the satisfaction relation for a local connective. The schemata

$$\frac{\begin{array}{ccc} [K_{j,1}^i] \cdots [K_{j,h_j^i}] & & \\ \vdots & \vdots & \vdots \\ J_1^i & J_j^i & J_{p_i}^i \end{array}}{J} \# I$$

for $1 \leq i \leq s$, corresponds to the satisfaction relation

$$\begin{aligned} w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} J \text{ if and only if either } & ((w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} K_{1,1}^1 \text{ and } \dots \text{ and } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} K_{1,h_1^1}^1) \\ & \text{imply } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} J_1^1) \text{ and } \dots \text{ and } ((w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} K_{p_1,1}^1 \text{ and } \dots \text{ and } \\ & w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} K_{p_1,h_{p_1}^1}^1) \text{ imply } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} J_{p_1}^1) \text{ or } \dots \text{ or } ((w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} K_{1,1}^s \text{ and } \\ & \dots \text{ and } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} K_{1,h_1^s}^s) \text{ imply } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} J_1^s) \text{ and } \dots \text{ and } ((w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} K_{p_s,1}^s \\ & \text{and } \dots \text{ and } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} K_{p_s,h_{p_s}^s}^s) \text{ imply } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} J_{p_s}^s) \end{aligned}$$

where \mathcal{M} is a Kripke model of \mathcal{O}_T and $J = j(\#(\phi_1, \dots, \phi_n))$.

The satisfaction relation for the non-local connectives comes from (Basin et al. 1997) and is simpler than the satisfaction relation for local connectives. It is also less general than Prawitz's schema. For a universal non-local connective, we have the satisfaction relation

$$w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j(\#(\phi_1, \dots, \phi_n)) \text{ if and only if for all worlds } w_1, \dots, w_n ((R_{\#} w w_1 \dots w_n \text{ and } w_1, \rho_1 \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j_1(\phi_1) \text{ and } \dots \text{ and } w_{n-1}, \rho_{n-1} \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j_{n-1}(\phi_{n-1})) \text{ imply } w_n, \rho_n \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j_n(\phi_n))$$

where \mathcal{M} is a Kripke model for \mathcal{O}_T .

For an existential non-local connective, we have the satisfaction relation

$$w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j(\#(\phi_1, \dots, \phi_n)) \text{ if and only if there exist worlds } w_1, \dots, w_n (R_{\#} w w_1 \dots w_n \text{ and } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j_1(\phi_1) \text{ and } \dots \text{ and } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j_n(\phi_n))$$

where \mathcal{M} is a Kripke model for \mathcal{O}_T .

We need a separate satisfaction relation for \perp and \neg , when they are present. Our definition of these satisfaction relations is more restrictive than the ones given in (Viganò 2000). This restriction, however, ensures that $\#$ cannot be a connective of a substructural logic. The satisfaction relation for \perp is defined as follows: for all Kripke models \mathcal{M} and all worlds w , $w, \rho \not\Vdash_{\mathcal{O}_T}^{\mathcal{M}} j(\perp)$. The satisfaction relation for \neg is defined as follows:

$$w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j(\neg\phi) \text{ if and only if } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j(\phi) \text{ implies } w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{M}} j(\perp)$$

where \mathcal{M} is a Kripke model.

We have not specified the relation $R_{\#}$ for each non-local connective. Basin et al. (1997) use a Horn relational theory to describe a relation for a given connective. We only examine two possibilities for this relation. The first is when the relation is \leq , the partial ordering on worlds. This is sufficient for us to define intuitionistic implication and universal and existential quantification. The second is a more general case, here we let the relation be an (i, j, m, n) -convergency axiom, *i.e.*, $R_{\#}$ is a closed formula of the form

$$\forall x \forall y \forall z ((x R^i y \wedge x R^j z) \supset \exists u (y R^m u \wedge z R^n u)),$$

where $x R^0 y$ means $x = y$ and $x R^{i+1} y$ means $\exists v (x R v \wedge v R^i y)$. We use a relation of this form when we are defining modal logics. Since, if $R_{\square} = R_{\diamond}$ is an (i, j, m, n) -convergency axiom, then it corresponds to an axiom of the form

$$\diamond^i \square^m \phi \supset \square^j \diamond^n \phi$$

for a proof of this result see; for example, (Viganò 2000), (Lemmon & Scott 1977) or (Chellas 1980). The axioms $D, T, B, 4, 5, \text{etc.}$, are all instances of this schema.

Basin et al. (1997) provide more relations, but all the other examples give substructural logics which cannot be adequately represented in LF. There is more scope here for examining the choice of relations in detail, but this is beyond the scope of this thesis. Correspondence theory examines the relationship between relations on Kripke models for logics that involves modalities, and axioms in Hilbert-type systems. The reader is referred to the survey paper (van Benthem 1984), for further details. A similar analysis is possible for intuitionistic modal logics, but the nice relationship between axiom schema and properties of R does not hold (*cf.* (Simpson 1994)).

The basic idea of our functorial treatment of object-logics is one that is familiar from our treatment of models of the $\lambda\Pi$ -calculus and its internal logic, *cf.* § 5. At each world, a prestructure provides, functorially, a functor from environments to values. In contrast, however, with the internal logic of the $\lambda\Pi$ -calculus, our formulation of object-logics, especially 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 Kripke model of \mathcal{O}_T , $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, which supports the principle

“if $\text{true}(\phi)$ then $\text{valid}(\phi)$ ”

by virtue of the existence of an arrow

$$\llbracket \langle \Box\phi, \text{true} \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{m} \llbracket \langle \Box\phi, \text{valid} \rangle \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

independently of the existence of a proof δ such that

$$w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \delta : (\langle \Box\phi, \text{true} \rangle \vdash_{\mathcal{O}_T} \langle \Box\phi, \text{valid} \rangle).$$

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 $V \otimes \mathbf{J}$, where \mathbf{J} is the category of judgements. The product here being the tensor product in \mathcal{Cat} .

Definition 7.15 (Category of Judgements)

A *category of judgements* \mathbf{J} is defined as follows:

Objects: Judgements j , including a terminal object null ;

Arrows: Identities and at least $\text{null} \xrightarrow{j} j$, for each judgement j . ■

Additionally, we may have arrows $j \xrightarrow{m} k$, which may be used in a model to validate principles of the kind discussed above, *i.e.*, rules in the object-logic which only change the judgement and not the formula. Note that although we have used the syntax of judgement symbols to describe the category of judgements, we can consider any category having the specified structure to be a category of judgements.

A judged proposition is interpreted in a category of judged values by taking the tensor product (in $\mathcal{C}at$) of a category of judgements and a category of values. The transition in 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 7.1.

Definition 7.16 (Judged Values)

A *category of judged values* is a category

$$V \otimes J$$

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

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

$$\begin{array}{ccc} & \text{proof} & \\ & \longrightarrow & \\ \circlearrowleft & & \circlearrowleft \\ \text{null} & & \text{proof} \end{array}$$

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

$$\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \otimes \text{null} \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \otimes \text{proof}} \llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \otimes \text{proof}$$

where δ denotes the corresponding unjudged proof-tree, which can be characterized by the techniques of Lemma 7.1. From now on we refer to $\langle \phi, j \rangle$ as $j(\phi)$.

Throughout our development of Kripke models for \mathcal{O}_T , we will just refer to V as the category of values rather than $V \otimes J$ and $\llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \otimes \text{proof}$ as $\llbracket \text{proof}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$.

Definition 7.17 (Kripke Prestructures for \mathcal{O}_T)

A *Kripke prestructure for \mathcal{O}_T* 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}_W is a small cartesian closed category with a terminal object;
3. $\mathcal{B}^{op} = \coprod_{W \in |\mathcal{W}|} \mathcal{B}_W^{op}$;
4. \mathcal{V} is a (sub)category (of $\mathcal{C}at$) of categories of judged values such that each $\mathcal{J}(W)(D)$ has a terminal object and finite products, which are preserved on the nose by inverse image functors;
5. The indexed category \mathcal{J} is strict. ■

We take base categories at each world and then define fibres over their coproduct to follow the structure of a Kripke $\lambda\Pi$ -prestructure. We recall that the reason for this is that we want our Kripke models to be analogous to Kripke models of intuitionistic logic where there is a model of classical logic at each world. We will find when we construct the term model of \mathcal{O}_T that we need to take the same category at each world. This is because we need to define constants and functions at each world.

This Kripke prestructure for \mathcal{O}_T has less structure than the Kripke prestructure we defined for the internal logic in § 5. We no longer require that there is a right adjoint to the inverse image of projections or that $\mathcal{J}(W)(D)$ to be cartesian closed. The reason for this is that the object-logic \mathcal{O}_T may not have universal quantification or implication, so we do not, in general, require the extra structure. Any structure required to interpret a connective arises from the enough structure condition (*cf.* Definition 7.19) and the satisfaction relation.

We still have the coproduct construction to ensure that our models are consistent throughout and that there are conceptually natural.

Definition 7.18 (Kripke Structures for \mathcal{O}_T)

Let \mathcal{J} be a Kripke prestructure for \mathcal{O}_T . A *Kripke structure for \mathcal{O}_T* is a functor

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

such that the category $\overline{\mathcal{V}}$ has the following properties:

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

Objects: Arrows

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

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

Arrows: Arrows

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

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

Arrows: Functors $\mathcal{K}_{\mathcal{J}}(f) : \mathcal{K}_{\mathcal{J}}(W)(Y) \rightarrow \mathcal{K}_{\mathcal{J}}(W)(X)$, where $f : X \rightarrow Y$ is an arrow in \mathcal{B}_W . These functors have 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 $\mathcal{K}_{\mathcal{J}}(W)(X)$,

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

2. The functor $\mathcal{K}_{\mathcal{J}}(W)(f)$ takes an arrow of $\mathcal{K}_{\mathcal{J}}(W)(Y)$, $\overline{A} \xrightarrow{\mu} \overline{B}$, and returns the arrow $\nu = \mathcal{J}(W)(f)(\mu)$, where $\overline{C} \xrightarrow{\nu} \overline{D}$, $\mathcal{J}(W)(f)(A_i) = C_i$, for $1 \leq i \leq m$, and $\mathcal{J}(W)(f)(B_j) = D_j$, for $1 \leq j \leq n$. \blacksquare

Before we can define the Kripke model of \mathcal{O}_T , we need to ensure that the Kripke prestructure for \mathcal{O}_T has enough structure to interpret the language of L of \mathcal{O}_T .

Definition 7.19 (Enough Structure)

Let \mathcal{O}_T be an object-logic with alphabet A . We say that a Kripke prestructure for \mathcal{O}_T has *enough structure* to interpret A if the following conditions hold:

- Each \mathcal{B}_w has at least as many objects as there are syntactic categories;
- For all worlds w and for each function symbol $e \in E/C$ with arity $(S_1, \dots, S_n) \rightarrow S$, there exists an arrow $(\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}) \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in \mathcal{B}_w ;
- For all worlds w and for each predicate letter $e \in E/C$ with arity $(S_1, \dots, S_n) \rightarrow S$, there exists an object $\llbracket j(e(x_1, \dots, x_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$, where $x_i : S_i$ for $1 \leq i \leq n$ and $j \in J$, in $\mathcal{J}(w)(\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$;
- For all worlds w and for each connective $c \in C$, with arity $(a_1, \dots, a_n) \rightarrow a$, there exists an operation op_c such that op_c takes n objects $\llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho_i}$ in $\mathcal{J}(w)(\prod_{j=1}^{m_i} \llbracket S_j^i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho_i})$, where the free variables of ϕ_i are in the set $\{x_1^i : S_1^i, \dots, x_{m_i}^i : S_{m_i}^i\}$, and returns an object $\llbracket j(c(\phi_1, \dots, \phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in $\mathcal{J}(w)(\prod_{k=1}^p \llbracket S_k \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$, where the free variables of $c(\phi_1, \dots, \phi_n)$ are in the set $\{x_1 : S_1, \dots, x_p : S_p\}$, each $x_k : S_k \in \{x_1^1 : S_1^1, \dots, x_{m_n}^n : S_{m_n}^n\}$ and ρ is the combination of each of the environments ρ_1, \dots, ρ_n with any variables which are bound in $c(\phi_1, \dots, \phi_n)$ but free in one of the ϕ_i s removed;

- For each judgement symbol $j \in J$, there is an object in the category of judgements j . ■

Since the third condition is rather complicated, we provide some examples which illustrate that it is correct.

- We take c to be \wedge and have op_\wedge act on $\llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in $\mathcal{J}(w)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$, where ϕ has free variable $x : S$, and $\llbracket j(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in $\mathcal{J}(w)(\llbracket T \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$, where ψ has free variable $y : T$. op_c returns an object $\llbracket j(\phi \wedge \psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in $\mathcal{J}(w)(\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \llbracket T \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$;
- The condition on the environments allows us to handle quantification correctly. We take c to be \forall , op_\forall acts on $\llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in $\mathcal{J}(w)(\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$, where the free variables of ϕ are in the set $\{x_1 : S_1, \dots, x_n : S_n, x : S\}$, and returns $\llbracket j(\forall x : S. \phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho/x}$ in $\mathcal{J}(w)(\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho/x})$. We observe that x is free in ϕ but bound in $\forall x : S. \phi$.

In order to define the structure carried by models of \mathcal{O}_T , we require an interpretation $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-,\rho}$ of the syntax \mathcal{O}_T in a structure $\mathcal{K}_{\mathcal{J}}$: the relations $R_\#$ and the satisfaction relation that define each connective $\#$ are predicated on the interpretations of the propositional arguments taken by ϕ .

Definition 7.20 (Kripke Models of \mathcal{O}_T)

A *Kripke model* of \mathcal{O}_T consists of a pair $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-,\rho} \rangle$, where $\mathcal{K}_{\mathcal{J}}$ is a Kripke structure for \mathcal{O}_T , which has enough structure to interpret the alphabet of A , and the partial function $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-,\rho}$ is an *interpretation* of \mathcal{O}_T in $\mathcal{K}_{\mathcal{J}}$. The interpretation is defined by induction on the structure of (i) sorts, which are interpreted as objects in \mathcal{B}_w ; (ii) terms, which are interpreted as arrows in \mathcal{B}_w ; and (iii) propositions with free variables in the set $X = \{x_1 : S_1, \dots, x_n : S_n\}$, which are interpreted in the fibre over $\llbracket S_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \dots \times \llbracket S_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. If $X = \emptyset$, then $\llbracket \phi(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is an object of $\mathcal{K}_{\mathcal{J}}(w)(1)$. The sorts, terms and functions are interpreted as follows:

1. For each sort S , $\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is an object of \mathcal{B}_w , defined by induction on sorts:
 - For each sort S , $\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is (a choice of) an object in \mathcal{B}_w ;
 - For each function sort, *i.e.*, $S = S_1, \dots, S_n \rightarrow T$,

$$\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = \llbracket T \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho(\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})}$$

the internal *hom* in \mathcal{B}_w

2. For each variable $x : S$, $\llbracket x \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is an arrow $\llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\rho(x)} \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in \mathcal{B}_w , not dependent on w ;
3. For each function symbol $f : S_1, \dots, S_n \rightarrow S$, $\llbracket f \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is the arrow given by Definition 7.19. We have that:

- Constants c with arity S are interpreted as arrows $\llbracket c \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} : 1 \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in \mathcal{B}_w ;
- Functions $f : S_1, \dots, S_n \rightarrow S$ are interpreted as arrows

$$\left(\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \right) \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

in \mathcal{B}_w , and given $t_i : S_i$ for $1 \leq i \leq n$, then

$$\llbracket f t_1 \dots t_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = \llbracket f \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \llbracket t_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \dots \llbracket t_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

using the cartesian closed structure of \mathcal{B}_w in the usual way (cf. (Lambek & Scott 1986));

- Tuples of terms are interpreted as arrows in \mathcal{B}_w :

$$\langle \llbracket t_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}, \dots, \llbracket t_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rangle : \llbracket A_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \dots \times \llbracket A_m \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket B_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \dots \times \llbracket B_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho},$$

where, for each $1 \leq i \leq n$, $x_1 : A_1, \dots, x_m : A_m \vdash_T t_i : B_i$;

- Term-formation by application is interpreted by function space application in \mathcal{B}_w .

The connectives are interpreted as objects in $\mathcal{J}(W)(X)$ by induction over the structure of formulæ:

- Atomic: For each predicate letter P with arity $(S_1, \dots, S_n) \rightarrow S$, $\llbracket j(P(x_1, \dots, x_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is an object of $\mathcal{J}(W)(\llbracket S_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \dots \times \llbracket S_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$ given by Definition 7.19;
- Connective: For each connective c , with arity $(S_1, \dots, S_n) \rightarrow S$, $\llbracket j(c(\phi_1, \dots, \phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is an object of $\mathcal{J}(W)(X)$, where $X = \prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and the free variables of $c(\phi_1, \dots, \phi_n)$ are in the set $\{x_1 : S_1, \dots, x_m : S_m\}$ and $j \in \mathbf{J}$, given by Definition 7.19.

4. For each local connective $\#$, we have the following satisfaction condition: there is an arrow

$$1 \xrightarrow{f} \llbracket j(\#(\phi_1, \dots, \phi_m)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

if and only if there are arrows

$$(((1 \xrightarrow{f_{1,1}^1} \llbracket \mathbf{K}_{1,1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ and } \dots \text{ and } 1 \xrightarrow{f_{h_1}^1} \llbracket \mathbf{K}_{1,h_1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}) \text{ which imply } 1 \xrightarrow{f_1^1} \llbracket \mathbf{J}_1^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$$

$$\text{and } \dots \text{ and arrows } ((1 \xrightarrow{f_{p_1,1}^1} \llbracket \mathbf{K}_{p_1,1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ and } \dots \text{ and } 1 \xrightarrow{f_{p_1,h_{p_1}^1}^1} \llbracket \mathbf{K}_{p_1,h_{p_1}^1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$$

which imply $1 \xrightarrow{f_{p_1}^1} \llbracket J_{p_1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) or ... or there are arrows ($((1 \xrightarrow{f_{1,1}^s} \llbracket K_{1,1}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$
and ... and $1 \xrightarrow{f_{1,h_1^s}^s} \llbracket K_{1,h_1^s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \xrightarrow{f_1^s} \llbracket J_1^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) and
 $((1 \xrightarrow{f_{p_s,1}^s} \llbracket K_{p_s,1}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and ... and $(1 \xrightarrow{f_{p_s,h_{p_s}^s}^s} \llbracket K_{p_s,h_{p_s}^s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply
 $1 \xrightarrow{f_{p_s}^s} \llbracket J_{p_s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$)).

Also, $\llbracket \#-I \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = f$;

5. For each universal non-local connective $\#$, we have the following satisfaction condition: there is an arrow

$$1 \xrightarrow{f} \llbracket j(\#(\phi_1, \dots, \phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

if and only if for all worlds w_1, \dots, w_n ($R_{\#}ww_1 \dots w_n$ and there exist arrows

$$1 \xrightarrow{f_1} \llbracket j_1(\phi_1) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_1,\rho_1} \text{ and } \dots \text{ and } 1 \xrightarrow{f_{n-1}} \llbracket j_{n-1}(\phi_{n-1}) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_{n-1},\rho_{n-1}})$$
 which imply

$$1 \xrightarrow{f_n} \llbracket j_n(\phi_n) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_n,\rho_n}.$$

Also, $\llbracket \#-I \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = f$;

6. For each existential non-local connective $\#$, we have the following satisfaction condition: there is an arrow

$$1 \xrightarrow{f} \llbracket j(\#(\phi_1, \dots, \phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

if and only if there exists worlds w_1, \dots, w_n such that $R_{\#}ww_1, \dots, w_n$ and there exist arrows

$$1 \xrightarrow{f_1} \llbracket j_1(\phi_1) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_1,\rho_1} \text{ and } \dots \text{ and } 1 \xrightarrow{f_n} \llbracket j_n(\phi_n) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_n,\rho_n}.$$

Also, $\llbracket \#-I \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = f$;

7. If $\perp \in C_G$, then we have the following satisfaction condition: for all worlds w , there are no arrows $1 \xrightarrow{f} \llbracket j(\perp) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. Also, $\llbracket \perp-I \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = f$;

8. If $\neg \in C_G$, then we have the following satisfaction condition: for all worlds w , there exists an arrow

$$1 \xrightarrow{f} \llbracket j(\neg\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ if and only if the arrow } 1 \xrightarrow{f_1} \llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ implies}$$

$$1 \xrightarrow{f_2} \llbracket j(\perp) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}.$$

Also, $\llbracket \neg I \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = f$.

9. If there is more than one judgement, we require the following satisfaction condition:

$$1 \xrightarrow{f} \llbracket \mathbf{j}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

implies

$$1 \xrightarrow{f'} \llbracket \mathbf{k}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho};$$

10. We require the following *syntactic monotonicity condition*: if $\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is defined, then so is $\llbracket X' \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$, for every subterm X' of X ;
11. We require the following *accessibility condition*: there is an arrow $W \xrightarrow{\alpha} W'$ in \mathcal{W} , then $\mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^W) = \mathcal{J}(W')(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W',\rho})$ and $\mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W,\rho}) = \mathcal{J}(W)(\llbracket \Gamma \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{W',\rho})$. ■

The definition of the structure in the model corresponding to each connective $\#$ relies on the definition of the interpretations of the propositional arguments taken by $\#$. Hence without the syntactic monotonicity condition, our definition would fail because we would be unable to start the induction. Condition (9) essentially defines an arrow $\mathbf{j} \rightarrow \mathbf{k}$ in the category of judgements.

As an example of a Kripke model for \mathcal{O}_T , we construct the term model. This is essentially the same construction as the one in § 5.2.3. We firstly fix the object-logic we are defining, so that we know the alphabet A . We define the category $\mathcal{B}(A)$ of contexts and realizations as follows:

Objects: Contexts of the form $x_1 : S_1, \dots, x_m : S_m$, for $m \geq 0$ ($m = 0$ gives the unique empty context, the terminal object of \mathcal{B});

Arrows: Tuples of the form

$$x_1 : S_1, \dots, x_m : S_m \xrightarrow{\langle t_1, \dots, t_n \rangle} y_1 : T_1, \dots, y_n : T_n$$

such that for $1 \leq i \leq n$, $x_1 : S_1, \dots, x_m : S_m \vdash_T t_i : T_i$ (Terms t_i will be of the form $f t_1 \dots t_n$. In particular, a variable x of sort S arises as an arrow $x : S \xrightarrow{\langle x \rangle} x : S$).

We define a category of worlds \mathcal{W} .

Definition 7.21

The category \mathcal{W} is defined as follows:

Objects: The empty context, $\langle \rangle$ is an object of \mathcal{W} . If X is an object of \mathcal{W} and there exists an arrow $X \rightarrow X'$ in $\mathcal{B}(A)$, then X, X' is an object of \mathcal{W} ;

Arrows: There is an arrow $X \rightarrow X'$ if and only if $X \subseteq X'$. ■

We have the obvious inclusion, $\mathcal{W} \hookrightarrow \mathcal{B}(A)$. Clearly, \mathcal{W} can be viewed as a posetal category of contexts ordered by inclusion, $X \subseteq X'$.

We define the category \mathcal{C}_X to be the category $\mathcal{B}(A)$ at each world. We then define $\mathcal{B} = \coprod_{X \in |\mathcal{W}|} \mathcal{C}_X$.

\mathcal{C}_X is cartesian closed, see the argument in § 5.2.3.

We define a functor $\mathcal{T} : [\mathcal{W}, [\mathcal{B}^{op}, \mathcal{V}]]$ as follows: at each object $X = x_1 : S_1, \dots, x_m : S_m$ of \mathcal{W} and each object $\Delta = y_1 : T_1, \dots, y_n : T_n$ of \mathcal{C}_X , we define a category $\mathcal{T}(X)(\Delta)$ as follows:

$$\mathcal{T}(X)(\Delta) = \begin{cases} \text{Objects: Judged propositions } j(\phi) \text{ such that } Fv(\phi) \subseteq X \bowtie \Delta; \\ \text{Arrows: Proofs } \Phi \text{ such that } j(\phi) \xrightarrow{\Phi} k(\psi) \text{ if and only if} \\ (X \bowtie \Delta) j(\phi) \vdash_T \Phi : k(\psi). \end{cases}$$

At each object Y of \mathcal{W} and each arrow $X' \xrightarrow{t} X$, we define the functor $t^*(= \mathcal{T}(Y)(t)) : \mathcal{T}(Y)(X) \rightarrow \mathcal{T}(Y)(X')$, as usual (Lawvere 1970) and (Seely 1983) this is given by substitution.

At each arrow $X \xrightarrow{\alpha} X'$ of \mathcal{W} , we must define a natural transformation $\mathcal{T}(X) \xrightarrow{\mathcal{T}(\alpha)} \mathcal{T}(X')$. As in the example constructed in § 3.2.1, inclusions will do:

$$\begin{array}{ccc} \Delta & & \mathcal{T}(X)(\Delta) \xrightarrow{\mathcal{I}_\Delta} \mathcal{T}(X')(\Delta) \\ \downarrow t & & \downarrow \mathcal{T}(X)(t) \quad \downarrow \mathcal{T}(X')(t) \\ \Delta' & & \mathcal{T}(X)(\Delta') \xrightarrow{\mathcal{I}_{\Delta'}} \mathcal{T}(X')(\Delta') \end{array}$$

Each fibre has a product, the proof is a slight generalization of the one given for the term model in § 5.2.3 to include judgements. There may be extra structure required in the Kripke prestructure. This will arise due to the structure having enough structure to interpret the connectives in the alphabet A and the satisfaction conditions in the definition of the interpretation (*cf.* examples below).

We are now able to define a Kripke structure for $\mathcal{L}_T, \mathcal{K}_T : [\mathcal{W}, [\mathcal{B}, \overline{\mathcal{V}}]]$. We define the category $\overline{\mathcal{V}}$ as follows:

Objects: Categories \overline{V} built out of $V = \mathcal{J}(X)(\Delta)$, which contain

Objects: Arrows $j_1(\phi_1) \times \dots \times j_n(\phi_n) \rightarrow k(\psi)$ in $\mathcal{J}(X)(\Delta)$;

Arrows: Arrows $\Gamma \rightarrow k(\psi)$ to $\Gamma' \rightarrow k(\tau)$ are given by arrows $\Gamma \rightarrow \Gamma'$ in $\mathcal{J}(X)(\Delta)$.

Arrows: Functors $f^* : \mathcal{K}_T(X)(\Delta) \rightarrow \mathcal{K}_T(X)(\Delta')$, where $f : \Delta' \rightarrow \Delta$ is an arrow in $\mathcal{T}(X)$ are defined to be the usual substitution.

It is straightforward to check that the functors f^* satisfy the definition. The interpretation is taken to be the identity function in the fibres, in the base categories \mathcal{C}_X , we interpret a sort S by $x:S$.

We now complete our semantic view of object-logics by defining the satisfaction of a proposition in a model. We have given the satisfaction relations for connectives $\#$ in the definition of the interpretation, here we are concerned with the satisfaction relation for the whole logic.

Definition 7.22 (Satisfaction)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{B}^{op}, \overline{\mathcal{V}}]]$, be a Kripke model of \mathcal{O}_T . $\mathcal{K}_{\mathcal{J}}$ satisfies $j(\phi)$ at world w with respect to environment ρ ,

$$w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$$

if and if there is an arrow

$$1 \xrightarrow{f} \llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$$

in $\mathcal{J}(W)(D)$, where $D = \prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$ and all the free variables of ϕ are in the set $\{x_1:S_1, \dots, x_n:S_n\}$. ■

If $\Gamma = \{y_1:j_1(\phi_1), \dots, y_n:j_n(\phi_n)\}$, then we write $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ (or more commonly $w, \rho, \Gamma \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$) if for each $1 \leq i \leq n$, $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j_i(\phi_i)$. Satisfaction is also monotone, if $w \xrightarrow{f} w'$ and $\llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w', \rho}$ is defined, then $w', \rho[f] \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$,

We consider three examples of our formulation of object-logic models: classical predicate logic, intuitionistic predicate logic and classical propositional modal K . We begin by giving the alphabet of classical predicate logic:

- $S = \{\iota, o\}$;
- $V = \{\iota\}$;
- $E = \{\wedge, \vee, \supset, \forall, \exists, \neg, \perp\}$;
- $C_L = \{\wedge, \vee, \supset, \neg\}$;
- $C_G = \{\forall, \exists, \perp\}$;
- $J = \{\text{true}\}$.

The satisfaction relation for each local condition for each local connective is as follows:

$$1 \xrightarrow{f} \llbracket \text{true}(\phi \wedge \psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho} \text{ if there exist arrows } 1 \xrightarrow{f_1} \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$$

$$\text{and } 1 \xrightarrow{f_2} \llbracket \text{true}(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$$

$$\begin{aligned}
1 &\xrightarrow{f} \llbracket \text{true}(\phi \vee \psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ if there exist arrows } 1 \xrightarrow{f_1} \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \\
&\quad \text{or } 1 \xrightarrow{f_2} \llbracket \text{true}(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \\
1 &\xrightarrow{f} \llbracket \text{true}(\phi \supset \psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ if there exist arrows } 1 \xrightarrow{f_1} \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \\
&\quad \text{implies } 1 \xrightarrow{f_2} \llbracket \text{true}(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}
\end{aligned}$$

The satisfaction relation for each non-local connective is as follows:

$$\begin{aligned}
1 &\xrightarrow{f} \llbracket \forall x : \iota . \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ if and only if for all worlds } w', w \xrightarrow{\alpha} w' \text{ and for all} \\
&\quad a : \llbracket \iota \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket \iota \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho} \text{ and terms } t \text{ such that } \llbracket t \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho[\alpha]}, \text{ then } 1 \rightarrow \llbracket \phi[t/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho[\alpha][x:=a]} \\
1 &\xrightarrow{f} \llbracket \exists x : \iota . \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ if and only if there exists a world } w', w \xrightarrow{\alpha} w' \text{ such that there} \\
&\quad \text{exists } a : \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho} \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}, \text{ a term } t \text{ such that } \llbracket t \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho} \text{ an arrow} \\
&\quad 1 \rightarrow \llbracket \phi[t/x] \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho[f][x:=a]}
\end{aligned}$$

We now show what properties we need \mathcal{J} to have in order for it to have enough structure to interpret the alphabet of classical predicate logic. We require each fibre $\mathcal{J}(W)(D)$ to have products, coproducts, exponentials and an initial object. Further, we require that each functor p^* induced by the projection $p:U \times V \rightarrow U$ in the base category, has a left and right adjoint which satisfy the (strict) Beck-Chevalley conditions. We show the argument for interpreting universal quantification as a right adjoint, the other arguments are similar. The satisfaction relation tells us that for every $\llbracket \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$, there is a bijection between arrows $op_c(\llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}) \rightarrow \llbracket \psi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in $\mathcal{J}(W)(\prod_{i=1}^n \llbracket \iota \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$ and arrows $\llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow p^*_{\prod_{i=1}^n \llbracket \iota \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \llbracket \iota \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}}$ in $\mathcal{J}(w)(\prod_{i=1}^n \llbracket \iota \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \llbracket \iota \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho})$. The functor p^* having a right adjoint op_v ensures that this condition holds. The Beck-Chevalley condition arises from the functoriality of substitution. Given an arrow $f: \prod_{i=1}^n \llbracket \iota \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow B$, we have, writing op_v as $\forall_p, \forall_{p \circ f}(\llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = \forall_g(\forall_f(\llbracket \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}))$.

The connectives are interpreted as follows:

$$\begin{aligned}
\llbracket \text{true}(\phi \wedge \psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} &= \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \llbracket \text{true}(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \\
\llbracket \text{true}(\phi \vee \psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} &= \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} + \llbracket \text{true}(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \\
\llbracket \text{true}(\phi \supset \psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} &= \llbracket \text{true}(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \\
\llbracket \text{true}(\perp) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} &= 1 \text{ (the initial object)} \\
\llbracket \text{true}(\neg \phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} &= \llbracket \text{true}(\perp) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \\
\llbracket \text{true}(\forall x : \iota . \phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} &= \forall_\iota \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \text{ where } \forall_\iota \text{ is the right adjoint to } p^*
\end{aligned}$$

$\llbracket \text{true}(\exists x : \iota . \phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho} = \exists_i \llbracket \text{true}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$ where \exists_i is the left adjoint to p^*

Exactly the same structure will be sufficient to ensure that the Kripke model of intuitionistic predicate logics has enough structure to interpret the alphabet of intuitionistic predicate logic, which apart from the naming of judgements has the same alphabet as classical predicate logic. The distinct between classical and intuitionistic predicate logic is obtained from the forcing condition on implication. For intuitionistic logic, we have $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \text{proof}(\phi \supset_I \psi)$ if and only if for all worlds w' such that $w \xrightarrow{f} w'$, $w', \rho[f] \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \text{proof}(\phi)$ implies $w', \rho[f] \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \text{proof}(\psi)$.

The alphabet for the classical propositional modal logic K is as follows:

- $S = \{o\}$;
- $V = \emptyset$;
- $E = \{\wedge, \vee, \supset, \perp, \Box\}$;
- $C_L = \{\wedge, \vee, \supset\}$;
- $C_G = \{\perp, \Box\}$;
- $J = \{\text{true}, \text{valid}\}$.

Here the category of judgements has an arrow $\text{valid} \rightarrow \text{true}$. A Kripke pre-structure in which the fibres are cartesian closed with coproducts and an initial object is enough to ensure that the Kripke model has enough structure to interpret \wedge, \vee, \perp and \supset . To satisfy the enough points condition for \Box we need a functor from $\mathcal{J}(W)(D)$ to itself. Once one moves to extensions of K , usually, extra conditions are imposed on this functor to make it satisfy the extra axioms, or, equivalently, the conditions on the relation R_{\Box} . An example of this is the monoidal co-monad (\Box, S, ϵ) , where $\Box : \mathcal{J}(W)(D) \rightarrow \mathcal{J}(W)(D)$ used to model \Box in (constructive) $S4$ in (Alechina, de Paiva & Ritter 1998).

We can extend our treatment of object-logic models to provide interpretations of proofs. Again, we exploit the satisfaction relation. If $\Gamma = \{y_1 : j_1(\phi_1), \dots, y_n : j_n(\phi_n)\}$, then $w, \rho, \bigwedge \Gamma \Vdash_{j(\phi)}^{\mathcal{K}_{\mathcal{J}}}$ if and only if, for all $w \xrightarrow{f} w'$, $(w', \rho[f] \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \bigwedge \Gamma$ implies $w', \rho[f] \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$), where $\bigwedge \Gamma = j_1(\phi_1) \times \dots \times j_n(\phi_n)$.

Lemma 7.23 (Satisfaction of Consequences)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$ be a Kripke model of \mathcal{O}_T . If $\Gamma = \{y_1 : j_1(\phi_1), \dots, y_n : j_n(\phi_n)\}$, then $w, \rho, \bigwedge \Gamma \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$ if and only if $w, \rho, \Gamma \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$.

Proof The same argument as Lemma 5.12 will work here. ■

Our models have enough structure to interpret not only the consequences but also the proofs, or realizers of consequence of \mathcal{O}_T . Let $\delta: ((X) y_1 : j_1(\phi_1), \dots, y_n : j_n(\phi_n) \vdash_{\mathcal{O}_T} j(\phi))$ be a proof in the axiomatization of \mathcal{O}_T . Let, for each $1 \leq i \leq n$, $\llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and $\llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ be defined. If $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$, the interpretation of δ , is defined, then it is an object

$$\left(\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \right) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

in $\mathcal{K}_{\mathcal{J}}(W)(Y)$, where $Y = \prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and $X = \{x_1 : S_1, \dots, x_n : S_n\}$. We write $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \delta: ((X) \Gamma \vdash_{\mathcal{O}_T} j(\phi))$ if and only if

$$\left(\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \right) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$$

is defined. This move to realizers is important for classical logic because the fibres of the Kripke prestructure for classical logic collapse, while the fibres of the Kripke structure for classical logic do not and so we are able to interpret and distinguish the proof-objects.

7.4 Meta-theorems for Object-logics

We now provide soundness and completeness results for object-logics. Before we do so, a few comments are needed on the axiomatization of Kripke models. The satisfaction relation for local connectives is defined in such a way that it always corresponds to the natural deduction rules for that connective as given by Prawitz's general schema. The satisfaction relation for non-local connectives has also been chosen so that it corresponds to a natural deduction rule. The satisfaction relation for universal non-local connectives corresponds to the introduction rule

$$\frac{\begin{array}{c} \llbracket j_1(\phi_1) \rrbracket \cdots \llbracket j_{n-1}(\phi_{n-1}) \rrbracket \\ \vdots \\ j_n(\phi_n) \end{array}}{\llbracket j(\#(\phi_1, \dots, \phi_n)) \rrbracket} \#-I$$

with elimination rule

$$\frac{\llbracket j(\#(\phi_1, \dots, \phi_n)) \rrbracket \quad \llbracket j_1(\phi_1) \rrbracket \cdots \llbracket j_{n-1}(\phi_{n-1}) \rrbracket}{\llbracket j_n(\phi_n) \rrbracket} \#-E$$

and the satisfaction relation for the existential non-local connectives corresponds to the introduction rule

$$\frac{j_1(\phi_1) \cdots j_n(\phi_n)}{j(\#(\phi_1, \dots, \phi_n))} \#-I$$

with elimination rule

$$\frac{j(\#(\phi_1, \dots, \phi_n)) \quad \begin{array}{c} [j_1(\phi_1)] \cdots [j_n(\phi_n)] \\ \vdots \\ k(\tau) \end{array}}{k(\tau)} \#-E$$

This is not the full picture since the relation $R_{\#}$ will correspond to an axiom of the judged proof system. This correspondence is where our general approach fails. We do not know which axiom a relation corresponds to in general. We are forced to deal with specific families of logics where we do know the relationship between $R_{\#}$ and axioms in the judged proof system. We have three families of logics which we shall analyse in detail. The first are logics which only have local connectives, so that the issue about the properties of the relation $R_{\#}$ do not arise. The second family contains minimal, intuitionistic and classical predicate logics together with all their fragments. The final family contains extensions of the classical propositional modal logic K , where the extensions are obtained by adding axioms of the form

$$\mathbf{true}(\diamond^i \Box^m \phi \supset \Box^j \diamond^n \phi)$$

which corresponds to $R_{\Box}(= R_{\diamond})$ satisfying

$$\forall x \forall y \forall z (x R_{\Box}^i y \wedge x R_{\Box}^j z \supset \exists u (y R_{\Box}^m u \wedge z R_{\Box}^n u))$$

where $x R_{\Box}^0 y$ means $x = y$ and $x R_{\Box}^{i+1} y$ means $\exists v (x R_{\Box}^i v \wedge v R_{\Box}^1 y)$. Other choices for R_{\Box} are possible but we restrict to this class because it covers all the usual modal logics and the relationship between conditions on R_{\Box} and axioms in the judged proof system is well known. This family of modal logics is called the Geach hierarchy by (Basin et al. 1997).

We can now prove soundness and completeness results for these families of logics. We initially prove soundness for the family of logics which only contain local connectives and then obtain the others as corollaries.

Theorem 7.24 (Soundness)

Let \mathcal{L}_T be a judged proof system with alphabet A which only contains local connectives. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{B}^{op}, \overline{\mathcal{V}}]]$, be a Kripke model of \mathcal{L}_T .

We have that $\Gamma \vdash_{\mathcal{L}_T} j(\phi)$ implies $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$.

Proof This is by induction on the structure of proofs in \mathcal{L}_T . We begin with

the case where $j(\phi) \in \Gamma$, that is, the inference rule is an axiom. By the induction hypothesis, we have that $\llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \in \prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. By the definition of product, we have an arrow $\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and so $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$.

We assume that the last rule used was $\#-I$. By the induction hypothesis we have that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies either $((w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{K}_{1,1}^1$ and \dots and $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{K}_{1,h_1^1}^1$) imply $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{J}_1^1$) and \dots and $((w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{K}_{p_1,1}^1$ and \dots and $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{K}_{p_1,h_{p_1}^1}^1$) imply $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{J}_{p_1}^1$) or \dots or $((w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{K}_{1,1}^s$ and \dots and $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{K}_{1,h_1^s}^s$) imply $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{J}_1^s$) and \dots and $((w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{K}_{p_s,1}^s$ and \dots and $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{K}_{p_s,h_{p_s}^s}^s$) imply $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{J}_{p_s}^s$) which by Definition 7.22 is equivalent to the following condition in terms of arrows. Either there are arrows $((1 \xrightarrow{f_{1,1}^1} \llbracket \mathbf{K}_{1,1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and \dots and $1 \xrightarrow{f_{1,h_1^1}^1} \llbracket \mathbf{K}_{1,h_1^1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \xrightarrow{f_1^1} \llbracket \mathbf{J}_1^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) and \dots and arrows $((1 \xrightarrow{f_{p_1,1}^1} \llbracket \mathbf{K}_{p_1,1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and \dots and $1 \xrightarrow{f_{p_1,h_{p_1}^1}^1} \llbracket \mathbf{K}_{p_1,h_{p_1}^1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \xrightarrow{f_{p_1}^1} \llbracket \mathbf{J}_{p_1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) or \dots or there exist arrows $((1 \xrightarrow{f_{1,1}^s} \llbracket \mathbf{K}_{1,1}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and \dots and $1 \xrightarrow{f_{1,h_1^s}^s} \llbracket \mathbf{K}_{1,h_1^s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \xrightarrow{f_1^s} \llbracket \mathbf{J}_1^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) and \dots and $((1 \xrightarrow{f_{p_s,1}^s} \llbracket \mathbf{K}_{p_s,1}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and \dots and $1 \xrightarrow{f_{p_s,h_{p_s}^s}^s} \llbracket \mathbf{K}_{p_s,h_{p_s}^s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \xrightarrow{f_{p_s}^s} \llbracket \mathbf{J}_{p_s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$), which by Definition 7.20, means $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} j(\phi)$ and therefore $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} j(\#(\phi_1, \dots, \phi_n))$.

Assuming that the last rule applied was $\#-E$, we imply the induction hypothesis to obtain $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $(w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} j(\#(e_1, \dots, e_n)))$ and $((((w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{H}_{1,1}^1$ and \dots and $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{H}_{1,h_1^1}^1$) implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{G}_1^1$) and \dots and $((w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{H}_{p_1,1}^1$ and \dots and $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{H}_{p_1,h_{p_1}^1}^1$) implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{G}_{p_1}^1$) implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{J}$) and \dots and $((((w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{H}_{1,1}^s$ and \dots and $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{H}_{1,h_1^s}^s$) implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{G}_1^s$) and \dots and $((w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{H}_{p_s,1}^s$ and \dots and $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{H}_{p_s,h_{p_s}^s}^s$) implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{G}_{p_s}^s$) implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{J}$), which by Definition 7.22 is equivalent to $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies the existence of arrows $(1 \xrightarrow{f} \llbracket j(\#(e_1, \dots, e_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) and arrows $((1 \rightarrow \llbracket \mathbf{H}_{1,1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and \dots and $1 \rightarrow \llbracket \mathbf{H}_{1,h_1^1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \rightarrow \llbracket \mathbf{G}_1^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) and \dots and arrows $(1 \rightarrow \llbracket \mathbf{H}_{p_1,1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and \dots and $1 \rightarrow \llbracket \mathbf{H}_{p_1,h_{p_1}^1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \rightarrow \llbracket \mathbf{G}_{p_1}^1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \rightarrow \llbracket \mathbf{J} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) and \dots and arrows $((1 \rightarrow \llbracket \mathbf{H}_{1,1}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and \dots and $1 \rightarrow \llbracket \mathbf{H}_{1,h_1^s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \rightarrow \llbracket \mathbf{G}_1^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) and \dots and arrows $((1 \rightarrow \llbracket \mathbf{H}_{p_s,1}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and \dots and $1 \rightarrow \llbracket \mathbf{H}_{p_s,h_{p_s}^s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \rightarrow \llbracket \mathbf{G}_{p_s}^s \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$) which imply $1 \rightarrow \llbracket \mathbf{J} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$). By Definition 7.20, we can use the left-to-right direction of condition (4) and observe that all the assumptions of the above implications correspond to arrows given by the expanded form of $1 \xrightarrow{f} \llbracket j(\#(e_1, \dots, e_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. We can conclude that $w, \rho \Vdash_{\Gamma}^{\mathcal{K}_{\mathcal{J}}}$ implies $1 \rightarrow \llbracket \mathbf{J} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and thus, by Definition 7.22, we have $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{J}$.

Assuming that the last rule was $\supset E$, we can apply the induction hypothesis to obtain $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \Gamma$ implies $(w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{J}(\phi \supset \psi)$ and $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{J}(\phi)$ and $(w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{J}(\psi)$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{J}(\tau))$). We can now apply Definitions 7.22 and 7.20 to get $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \Gamma$ implies $((1 \rightarrow \llbracket \mathbf{J}(\phi) \rrbracket_{\mathcal{K}_T}^{w, \rho}$ implies $1 \rightarrow \llbracket \mathbf{J}(\psi) \rrbracket_{\mathcal{K}_T}^{w, \rho}$) and $1 \rightarrow \llbracket \mathbf{J}(\phi) \rrbracket_{\mathcal{K}_T}^{w, \rho}$ and $(1 \rightarrow \llbracket \mathbf{J}(\psi) \rrbracket_{\mathcal{K}_T}^{w, \rho}$ implies $1 \rightarrow \llbracket \mathbf{J}(\tau) \rrbracket_{\mathcal{K}_T}^{w, \rho}$). It follows that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{J}(\tau)$.

We assume that we have used the judgement rule

$$\frac{\mathbf{j}(\phi)}{\mathbf{k}(\phi)}$$

We apply the induction hypothesis to obtain $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{j}(\phi)$. By Definition 7.22, we have an arrow $1 \xrightarrow{f} \llbracket \mathbf{j}(\phi) \rrbracket_{\mathcal{K}_T}^{w, \rho}$. The satisfaction condition in Definition 7.20 gives us an arrow $1 \xrightarrow{f'} \llbracket \mathbf{k}(\phi) \rrbracket_{\mathcal{K}_T}^{w, \rho}$ and hence $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{k}(\phi)$.

We assume that we have the introduction rule for \perp , here there are no premisses. We apply the induction hypothesis to obtain that $w, \rho \not\vdash_{\mathcal{O}_T}^{\mathcal{K}_T} \mathbf{j}(\perp)$.

If the last rule applied was $\perp E$, then we can apply the induction hypothesis to obtain $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{j}(\perp)$. Since \perp is interpreted as the initial object, if we have an arrow $1 \xrightarrow{f} \llbracket \mathbf{j}(\perp) \rrbracket_{\mathcal{K}_T}^{w, \rho}$, then we have arrow $1 \xrightarrow{f'} D$ for all objects D in $\mathcal{J}(W)(X)$. Hence w forces any formula.

Finally, we assume that we have the introduction rule for \neg . Applying the induction hypothesis tells us that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \Gamma$ implies $(w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{j}(\phi)$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{j}(\perp)$). By Definition 7.22, we have that an arrow $1 \xrightarrow{f_1} \llbracket \mathbf{j}(\phi) \rrbracket_{\mathcal{K}_T}^{w, \rho}$ implies that there is an arrow $1 \xrightarrow{f_2} \llbracket \mathbf{j}(\perp) \rrbracket_{\mathcal{K}_T}^{w, \rho}$. Definition 7.20 tells us that we have an arrow $1 \xrightarrow{f} \llbracket \mathbf{j}(\neg\phi) \rrbracket_{\mathcal{K}_T}^{w, \rho}$. Hence $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{j}(\neg\phi)$. \blacksquare

Corollary 7.25

Let \mathcal{L}_T be the judged proof system for classical predicate logic. Let $\langle \mathcal{K}_T, \llbracket - \rrbracket_{\mathcal{K}_T}^{w, \rho} \rangle$ be a Kripke model of \mathcal{L}_T . We have that $\Gamma \vdash_{\mathcal{L}_T} \mathbf{true}(\phi)$ implies $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{true}(\phi)$.

Proof We only need to deal with the cases when $\mathbf{j}(\phi)$ has been inferred from $\forall I$, $\forall E$, $\exists I$ and $\exists E$. We begin by assuming that the last rule used was $\forall I$. By the induction hypothesis we have that for all worlds w' such that $w \xrightarrow{f} w'$, all $a : \llbracket S \rrbracket_{\mathcal{K}_T}^w \rightarrow \llbracket s \rrbracket_{\mathcal{K}_T}^{w, \rho}$ and for all terms t such that $\llbracket s \rrbracket_{\mathcal{K}_T}^{w, \rho} = a$, $(w', \rho[f][x := a] \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \Gamma$ implies $w', \rho[f][x := a] \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{true}(\phi[x/t])$). By Definitions 7.22 and 7.20, we have that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{true}(\forall x : S\phi)$.

We assume that the last rule used was $\forall E$, we apply the induction hypothesis to obtain $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{true}(\forall x : S\phi)$. Definitions 7.22 and 7.20 tells us that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_T} \mathbf{true}(\phi)$.

Next, we assume that the last rule used was $\exists I$, applying the induction hypothesis tells us that there exists a world w' , where $w \xrightarrow{f} w'$, an arrow $a : \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho}$ and a term t such that $\llbracket t \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = a$, and $w', \rho[f][x := a] \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $w', \rho[f][x := a] \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{true}(\phi[t/x])$. Definition 7.22 and 7.20 tell us that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{true}(\exists x : S\phi)$.

Finally, we take the last rule to be used to be $\exists E$. Applying the induction hypothesis tells us that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{true}(\exists x : S\phi)$. By Definitions 7.22 and 7.20, we have that there exists a world w' , $w \xrightarrow{f} w'$, such that there exists an arrow $a : \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket S \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho}$ and a term t such that $\llbracket t \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} = a$ and $(w', \rho[f][x := a]) \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $w', \rho[f][x := a] \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{true}(\phi[x/t])$. It follows that $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{true}(\phi)$. \blacksquare

Corollary 7.26

Let \mathcal{L}_T be the judged proof system for minimal or intuitionistic predicate logic. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\bar{\rho}} \rangle$ be a Kripke model of \mathcal{L}_T . $\Gamma \vdash_{\mathcal{L}_T} \mathbf{proof}(\phi)$ implies $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\phi)$.

Proof We only show the cases for the rules $\supset I$ and $\supset E$, since \supset is non-local. The cases for the quantifiers are the same as Corollary 7.25. We assume that the last rule used was $\supset I$ and apply the induction hypothesis. We have that for all worlds w' , $w \xrightarrow{f} w'$, $(w', \rho[f][x := a]) \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $(w', \rho[f][x := a]) \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\psi)$ implies $w', \rho[f][x := a] \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\phi)$. We apply Definitions 7.22 and 7.20 to obtain $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\phi \supset \psi)$, Hence $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\phi \supset \psi)$.

Finally, we assume that the last rule used was $\supset E$. We apply the induction hypothesis to see that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{j}(\phi \supset \psi)$ and $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\psi)$ and $(w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\phi))$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\tau)$. We apply Definition 7.22 and 7.20 to see that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $((w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\phi))$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\psi))$ and $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\phi)$ and $(w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\psi))$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\tau)$. We observe that this is equivalent to $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{proof}(\tau)$ and we are done. \blacksquare

Corollary 7.27

Let \mathcal{L}_T be a classical modal propositional logic which is an extension of K by axioms of the form $\mathbf{valid}(\diamond^i \Box^m \phi \supset \Box^j \diamond^n \phi)$ and let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\bar{\rho}} \rangle$ be a Kripke model for \mathcal{L}_T . Then $\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi)$ implies $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{j}(\phi)$ where $\mathbf{j} \in \{\mathbf{true}, \mathbf{valid}\}$.

Proof The main part of this proof is showing that if $R_{\Box}(= R_{\diamond})$ satisfies (i, j, m, n) -convergency then $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{valid}(\diamond^i \Box^m \phi \supset \Box^j \diamond^n \phi)$. The proof we give comes from (Chellas 1980). Let x be a world in \mathcal{W} , and suppose that $x, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{valid}(\diamond^i \Box^m \phi)$. This means that

(a) For some world y , $xR_{\square}^i y$, every world u , $yR_{\square}^m u$, is such that $u, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \text{valid}(\phi)$.

We wish to prove that $x, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \text{valid}(\square^j \diamond^m \phi)$, that is,

(b) For every world z , $xR_{\square}^i z$, there is a world u , $zR_{\square}^n u$, such that $u, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \text{valid}(\phi)$.

To show this, we suppose z is a world, $xR_{\square}^i z$ and argue that there is a world u , $zR_{\square}^n u$ such that $u, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \text{valid}(\phi)$.

By our assumptions, y and z are such that $xR_{\square}^i y$ and $xR_{\square}^m z$. So, by (i, j, m, n) -convergency of R_{\square} , there is a world v such that $yR_{\square}^j v$ and $zR_{\square}^n v$. From the first half of this and (a), it follows that $v, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \text{valid}(\phi)$.

It remains to show that soundness holds for the rules for $\square I$, $\square E$, $\diamond I$ and $\diamond E$. These follow the same pattern as the other non-local connectives. \blacksquare

As usual, we obtain completeness from the construction of a term model. To make the statements of the theorems easier until the end of the chapter, unless we specify otherwise, we assume that \mathcal{L}_T is one of the following judged proof systems:

- A judged proof system which only contains local connectives;
- The judged proof system for classical, minimal or intuitionistic predicate logic;
- The judged proof system for a modal logic in the Geach hierarchy.

Lemma 7.28 (Model Existence)

Let \mathcal{L}_T be a judged proof system as specified above, then there exists a Kripke model of \mathcal{L}_T , $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, together with a world w_0 such that if $\Gamma \not\vdash_{\mathcal{L}_T} \mathbf{j}(\phi)$ then $w_0, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \Gamma$ and $w_0, \rho \not\Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{j}(\phi)$.

Proof The term model we constructed after Definition 7.20 is the required Kripke model of \mathcal{L}_T . We take $W_0 = \langle \rangle$. \blacksquare

Before we can prove completeness, we need to define validity. As usual, we write $\Gamma \Vdash_{\mathcal{L}_T} \mathbf{j}(\phi)$ if for all models and all worlds w , $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \mathbf{j}(\phi)$.

Theorem 7.29 (Completeness for \Vdash)

Let \mathcal{L}_T be a judged proof system as specified above, then $\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi)$ if and only if $\Gamma \Vdash_{\mathcal{L}_T} \mathbf{j}(\phi)$, for appropriate \mathbf{j} .

Proof

Only If This is soundness, Lemma 7.24 and Corollaries 7.25, 7.26 and 7.27.

If Suppose $\Gamma \not\vdash_{\mathcal{L}_T} \mathbf{j}(\phi)$, then Lemma 7.28 yields a contradiction. \blacksquare

We now turn to soundness for consequences.

Theorem 7.30 (Soundness for \Vdash)

Let \mathcal{L}_T be a judged proof system which only contains local connectives. Let $\langle \mathcal{L}_T, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rangle$ be a Kripke model for \mathcal{L}_T . If $\delta : (\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi))$ is a natural deduction proof and $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is defined, then $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi))$.

Proof We proceed by induction on the structure of proof-terms of \mathcal{L}_T . The case where $\delta = \text{HYP}_\phi(y)$ follows from the fact that every product comes with projections. We assume that the last rule used was $v\text{-}\#$ -I. We apply the induction hypothesis and Definition 7.22 to obtain one of $1 \leq i \leq s$ sets of arrows

$\{\prod_{l=1}^n \llbracket \mathbf{j}_l(\phi_l) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times (\prod_{k=1}^{h_j^i} \llbracket \mathbf{K}_{j,k}^i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \delta_j^i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket \mathbf{J}_j \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \mid 1 \leq j \leq p_i\}$. Each of these arrows are in different fibres of $\mathcal{K}_{\mathcal{J}}$. We can apply $op\#$ to the objects $\llbracket \mathbf{J}_j^i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ to obtain

an arrow $\prod_{i=1}^n \llbracket \mathbf{j}_n(\phi_n) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \# \text{-I} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket \mathbf{j}(\#(\phi_1, \dots, \phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ in the fibre $\mathcal{K}_{\mathcal{J}}(w)(X)$, where $X = \prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and the free variables of $\mathbf{j}_1(\phi_1), \dots, \mathbf{j}_n(\phi_n)$, $\mathbf{j}(\#(\phi_1, \dots, \phi_n))$ are in the set $X' = \{x_1 : S_1, \dots, x_n : S_n\}$.

We assume that $\delta = v\text{-}\#$ -E. We apply the induction hypothesis and Definition 7.22 to obtain an arrow $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket \mathbf{j}(\#(\phi_1, \dots, \phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and s arrows $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \times \llbracket \Gamma_j \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket \mathbf{K} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ for $1 \leq j \leq s$. By Definition 7.20 and the definition of Γ_i , we observe that the Γ_i correspond to clauses in the right hand side of the implication in condition (4) of Definition 7.20, so that we obtain

an arrow $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket v\text{-}\#$ -E \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket \mathbf{k}(\tau) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}.

We assume that $\delta = v\text{-}\supset$ -E, we apply the induction hypothesis to obtain $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi \supset \psi))$, $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_1} \delta_1 : (\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi))$ and $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \delta_2 : (\Gamma, y_2 : \mathbf{j}(\psi) \vdash_{\mathcal{L}_T} \mathbf{j}(\tau))$. By Definitions 7.22 and 7.20, we have that $(\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket \mathbf{j}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ implies $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket \mathbf{j}(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket \mathbf{j}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and $(\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket \mathbf{j}(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ implies $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket \mathbf{j}(\tau) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$). We thus have that $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rightarrow \llbracket \mathbf{j}(\tau) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$, that is, $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} v\text{-}\supset$ -E($\delta, \delta_1, y_2 : \delta_2$) : $(\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\tau))$.

We assume that $\delta = \text{BOT}$ -I. We apply the induction hypothesis to see that there are no arrows into $\llbracket \perp \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$, as required.

We assume that $\delta = \text{BOT}$ -E. We apply the induction hypothesis and obtain an arrow $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\delta} \llbracket \mathbf{j}(\perp) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and since $\llbracket \mathbf{j}(\perp) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is the initial object, there is an arrow to every object in $\mathcal{K}_{\mathcal{J}}$ and so we can compose them to obtain an arrow $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \text{BOT-E} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} D$ as required.

Next, we assume that $\delta = \text{NEG}$ -I and we apply the induction hypothesis to obtain $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket \mathbf{j}(\perp) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. We can curry here to obtain $\prod_{i=1}^n \llbracket \mathbf{j}_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \text{NEG-I} \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket \mathbf{j}(\neg\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and can conclude that $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_{\mathcal{L}_T}$

$j(\phi)$.

We assume that $\delta = NEG-E$. We apply the induction hypothesis to obtain arrows $\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket j(\neg\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$, $\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \delta_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket j_1(\phi_1) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ and $\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \delta_2 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket k(\tau) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. We use Definition 7.20 and the evaluation arrow twice to obtain an arrow $\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket NEG-E \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket k(\psi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$.

Finally, the case for a judgement rule, follows from the fact that there is an arrow $j \rightarrow k$ in the category of judgements. \blacksquare

Corollary 7.31

Let \mathcal{L}_T be the judged proof system for classical predicate logic and $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rangle$ be a Kripke model of \mathcal{L}_T . If $\delta : (\Gamma \vdash_{\mathcal{L}_T} \mathbf{true}(\phi))$ be a natural deduction proof and $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is defined then $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_{\mathcal{L}_T} \mathbf{true}(\phi))$.

Proof (Sketch) We only need to prove the cases when δ is either *FORALL-I*, *FORALL-E*, *EXISTS-I* or *EXISTS-E*. We just show *FORALL-I*, since the rest are similar. We apply the induction hypothesis to obtain the arrow

$\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket j(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. We first apply the natural transformation induced by the arrow $w \xrightarrow{f} w'$ in \mathcal{W} and then the right adjoint \forall_S to obtain $\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho[f][x:=a]}$
 $\xrightarrow{\forall_S \llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho[f][x:=a]}} \llbracket \forall x : S \phi \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. \blacksquare

We do not prove soundness for minimal and intuitionistic logic because the only difference between the proof and that of the above Corollary is the way implication is handled and the proof is straightforward.

Corollary 7.32

Let \mathcal{L}_T be the judged proof system for classical propositional modal logic *K* extended with axioms of the form $\mathbf{valid}(\diamond^i \Box^m \phi \supset \Box^j \diamond^n \phi)$. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho} \rangle$ be a Kripke model of \mathcal{L}_T . If $\delta : (\Gamma \vdash_{\mathcal{L}_T} j(\phi))$ be a natural deduction proof $\Gamma \vdash_{\mathcal{L}_T} j(\phi)$ and $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$ is defined then $w, \rho, \Gamma \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \delta : (\Gamma \vdash_{\mathcal{L}_T} j(\phi))$, where $j \in \{\mathbf{true}, \mathbf{valid}\}$.

Proof (Sketch) We just need to show the cases where δ is *BOX-I*, *BOX-E*, *LOZENGE-I* and *LOZENGE-E*. The case where δ is a natural deduction proof $\Gamma \vdash_{\mathcal{L}_T} \mathbf{valid}(\diamond^i \Box^m \phi \supset \Box^j \diamond^n \phi)$ follows from the argument given in Corollary 7.27. We just show the case for *BOX-I*, the others being similar. We apply

the induction hypothesis to obtain $(\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}} \llbracket \mathbf{valid}(\phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho}$. We apply the natural transformation induced by $w \xrightarrow{f} w'$ in \mathcal{W} and then the functor \Box to obtain an arrow $(\prod_{i=1}^n \llbracket j_i(\phi_i) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho[f][x:=a]}) \xrightarrow{\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w',\rho[f][x:=a]}} \llbracket \mathbf{valid}(\Box \phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w,\rho[f][x:=a]}$. \blacksquare

Lemma 7.33 (Model Existence)

Let \mathcal{L}_T be one of the following:

- A judged proof system which only contains local connectives;
- The judged proof system for classical, minimal or intuitionistic predicate logic;
- The judged proof system for a modal logic in the Geach hierarchy.

There exists a Kripke model of \mathcal{L}_T , $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, together with a world w_0 where, if $\delta: (\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi))$ is not a natural deduction proof, $w_0, \rho \not\Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \delta$,

Proof We use the term model constructed after Definition 7.20 and take $w_0 = \langle \rangle$. ■

Theorem 7.34 (Completeness for \Vdash)

Let \mathcal{L}_T be one of the following:

- A judged proof system which only contains local connectives;
- The judged proof system for classical, minimal or intuitionistic predicate logic;
- The judged proof system for a modal logic in the Geach hierarchy.

Let $\delta: (\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi))$ be a natural deduction proof. Then $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$ is defined if and only if $w, \rho \Vdash_{\mathcal{L}_T}^{\mathcal{K}_{\mathcal{J}}} \delta: (\Gamma \vdash_{\mathcal{L}_T} \mathbf{j}(\phi))$.

Proof

Only If By soundness, Lemma 7.30 and Corollaries 7.31 and 7.32.

If Suppose $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$ is not defined, then Lemma 7.33 yields a contradiction. ■

Chapter 8

Judgements-as-types Correspondence

After providing providing proof- and model-theoretic characterizations of object-logics which are suitable for representing in LF, we now turn to a characterization of the representation mechanism. We restrict our attention to the judgements-as-types representation mechanism, recalling that the previous chapter was developed with this in mind. We begin by describing the usual method of representing object-logics in LF (*cf.* (Harper et al. 1993), (Avron et al. 1992), (Avron et al. 1997), *etc.*), using an encoding function induced by the judgements-as-types correspondence. We then turn our attention to the model-theoretic counterpart of this encoding. The judgements-as-types correspondence induces an (indexed) epimorphism between Kripke models of the object-logic and Kripke models of the $\lambda\Pi$ -calculus. We conclude this chapter with a section on representation theorems. A representation theorem proves that a representation is adequate, that is, every proof in the object-logic has a corresponding proof in LF (*fullness*) and every derivation in LF is (essentially) the representation of a proof in the object-logic (*faithfulness*).

Traditionally, adequacy has been proven proof-theoretically: fullness is proven by induction over the structure of proofs in the object-logic; and, faithfulness is proven by analysis of normal forms for derivations in LF. Faithfulness proofs are often quite involved. We claim that faithfulness should follow (intuitively) from the semantics of LF. This idea is not new and can be found in (Simpson 1993). We provide evidence for this claim by using the judgements-as-types epimorphism to establish faithfulness. We prove faithfulness by constructing Kripke models of the $\lambda\Pi$ -calculus out of the encoded syntax of the object-logic. We then use soundness, the judgements-as-types epimorphism and completeness of Kripke models of the object-logic to complete the proof.

The judgements-as-types encoding is not original work, the generality of our presentation is, however, original. The section on the judgements-as-types epi-

morphism is original. We do not establish any new adequacy results, apart from a couple of fairly obscure modal logics, and the originality of our work is in the methodology not the result.

Our work is similar to Gardner (1992b), (Gardner 1992a). She presents indexed categories as models of LF^+ (LF restricted to canonical terms) and object-logics and shows that adequate encodings induce (indexed) isomorphisms between the models. She does not, however, work with LF or have any notion of satisfaction in her models, so is unable to provide model-theoretic proofs of faithfulness.

8.1 Syntactic Judgements-as-types: Encoding Logics in LF

The judgements-as-types correspondence tells us that the basic judgements of an object-logic correspond to primitive types in the $\lambda\Pi$ -calculus and hypothetico-general judgements correspond to Π -types.

One of the main reasons for using logical frameworks is that the handling of variables and binding is dealt with by the variables and binding of the framework. Recalling the definition of expressions in a judged proof system, we see that the expressions were generated by abstraction and application. These correspond to abstraction and application in the $\lambda\Pi$ -calculus. Presentations of syntax in this manner are often called higher-order abstract syntax (*cf.* (Pfenning & Elliott 1988)).

Definition 8.1 (Encoding)

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 signature $\Sigma_{\mathcal{O}_T}$, which determines a pair $\epsilon = (\epsilon_s, \epsilon_j)$ of functions. ϵ_s maps the syntax of \mathcal{O}_T to $\lambda\Pi$ -terms and ϵ_j maps the proof expressions of \mathcal{O}_T to $\lambda\Pi$ -terms. Thus we obtain that

$$\underbrace{(x_1 : S_1, \dots, x_n : S_n)}_X \underbrace{y_1 : j_1(\phi_1), \dots, y_m : j_m(\phi_m)}_\Delta \vdash_{\mathcal{O}_T} \delta : j(\phi)$$

gets sent to

$$\underbrace{\epsilon_s(x_1 : S_1), \dots, \epsilon_s(x_n : S_n)}_{\Gamma_X}, \underbrace{\epsilon_s(y_1 : j_1(\phi_1)), \dots, \epsilon_s(y_m : j_m(\phi_m))}_{\Gamma_\Delta} \vdash_{\Sigma_{\mathcal{O}_T}} \epsilon_j(\delta) : \epsilon_s(j(\phi))$$

An alphabet $A = (S, V, E, C, J)$ is encoded as follows:

- Let $s \in S/V$: if s has zero arity, then s is encoded by a constant $s : \text{Type} \in \Sigma_{\mathcal{O}_T}$ or by a constant $s : s' \in \Sigma_{\mathcal{O}_T}$, where $s' : \text{Type}$ is declared to the left of

$s : s'$. If s has arity m , then s is encoded in LF either by a constant

$$s : (s_1 \rightarrow \dots \rightarrow s_m) \rightarrow \mathbf{Type} \in \Sigma_{\mathcal{O}_T}$$

or by a constant

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

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

- Let $s \in V$: a variable $x : V_s$ (with zero arity) 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 $(a_1, \dots, a_m) \rightarrow s$, then e is encoded in LF by a constant

$$e : a_1 \rightarrow \dots \rightarrow a_m \rightarrow s \in \Sigma_{\mathcal{O}_T}$$

where if any a_i are of the form $a \rightarrow a'$, we put brackets around the a_i and if any of the a_i stand for more than one syntactic category, we quantify the whole expression by $\Pi a_i : \mathbf{Type}$, for each a_i ranging over multiple syntactic categories;

- Let $j \in J$: if j has arity (s_1, \dots, s_m) , then j is encoded in LF by a constant

$$j : s_1 \rightarrow \dots \rightarrow s_m \rightarrow \mathbf{Type} \in \Sigma_{\mathcal{O}_T}.$$

Application in L is encoded in LF by application in the $\lambda\Pi$ -calculus and abstraction in L is encoded in LF by abstraction in the $\lambda\Pi$ -calculus.

1. Hilbert-type systems:

- An axiom Ax of the form $j(e(\phi_1, \dots, \phi_n))$ is encoded in LF by a constant of the form

$$Ax : \Pi p_1 : o_1 \dots \Pi p_m : o_m \cdot j(e_1(p_1, \dots, p_m)) \in \Sigma_{\mathcal{O}_T}$$

where each o_i is one of the syntactic categories of propositions distinguished in A ;

- A rule R of the form

$$\frac{J_1 \dots J_m}{J} R$$

where each J_i and J are basic judgements, is encoded in LF by a constant of the form

$$R : \Pi p_1 : o_1 \dots \Pi p_r : o_r \cdot J_1 \rightarrow \dots \rightarrow J_m \rightarrow J \in \Sigma_{\mathcal{O}_T}$$

where each o_i is one of the distinguished syntactic categories of propositions distinguished in A , and there are r distinct formulæ $\phi_i : o_i$ in the rule, with each occurrence replaced by $p_i : o_i$ in the constant.

2. Natural deduction systems:

- An axiom Ax of the form $j(e(\phi_1, \dots, \phi_n))$ is encoded in LF by a constant of the form

$$Ax : \Pi p_1 : o_1 \dots \Pi p_m : o_m \cdot j(e_1(p_1, \dots, p_m)) \in \Sigma_{\mathcal{O}_T}$$

where each o_i is one of the syntactic categories of propositions distinguished in A ;

- The $\#$ -introduction rule schema of the form

$$\frac{\begin{array}{ccc} & [H_{j,1}^i] \cdots [H_{j,h_j^i}^i] & \\ \vdots & \vdots & \vdots \\ G_1^i & G_j^i & G_{p_i}^i \end{array}}{J} \# I$$

where each G_j^i , $H_{j,k}^i$ and $J = j(\#(e_1, \dots, e_n))$ are basic judgements, is encoded in LF by a constant of the form

$$\#-I : \Pi p_1 : o_1 \dots \Pi p_m : o_m \cdot ((H_{1,1}^i \rightarrow \dots \rightarrow H_{1,h_1^i}^i) \rightarrow G_1^i) \rightarrow \dots \rightarrow ((H_{p_i,1}^i \rightarrow \dots \rightarrow H_{p_i,h_{p_i}^i}^i) \rightarrow G_{p_i}^i) \rightarrow J \in \Sigma_{\mathcal{O}_T}$$

where each o_i is one of the syntactic categories of propositions distinguished in A and there are m distinct formulæ $\phi_i : o_i$ in the schema, each replaced by $p_i : o_i$ in the encoded rule;

- The $\#$ -elimination rule schema of the form

$$\frac{\begin{array}{ccc} & [\Gamma_1] & [\Gamma_s] \\ & \vdots & \vdots \\ K & J & \dots & J \end{array}}{J} \# E$$

where each Γ_i is $\bigwedge_{k=1}^{h_1^i} H_{1,k}^i \supset G_1^i, \dots, \bigwedge_{k=1}^{h_{p_i}^i} H_{1,k}^i \supset G_{p_i}^i$ and each $K = k(\#(e_1, \dots, e_n))$, J , $H_{j,k}^i$ and G_j^i are basic judgements, is encoded in LF by a constant of the form

$$\#-E : \Pi p_1 \dots \Pi p_m : o_m \cdot (K \rightarrow (\Gamma_1 \rightarrow J) \rightarrow \dots \rightarrow (\Gamma_s \rightarrow J)) \rightarrow J$$

$$\in \Sigma_{\mathcal{O}_T}$$

where each o_i is one of the syntactic categories of propositions distinguished in A and there are m distinct formulæ $\phi_i : o_i$ in the rule each replaced by $p_i : o_i$ in the encoded rule and, for brevity, we have elided the currying of the Γ_i 's;

- The \supset elimination rule

$$\frac{\begin{array}{c} [\mathbf{j}(\psi)] \\ \vdots \\ \mathbf{j}(\phi \supset \psi) \quad \mathbf{j}(\phi) \quad \mathbf{j}(\tau) \end{array}}{\mathbf{j}(\tau)} \supset E$$

is encoded in LF by a constant of the form

$$\supset -E : \Pi p : o . \Pi p : o . \Pi r : o . (\mathbf{j}(\supset pq) \rightarrow \mathbf{j}(p) \rightarrow (\mathbf{j}(q) \rightarrow \mathbf{j}(r))) \rightarrow \mathbf{j}(r).$$

The function ϵ_s is defined inductively on the structure of syntactic categories, expressions and basic judgements of the language L generated by A (cf. Definitions 7.5, 7.7 and 7.8):

- For a syntactic category $sc_1 \dots c_m$, $\epsilon_s(sc_1 \dots c_m) = \epsilon_s(s)\epsilon_s(c_1) \dots \epsilon_s(c_m)$, where $s \in S$ and c_1, \dots, c_m are syntactic categories;
- Expression application: $\epsilon_s(ee_1 \dots e_n) = \epsilon_s(e)\epsilon_s(e_1) \dots \epsilon_s(e_n)$, where e and each e_i are expressions;
- Expression abstraction: $\epsilon_s((x_1, \dots, x_m)e) = \lambda x_1 : s_1 . \dots . \lambda x_m : s_m . \epsilon_s(e)$, where e is an expression and each $x_i \in V_{s_i}$;
- For a basic judgement $\mathbf{j}(e_1, \dots, e_m)$, $\epsilon_s(\mathbf{j}(e_1, \dots, e_m)) = \epsilon_s(\mathbf{j})\epsilon_s(e_1) \dots \epsilon_s(e_m)$.

Also, a context $(x_1, \dots, x_m) y_1 : \mathbf{j}_1(\phi_1), \dots, y_n : \mathbf{j}_n(\phi_n)$ is encoded by ϵ_s as

$$\underbrace{x_1 : s_1, \dots, x_m : s_m}_{\Gamma_X}, \underbrace{y_1 : \epsilon_s(\mathbf{j}_1(\phi_1)), \dots, y_n : \epsilon_s(\mathbf{j}_n(\phi_n))}_{\Gamma_\Delta}.$$

ϵ_j is defined inductively over proof expressions:

- For $HYP_\phi(y)$, an hypothesis, $\epsilon_j(HYP_\phi(y)) = y : \mathbf{j}(\phi)$;
- For $\#-I$ or $\#-E$ applied to proofs, we describe the function in two steps:

1. Instantiation of variables, formulæ and terms. These are written below the proof expression. If a variable is present it is abstracted. For example,

$$\epsilon_j(\#-I_{x_1, \dots, x_n, \phi_1, \dots, \phi_n, x_{n+1}, \phi}) = \#-I(\lambda x_1 : s_1 . \dots . \lambda x_n : s_n . \epsilon_s(\phi_1))\epsilon_s(\phi_2)$$

$$\dots \epsilon_s(\phi_n)(\lambda x_{n+1} : s_{n+1} . \epsilon_s(\phi))$$

2. Application of proof-terms. Any discharged hypothesis is λ -abstracted. For example,

$$\begin{aligned} \epsilon_j(\#-I(\Pi_1, \dots, \Pi_n, (y_1, \dots, y_n) : \Pi)) = \#-I(\epsilon_j(\Pi_1), \dots, \epsilon_j(\Pi_n)(\lambda y_1 : \epsilon_s \\ (j_1(\phi_1)) \dots \lambda y_n : \epsilon_s(j_n(\phi_n) . \epsilon_j(\Pi))) \end{aligned}$$

If we have a proof which contains a formula with a free variable, then we have to abstract this variable as well, using scoping to capture which formulæ it is free in. ■

As we mentioned earlier, abstraction and application in the $\lambda\Pi$ -calculus handle any application and abstraction in the object-logic. We have the following example:

In classical logic we have $(X, x : \iota) \vdash_{\mathcal{O}_T} \phi : o$, which is encoded in LF as $\Gamma_X, x : \iota \vdash_{\Sigma_{\mathcal{O}_T}} \epsilon_s(\phi) : o$. We can abstract to obtain $\Gamma_X \vdash_{\Sigma_{\mathcal{O}_T}} \lambda x : \iota . \epsilon_s(\phi) : \iota \rightarrow o$, since x is free in o . Since $\forall : (\iota \rightarrow o) \rightarrow o \in \Sigma_{\mathcal{O}_T}$, $\Gamma_X \vdash_{\mathcal{O}_T} \forall : (\iota \rightarrow o) \rightarrow o$. Application gives $\Gamma_X \vdash_{\mathcal{O}_T} \forall(\lambda x : \iota . \epsilon_s(\phi)) : o$, which is the encoding of $X \vdash_{\mathcal{O}_T} \forall x : \iota . \phi : o$.

λ -abstraction is also used to handle discharge in natural deduction rules. Take, for example, the rule for $\supset I$ in classical logic, which gets encoded as the constant

$$IMP-I : \Pi p : o . \Pi q : o . (\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(p \supset q) \in \Sigma_{\mathcal{O}_T}$$

Given $\Gamma_X, \Gamma_\Delta \vdash_{\mathcal{O}_T} M_\delta : \text{true}(\psi)$ in LF, we abstract to obtain $\Gamma_X, \Gamma_\Delta \vdash_{\mathcal{O}_T} \lambda y : \text{true}(\phi) . M_\delta : \text{true}(\phi) \rightarrow \text{true}(\psi)$, which can then be applied to (the instantiated) constant $IMP-I$ to obtain $\Gamma_X, \Gamma_\Delta \vdash_{\mathcal{O}_T} IMP-I \epsilon_s(\phi) \epsilon_s(\psi) (\lambda y : j(\phi) . M_\delta) : \text{true}(\phi \supset \psi)$, which is the encoding of $(X) \Delta \vdash_{\mathcal{O}_T} IMP-I \epsilon_s(\phi) \epsilon_s(\psi) ((y) : \delta) : \text{true}(\phi \supset \psi)$.

Definition 8.1 is general enough to encompass the worlds-as-parameters representation mechanism. We are able to encode judgements of the form $U \rightarrow o \rightarrow \text{Type}$, where U is a syntactic category of worlds. We provide a detailed treatment of worlds-as-parameters in § 9.

Theories in which induction is restricted; for example, to Σ_1^0 -induction, present a further syntactic challenge for the definition of an object-logic. In an informal meta-theory, we restrict our attention to induction formulæ of the appropriate class; for example, Σ_1^0 -induction. Our object-logics, however, are intended to be the logics which can be adequately represented in LF. The strength of the induction of the theories built on top of an object-logic are restricted by the meta-logic. We thus have a restriction imposed on us by the $\lambda\Pi$ -calculus and so we cannot adequately represent theories whose induction is stronger than Π_2 -sentences.

We provide the $\lambda\Pi$ -signature Σ_{cL} for classical predicate logic with the $\lambda\Pi$ -signature for minimal and intuitionistic predicate logic, our usual family of classi-

cal propositional modal logics and higher-order logic being given in Appendix A.

Definition 8.2 (Σ_{CL})

The $\lambda\Pi$ -signature Σ_{CL} contains the following constants:

$$\begin{aligned}
& o : \text{Type} \\
& \text{true} : o \rightarrow \text{Type} \\
& \wedge : o \rightarrow o \rightarrow o \\
& \vee : o \rightarrow o \rightarrow o \\
& \supset : o \rightarrow o \rightarrow o \\
& \neg : o \rightarrow o \\
& \perp : o \\
& \forall : (\iota \rightarrow o) \rightarrow o \\
& \exists : (\iota \rightarrow o) \rightarrow o \\
& Ex : \Pi p : o . \text{true}(p \vee \neg p) \\
& \wedge I : \Pi p : o . \Pi q : o . \text{true}(p) \rightarrow \text{true}(q) \rightarrow \text{true}(p \wedge q) \\
& \vee I1 : \Pi p : o . \Pi q : o . \text{true}(p) \rightarrow \text{true}(p \vee q) \\
& \vee I2 : \Pi p : o . \Pi q : o . \text{true}(q) \rightarrow \text{true}(p \vee q) \\
& \supset I : \Pi p : o . \Pi q : o . (\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(p \supset q) \\
& \neg I : \Pi p : o . (\text{true}(p) \rightarrow \text{true}(\perp)) \rightarrow \text{true}(\neg p) \\
& \forall I : \Pi F : \iota \rightarrow o . (\Pi x : \iota . \text{true}(Fx)) \rightarrow \text{true}(\forall(\lambda x : \iota . Fx)) \\
& \exists I : \Pi F : \iota \rightarrow o . (\Pi x : o . \text{true}(Fx)) \rightarrow \text{true}(\exists(\lambda x : \iota . Fx)) \\
& \wedge E : \Pi p : o . \Pi q : o . \Pi r : o . \text{true}(p \wedge q) \rightarrow ((\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(r)) \rightarrow \text{true}(r) \\
& \vee E : \Pi p : o . \Pi q : o . \Pi r : o . \text{true}(p \vee q) \rightarrow (\text{true}(p) \rightarrow \text{true}(r)) \rightarrow (\text{true}(q) \rightarrow \text{true}(r)) \\
& \quad \rightarrow \text{true}(r) \\
& \supset E : \Pi p : o . \Pi q : o . \Pi r : o . \text{true}(p \supset q) \rightarrow \text{true}(p) \rightarrow (\text{true}(q) \rightarrow \text{true}(r)) \rightarrow \text{true}(r) \\
& \quad \neg E : \Pi p : o . \Pi q : o . \text{true}(\neg p) \rightarrow (\text{true}(\phi \supset \perp) \rightarrow \text{true}(q)) \rightarrow \text{true}(q) \\
& \forall E : \Pi F : \iota \rightarrow o . \Pi x : \iota . \Pi r : o . \text{true}(\forall(\lambda x : \iota . Fx)) \rightarrow (\text{true}(Fx) \rightarrow \text{true}(r)) \\
& \quad \rightarrow \text{true}(r) \\
& \exists E : \Pi F : \iota \rightarrow o . \Pi r : o . \text{true}(\exists(\lambda x : \iota . Fx)) \rightarrow (\Pi x : \iota . \text{true}(Fx) \rightarrow \text{true}(r)) \\
& \quad \rightarrow \text{true}(r)
\end{aligned}$$

■

We need to show that our encoding sends every proof in the object-logic to a derivation in LF and that every derivation in LF corresponds to a proof in the object-logic. The next two definitions are closely related to those in (Harper et al. 1994) and (Simpson 1993).

Definition 8.3 (Full and Faithful Encodings)

An encoding of an object-logic \mathcal{O}_T is *full* if the valid proof

$$\underbrace{(x_1 : S_1, \dots, x_n : S_n)}_X \quad \underbrace{y_1 : j_1(\phi_1), \dots, y_m : j_m(\phi_m)}_\Delta \vdash_{\mathcal{O}_T} \delta : j(\phi)$$

in \mathcal{O}_T implies the derivation

$$\underbrace{\epsilon_s(x_1 : S_1), \dots, \epsilon_s(x_n : S_n)}_{\Gamma_X}, \underbrace{\epsilon_s(y_1 : j_1(\phi_1)), \dots, \epsilon_s(y_m : j_m(\phi_m))}_{\Gamma_\Delta} \vdash_{\mathcal{O}_T} \epsilon_j(\delta) : \epsilon_s(j(\phi))$$

in LF, where $\epsilon_j(\delta)$ is in long $\beta\eta$ -normal form.

An encoding of an object-logic is *faithful* if the derivation

$$\Gamma \vdash_{\Sigma_{\mathcal{O}_T}} M : A$$

in LF, where M is in long $\beta\eta$ -normal form, implies the proof

$$(X) \Delta \vdash_{\mathcal{O}_T} \delta : j(\phi)$$

is valid, where $\epsilon_j(X) = \Gamma_X$, $\epsilon_j(\Delta) = \Gamma_\Delta$, $\Gamma = \Gamma_X, \Gamma_\Delta$, $\epsilon_j(\delta) = M$ and $\epsilon_s(j(\phi)) = A$.

An encoding is *adequate* if it is both full and faithful. \blacksquare

The terms full and faithful are not, perhaps, the best terminology here. When we start ‘thinking semantically’, in the next section, the notion of full and faithful functor may provide false intuition. We instead suggest *relative soundness* and *relative completeness* as better terminology but keep the traditional terminology to avoid confusion.

Definition 8.4 (Encoding Uniformly)

An encoding of an object-logic is *uniformly full* if the encoding is full and ϵ is surjective, and *uniformly faithful* if the encoding is faithful and ϵ is surjective. \blacksquare

In (Harper et al. 1994), the term ‘uniform encoding’ is used to denote a stronger property than our ‘uniformly faithful’, requiring a quantification over all possible signatures $\Sigma_{\mathcal{O}_T}$ which ‘present’ the logic \mathcal{O}_T . The details of this approach to the representation of logics, described in (Harper et al. 1994), are beyond the scope of the thesis.

From now on, we require every encoding to be uniformly full because we need this property when we establish faithfulness model-theoretically.

Lemma 8.5 (Fullness)

Let \mathcal{O}_T be a judged proof system with either natural deduction or Hilbert-type rules. Then the encoding ϵ given in Definition 8.1 is full.

Proof (Sketch) By induction on the structure of proofs in \mathcal{O}_T and observing that Definition 8.1 covers all the possible cases. ■

8.2 Semantic Judgements-as-types: an Epimorphism of Models

Our work in this section is similar to that of Gardner (1992b). She obtains an indexed isomorphism between models of judged object-logics and models of the $\lambda\Pi$ -calculus rather than the epimorphism because she restricts the $\lambda\Pi$ -calculus to long $\beta\eta$ -normal terms; that is, those terms which represent object-logics. We do not make such a restriction and thus obtain an epimorphism.

Finally, we are able to set up what we shall call the judgements-as-types epimorphism between suitable Kripke models, induced by the judgements-as-types correspondence. We recall the definition of an indexed functor (Definition 5.16) and indexed isomorphisms (Definition 5.17). We need to introduce a new definition, that of an indexed epimorphism.

Definition 8.6 (Indexed Epimorphism)

An indexed functor $\tau = (\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$ is an *indexed epimorphism* if α and β are epimorphisms and each ϵ_w is a natural epimorphism. ■

We now provide a new definition of the category of models because we are now working with a more general notion of Kripke model of object-logics.

Definition 8.7 (Category of Models)

We define the category \mathcal{M} of *models* as follows:

Objects: each object of \mathcal{M} is either a Kripke $\Sigma_{\mathcal{O}_T}$ - $\lambda\Pi$ -model or a Kripke model of \mathcal{O}_T ;

Arrows: there are four cases:

1. An arrow

$$\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \xrightarrow{h} \langle \mathcal{K}'_{\mathcal{J}'}, \llbracket - \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^- \rangle$$

is given by an indexed functor $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|}) : \mathcal{K}_{\mathcal{J}} \rightarrow \mathcal{K}'_{\mathcal{J}'}$ such that, if $\alpha w = w'$, then $h(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w) = \llbracket X \rrbracket_{\mathcal{K}'_{\mathcal{J}'}}^{w'}$;

2. An arrow

$$\langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle \xrightarrow{h} \langle \mathcal{R}'_{\mathcal{S}'}, \llbracket - \rrbracket_{\mathcal{R}'_{\mathcal{S}'}}^{-,\rho'} \rangle$$

is given by an indexed functor $(\alpha, \beta, (\epsilon_x)_{x \in |\mathcal{X}|}) : \mathcal{R}_{\mathcal{S}} \rightarrow \mathcal{R}'_{\mathcal{S}'}$ such that, if $\alpha x = x'$, then $h(\llbracket X \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{x,\rho}) = \llbracket X \rrbracket_{\mathcal{R}'_{\mathcal{S}'}}^{x',\rho'}$;

3. An arrow

$$\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \xrightarrow{h} \langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-,\rho} \rangle$$

is given by an indexed functor $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|}) : \mathcal{K}_{\mathcal{J}} \rightarrow \mathcal{R}_{\mathcal{S}}$ such that, if $\alpha w = x$, then $h(\llbracket \epsilon(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w) = \llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{x,\rho}$;

4. An arrow

$$\langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle \xrightarrow{h} \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$$

is given by an indexed functor $(\alpha, \beta, (\epsilon_x)_{x \in |\mathcal{X}|}) : \mathcal{R}_{\mathcal{S}} \rightarrow \mathcal{K}_{\mathcal{J}}$ such that, if $\alpha x = w$, then $h(\llbracket X \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{x,\rho}) = \llbracket \epsilon(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w$. ■

Lemma 8.8 (\mathcal{M} is well-defined)

The category \mathcal{M} described in Definition 8.7 is well-defined.

Proof (Sketch) Is essentially the same as that given in the proof of Lemma 5.19. ■

The definition of Kripke prestructures and prestructures, for both the $\lambda\Pi$ -calculus and the judged object-logic, involve the categories satisfying certain properties. The parts of the categories which satisfy these properties are those which interpret the syntax of either the $\lambda\Pi$ -calculus or the judged object-logic. We restrict our attention to morphisms between these parts of the Kripke models.

We are now in a position to define an epimorphism of models.

Definition 8.9 (Epimorphism of Models)

Let \mathcal{O}_T be a judged object-logic. Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$ be a Kripke $\Sigma_{\mathcal{O}_T}$ - $\lambda\Pi$ -model and $\langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle$ be a Kripke model of \mathcal{O}_T . Let $h : \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \rightarrow \langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle$ be a morphism of models. We say that h is an *epimorphism of models* if the indexed functor $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|}) : \mathcal{K}_{\mathcal{J}} \rightarrow \mathcal{R}_{\mathcal{S}}$ (corresponding to h) is an indexed epimorphism when its domain is restricted to those objects and arrows in $\mathcal{K}_{\mathcal{J}}$ which interpret the syntax of the $\lambda\Pi$ -calculus and its range is restricted to those objects and arrows in $\mathcal{R}_{\mathcal{S}}$ which interpret the syntax of \mathcal{O}_T . ■

Proposition 8.10 (Judgements-as-types Epimorphism)

Let \mathcal{O}_T be a judged object-logic as defined in Definition 7.13 and let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, 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 . Then, there is a Kripke model of \mathcal{O}_T , $\langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle$, where $\mathcal{R}_{\mathcal{S}} : [\mathcal{X}, [\mathcal{E}^{op}, \bar{\mathcal{U}}]]$, together with an epimorphism of models

$$h : \langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle \rightarrow \langle \mathcal{R}_{\mathcal{S}}, \llbracket - \rrbracket_{\mathcal{R}_{\mathcal{S}}}^{-,\rho} \rangle$$

induced by the judgements-as-types correspondence. Specifically, abusing notation by suppressing information about worlds, if $\llbracket X \rrbracket_{\mathcal{R}_{\mathcal{S}}}$ and $\llbracket \epsilon(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}$ are defined, then

$$h(\llbracket \epsilon(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}) = \llbracket X \rrbracket_{\mathcal{R}_{\mathcal{S}}}.$$

Proof (Sketch) Given $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, we sketch the construction of $\mathcal{R}_{\mathcal{S}} : [\mathcal{X}, [\mathcal{E}^{op}, \bar{\mathcal{U}}]]$, together with an indexed epimorphism $(\alpha, \beta, (\epsilon_x)_{x \in |\mathcal{X}|}) : \mathcal{K}_{\mathcal{J}} \rightarrow \mathcal{R}_{\mathcal{S}}$.

- We take $\mathcal{X} = \mathcal{W}$ with $\alpha = 1_{\mathcal{W}}$. It should be clear that α is an epimorphism.
- We take \mathcal{E} to be the subcategory of \mathcal{D} defined as follows:

Objects: Objects D in \mathcal{D} such that $D = \llbracket \epsilon(S) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w$, where S is a sort of \mathcal{O}_T ;

Arrows: all arrows in \mathcal{D} whose domain and co-domain are objects in \mathcal{E} .

We define the functor $\beta : \mathcal{D} \rightarrow \mathcal{E}$ to be the functor which is the identity functor on all objects and arrows in \mathcal{D} which are also in \mathcal{E} and sends any other objects in \mathcal{D} to the terminal object in \mathcal{E} and any other arrows in \mathcal{D} to the identity arrow on the terminal object in \mathcal{E} . It should be clear that β is an epimorphism.

- We take \mathcal{U} and $\bar{\mathcal{U}}$ to be the subcategories of \mathcal{V} and $\bar{\mathcal{V}}$ defined as follows:

Objects of \mathcal{U} : objects $\mathcal{J}(W)(D)$ in \mathcal{V} , where for each object A in $\mathcal{J}(W)(D)$, $A = \llbracket \epsilon(j(\phi)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w$, where $j(\phi)$ is a judged proposition in \mathcal{O}_T , and for each arrow $A \xrightarrow{m} B$ in $\mathcal{J}(W)(D)$, $m = \llbracket \epsilon(\delta) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w$, where δ is a proof in \mathcal{O}_T ;

Arrows of \mathcal{U} : arrows in \mathcal{V} whose domain and codomain are objects in \mathcal{U} .

Objects of $\bar{\mathcal{U}}$: objects $\mathcal{K}_{\mathcal{J}}(W)(D)$ in $\bar{\mathcal{V}}$ such that each object $\bar{A} \xrightarrow{m} A = \prod_{i=1}^n \llbracket \epsilon(j_i(\phi_i)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w \xrightarrow{\llbracket \epsilon(\delta) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w} \llbracket \epsilon(j(\phi)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w$ and $\delta : y_1 : j_1(\phi_1), \dots, y_n : j_n(\phi_n) \vdash_{\mathcal{O}_T} j(\phi)$ is a natural deduction proof in \mathcal{O}_T ;

Arrows of $\bar{\mathcal{U}}$: arrows of $\bar{\mathcal{V}}$, whose domains and codomains are objects of $\bar{\mathcal{U}}$.

This completes our construction of $\mathcal{R}_{\mathcal{S}}$ and it is straightforward to show that $\mathcal{R}_{\mathcal{S}}$ is a Kripke structure for \mathcal{O}_T . We continue with the construction of an indexed epimorphism $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$.

We now define a family of natural transformations $(\epsilon_w)_{w \in |\mathcal{W}|} : \mathcal{K}_{\mathcal{J}}(w) \Rightarrow \beta^{op}; \mathcal{R}_{\mathcal{S}}(\alpha(w))$. We fix w and define each component of ϵ_w , $\eta_a^w : \mathcal{K}_{\mathcal{J}}(w)(a) \rightarrow (\beta^{op}; \mathcal{R}_{\mathcal{S}}(\alpha(w)))a$, where $a \in |\mathcal{D}|$, to be the functor which is the identity functor on objects and arrows in $\mathcal{K}_{\mathcal{J}}(w)(a)$ which are also in $\mathcal{R}_{\mathcal{S}}(\alpha(w))(\beta^{op}(a))$ and sends objects in $\mathcal{K}_{\mathcal{J}}(w)(a)$ which are not in $\mathcal{R}_{\mathcal{S}}(\alpha(w))(\beta^{op}(a))$ to the terminal object in $\mathcal{R}_{\mathcal{S}}(\alpha(w))(\beta^{op}(a))$ and arrows in $\mathcal{K}_{\mathcal{J}}(w)(a)$ which are not in $\mathcal{R}_{\mathcal{S}}(\alpha(w))(\beta^{op}(a))$

to the identity arrow on $\mathcal{R}_S(\alpha(w))(\beta^{op}(a))$. We need to show that the diagram

$$\begin{array}{ccccc}
a & & \mathcal{K}_{\mathcal{J}}(w)(a) & \xrightarrow{\eta_a^w} & \mathcal{R}_S(\alpha(w))(\beta^{op}(a)) \\
\downarrow f & & \downarrow \mathcal{K}_{\mathcal{J}}(w)(f) & & \downarrow \mathcal{R}_S(\alpha(w))(\beta^{op}(f)) \\
b & & \mathcal{K}_{\mathcal{J}}(w)(b) & \xrightarrow{\eta_b^w} & \mathcal{R}_S(\alpha(w))(\beta^{op}(b))
\end{array}$$

commutes. This follows from the definition of η_-^w . It should be clear that each ϵ_w is a natural epimorphism. Once we have shown that the diagram

$$\begin{array}{ccccc}
v & & \mathcal{K}_{\mathcal{J}}(v) & \xrightarrow{\epsilon_v} & \beta^{op}; \mathcal{R}_S(\alpha(v)) \\
\downarrow f & & \Downarrow \mathcal{K}_{\mathcal{J}}(f) & & \Downarrow \mathcal{R}_S(\alpha(f)) \\
w & & \mathcal{K}_{\mathcal{J}}(w) & \xrightarrow{\epsilon_w} & \beta^{op}; \mathcal{R}_S(\alpha(w))
\end{array}$$

commutes, we have shown that $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$ is an indexed epimorphism. The commutativity of the diagram follows from the definition of each natural transformation ϵ_w .

It remains to show that there is a Kripke model of \mathcal{O}_T which uses \mathcal{R}_S and that there is an epimorphism of models h . We use the judgements-as-types correspondence and the interpretation function $\llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^-$ to define the interpretation function $\llbracket - \rrbracket_{\mathcal{R}_S}^{-,\rho}$. Letting X range over the syntax of \mathcal{O}_T , we define $\llbracket X \rrbracket_{\mathcal{R}_S}^{x,\rho} = \llbracket \epsilon(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^w$, where $\alpha(w) = x$. Showing that $\langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-,\rho} \rangle$ is a Kripke model of \mathcal{O}_T is straightforward. h is then defined to be the morphism of models which sends $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$ to $\langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-,\rho} \rangle$ using the indexed epimorphism $(\alpha, \beta, (\epsilon_w)_{w \in |\mathcal{W}|})$. We observe that the required condition on the interpretation function holds for h to be a morphism of models. \blacksquare

We call the epimorphism constructed in Proposition 8.10, the *judgements-as-types epimorphism*.

Next, we show, as a corollary of the existence of the judgements-as-types epimorphism, that a model of the *representation* of an object-logic can be uniformly constructed from a model of the object-logic. The result generalizes one of Simpson (1993), and will be crucial when we prove representation theorems in the next section.

Corollary 8.11 (Induced Models)

Let \mathcal{O}_T be an object-logic as defined in Definition 7.13 and let $\langle \mathcal{R}_S, \llbracket - \rrbracket_{\mathcal{R}_S}^{-,\rho} \rangle$, where

$\mathcal{R}_S : [\mathcal{X}, [\mathcal{E}^{op}, \bar{\mathcal{U}}]]$, be a Kripke model of \mathcal{O}_T . 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, $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \bar{\mathcal{V}}]]$, induced by the correspondence.

Proof (Sketch) It follows from Proposition 8.10 that we can define a Kripke $\Sigma_{\mathcal{O}_T}$ - $\lambda\Pi$ -structure $\mathcal{K}_{\mathcal{J}} : [\mathcal{X}, [\mathcal{E}^{op}, \bar{\mathcal{U}}]]$ and further that if we take the interpretation function $\llbracket \epsilon(X) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^- = \llbracket X \rrbracket_{\mathcal{R}_S}^-$ then we have a Kripke $\Sigma_{\mathcal{O}_T}$ - $\lambda\Pi$ -model. ■

8.3 Representation Theorems

A representation theorem is a theorem stating that a given object-logic can be adequately represented in a logical framework. Usually, representation theorems for LF are proven proof-theoretically. We, however, are now in a position to prove faithfulness model-theoretically. The idea behind this proof can be found in (Simpson 1993). One constructs a $\Sigma_{\mathcal{O}_T}$ - $\lambda\Pi$ -term model out of the syntax of the encoded logic. Providing we restrict our attention to realizations of the from $\Gamma \rightarrow z : A$ then this term model will be the part of the domain of the judgements-as-types epimorphism on which the indexed functor is the identity. Hence we obtain an (indexed) isomorphism between these two models. We can then use soundness for Kripke $\Sigma_{\mathcal{O}_T}$ - $\lambda\Pi$ -models and completeness for Kripke models of \mathcal{O}_T to prove faithfulness.

We now have the following representation theorems.

Theorem 8.12 (Classical, Intuitionistic and Minimal Predicate Logic)

Classical, intuitionistic and minimal predicate logic presented as judged proof systems with natural deduction rules can be adequately represented in LF. ■

Theorem 8.13 (Classical Propositional Modal Logics)

Extensions of the classical propositional modal logic K by axioms of the form $\text{valid}(\diamond^j \square^m \phi \supset \square^i \diamond^n \phi)$ presented as judged proof systems with natural deduction rules can be adequately represented in LF. ■

Theorem 8.14 (Higher-order Classical, Intuitionistic and Minimal Logics)

Higher-order classical, intuitionistic and minimal logic presented as judged proof systems with natural deduction rules can be adequately represented in LF. ■

Theorem 8.15 (Local Connectives)

Logics with only local connectives presented as judged proof systems with natural deduction rules can be adequately represented in LF. ■

Chapter 9

Worlds-as-parameters

In this chapter, we provide a quick overview of how the worlds-as-parameters encoding mechanism can be understood in terms of our previous work on the judgements-as-types correspondence. The worlds-as-parameters encoding was introduced in (Avron et al. 1997). It is used to encode logics whose judgements involve a parameter. This parameter is rather suggestively called a ‘world’ and has sort U , called the ‘universe’. This terminology is intended to be entirely syntactic, although a clear link is being made with intuition obtained from the Kripke semantics of (modal) logics.

The treatment of the judgements-as-types correspondence in the previous chapter is sufficiently general to include the worlds-as-parameters encoding as a special case. We illustrate this point by examining the world-as-parameters encoding of the modal logic K given in (Avron et al. 1997). Their signature, $\Sigma_w(K)$, for K contains the following constants:

$$U : \text{Type}$$
$$o : \text{Type}$$
$$\top : U \rightarrow o \rightarrow \text{Type}$$
$$A_1 : \Pi x : o . \Pi y : o . \Pi w : U . (\top w(\epsilon_{(x,y)}(A_{1x,y})))$$
$$A_2 : \Pi x : o . \Pi y : o . \Pi z : o . \Pi w : U . (\top w(\epsilon_{(x,y,z)}(A_{2x,y,z})))$$
$$A_3 : \Pi x : o . \Pi y : o . \Pi w : U . (\top w(\epsilon_{(x,y)}(A_{3x,y})))$$
$$K : \Pi x : o . \Pi y : o . \Pi w : U . (\top w(\epsilon_{(x,y)}(K_{x,y})))$$
$$MP : \Pi x : o . \Pi y : o . \Pi w : U . (\top w x) \rightarrow (\top w(\supset xy)) \rightarrow (\top w y)$$
$$NEC : \Pi x : o . \Pi y : o . (\Pi w : U . (\top w x)) \rightarrow \Pi w : U . (\top w(\Box x))$$

where $A_{1x,y}$, $A_{2x,y,z}$, $A_{3x,y}$ and $K_{x,y}$ are the axioms in the usual Hilbert presentation of K (cf. (Troelstra & Schwichtenberg 1996)), $\epsilon_{(x,y)}$ and $\epsilon_{(x,y,z)}$ are encoding

functions which are essentially the same as ϵ_s in § 8.1.

The judged proof system for K with a world parameter is given by the alphabet

$$\begin{aligned} S &= \{U, o\} \\ V &= \{U\} \\ E &= \{\supset, \square\} \\ C &= \{\supset, \square\} \\ J &= \{\top\} \end{aligned}$$

where U and o have arity 0, \supset has arity $(o, o) \rightarrow o$ and level 1, \square has arity $o \rightarrow o$ and level 1 and \top has arity (U, o) . Together with the Hilbert-type system with axioms

$$\begin{aligned} \top w(A_1) \\ \top w(A_2) \\ \top w(A_3) \\ \top w(K) \end{aligned}$$

and rules

$$\begin{aligned} \frac{\top wx \quad \top w(x \supset y)}{\top wy} \text{MP} \\ \frac{\top wx}{\top w\square x} \text{NEC} \end{aligned}$$

It should be clear that the encoding of the above proof system using the judgements-as-types encoding in § 8.1 will give the signature $\Sigma_w(K)$.

Since our treatment of judgements in a judged proof system will always allow us to do this analysis (adding an extra syntactic category U); we claim that the worlds-as-parameters representation mechanism is a special case of the worlds-as-parameters encoding is a special case of the judgements-as-types encoding.

We further claim that the appropriate judged proof systems in which to present logics that will be encoded in LF using the worlds-as-parameters representation mechanism are labelled natural deduction systems (themselves presented as judged proof systems).

This chapter intends to prove evidence for the second claim. The first being sufficiently clear from the above discussion.

In the next section, we introduce labelled natural deduction systems. They are presented as judged proof systems and are based on the unjudged labelled natural deduction systems in (Basin et al. 1997), (Basin et al. 1998) and (Viganò 2000). A consequence of presenting them as judged proof systems is that the results of the previous chapter hold: the judgements-as-types epimorphism and the semantic

proof of fullness.

We conclude this chapter with a section on the worlds-as-parameters encoding. We exploit the fact that labelled natural deduction systems are designed to be sound and complete with respect to Kripke models in which labels are interpreted as worlds to modify Kripke \mathcal{O}_T -models so that labels are interpreted as worlds and soundness and completeness holds. Further, we show that we can also alter our definition of a Kripke $\lambda\Pi$ -model so that encoded labels are interpreted as worlds. We are then able to set up a worlds-as-parameters epimorphism between the modified Kripke \mathcal{O}_T -models and Kripke $\lambda\Pi$ -models.

9.1 Labelled Natural Deduction Systems

The idea behind a labelled natural deduction system is that the semantic information (from the appropriate transition system) is embedded into the syntax. Each proposition is labelled, with the intended meaning that the proposition is true at the world corresponding to that label in the transition system. Further, relations are introduced between labels. These allow one to write down natural deduction rules for local and non-local connectives (*cf.* § 7) where the rules for non-local connectives involve relations between labels. A labelled natural deduction system consists of two parts: a base system $N(\mathcal{B})$ consisting of introduction and elimination rules for each of the connectives; and, a relational theory $N(\mathcal{T})$ consisting of Horn relational rules. A Horn relational rule is of the form

$$\frac{R(t_0^1 \dots t_n^1) \cdots R(t_0^m \dots t_n^m)}{R(t_0^0 \dots t_n^0)}$$

where the t_j^i are terms built from labels and function symbols.

Labelled natural deduction systems can be used to describe a wide range of non-classical logics, *cf.* (Viganò 2000). We restrict our attention to structural logics, since substructural logics are not suitable for representation in LF. We achieve this restrict by taking \perp to not hold at any world and \neg to have the introduction rule

$$\frac{\begin{array}{c} [\text{true}(w, \phi)] \\ \vdots \\ \text{true}(z, \perp) \end{array}}{\text{true}(w, \neg\phi)} \neg I$$

More information about the different choices of \perp and \neg can be found in (Viganò 2000). The only examples which use these rules can be found in Appendix B.

We now show how to present a labelled natural deduction system as a judged proof system. From the above discussion, we observe that the labelled propositions $a:\phi$ can be understood as being judged by the judgement **true**, with arity

(U, o) ; that is, $\text{true}(a, \phi)$. Similarly, each relation $R(t_1, \dots, t_n)$ can be viewed as being judged by the proposition related with arity (U, \dots, U) . Hence provided we have a syntactic category U of labels and judgements is just described, we can express a labelled natural deduction system as a judged proof system. We have to split the connectives into local and non-local connectives as in § 7. The local connectives have introduction and elimination rules given by the general schema in Definition 7.11. The general schema has to be altered so that each judgement also contains a world. Since the connectives are local, the world is fixed for each connective and so this is just a notational issue. The rules for non-local connectives are given by the following definition:

Definition 9.1 (Labelled Natural Deduction Rules for Non-local Connectives)

Let $\#$ be a universal non-local connective with arity u . The introduction rule for $\#$ is

$$\frac{[j_1(w_1, \phi_1)] \cdots [j_{u-1}(w_{u-1}, \phi_{u-1})][\text{related}^u(w_1, \dots, w_u, y)] \quad \begin{array}{c} \vdots \\ j_u(w_u, \phi_u) \end{array}}{j(y, \#(\phi_1, \dots, \phi_u))} \# I$$

and the elimination rule is

$$\frac{j(y, \#(\phi_1, \dots, \phi_u)) \quad j_1(w_1, \phi_1) \cdots j_{u-1}(w_{u-1}, \phi_{u-1}) \quad \text{related}(w, w_1, \dots, w_u, y)}{j_u(\phi_u)} \# E$$

Let $\#$ be a non-local connective with arity e . The natural deduction introduction rule for $\#$ is

$$\frac{j_1(w_1, \phi_1) \cdots j_e(w_e, \phi_e) \quad \text{related}(y, w_1, \dots, w_e)}{j(y, \#(\phi_1, \dots, \phi_e))} \# I$$

and the elimination rule is

$$\frac{j(y, \#(\phi_1, \dots, \phi_e)) \quad \begin{array}{c} [j_1(w_1, \phi_1)] \cdots [j_e(w_e, \phi_e)][\text{related}^e(y, w_1, \dots, w_e)] \\ \vdots \\ k(z, \tau) \end{array}}{k(z, \tau)} \# E \quad \blacksquare$$

We conclude this section with an example. We define the judged proof system for al modal logics in the Geach hierarchy (Viganò 2000); that is, K extended by axioms of the form $\diamond^j \square^m \phi \supset \square^i \diamond^n \phi$. The judged proof system for K is defined as follows:

Definition 9.2 (K)

The judged proof system for K is given by the alphabet

$$S = \{U, o\}$$

$$V = \{U\}$$

$$E = \{\Box, \supset\}$$

$$C = \{\Box, \supset\}$$

$$J = \{\text{true}\}$$

together with the natural deduction rules

$$\begin{array}{c}
 [\text{true}(w, \phi \supset \psi)] \\
 \vdots \\
 \frac{\text{true}(y, \perp)}{\text{true}(w, \phi)} \perp E
 \end{array}$$

$$\begin{array}{c}
 [\text{true}(w, \phi)] \\
 \vdots \\
 \frac{\text{true}(w, \psi)}{\text{true}(w, \phi \supset \psi)} \supset I
 \end{array}
 \quad
 \begin{array}{c}
 [\text{true}(w, \phi)] \\
 \vdots \\
 \frac{\text{true}(w, \phi \supset \psi)\text{true}(w, \phi) \quad \text{true}(w, \tau)}{\text{true}(w, \tau)} \supset E
 \end{array}$$

$$\begin{array}{c}
 [\text{related}(w, z)] \\
 \vdots \\
 \frac{\text{true}(z, \phi)}{\text{true}(w, \Box\phi)} \Box I
 \end{array}
 \quad
 \begin{array}{c}
 \frac{\text{true}(w, \Box\phi) \quad \text{related}(w, z)}{\text{true}(z, \phi)} \Box E
 \end{array}
 \quad \blacksquare$$

There is no Horn relational theory for K . This corresponds to there being no frame condition on the transition system for K . Other modal logics in the Geach hierarchy do have a relational theory and this arises from the relationship between axioms of the form $\Diamond^j \Box^m \phi \supset \Box^i \Diamond^n \phi$ and (i, j, m, n) -convergency axioms and the following proposition.

Proposition 9.3 (Basin, Matthews and Viganò)

If \mathcal{T} is a theory corresponding to a collection of restricted (i, j, m, n) -convergency axioms, then there is a Horn relational theory $\mathcal{N}(\mathcal{T})$ conservatively extending it.

■

Restricted (i, j, m, n) -convergency axioms are a special case of (i, j, m, n) -

convergency axioms where if $m = n = 0$, then $i = j = 0$. The axioms

$$\begin{aligned}
D &: \text{valid}(\Box\phi \supset \Diamond\phi) \\
T &: \text{valid}(\Box\phi \supset \phi) \\
B &: \text{valid}(\phi \supset \Box\Diamond\phi) \\
4 &: \text{valid}(\Box\phi \supset \Box\Box\phi) \\
5 &: \text{valid}(\Diamond\phi \supset \Box\Diamond\phi)
\end{aligned}$$

correspond to the Horn relational rules

$$\begin{aligned}
&\frac{}{\text{related}(x, f(x))} \textit{Ser} \\
&\frac{}{\text{related}(x, x)} \textit{Refl} \\
&\frac{\text{related}(x, y)}{\text{related}(y, x)} \textit{symm} \\
&\frac{\text{related}(z, y) \quad \text{related}(y, z)}{\text{related}(x, z)} \textit{trans} \\
&\frac{\text{related}(x, y) \quad \text{related}(y, z)}{\text{related}(z, y)} \textit{eucl}
\end{aligned}$$

More of these correspondences can be found in (Viganò 2000).

This example illustrates how labelled natural deduction systems are used to describe families of logics. One chooses a base system and then different relational theories are used to describe different logics in the family.

More labelled natural deduction systems can be found in Appendix B.

9.2 Soundness and Completeness of Labelled Natural Deduction Systems

The soundness and completeness results of § 7.3 also hold for labelled natural deduction systems. Judgements $\text{true}(w, \phi)$ are interpreted in the fibre $\mathcal{J}(z)(\llbracket U \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{z, \rho} \times \prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{z, \rho})$, where the free variables of ϕ are in the set $\{x_1 : S_1, \dots, x_n : S_n\}$. The judgement $\text{related}(w, y)$ is interpreted in $\mathcal{J}(z)(\llbracket U \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{z, \rho} \times \llbracket U \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{z, \rho})$. The labels have no relationship to the world in the Kripke model of the labelled natural deduction system.

We now turn to an alternative presentation of the Kripke models for judged proof systems. Keeping the categorical structure the same, we interpret labels as the worlds. We then interpret the judgement $\text{related}(w, y)$ as an arrow between the objects w and y in \mathcal{W} . Since \mathcal{W} is a category, the relation is forced to be

transitive. So we can only prove soundness and completeness results for labelled natural deduction systems where the relation is transitive. This restricts our family of classical propositional modal logics to those which are extensions of $K4$ by axioms of the form $\text{valid}(\diamond^j \Box^m \phi \supset \Box^j \diamond^n \phi)$. We also have to change Definition 7.19 so that it includes the following:

- For each label $w \in U$, there is an object w in \mathcal{W} ;
- For each basic judgement $\text{related}(w_1, w_2)$, there is an arrow $w_1 \rightarrow w_2$ in \mathcal{W} ;
- For each basic judgement $\text{j}(w, \phi)$, where the free variables of ϕ are in the set $\{x_1 : S_1, \dots, x_n : S_n\}$, we have an object $\llbracket \text{j}(w, \phi) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$ in $\mathcal{J}(w)(\prod_{i=1}^n \llbracket S_i \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho})$.

Definition 7.20 also needs to be changed, so that it contains the following rules:

- For each inference rule in the Horn relational theory,

$$\frac{\text{related}(t_0^1, t_1^1) \cdots \text{related}(t_0^m, t_1^m)}{\text{related}(t_0^0, t_1^0)}$$

we have an arrow $t_0^0 \rightarrow t_1^0$ in \mathcal{W} if there are arrows $t_0^1 \rightarrow t_1^1$ and \dots and $t_0^m \rightarrow t_1^m$;

- For each predicate letter P with arity $(S_1, \dots, S_n) \rightarrow S$, $\llbracket \text{j}(w, \phi_1, \dots, \phi_n) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$ is an object in $\mathcal{J}(w)(\llbracket S_1 \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho} \times \dots \times \llbracket S_n \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho})$ given by the above extension to Definition 7.19;
- For each universal non-local connective $\#$, we have the following satisfaction condition: there is an arrow

$$1 \xrightarrow{f} \llbracket \text{j}(\#(\phi_1, \dots, \phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$$

if and only if for all worlds w_1 (there is an arrow $w \rightarrow w_1$ in \mathcal{W} and there exist arrows

$$1 \xrightarrow{f_1} \llbracket \text{j}_1(\phi_1) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_1, \rho_1} \text{ and } \dots \text{ and } 1 \xrightarrow{f_{n-1}} \llbracket \text{j}_{n-1}(\phi_{n-1}) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_1, \rho_{n-1}} \text{ which imply}$$

$$1 \xrightarrow{f_n} \llbracket \text{j}_n(\phi_n) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_1, \rho_n};$$

- For each existential non-local connective $\#$, we have the following satisfaction condition: there is an arrow

$$1 \xrightarrow{f} \llbracket \text{j}(\#(\phi_1, \dots, \phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$$

if and only if there exist worlds w_1 such that there is an arrow $w \rightarrow w_1$ in \mathcal{W} and there exist arrows

$$1 \xrightarrow{f_1} \llbracket j_1(\#(\phi_1)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_1, \rho_1} \text{ and } \dots \text{ and } 1 \xrightarrow{f_n} \llbracket j_n(\#(\phi_n)) \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w_1, \rho_n}.$$

For the remainder of this section, we assume that any interpretation of a labelled natural deduction system in a Kripke model takes into account the above changes. We have the following soundness results. The proofs are essentially the same as those of Lemma 7.24 and Corollaries 7.25, 7.26 and 7.27. We just have to be careful with the worlds: we have to ensure that we are always interpreting the judgement $j(w, \phi)$ at the world w and any change to another world w' means that we are now interpreting the judgement $j(w', \phi)$.

Lemma 9.4

Let \mathcal{O}_T be a labelled natural deduction system with alphabet A which only contains local connectives and let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-, \rho} \rangle$, where $\mathcal{K}_{\mathcal{J}}: [\mathcal{W}, [\mathcal{B}^{op}, \bar{\mathcal{V}}]]$, be a Kripke model of \mathcal{O}_T . If $\Gamma \vdash_{\mathcal{O}_T} j(w, \phi)$ then $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(w, \phi)$. ■

Corollary 9.5

Let \mathcal{O}_T be the labelled natural deduction system for classical predicate logic and let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-, \rho} \rangle$ be a Kripke model of $\mathcal{K}_{\mathcal{J}}$. If $\Gamma \vdash_{\mathcal{O}_T} \text{true}(w, \phi)$ then $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \text{true}(w, \phi)$. ■

Corollary 9.6

Let \mathcal{O}_T be the labelled natural deduction system for minimal or intuitionistic predicate logic and let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-, \rho} \rangle$ be a model of \mathcal{O}_T . If $\Gamma \vdash_{\mathcal{O}_T} \text{proof}(w, \phi)$ then $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \text{proof}(w, \phi)$. ■

Corollary 9.7

Let \mathcal{O}_T be the labelled natural deduction system for a classical modal propositional logic which is an extension of $K4$ by Horn relational rules corresponding to axioms of the form $\text{valid}(\diamond^i \Box^m \phi \supset \Box^j \diamond^n \phi)$ and let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{-, \rho} \rangle$ be a Kripke model of \mathcal{O}_T . If $\Gamma \vdash_{\mathcal{O}_T} \text{true}(w, \phi)$ then $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \text{true}(w, \phi)$. ■

To be able to prove completeness, we need to construct a term model. The one we constructed after Definition 7.20 will work with some slight modification. We just need to define the objects in the category $\mathcal{J}(W)(X)$ to be judged propositions $j(a, \phi(X'))$, where $X = X' \times U$ and define arrows $w_1 \rightarrow w_2$ in \mathcal{W} , whenever we can derive $\text{related}(w_1, w_2)$. We now have a model existence result. From now on we take \mathcal{O}_T to be one of the following labelled deduction systems: one which only contains local connectives; minimal propositional logic; intuitionistic propositional logic; classical propositional logic; or, a modal logic in the Geach hierarchy.

Lemma 9.8 (Model Existence)

There exists a Kripke model of \mathcal{O}_T , $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$, together with a world w_0 such that if $\Gamma \not\vdash_{\mathcal{O}_T} j(w, \phi)$ then $w_0, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(w, \phi)$ implies $w_0, \rho \not\Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} j(w_0, \phi)$.

Proof (Sketch) The term model constructed after Definition 7.20 together with the modifications described above is the required Kripke model of \mathcal{O}_T . We take the world w_0 to be the world which labels each $j_i(w_0, \phi_i) \in \Gamma$ but does not label $j(w, \phi)$. ■

Theorem 9.9 (Completeness for \Vdash)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$ be a Kripke model of \mathcal{O}_T . Then $\Gamma \vdash_{\mathcal{O}_T} j(w, \phi)$ if and only if $w, \rho \Vdash_{\mathcal{O}_T} j(w, \phi)$.

Proof

Only If This is soundness, Lemma 9.4 and Corollaries 9.5,9.6 and 9.7.

If Suppose $\Gamma \not\vdash_{\mathcal{O}_T} j(w, \phi)$, then Lemma 9.8 yields a contradiction. ■

The proofs of soundness and model existence for \Vdash are essentially the same as those in § 7.3. We thus have

Theorem 9.10 (Completeness for \Vdash)

Let $\langle \mathcal{K}_{\mathcal{J}}, \llbracket - \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{\rho} \rangle$ be a Kripke model of \mathcal{O}_T . Then if $\delta: (\Gamma \vdash_{\mathcal{O}_T} j(w, \phi))$ is a natural deduction proof and $\llbracket \delta \rrbracket_{\mathcal{K}_{\mathcal{J}}}^{w, \rho}$ is defined, then $w, \rho \Vdash_{\mathcal{O}_T}^{\mathcal{K}_{\mathcal{J}}} \delta: (\Gamma \vdash_{\mathcal{O}_T} j(w, \phi))$. ■

9.3 Worlds-as-parameters Encoding

It should be clear that we can use the judgements-as-types encoding outlined in § 8.1 to encode labelled natural deduction systems in LF. The encoding of K is given by the signature Σ_K .

Definition 9.11 (Basin & Matthews (2002))

The signature Σ_K is defined as follows:

$$\begin{aligned} U &: \text{Type} \\ o &: \text{Type} \\ \text{true} &: U \rightarrow o \rightarrow \text{Type} \\ \text{related} &: U \rightarrow U \rightarrow \text{Type} \\ \perp &: o \\ \supset &: o \rightarrow o \rightarrow o \\ \square &: o \rightarrow o \end{aligned}$$

$BOT-E: \Pi p: o . \Pi a: U . \Pi b: U . (\text{true}(a)(p \supset \perp) \rightarrow \text{true}(b)\perp) \rightarrow \text{true}(a)p$

$IMP-I: \Pi p: o . \Pi q: o . \Pi a: U . (\text{true}(a)p \rightarrow \text{true}(a)q) \rightarrow \text{true}(a)(p \supset q)$

$IMP-E: \Pi p: o . \Pi q: o . \Pi r: o . \Pi a: U . \Pi b: U . \text{true}(a)(p \supset q) \rightarrow \text{true}(a)p \rightarrow$
 $(\text{true}(a)q \rightarrow \text{true}(b)r) \rightarrow \text{true}(b)(r)$

$BOX-I: \Pi p: o . \Pi a: U . (\Pi b: U . \text{related}(a)b \rightarrow \text{true}(b)p) \rightarrow \text{true}(a)(\Box p)$

$BOX-E: \Pi p: o . \Pi a: U . \Pi b: U . \text{true}(a)(\Box p) \rightarrow \text{related}(a)b \rightarrow \text{true}(b)p$ ■

The encoding in (Basin & Matthews 2002) is also joint work with Luca Viganó.

Signatures for minimal, intuitionistic and classical predicate logics as well as more modal logics can be found in Appendix A.

Since we are working with judged proof systems represented in LF using the judgements-as-types correspondence, all the results of § 8.2 hold for labelled natural deduction systems. Specifically, the judgements-as-types epimorphism. One might be tempted to call this special case the worlds-as-parameters epimorphism, but we have another epimorphism which is more deserving of the name.

Recalling the new class of Kripke \mathcal{O}_T -models introduced in the previous section, we show that there is an epimorphism between these Kripke \mathcal{O}_T -models and suitably modified Kripke $\lambda\Pi$ -models.

Given a signature $\Sigma_{\mathcal{L}}^w$, which is the encoding of a labelled natural deduction system using the worlds-as-parameters encoding, we are able to define a Kripke $\Sigma_{\mathcal{L}}^w$ - $\lambda\Pi$ -model in which encoded labels are interpreted as worlds. We are able to define this model because when given the signature $\Sigma_{\mathcal{L}}^w$, we are told which constant encodes the syntactic category of labels and which constant encodes the judgement which relates labels. This information is not available *a priori*. We now include the following conditions in Definition 3.17:

- $\llbracket a: U \rrbracket_{\mathcal{K}_{\mathcal{J}}}^a = a \in |\mathcal{W}|$, where U is the type encoding the universe;
- Each type of the form $\text{related}(a, b)$, where $a: U$ and $b: U$, is interpreted as an arrow $a \rightarrow b$ in \mathcal{W} .

Care has to be taken with the interpretation of types of the form $j(a)\phi$. We need to interpret these as types $j'(\phi)$, where $j': o \rightarrow \text{Type}$, in the fibres over a . Roughly speaking, we ignore all the labels when we interpret any objects of the $\lambda\Pi$ -calculus apart from interpreting them at the world which interprets their label.

We then have essentially the same structure in this Kripke Σ - $\lambda\Pi$ -model, apart from \mathcal{W} , as we have in the usual Kripke Σ - $\lambda\Pi$ -model. Since \mathcal{W} has the same structure as the category of worlds in the Kripke model of the object-logic, we claim it is possible to construct an epimorphism of models. We call this epimorphism the worlds-as-parameters epimorphism. We also claim that soundness and completeness holds for the Kripke Σ - $\lambda\Pi$ -model sketched above. Furthermore, all

the results of § 8.3, provided they are put in terms of labelled natural deduction systems, hold.

Chapter 10

Introduction to Logic Programming

Modern symbolic logic stands firmly in the Aristotelean tradition and takes *deduction* as the primary proof-theoretic notion. *Rules of inference* are used to construct *consequences* from a collection of *assumptions*. These rules are applied to known propositions to establish further propositions.

An alternative viewpoint takes *proof-search* as the primary proof-theoretic notion. A logic is then seen as a system for *reduction*. One then uses inference rules as reduction operators and attempts to construct a proof from a given *judgement*. We are working from a conclusion to premisses and each step simplifies the proof.

Proof-search is inherently *non-deterministic* and any algorithm designed to calculate proofs, or decide putative consequences, must deal with this. The non-determinism arises from arbitrary choices involved in constructing a proof, *e.g.*, which proposition is reduced at each step and which proof is reduced when the proof branches. Logic viewed computationally can be summarized by the slogan

$$\textit{Logic} = \textit{Inference} + \textit{Control}.$$

Thus the nature of reasoning determined by a system of logic depends on the régime which controls their use as well as the inference rules (and indeed the satisfaction relation). (The above argument is taken from (Pym & Ritter 2004).)

Often (Pym & Wallen 1992) one takes the result of the computation of a logic program P , *i.e.*, a collection of clauses, together with a query $\exists x. g$ (written $g(X)$ in Prolog) to be a substitution of a term t for the (existentially quantified) variable of the query such that

$$P \text{ entails } g[t/x].$$

Pure logic programmes are usually modelled as sets of such substitutions (Lloyd 1984). This is a natural view to take since the substitutions typically

carry the information required by the user of the program.

Logical consequence, however, is an abstract notion and our access to it is via concrete notions of deductions; that is, we construct a (finitary) proof that $\exists x.g$ follows from P and extract a term t from this proof. There are two distinct phases of computation:

proof-search: the computation of the proof; and

residual computation: the extraction of witnesses from that proof.

It is clear that the residual computation depends on the result of the proof-search.

Following (Pym & Wallen 1992), we believe that the proof (*i.e.*, the output from the first computation) is more naturally seen as the result of the computation. A consequence of this view is that any sound proof procedure gives rise to a notion of computation with the values computed being proofs. If we take logical systems as represented in LF, then the calculation of objects satisfying predicates and of proofs proving formulæ is the same mechanism as that employed in logic programming. An elementary notion of logic programming for the type theory of LF is discussed in (Pym 1990) and (Pym & Wallen 1991); a more elaborate notion is developed in (Pfenning 1991).

Following (Pym & Ritter 2004), we believe that the view of logic as a reductive system is (at least) as fundamental as the deductive view and that the semantics of reductions should be as closely linked to proof-search as the semantics of proof is to model-theory. The semantics of deterministic strategies is beyond the scope of this thesis — indeed, it is a current research topic. Here we present a proof-theoretic operational semantics.

In type-theoretic analyses, such as those presented in (Pym 1990), (Pym & Wallen 1991), (Ritter, Pym & Wallen 1996*b*) and (Ritter, Pym & Wallen 1996*a*), the realizing term provides the proof-object and answer substitution in a single construction. It is the operational, or procedural, description of concrete notions of deduction that traditionally lie outwith the declarative realm. Even if the “computational” proof system, such as Prolog’s resolution, is sound and complete with respect to consequence in the underlying logic, these soundness and completeness results are highly non-deterministic and the non-determinism must be resolved. The non-determinism, as we discussed above, arising via the choice of which proposition to work on and the order in which to deal with branches. *Backtracking*, the principal control mechanism for proof-search is one way to resolve this non-determinism. When presented with a branch, choose one and if it fails, backtrack to the choice and choose another branch. Clearly, there is an ordering of the choice of branches. In Prolog, resolution computations are executed using a depth first search strategy and a leftmost-first clause selection strategy. But in practice, there is more to logic programming than search and clause-selection strategies. In fact, one is required to deal with; for example, cut and assert within a logic programming language.

Historically, the starting point for resolution is Herbrand’s *Investigations in Proof Theory: The Properties of True Propositions* (reprinted in (Herbrand 1967)). ‘Herbrand’s theorem’ provides a basis for a mechanical proof procedure for first-order logic of manageable complexity. Perhaps, the most significant development in the 1960s was Alan Robinson’s *resolution* procedure (Robinson 1965). For formulæ in a certain, functionally complete, clausal form, the resolution rule is, together with the use of unification to calculate terms, both computationally appealing and logically complete. This use of resolution followed by unification reminds us that there are two stages to a computation; proof-search and the residual computation. The resolution procedure together with a control régime for selecting which clauses from a set the resolution rule should next be applied to, forms the basis of the programming language prolog and Kowalski’s famous dictum

$$\textit{Programming} = \textit{Logic} + \textit{Control}.$$

In recent years, Miller et al. (1991), Pym & Harland (1994) and Ritter et al. (1996b) have provided more systematic accounts of logic programming via the sequent calculus, and the proof-theoretic basis of their work provides a point of departure for the remaining chapters of the thesis.

10.1 Logic Programming in $\lambda\Pi$

In (Pym 1990) and (Pym & Wallen 1991), it was shown that the $\lambda\Pi$ -calculus admits a natural interpretation as a logic programming language, based on sequents of the form $\Gamma \Rightarrow_{\Sigma} A(\alpha)$, where α is an indeterminate. Such sequents are interpreted as request to calculate terms M and N such that $\Gamma \vdash_{\Sigma} M : A[N/\alpha]$ is provable. Here $[N/\alpha]$ corresponds to the usual notion of *answer substitution*, intended to be calculated by unification. An alternative formulation of logic programming for LF has been presented by Pfenning (1991) and implemented in Elf and Twelf (Pfenning & Schürmann 1999). (In the hope of clarifying some previous confusion over the names of frameworks and implementations, we have used the following terminology in this thesis:

- LF - the $\lambda\Pi$ -calculus together with the judgements-as-types representation mechanism;
- ELF - the implementation of LF (Arnon Avron & Mason 1996);
- Elf - Pfenning’s first implementation (Pfenning 1991);
- Twelf - Pfenning’s second implementation (Pfenning & Schürmann 1999).

This appears to match the current naming conventions. We have probably added further confusion to the naming of different logical frameworks by calling all the logical frameworks in this thesis which have the same proof-terms LF, when technically they are different frameworks due to different choices of representation mechanisms or languages.)

As we discussed in the previous section, there are two distinct phases of computation in logic programming: proof-search and residual computation. We are interested in providing a proof-theoretic operational semantics and hence are interested in resolution and unification. Our treatment of resolution follows Miller et al. (1991) by dealing with uniform proof. We are not concerned with unification in our current treatment, this is mainly because a lot of the issues have been addressed elsewhere. A full and complete unification algorithm for the $\lambda\Pi$ -calculus was discovered independently by Elliott (1990) and Pym (1990) (and (Pym 1992)). These algorithms would not be used in practice because they are too expensive computationally. The work of Pfenning *et al.* ((Pientka & Pfenning 2003), (Dowek, Hardin, Kirchner & Pfenning 1996) and (Pfenning & Schürmann 1998)) provides more efficient algorithms. A treatment of unification from the point of view of the $\lambda\Pi$ -calculus as the language of a logical framework and the relationship between the unification of the $\lambda\Pi$ -calculus and the encoded logic can be found in (Brown & Wallen 1995).

The calculus we would use as a basis for computation is **L** (Pym & Wallen 1991). This is a system for the semi-decidable relation of inhabitation: $\Gamma \Rightarrow_{\Sigma} A$, with the meaning $(\exists M)(\Gamma \vdash_{\Sigma} M : A)$. The judgements of this calculus assert the existence of proofs of the judgements of **N**. **L** is almost logistic in the sense of Gentzen and has a subformula property with respect to the Π -type structure.

Definition 10.1 (Sequent)

A sequent is a triple $\langle \Sigma, \Gamma, A \rangle$, written $\Gamma \Rightarrow_{\Sigma} A$, where Σ is a signature, Γ is a context and A is a type (family). The intended interpretation of the sequent is the (meta-)assertion;

$$(\exists M)\mathbf{N} \text{ proves } \Gamma \vdash_{\Sigma} M : A \quad \blacksquare$$

Definition 10.2 (The System **L** (Pym & Wallen 1991))

The following axioms and rules define the semi-logistic calculus **L**.

$$\begin{array}{l}
Ax1 \quad \Gamma, x:A, \Gamma' \Rightarrow_{\Sigma} A \\
Ax2 \quad \Gamma \Rightarrow_{\Sigma, c:A, \Sigma'} A \\
\rightarrow r \quad \frac{\Gamma, x:A \Rightarrow_{\Sigma} B}{\Gamma \Rightarrow_{\Sigma} A \rightarrow B} \quad (a) \ x \notin Dom(\Gamma) \\
\Pi r \quad \frac{\Gamma, x:A \Rightarrow_{\Sigma} B}{\Gamma \Rightarrow_{\Sigma} \Pi x:A. B} \quad (a) \ x \notin Dom(\Gamma) \\
\rightarrow l \quad \frac{\Gamma \Rightarrow_{\Sigma} A \quad \Gamma, y:B \Rightarrow_{\Sigma} C}{\Gamma \Rightarrow_{\Sigma} C} \quad (a) \ @:A \rightarrow B \in \Sigma \cup \Gamma \\
\quad \quad \quad (b) \ y \notin Dom(\Gamma) \\
\Pi l \quad \frac{\Gamma, y:D \Rightarrow_{\Sigma} C}{\Gamma \Rightarrow_{\Sigma} C} \quad (a) \ @:\Pi x:A. B \in \Sigma \cup \Gamma \\
\quad \quad \quad (b) \ y \notin Dom(\Gamma) \\
\quad \quad \quad (c) \ \mathbf{G}/cut \text{ proves } \Gamma \vdash_{\Sigma} M:A \\
\quad \quad \quad (d) \ B[N/x] \rightarrow_{\beta\eta} D \quad \blacksquare
\end{array}$$

\mathbf{G}/cut is the calculus obtained from **N** by replacing (2.15) with the rule

$$\frac{@:\Pi x:A. B \in \Sigma \cup \Gamma \quad \Gamma \vdash_{\Sigma} N:A \quad B[N/x] =_{\beta\eta} C \quad \Gamma, y:C \vdash_{\Sigma} M:D}{\Gamma \vdash_{\Sigma} M[@N/y]:D}$$

where $y \notin FV(D)$.

We have the following result which describes the relationship between **L** and **N**. We begin with a definition.

Definition 10.3 (Well-formed Sequent (Pym 1990))

A sequent $\Gamma \Rightarrow_{\Sigma} A$ is said to be *well-formed* just in case \mathbf{G}/cut proves $\Gamma \vdash_{\Sigma} A$: Type. \blacksquare

Proposition 10.4 ((Pym 1990))

For well-formed sequents $\Gamma \Rightarrow_{\Sigma} A$

L proves $\Gamma \Rightarrow_{\Sigma} A$ if and only if $(\exists M)$ **N** proves $\Gamma \vdash_{\Sigma} M:A$ \blacksquare

Chapter 11

Σ - $\lambda\Pi$ -Herbrand Models

Having discussed how the $\lambda\Pi$ -calculus can be used as a logic programming language, we provide a suitable semantics for it. We do this in terms of Σ - $\lambda\Pi$ -Herbrand models. In this chapter, we introduce Σ - $\lambda\Pi$ -Herbrand models and provide a least fixed-point construction. Σ - $\lambda\Pi$ -Herbrand models are a special class of Kripke Σ - $\lambda\Pi$ -models and are more concrete. In a Kripke Σ - $\lambda\Pi$ -model realizations $\Gamma \xrightarrow{\sigma} \Delta$ are constructed out of the Kripke $\lambda\Pi$ -prestructure. A consequence of this is that there may be realizations in the fibre over Γ , say, in the absence of putatively corresponding arrows in the Kripke $\lambda\Pi$ -prestructure. Σ - $\lambda\Pi$ -Herbrand models take the structure at a world Δ and base Γ to be a subset of $\text{hom}_{\mathcal{D}}(\Gamma, \Delta)$. Thus ensuring that all arrows in the Σ - $\lambda\Pi$ -Herbrand model have a corresponding arrow in the Σ - $\lambda\Pi$ -Herbrand prestructure.

The worlds in Σ - $\lambda\Pi$ -Herbrand models consist of collections of propositions (actually encodings of judged proof-variables). The motivation for taking worlds to be collections of propositions can be found in (van Emden & Kowalski 1976). van Emden and Kowalski's use of atoms in a least Herbrand model is analogous to our use of worlds to form axiom sequents.

The fixed-point construction is a generalization of a fixed-point construction found in (Miller 1989) for intuitionistic logic. We generalize his construction to our setting, taking full advantage of the fact that he uses Kripke models.

11.1 Herbrand Prestructures and Structures

We begin by defining a clausal form for terms in the $\lambda\Pi$ -calculus. From now on we assume that all terms used in the application rule in \mathbf{C} are clausal and we refer to this rule as resolution.

Definition 11.1 (Clausal Form)

We say that any constant or variable $@ : A \in \Sigma \cup \Gamma$ is in *clausal form* if it is of

the form:

$$\textcircled{\text{A}}: \Pi z_1: B_1 \dots \Pi z_m: B_m \cdot (C_1 \rightarrow (C_2 \rightarrow (\dots (C_n \rightarrow D) \dots)))$$

where m and n may be 0 and each B_j , C_i and D are clausal. The rule is said to be strongly clausal if D is atomic. ■

We are now able to define a Herbrand Σ - $\lambda\Pi$ -prestructure. A Herbrand Σ - $\lambda\Pi$ -prestructure is the Kripke $\lambda\Pi$ -prestructure, \mathcal{T}_Σ ; that is, the term model, we constructed in § 3.2.1.

Definition 11.2 (Σ - $\lambda\Pi$ -Herbrand Structure)

Let Σ be a $\lambda\Pi$ -signature. A Σ - $\lambda\Pi$ -Herbrand structure is a Kripke $\lambda\Pi$ -structure, $\mathcal{I}_{\mathcal{H}(\Sigma)}$, on $\mathcal{H}(\Sigma)$ such that the objects of each $\mathcal{I}_{\mathcal{H}(\Sigma)}(\Delta)(\Delta \bowtie \Gamma)$ are given by arrows $\Gamma \xrightarrow{\sigma} \Delta$ of $\mathcal{B}(\Sigma)$ (Definition 2.2). The arrows of $\mathcal{I}_{\mathcal{H}(\Sigma)}(\Delta)(\Gamma)$ are given by Definition 3.2. ■

The proof that the above Σ - $\lambda\Pi$ -Herbrand structure is well-defined; that is, it is a Kripke $\lambda\Pi$ -prestructure is very similar to that given for the term model in § 3.2.1.

11.2 Σ - $\lambda\Pi$ -Herbrand Models

We now define a Σ - $\lambda\Pi$ -Herbrand model. The partiality in the definition enables us to interpret incomplete proofs.

Definition 11.3 (Σ - $\lambda\Pi$ -Herbrand Model)

Let Σ be a $\lambda\Pi$ -signature. A Σ - $\lambda\Pi$ -Herbrand model is an ordered pair, $\langle \mathcal{I}_{\mathcal{H}(\Sigma)}, \llbracket - \rrbracket_{\mathcal{I}_{\mathcal{H}(\Sigma)}}^- \rangle$, where $\mathcal{I}_{\mathcal{H}(\Sigma)} : [\mathcal{P}(\Sigma), [\mathcal{B}(\Sigma)^{op}, \mathcal{V}(\Sigma)]]$, is a Σ - $\lambda\Pi$ -Herbrand structure. $\llbracket - \rrbracket_{\mathcal{I}_{\mathcal{H}(\Sigma)}}^-$ is the standard term model interpretation (we give only a sketch, the details being obvious):

- $\llbracket \Gamma \rrbracket_{\mathcal{I}_{\mathcal{H}(\Sigma)}}^\Delta = \Gamma$;
- $\llbracket A_\Gamma \rrbracket_{\mathcal{I}_{\mathcal{H}(\Sigma)}}^\Delta = A$;
- $\llbracket c_\Gamma \rrbracket_{\mathcal{I}_{\mathcal{H}(\Sigma)}}^\Delta = op_c(= c)$;
- $\llbracket x_\Gamma \rrbracket_{\mathcal{I}_{\mathcal{H}(\Sigma)}}^\Delta = \langle \rangle \xrightarrow{x} A$;

etc. A consequence $\Gamma \xrightarrow{\sigma} \Delta$ is defined at Δ with respect to Γ just in case $\llbracket \sigma_\Gamma \rrbracket_{\mathcal{I}_{\mathcal{H}(\Sigma)}}^\Delta$ is an object of $\mathcal{I}_{\mathcal{H}(\Sigma)}(\Delta)(\Gamma)$. ■

We now adapt the ideas of a Herbrand universe and Herbrand base to our setting. The usual definitions (*cf.* (Lloyd 1984)) are as follows: a Herbrand universe is the set of all ground terms which can be formed out of constants and functions appearing in a given logic. Ground terms are terms that contain no variables. A Herbrand base is the set of all ground atoms, *i.e.*, atoms that contain no variables, which can be formed by using predicates of the logic and ground terms from the Herbrand universe as arguments. The idea is then that for any given Herbrand interpretation, the assignments of constants and functions are fixed but not the assignments of predicates. This means that we are not instantiating the symbols in a given set of clauses and so are restricting to an easier case. The appropriate definition for our setting is given below.

Definition 11.4 (Ground Homset)

Let Σ be a $\lambda\Pi$ -signature. Let Δ and Γ be contexts. The *ground homset* $[\Sigma, \Delta, \Gamma]$ is the subset of $\text{hom}_{\mathcal{B}(\Sigma)}(\Gamma, \Delta)$ consisting of arrows of the form $\langle @_1, \dots, @_n \rangle$, where each $@_i \in \Gamma \cup \Sigma$. ■

11.3 Least Fixed-Point Construction

We are moving towards a least fixed-point construction. The material that follows is similar to the basic least fixed-point construction for intuitionistic logic in (Miller 1989). We begin by showing that Σ - $\lambda\Pi$ -Herbrand structures form a lattice.

Definition 11.5 (Lattice Operations)

Let Σ be a $\lambda\Pi$ -signature. We define at each object Δ in $\mathcal{P}(\Sigma)$ and Γ in $\mathcal{B}(\Sigma)$, the following operations on Σ - $\lambda\Pi$ -Herbrand structures:

- Meet:
Objects: $(\mathcal{I}_{\mathcal{H}(\Sigma)_1} \sqcap \mathcal{I}_{\mathcal{H}(\Sigma)_2})(\Delta)(\Gamma) =_{def} \mathcal{I}_{\mathcal{H}(\Sigma)_1}(\Delta)(\Gamma) \cap \mathcal{I}_{\mathcal{H}(\Sigma)_2}(\Delta)(\Gamma)$;
Arrows: post- and pre-composition, respectively, for the first and second arguments;
- Join:
Objects: $(\mathcal{I}_{\mathcal{H}(\Sigma)_1} \sqcup \mathcal{I}_{\mathcal{H}(\Sigma)_2})(\Delta)(\Gamma) =_{def} \mathcal{I}_{\mathcal{H}(\Sigma)_1}(\Delta)(\Gamma) \cup \mathcal{I}_{\mathcal{H}(\Sigma)_2}(\Delta)(\Gamma)$;
Arrows: post- and pre-composition, respectively, for the first and second arguments;
- Order: $\mathcal{I}_{\mathcal{H}(\Sigma)_1} \sqsubseteq \mathcal{I}_{\mathcal{H}(\Sigma)_2}$ if and only if, for all Δ in $\mathcal{P}(\Sigma)$ and Γ in $\mathcal{B}(\Sigma)$, $\mathcal{I}_{\mathcal{H}(\Sigma)_1}(\Delta)(\Gamma) \subseteq \mathcal{I}_{\mathcal{H}(\Sigma)_2}(\Delta)(\Gamma)$;
- Bottom: $\perp(\Sigma)(\Delta)(\Gamma)$:

Objects: $\langle \rangle \xrightarrow{\langle \rangle} \langle \rangle$;

Arrows: $\langle \rangle \xrightarrow{\langle \rangle} \langle \rangle$. ■

$\perp(\Sigma)$ is the Σ - $\lambda\Pi$ -Herbrand structure where for any Δ in $\mathcal{P}(\Sigma)$ and Γ in $\mathcal{B}(\Sigma)$ the fibre $\perp(\Sigma)(\Delta)(\Sigma)$ is the empty category.

Lemma 11.6

With the operations of Definition 11.5, Σ - $\lambda\Pi$ -Herbrand structures form a complete lattice.

Proof We observe that all the above structures are well-defined and note that a powerset ordered by \subseteq forms a complete lattice. ■

To be able to obtain a least fixed-point, we need to have a monotone function between lattices (*cf.* (Miller 1989) and (Lloyd 1984)). In our situation, we need a natural transformation between Σ - $\lambda\Pi$ -Herbrand structures, which is monotone with respect to the ordering, \sqsubseteq . We need a natural transformation because Σ - $\lambda\Pi$ -Herbrand structures are functors. The intention is that one application of this natural transformation corresponds to one application of the resolution rule (3.31), so that we obtain a stratification of the search-space.

Definition 11.7

We define the operator \mathbb{T} on Herbrand structures as follows:

Objects:

$$\begin{aligned} \mathbb{T}(\mathcal{I}_{\mathcal{H}}(\Sigma)(\Delta)(\Gamma)) &=_{def} [\Sigma, \Delta, \Gamma] \cup \mathcal{I}_{\mathcal{H}}(\Sigma)(\Delta)(\Gamma) \cup \\ &\quad \{ \langle M_1, \dots, M'_i, \dots, M_n \rangle | \Gamma \xrightarrow{\langle \overline{M} \rangle} \Delta, \text{ where } M'_i \text{ is} \\ &\quad \text{such that } @_{\overline{P}\overline{Q}} \rightarrow_{\beta\eta} M'_i, \text{ for some appropriate} \\ &\quad \overline{P}\overline{Q} \text{ and } @_i : \Pi z_{i1} : B_{i1} \dots \Pi z_{ip} : B_{ip} \cdot (C_{i1} \\ &\quad \rightarrow (C_{i2} \rightarrow \dots (C_{iq} \rightarrow D_i) \dots)) \in \Sigma \cup \Gamma. M'_i \text{ the} \\ &\quad \text{object replaced by } M_i \text{ is such that} \\ &\quad \Delta \Vdash_{\Sigma}^{\mathcal{I}_{\mathcal{H}}(\Sigma)} (\Gamma \xrightarrow{\langle M_1, \dots, M_i, \dots, M_n \rangle} \Delta) [\Gamma] \text{ and where} \\ &\quad \Gamma, x : B_i [M_j / y_j]_{j=1}^{i-1} \Vdash_{\Sigma}^{\mathcal{I}_{\mathcal{H}}(\Sigma)} ((\Gamma \xrightarrow{\langle x_1, \dots, x_m, M'_i \rangle} \Gamma, \\ &\quad x : B_i) [M_j / y_j]_{j=1}^{i-1}) [\Gamma] \}; \end{aligned}$$

Arrows: Given by the natural transformation $\mathbb{T}(\mathcal{I}_{\mathcal{H}}(\Sigma))(\Delta \xrightarrow{\alpha} \Delta')$, where $\Delta' = \Delta \cup \Gamma, x : B_i [M_j / y_j]_{j=1}^{i-1}$, with components

$$\mathbb{T}(\mathcal{I}_{\mathcal{H}}(\Sigma))(\Delta)(\Gamma) \xrightarrow{(\mathbb{T}(\mathcal{I}_{\mathcal{H}}(\Sigma))(\alpha))_{\Gamma}} \mathbb{T}(\mathcal{I}_{\mathcal{H}}(\Sigma))(\Delta')(\Gamma),$$

given by $\Gamma \xrightarrow{\sigma} \Delta \mapsto \Gamma \xrightarrow{\sigma'} \Delta'$, where $\sigma' = \sigma; \alpha$. ■

Lemma 11.8

The operator T is a natural transformation between Σ - $\lambda\Pi$ -Herbrand structures.

Proof It is straightforward to check that if $\mathcal{I}_{\mathcal{H}(\Sigma)}$ is a Σ - $\lambda\Pi$ -Herbrand structure, then so is $\mathsf{T}(\mathcal{I}_{\mathcal{H}(\Sigma)})$. Naturality follows straightforwardly. \blacksquare

We now give two lemmas concerning the relationship between the ordering \sqsubseteq and the satisfaction relation \models . For simplicity, henceforth, we shall drop the subscript $\mathcal{H}(\Sigma)$ from Σ - $\lambda\Pi$ -Herbrand models, writing just \mathcal{I} or \mathcal{I}_i , etc., where no confusion can occur.

Lemma 11.9

If $\Delta \models_{\Sigma}^{\mathcal{I}_1} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$ and $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$, then $\Delta \models_{\Sigma}^{\mathcal{I}_2} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$.

Proof If $\Delta \models_{\Sigma}^{\mathcal{I}_1} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$, then $\Gamma \xrightarrow{\sigma} \Delta \in \mathcal{I}_1(\Delta)(\Gamma)$. By the definition of \sqsubseteq , $\Gamma \xrightarrow{\sigma} \Delta \in \mathcal{I}_2(\Delta)(\Gamma)$ and thus $\Delta \models_{\Sigma}^{\mathcal{I}_2} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$. \blacksquare

Lemma 11.10

Let $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2 \sqsubseteq \dots$ be an ω -chain. If $\Delta \models_{\Sigma}^{(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i)} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$, then there is a $k \geq 1$ such that $\Delta \models_{\Sigma}^{\mathcal{I}_k} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$.

Proof If $\Delta \xrightarrow{(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i)} \Sigma(\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$, then there is an arrow $\Gamma \xrightarrow{\sigma} \Delta$ in $(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i)(\Delta)(\Gamma)$. Therefore, there is a $k \geq 1$ such that the arrow $\Gamma \xrightarrow{\sigma} \Delta$ is an object of $\mathcal{I}_k(\Delta)(\Gamma)$, from which it follows immediately that $\Delta \models_{\Sigma}^{\mathcal{I}_k} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$. \blacksquare

We now show the key lemma in establishing that the operator T has a fixed-point.

Lemma 11.11

The operator T is monotone with respect to \sqsubseteq .

Proof Suppose that $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$. We must show that $\mathsf{T}(\mathcal{I}_1) \sqsubseteq \mathsf{T}(\mathcal{I}_2)$. Let $\langle \overline{M} \rangle \in \mathsf{T}(\mathcal{I}_1)(\Delta)(\Gamma)$. Either $\langle \overline{M} \rangle \in [\Sigma, \Delta, \Gamma]$ or $\mathcal{I}_1(\Delta)(\Gamma)$, in which case $\langle \overline{M} \rangle \in \mathsf{T}(\mathcal{I}_2)(\Delta)(\Gamma)$, or $\langle \overline{M} \rangle \equiv \langle M_1, \dots, M'_i, \dots, M_n \rangle$ and there is an $@_i \in \Sigma \cup \Gamma$ such that $@_i \overline{P} \overline{Q} \rightarrow_{\beta\eta} M'_i$,

$$\Delta \models_{\Sigma}^{\mathcal{I}_1} (\Gamma \xrightarrow{\langle M_1, \dots, M_i, \dots, M_n \rangle} \Delta)[\Gamma],$$

and

$$\Gamma, x : B_i[M_j/y_j]_{j=1}^{i-1} \models_{\Sigma}^{\mathcal{I}_2} ((\Gamma \xrightarrow{\langle x_1, \dots, x_m, M'_i \rangle} \Gamma, x : B_i)[M_j/y_j]_{j=1}^{i-1})[\Gamma].$$

By Lemma 11.9, we have

$$\Delta \models_{\Sigma}^{\mathcal{I}_2} (\Gamma \xrightarrow{\langle M_1, \dots, M_i, \dots, M_n \rangle} \Delta)[\Gamma],$$

and

$$\Gamma, x : B_i[M_j/y_j]_{j=1}^{i-1} \models_{\Sigma}^{\mathcal{I}_2} ((\Gamma \xrightarrow{\langle x_1, \dots, x_m, M'_i \rangle} \Gamma, x : B_i)[M_j/y_j]_{j=1}^{i-1})[\Gamma]$$

so that $\langle \overline{M} \rangle \in \mathsf{T}(\mathcal{I}_2)(\Delta)(\Gamma)$. \blacksquare

The following results allows us to define the least fixed-point.

Lemma 11.12

The operator \mathbb{T} is continuous.

Proof We must show that if $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2 \sqsubseteq \dots$ is an ω -chain, then

$$\mathbb{T}(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i) = \bigsqcup_{i=1}^{\omega} (\mathbb{T}(\mathcal{I}_i)).$$

We establish the inclusion in two directions.

For any $j \geq 1$, $\mathcal{I}_j \sqsubseteq \bigsqcup_{i=1}^{\omega} \mathcal{I}_i$. So, by Lemma 11.11, $\mathbb{T}(\mathcal{I}_j) \sqsubseteq \mathbb{T}(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i)$. Since j is arbitrary, it follows that $\bigsqcup_{i=1}^{\omega} \mathbb{T}(\mathcal{I}_i) \sqsubseteq \mathbb{T}(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i)$.

For the converse, suppose $\langle \overline{M} \rangle \in \mathbb{T}(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i)(\Delta)(\Gamma)$. If $\langle \overline{M} \rangle \in [\Sigma, \Delta, \Gamma]$ or $(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i)(\Delta)(\Gamma)$, then $\langle \overline{M} \rangle \in \mathbb{T}(\mathcal{I}_j)(\Delta)(\Gamma)$, for any $j \geq 1$, so that $\langle \overline{M} \rangle \in (\bigsqcup_{i=1}^{\omega} \mathbb{T}(\mathcal{I}_i)(\Delta)(\Gamma))$. Otherwise, we must have that $\langle \overline{M} \rangle \equiv M_1, \dots, M'_i, \dots, M_n$ and there is some $@_i \in \Sigma \cup \Gamma$ such that $@_i \overline{P} \overline{Q} \rightarrow M'_i$,

$$\Delta \xRightarrow{\Sigma} (\bigsqcup_{h=1}^{\omega} \mathcal{I}_h) (\Gamma \xrightarrow{\langle M_1, \dots, M_i, \dots, M_n \rangle} \Delta) [\Gamma],$$

for some M_i , and

$$\Gamma, x : B_i [M_j / y_j]_{j=1}^{i-1} \xRightarrow{\Sigma} ((\Gamma \xrightarrow{\langle x_1, \dots, x_k, M \rangle} \Gamma, x : B_i) [M_j / y_j]_{j=1}^{i-1}) [\Gamma].$$

By Lemmas 11.9 and 11.10, there is a $g \geq 1$ such that

$$\Delta \xRightarrow{\Sigma} \mathcal{I}_g (\Gamma \xrightarrow{\langle M_1, \dots, M_i, \dots, M_n \rangle} \Delta) [\Gamma],$$

and

$$\Gamma, x : B_i [M_j / y_j]_{j=1}^{i-1} \xRightarrow{\Sigma} ((\Gamma \xrightarrow{\langle x_1, \dots, x_m, M'_i \rangle} \Gamma, x : B_i) [M_j / y_j]_{j=1}^{i-1}) [\Gamma].$$

So $\langle \overline{M} \rangle \in \mathbb{T}(\mathcal{I}_g)(\Delta)(\Gamma) \sqsubseteq (\bigsqcup_{i=1}^{\omega} \mathbb{T}(\mathcal{I}_i)(\Delta)(\Gamma))$. Since $\langle \overline{M} \rangle$, Δ and Γ are arbitrary. It follows that $\mathbb{T}(\bigsqcup_{i=1}^{\omega} \mathcal{I}_i) \sqsubseteq \bigsqcup_{i=1}^{\omega} \mathbb{T}(\mathcal{I}_i)$. \blacksquare

Recall Tarski's (1955) theorem that if X is a complete lattice, then any monotone mapping $X \xrightarrow{f} X$ has a least fixed-point. Since \mathbb{T} is also continuous, its least fixed-point is

$$\mathbb{T}^{\omega}(\perp(\Sigma)) =_{def} \bigsqcup_{i=1}^{\omega} \mathbb{T}^i(\perp(\Sigma))$$

i.e., the closure ordinal of \mathbb{T} is ω . We abbreviate $\mathbb{T}^{\omega}(\perp(\Sigma))$ to \mathbb{T}^{ω} , where no confusion can arise. We obtain (recalling Lemma 11.8) the following;

Proposition 11.13

\mathbb{T}^{ω} is a Kripke Σ - $\lambda\Pi$ -structure.

Proof (Sketch) Definition 11.2 tells us that a Σ - $\lambda\Pi$ -Herbrand structure is always a Kripke $\lambda\Pi$ -structure. Lemma 11.8 tells us that the operation T is a natural transformation between Σ - $\lambda\Pi$ -Herbrand structures. Applying T to a Σ - $\lambda\Pi$ -Herbrand structure always returns a Σ - $\lambda\Pi$ -Herbrand structure. A limiting argument tells us that applying it ω -times still returns a Σ - $\lambda\Pi$ -Herbrand structure. \blacksquare

The model $\langle \mathsf{T}^\omega, \llbracket - \rrbracket_{\mathsf{T}^\omega}^- \rangle$ is thus given by the Kripke $\lambda\Pi$ -structure together with the standard term interpretation. We must show that this model is well-defined. Since it is a term model, defined by constructing realizers, this will amount to a soundness argument.

Proposition 11.14

$\langle \mathsf{T}^\omega, \llbracket - \rrbracket_{\mathsf{T}^\omega}^- \rangle$ is well-defined.

Proof It is clear that all the contexts (programs) are properly interpreted. It remains to show that consequences are properly interpreted over each context and world (when they are defined in the model). The argument proceeds by induction on the structure of derivations in \mathbf{C} .

Suppose we have a derivation in \mathbf{C} consisting of an axiom, *i.e.*,

$$\frac{}{\vdash_\Sigma \Gamma \xrightarrow{\langle @_1, \dots, @_n \rangle} \Delta}$$

where each $@_i \in \Sigma \cup \Gamma$ and $\Gamma \vdash_\Sigma @_i : B_i[M_j/y_j]_{j=1}^{i-1}$ for $1 \leq i \leq n$. We have to show that

$$\Delta \Vdash_\Sigma^{\mathsf{T}^\omega} (\Gamma \xrightarrow{\langle @_1, \dots, @_n \rangle} \Delta)[\Gamma]$$

holds. By definition, $\Gamma \xrightarrow{\langle @_1, \dots, @_n \rangle} \Delta \in [\Sigma, \Delta, \Gamma]$ and so by the definition of T belong to T^ω .

Suppose we have a derivation in \mathbf{C} , the last rule applied being the following:

$$\frac{\vdash_\Sigma \Gamma \xrightarrow{\langle M_1, \dots, M_i, \dots, M_n \rangle} \Delta}{\vdash_\Sigma \Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_n \rangle} \Delta}$$

where \mathbf{N} proves $\Gamma \vdash_\Sigma M'_i : B_i[M_j/y_j]_{j=1}^{i-1}$, $@_i : \Pi z_{i1} : B_{i1} \dots \Pi z_{ip} : B_{ip} \cdot (C_{i1} \rightarrow (C_{i2} \rightarrow (\dots (C_{iq} \rightarrow D_i) \dots))) \in \Sigma \cup \Gamma$, $@_i \overline{P} \overline{Q} \rightarrow_{\beta\eta} M'_i$, for some $1 \leq i \leq n$ and p, q possibly 0. We must show that if

$$\Delta \Vdash_\Sigma^{\mathsf{T}^\omega} (\Gamma \xrightarrow{\langle M_1, \dots, M_i, \dots, M_n \rangle} \Delta)[\Gamma] \tag{11.1}$$

and

$$\Gamma, x : B_i[M_j/y_j]_{j=1}^{i-1} \Vdash_\Sigma^{\mathsf{T}^\omega} ((\Gamma \xrightarrow{\langle x_1, \dots, x_m, M'_i \rangle} \Gamma, x : B_i)[M_j/y_j]_{j=1}^{i-1})[\Gamma] \tag{11.2}$$

then

$$\Delta \Vdash_{\Sigma}^{\mathbb{T}^{\omega}} (\Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_n \rangle} \Delta) [\Gamma] \quad (11.3)$$

It immediately follows from the definition of \mathbb{T} that if (11.1) and (11.2) obtain, then $\Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_n \rangle} \Delta(\mathbb{T}(\mathbb{T}^{\omega}))(\Delta)(\Gamma)$ holds. Also $\mathbb{T}(\mathbb{T}^{\omega}) = \mathbb{T}^{\omega}$, so it follows that (11.3) obtains.

Assume that the last rule applied was introduction:

$$\frac{\vdash_{\Sigma} \Gamma, x : A \xrightarrow{\langle M_1, \dots, M_n, x, M \rangle} \Gamma, x : A, y : B}{\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M_n, \lambda x : A. M \rangle} \Gamma, y : \Pi x : A. B}$$

We have to show that if

$$\Gamma, x : A, y : B \Vdash_{\Sigma}^{\mathbb{T}^{\omega}} (\Gamma, x : A \xrightarrow{\langle M_1, \dots, M_n, x, M \rangle} \Gamma, x : A, y : B) [\Gamma, x : A]$$

then

$$\Gamma, y : \Pi x : A. B \Vdash_{\Sigma}^{\mathbb{T}^{\omega}} (\Gamma \xrightarrow{\langle M_1, \dots, M_n, \lambda x : A. M \rangle} \Gamma, y : \Pi x : A. B) [\Gamma]$$

We apply the adjoint $\Pi_{\Gamma, A}$ to $\mathcal{I}(\Gamma, x : A, y : B)(\Gamma, x : A)$ and also note that the realization $\Gamma, x : A, y : B \rightarrow \Gamma, y : \Pi x : A. B$ is an arrow in $\mathcal{P}(\Sigma)$. We are thus able to move to the fibre $\mathbb{T}^{\omega}(\Gamma, y : \Pi x : A. B)(\Gamma)$ in which the arrow $\Gamma \xrightarrow{\langle M_1, \dots, M_n, \lambda x : A. M \rangle} \Gamma, y : \Pi x : A. B$ is defined and hence $\Gamma, y : \Pi x : A. B \Vdash_{\Sigma}^{\mathbb{T}^{\omega}} (\Gamma \xrightarrow{\langle M_1, \dots, M_n, \lambda x : A. M \rangle} \Gamma, y : \Pi x : A. B) [\Gamma]$. The term $\lambda x : A. M$ arises when we apply the adjunction.

Finally, we consider the case where we have applied the rule for equality:

$$\frac{\vdash_{\Sigma} \Gamma \xrightarrow{\sigma} \Delta}{\vdash_{\Sigma} \Gamma' \xrightarrow{\sigma'} \Delta'}$$

where $\Gamma =_{\beta\eta} \Gamma'$, $\sigma =_{\beta\eta} \sigma'$ and $\Delta =_{\beta\eta} \Delta'$ are all defined componentwise. We have to show that if

$$\Delta \Vdash_{\Sigma}^{\mathbb{T}^{\omega}} (\Gamma \xrightarrow{\sigma} \Delta) [\Gamma]$$

then

$$\Delta \Vdash_{\Sigma}^{\mathbb{T}^{\omega}} (\Gamma' \xrightarrow{\sigma'} \Delta') [\Gamma']$$

where $\Gamma =_{\beta\eta} \Gamma'$, $\sigma =_{\beta\eta} \sigma'$ and $\Delta =_{\beta\eta} \Delta'$. By the definition of $\mathcal{P}(\Sigma)$ and $\mathcal{B}(\Sigma)$ there are realizations $\Delta \rightarrow \Delta'$ and $\Gamma \rightarrow \Gamma'$ in each respectively. We are then able to move to the fibre $\mathbb{T}^{\omega}(\Delta')(\Gamma')$ which contains the realization $\Gamma' \xrightarrow{\sigma'} \Delta'$ and hence $\Delta' \Vdash_{\Sigma}^{\mathbb{T}^{\omega}} (\Gamma' \xrightarrow{\sigma'} \Delta') [\Gamma']$. \blacksquare

We are now able to give the completeness theorem. Its proof follows the pattern found in (Miller 1989).

Theorem 11.15 (Completeness)

Let Σ be a signature. If, for some Δ in $\mathcal{P}(\Sigma)$ $\Delta \Vdash_{\Sigma}^{\mathsf{T}^{\omega}} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$, then \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\sigma} \Delta$, where $\Gamma \equiv x_1 : A_1, \dots, x_m : A_m$, $\Delta \equiv y_1 : B_1, \dots, y_n : B_n$ and $\sigma \equiv \langle M_1, \dots, M_n \rangle$.

Proof If

$$\Delta \Vdash_{\Sigma}^{\mathsf{T}^{\omega}} \underbrace{(x_1 : A_1, \dots, x_m : A_m)}_{\Gamma} \xrightarrow{\langle M_1, \dots, M_n \rangle^{\sigma}} \underbrace{(y_1 : B_1, \dots, y_n : B_n)}_{\Delta} [\Gamma]$$

then, according to Definitions 3.26 and 11.3, there is an arrow $\Gamma \xrightarrow{\sigma} \Delta$ in $\mathsf{T}^{\omega}(\Delta)(\Gamma)$. We must show that the realization $\vdash_{\Sigma} \Gamma \xrightarrow{\sigma} \Delta$ is derivable in \mathbf{C} .

We proceed by induction on the structure of the types of the components of the arrows and the number of applications of T . To this end, we proceed in a manner similar to (Miller 1989) and (Pym 1990) by assigning to each triple (Θ, Ξ, ρ) , where $\rho = \langle N_1, \dots, N_p \rangle$, the ordinal measure $\omega \cdot (r - 1) + \sum_{i=1}^p s_i$, where each $s_i \geq 0$ is the number of λ -abstractions in the head of the object N_i and $r \geq 0$ is the least r such that

$$\Xi \Vdash_{\Sigma}^{\mathsf{T}^r(\perp(\Sigma))} (\Theta \xrightarrow{\rho} \Xi)[\Theta],$$

and proceed by induction on this measure.

(Base Case) The measure of (Δ, Γ, σ) is $0 (= \omega \cdot 0 + 0)$. Each of the objects M_i must be atomic, *i.e.*, of the form $@_i \in \Sigma \cup \Gamma$ and $\Delta \Vdash_{\Sigma}^{\mathsf{T}^1(\perp(\Sigma))} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$.

Therefore, there must be an arrow $\Gamma \xrightarrow{\langle @_1, \dots, @_n \rangle} \Delta$, such that $\langle @_1, \dots, @_n \rangle$ in $[\Sigma, \Delta, \Gamma]$. It follows that \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle @_1, \dots, @_n \rangle} \Delta$.

(Inductive Case) The measure of (Δ, Γ, σ) is $\omega \cdot \alpha + \beta$, so that $\Delta \Vdash_{\Sigma}^{\mathsf{T}^{\alpha+1}(\perp(\Sigma))} s(\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$. Either this measure is a limit ordinal, *i.e.* $\beta = 0$ or it is a successor, *i.e.*, $\beta > 0$.

($\beta = 0$) In this case we have that $\alpha > 0$ or σ is atomic. There are two possibilities:

1. There is an arrow $\Gamma \xrightarrow{\langle \bar{@} \rangle} \Delta$ in $[\Sigma, \Delta, \Gamma]$, with $\langle \bar{@} \rangle = \sigma$. It then follows that \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle @_1, \dots, @_n \rangle} \Delta$;
2. There is in $\mathsf{T}^{\alpha+1}(\perp(\Sigma))(\Delta)(\Gamma)$ some $\Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_p \rangle} \Delta$, constructed from some $\Gamma \xrightarrow{\langle M_1, \dots, M_n \rangle} \Delta$, in $\mathsf{T}^{\alpha}(\perp(\Sigma))(\Delta)(\Gamma)$, where M'_i is such that

$$\Gamma, x : B_i [M_j / y_j]_{j=1}^{i-1} \Vdash_{\Sigma}^{\mathsf{T}^{\alpha}} ((\Gamma \xrightarrow{\langle x_1, \dots, x_m, M'_i \rangle} \Gamma, x : B_i) [M_j / y_j]_{j=1}^{i-1}) [\Gamma]$$

Since this satisfaction requires fewer applications of \mathbf{T} , we have, by the induction hypothesis that \mathbf{N} proves $\Gamma \vdash_{\Sigma} M'_i : B_i[M_j/y_j]_{j=0}^{i-1}$,

Therefore, by resolution \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_p \rangle} \Delta$.

($\beta > 0$) In this case, we have that σ is not atomic. In particular, suppose that M_i is not atomic. There is in $\mathbf{T}^{\alpha+1}(\perp(\Sigma))(\Delta)(\Gamma)$ some $\Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_p \rangle} \Delta$, where M'_i is such that

$$\Gamma, x : B_i[M_j/y_j]_{j=1}^{i-1} \xRightarrow{\mathbf{T}^{\alpha}} ((\Gamma \xrightarrow{\langle x_1, \dots, x_m, M'_i \rangle} \Gamma, x : B_i)[M_j/y_j]_{j=1}^{i-1})[\Gamma].$$

Since M'_i is of the form $\textcircled{P} \overline{P} \overline{Q}$, it follows that $\langle x_1, \dots, x_m, M'_i \rangle$ and $\langle M_1, \dots, M'_i, \dots, M_p \rangle$ have fewer abstractions than $\Gamma \xrightarrow{\sigma} \Delta$. So, by the induction hypothesis, it follows that \mathbf{N} proves $\Gamma \vdash_{\Sigma} M'_i : B_i[M_j/y_j]_{j=1}^{i-1}$ and \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_p \rangle} \Delta$. Therefore, by resolution, \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\langle M_1, \dots, M'_i, \dots, M_p \rangle} \Delta$. By Lemma 11.10, if $\Delta \xRightarrow{\mathbf{T}^{\omega}} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$, then there is a $k \geq 1$ such that $\Delta \xRightarrow{\mathbf{T}^k(\perp(\Sigma))} (\Gamma \xrightarrow{\sigma} \Delta)[\Gamma]$. Therefore, by the inductive argument above, \mathbf{C} proves $\vdash_{\Sigma} \Gamma \xrightarrow{\sigma} \Delta$. ■

Chapter 12

Encoding Sequent Systems in LF

We wish to examine the relationship between resolution in the object-logic and resolution in the meta-logic. To achieve this we need a method of encoding sequent systems into LF. We are faced with a problem since LF does not provide a suitable meta-theory for sequent systems. This is because one of the desired properties of logical frameworks is that the structural rules of the logical framework correspond to the rules of the object-logic and the structural rules of a multi-conclusioned sequent system do not correspond to the structural rules of the $\lambda\Pi$ -calculus. This strongly suggests that it may not be possible to represent sequent systems adequately in LF. Pfenning (2000), however, has shown that it is possible to represent both the sequent presentation of classical and intuitionistic logic in LF. We reconstruct Pfenning's encoding in our setting and extend it to higher-order intuitionistic logic. The key idea of this encoding is that instead of using the 'usual' sequent rules, (Gentzen 1934), one uses the rules in Kleene's (1952) G_3 (System GKi in (Troelstra & Schwichtenberg 1996)). The reason for this choice is that if one views the sequent calculus for intuitionistic logic as a calculus of proof-search for natural deduction proofs, then one can systematically derive G_3 . Details of this derivation can be found in (Pfenning 2000). Thus the proof-objects of intuitionistic G_3 are essentially the same as those for the corresponding intuitionistic natural deduction system.

The encoding of classical sequent calculus is based on the extension of intuitionistic G_3 to multi-conclusioned sequents. This breaks the relationship between the sequent system and the corresponding natural deduction system. The G_3 systems differ from the 'usual' presentation of the sequent calculus in that all the rules have the principal formula of the rule present in each premiss. Having the principal formula in each premiss is also essential to the proof of adequacy. The encoding itself requires us to ignore any context and principal formulæ occurring in the premiss. The logical framework then handles the context correctly.

Both our presentation and encoding of higher-order intuitionistic logic is original, although higher-order classical logic has been encoded in LF before, (Harper

et al. 1993).

12.1 Intuitionistic Logic

We present the intuitionistic system G_{3i} , which is a judged proof system based on Kleene's (1952) G_3 . Here, unlike Pfenning, we use a single judgement and do not include cut in our definition.

Definition 12.1 (G_{3i})

The judged proof system G_{3i} is given by the alphabet $A = (S, V, E, C, J)$, where:

- $S = \{\iota, o\}$;
- $V = \{\iota\}$;
- $E = \{\top, \wedge, \vee, \supset, \forall, \exists\}$;
- $C = \{\top, \wedge, \vee, \supset, \forall, \exists\}$;
- $J = \{\text{proof}\}$.

Each connective is assigned an arity and a level. \top has arity o and level 1, \wedge, \vee and \supset all have arity $(o, o) \rightarrow o$ and level 1. \forall and \exists have arity $(\iota \rightarrow o) \rightarrow o$ and level 2. The judgement **proof** has arity 0. Together with the following rules:

$$\frac{}{\Delta, \text{proof}(\phi) \vdash_{G_{3i}} \text{proof}(\phi)} Ax$$

$$\frac{\Delta, \text{proof}(\phi \wedge \psi), \text{proof}(\phi), \text{proof}(\psi) \vdash_{G_{3i}} \text{proof}(\chi)}{\Delta, \text{proof}(\phi \wedge \psi) \vdash_{G_{3i}} \text{proof}(\chi)} \wedge l$$

$$\frac{\Delta \vdash_{G_{3i}} \text{proof}(\phi) \quad \Delta \vdash_{G_{3i}} \text{proof}(\psi)}{\Delta \vdash_{G_{3i}} \text{proof}(\phi \wedge \psi)} \wedge r$$

$$\frac{\Delta, \text{proof}(\phi \vee \psi), \text{proof}(\phi) \vdash_{G_{3i}} \text{proof}(\chi) \quad \Delta' \vdash_{G_{3i}} \text{proof}(\psi)}{\Delta, \text{proof}(\phi \vee \psi) \vdash_{G_{3i}} \text{proof}(\chi)} \vee l$$

where $\Delta' = \Delta, \text{proof}(\phi \vee \psi), \text{proof}(\psi)$.

$$\frac{\Delta \vdash_{G_{3i}} \text{proof}(\phi_i)}{\Delta \vdash_{G_{3i}} \text{proof}(\phi_1 \vee \phi_2)} \vee r_i$$

$$\frac{\Delta, \text{proof}(\phi \supset \psi) \vdash_{G_{3i}} \text{proof}(\phi) \quad \Delta, \text{proof}(\phi \supset \psi), \text{proof}(\psi) \vdash_{G_{3i}} \text{proof}(\chi)}{\Delta, \text{proof}(\phi \supset \psi) \vdash_{G_{3i}} \text{proof}(\chi)} \supset l$$

$$\frac{\Delta, \text{proof}(\phi) \vdash_{G_3i} \text{proof}(\psi)}{\Delta \vdash_{G_3i} \text{proof}(\phi \supset \psi)} \supset r$$

$$\frac{}{\Delta \vdash_{G_3i} \top} \top$$

$$\frac{\Delta, \text{proof}(\forall x\phi), \text{proof}(\phi) \vdash_{G_3i} \text{proof}(\chi)}{\Delta, \text{proof}(\forall x\phi) \vdash_{G_3i} \text{proof}(\chi)} \forall l$$

x free in ϕ

$$\frac{\Delta \vdash_{G_3i} \text{proof}(\phi)}{\Delta \vdash_{G_3i} \text{proof}(\forall x\phi)} \forall r$$

x free in ϕ

$$\frac{\Delta, \text{proof}(\exists x\phi), \text{proof}(\phi) \vdash_{G_3i} \text{proof}(\chi)}{\Delta, \text{proof}(\exists x\phi) \vdash_{G_3i} \text{proof}(\chi)} \exists l$$

x free in ϕ

$$\frac{\Delta \vdash_{G_3i} \text{proof}(\phi)}{\Delta \vdash_{G_3i} \text{proof}(\exists x\phi)} \exists r$$

x free in ϕ . The antecedent is a multiset throughout. ■

We now show that G_3i is provably equivalent to the usual sequent presentation of intuitionistic logic. The rules for the sequent calculus LJ can be found in Appendix C.

The following result is a simplified version of Kleene's proof that LJ and G_3 are provably equivalent. Kleene takes contexts in LJ to be sequences of formulæ rather than sets. This makes his proof more complicated.

Lemma 12.2 (Kleene (1952))

Let ϕ be a formula in G_3i , Δ a multiset of formulæ of G_3i and Δ' a set of formulæ of LJ . Then

$$\Delta \vdash_{G_3i} \text{proof}(\phi) \text{ if and only if } \Delta' \vdash_{LJ} \text{proof}(\phi)$$

where Δ and Δ' contain the same formulæ.

Proof We begin by proving the left to right direction. We show that each inference rule in G_3i can be translated to a proof-tree in LJ with the same premisses and conclusion. The right rules, Ax and \top , are identical for each system. We only need to show the translation for the left rules. We list the rules of G_3i on the left and the corresponding proof-tree in LJ on the right. We drop

the judgements for readability.

$$\begin{array}{c}
\frac{\Delta, \phi \wedge \psi, \phi, \psi \vdash_{G_{3i}} \chi}{\Delta, \phi \wedge \psi \vdash_{G_{3i}} \chi} \wedge l \quad \frac{\Delta, \phi \wedge \psi, \phi, \psi \vdash_{LJ} \chi}{\Delta, \phi \wedge \psi, \phi \wedge \psi \vdash_{LJ} \chi} \wedge l}{\Delta, \phi \wedge \psi \vdash_{G_{3i}} \chi} \text{contraction} \\
\\
\frac{\Delta, \phi \vee \psi, \phi \vdash_{G_{3i}} \chi \quad \Delta, \phi \vee \psi, \psi \vdash_{G_{3i}} \chi}{\Delta, \phi \vee \psi \vdash_{G_{3i}} \chi} \vee l \quad \frac{\Delta, \phi \vee \psi, \phi \vdash_{G_{3i}} \chi \quad \Delta, \phi \vee \psi, \psi \vdash_{G_{3i}} \chi}{\Delta, \phi \vee \psi, \phi \vee \psi \vdash_{G_{3i}} \chi}}{\Delta, \phi \vee \psi \vdash_{G_{3i}} \chi} \\
\\
\frac{\Delta, \phi \supset \psi \vdash_{G_{3i}} \phi \quad \Delta, \phi \supset \psi, \psi \vdash_{G_{3i}} \chi}{\Delta, \phi \supset \psi \vdash_{G_{3i}} \chi} \supset l \quad \frac{\Delta, \phi \supset \psi \vdash_{G_{3i}} \chi \quad \Delta, \phi \supset \psi, \psi \vdash_{G_{3i}} \chi}{\Delta, \phi \supset \psi, \phi \supset \psi \vdash_{G_{3i}} \chi}}{\Delta, \phi \supset \psi \vdash_{G_{3i}} \chi} \\
\\
\frac{\Delta, \forall x \phi, \phi \vdash_{G_{3i}} \chi}{\Delta, \forall x \phi \vdash_{G_{3i}} \chi} \forall l \quad \frac{\Delta, \forall x \phi, \phi \vdash_{G_{3i}} \chi}{\Delta, \forall x \phi, \forall x \phi \vdash_{G_{3i}} \chi} \forall l}{\Delta, \forall x \phi \vdash_{G_{3i}} \chi} \text{contraction} \\
\\
\frac{\Delta, \exists x \phi, \phi \vdash_{G_{3i}} \chi}{\Delta, \exists x \phi \vdash_{G_{3i}} \chi} \exists l \quad \frac{\Delta, \exists x \phi, \phi \vdash_{G_{3i}} \chi}{\Delta, \exists x \phi, \forall x \phi \vdash_{G_{3i}} \chi} \exists l}{\Delta, \forall x \phi \vdash_{G_{3i}} \chi} \text{contraction}
\end{array}$$

We have stated the use of contraction explicitly. It is not a separate rule but follows directly from the fact that contexts are either multisets or sets. We now have to show the right to left direction. We show that for every rule in LJ there is a proof-tree in $G_{3i'}$ with the same premisses and conclusion. $G_{3i'}$ is G_{3i} plus weakening and contraction. Again, we just show the left rules.

$$\begin{array}{c}
\frac{\Delta, \phi, \psi \vdash_{LJ} \chi}{\Delta, \phi \wedge \psi \vdash_{LJ} \chi} \wedge l \quad \frac{\Delta, \phi, \psi \vdash_{G_{3i'}} \chi}{\Delta, \phi \wedge \psi, \phi, \psi \vdash_{G_{3i'}} \chi} \text{weakening}}{\Delta, \phi \wedge \psi \vdash_{G_{3i'}} \chi} \wedge l \\
\\
\frac{\Delta, \phi \vdash_{LJ} \chi \quad \Delta, \psi \vdash_{LJ} \chi}{\Delta, \phi \vee \psi \vdash_{LJ} \chi} \vee l \quad \frac{\Delta, \phi \vdash_{G_{3i'}} \chi \quad \Delta, \psi \vdash_{G_{3i'}} \chi}{\Delta, \phi \vee \psi, \phi \vdash_{G_{3i'}} \chi \quad \Delta, \phi \vee \psi, \psi \vdash_{G_{3i'}} \chi}}{\Delta, \phi \vee \psi \vdash_{G_{3i'}} \chi} \\
\\
\frac{\Delta \vdash_{LJ} \phi \quad \Delta, \psi \vdash_{LJ} \chi}{\Delta, \phi \supset \psi \vdash_{LJ} \chi} \supset l \quad \frac{\Delta \vdash_{G_{3i'}} \phi \quad \Delta, \psi \vdash_{G_{3i'}} \chi}{\Delta, \phi \supset \psi \vdash_{G_{3i'}} \phi \quad \Delta, \phi \supset \psi, \psi \vdash_{G_{3i'}} \chi}}{\Delta, \phi \supset \psi \vdash_{G_{3i'}} \chi}
\end{array}$$

$$\frac{\Delta, \phi \vdash_{LJ} \chi}{\Delta, \forall x \phi \vdash_{LJ} \chi} \forall l \quad \frac{\frac{\Delta, \phi \vdash_{LJ} \chi}{\Delta, \forall x \phi, \phi \vdash_{LJ} \chi}}{\Delta, \forall x \phi \vdash_{LJ} \chi}$$

$$\frac{\Delta, \phi \vdash_{LJ} \chi}{\Delta, \exists x \phi \vdash_{LJ} \chi} \exists l \quad \frac{\frac{\Delta, \phi \vdash_{G_3i'} \chi}{\Delta, \exists x \phi, \phi \vdash_{G_3i'} \chi}}{\Delta, \exists x \phi \vdash_{G_3i'} \chi}$$

We now need to show that weakening and contraction are admissible in G_3i . These are proven by induction over the depth of the proof-tree. We start with the base case, *i.e.* we have a proof-tree of depth 1:

$$\overline{\Delta, \phi, \phi \vdash_{G_3i} \chi}$$

where either χ is atomic or χ is \top . If χ is \top then we can use the rule \top ,

$$\overline{\Delta, \phi \vdash_{G_3i} \top} \top$$

and we are done. If χ is atomic then either $\chi \in \Delta$ or $\chi = \phi$. If $\chi = \phi$ then we also have the axiom

$$\overline{\Delta, \phi \vdash_{G_3i} \phi} Ax$$

and we are done. If $\chi \in \Delta$ then we have the axiom

$$\overline{\Delta, \phi \vdash_{G_3i} \chi} Ax$$

since χ is still in Δ and we complete the base case.

We prove $\wedge r$ and $\wedge l$ to illustrate how the induction step works. The other cases are similar.

We begin with $\wedge r$. We have the proof-tree

$$\frac{\Delta, \chi, \chi \vdash_{G_3i} \phi \quad \Delta, \chi, \chi \vdash_{G_3i} \psi}{\Delta, \chi, \chi \vdash_{G_3i} \phi \wedge \psi} \wedge r$$

and we can apply the induction hypothesis to the premiss. We obtain $\Delta, \chi \vdash_{G_3i} \phi$ and $\Delta, \chi \vdash_{G_3i} \psi$. We now apply $\wedge r$ to obtain

$$\frac{\Delta, \chi \vdash_{G_3i} \phi \quad \Delta, \chi \vdash_{G_3i} \psi}{\Delta, \chi \vdash_{G_3i} \phi \wedge \psi} \wedge r$$

which completes this case.

We now look at $\wedge l$. There are three cases here. The first is when the formulæ being contracted are distinct from the principal formula. The second is when the principal formula is one of the contracted formulæ. The third is when one of the contracted formulæ is one of the formulæ ‘used up’ by the rule. We begin with the first case. We have a proof-tree

$$\frac{\Delta, \chi, \chi, \phi \wedge \psi, \phi, \psi \vdash_{G_{3i}} \tau}{\Delta, \chi, \chi, \phi \wedge \psi \vdash_{G_{3i}} \tau} \wedge l$$

Applying the induction hypothesis to the premiss we obtain $\Delta, \chi, \phi \wedge \psi, \phi, \psi \vdash_{G_{3i}} \tau$. We apply $\wedge l$ to obtain

$$\frac{\Delta, \chi, \phi \wedge \psi, \phi, \psi \vdash_{G_{3i}} \tau}{\Delta, \chi, \phi \wedge \psi \vdash_{G_{3i}} \tau} \wedge l$$

thus completing the case.

We now assume that we have the proof-tree

$$\frac{\Delta, \phi \wedge \psi, \phi \wedge \psi, \phi, \psi \vdash_{G_{3i}} \tau}{\Delta, \phi \wedge \psi, \phi \wedge \psi \vdash_{G_{3i}} \tau} \wedge l$$

We apply the induction hypothesis to the premiss to obtain $\Delta, \phi \wedge \psi, \phi, \psi \vdash_{G_{3i}} \tau$. We apply $\wedge l$ to obtain

$$\frac{\Delta, \phi \wedge \psi, \phi, \psi \vdash_{G_{3i}} \tau}{\Delta, \phi \wedge \psi \vdash_{G_{3i}} \tau} \wedge l$$

which completes this case.

When the contracted formula is a formula used up by the rule, the rule automatically does the contraction .

$$\frac{\Delta, \phi \wedge \psi, \phi, \phi, \psi \vdash_{G_{3i}} \tau}{\Delta, \phi \wedge \psi, \phi \vdash_{G_{3i}} \tau} \wedge l$$

We now look at weakening. For the base case we have the Ax rule and \top . If we have a proof-tree

$$\frac{}{\Delta \vdash_{G_{3i}} \chi} Ax$$

then we also have a proof-tree

$$\frac{}{\Delta, \phi \vdash_{G_{3i}} \chi} Ax$$

Similarly for \top , if we have a proof-tree

$$\frac{}{\Delta \vdash_{G_{3i}} \top} \top$$

then we also have

$$\frac{}{\Delta, \phi \vdash_{G_{3i}} \top} \top$$

The induction steps are straightforward, we provide $\supset r$ and $\supset l$ as examples. We begin with $\supset r$ and thus have the proof-tree

$$\frac{\Delta, \phi \vdash_{G_{3i}} \psi}{\Delta \vdash_{G_{3i}} \phi \supset \psi} \supset r$$

We apply the induction hypothesis to the premiss to obtain $\Delta, \chi, \phi \vdash_{G_{3i}} \psi$. We can now use $\supset r$ to obtain

$$\frac{\Delta, \chi, \phi \vdash_{G_{3i}} \psi}{\Delta, \chi \vdash_{G_{3i}} \phi \supset \psi} \supset r$$

which finishes the case.

We now prove $\supset l$. We have the proof-tree

$$\frac{\Delta, \phi \supset \psi \vdash_{G_{3i}} \phi \quad \Delta, \phi \supset \psi, \psi \vdash_{G_{3i}} \chi}{\Delta, \phi \supset \psi \vdash_{G_{3i}} \chi} \supset l$$

We apply the induction hypothesis to the premiss to obtain $\Delta, \phi \supset \psi, \tau \vdash_{G_{3i}} \phi$ and $\Delta, \phi \supset \psi, \psi, \tau \vdash_{G_{3i}} \chi$. Applying $\supset l$ gives the following proof-tree

$$\frac{\Delta, \phi \supset \psi, \tau \vdash_{G_{3i}} \phi \quad \Delta, \phi \supset \psi, \psi, \tau \vdash_{G_{3i}} \chi}{\Delta, \phi \supset \psi, \tau \vdash_{G_{3i}} \chi} \supset l$$

which completes this case and we are finished. ■

We now extend the rules of G_{3i} to rules for valid proof expressions. We label each hypothesis in Δ with a distinct proof variable y_i , introduce a set X of syntactic variables found in each formula in the judgement and label each succedent with a proof-object δ .

$$\frac{}{(X) \Delta, y : \mathbf{proof}(\phi) \vdash_{G_{3i}} y : \mathbf{proof}(\phi)} v\text{-HYP}$$

$$\frac{(X) \Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi) \vdash_{G_{3i}} \delta : \mathbf{proof}(\chi)}{(X) \Delta, y : \mathbf{proof}(\phi \wedge \psi) \vdash_{G_{3i}} \mathbf{AND-L}((y_1, y_2) : \delta, y) : \mathbf{proof}(\chi)} v\text{-AND-L}$$

$$\frac{(X) \Delta \vdash_{G_{3i}} \delta_1 : \mathbf{proof}(\phi) \quad (X) \Delta \vdash_{G_{3i}} \delta_2 : \mathbf{proof}(\psi)}{(X) \Delta \vdash_{G_{3i}} \mathbf{AND-R}(\delta_1, \delta_2) : \mathbf{proof}(\phi \wedge \psi)} \quad v\text{-AND-R}$$

$$\frac{(X) \Delta_1 \vdash_{G_{3i}} \delta_1 : \mathbf{proof}(\chi) \quad \Delta_2 \vdash_{G_{3i}} \delta_2 : \mathbf{proof}(\chi)}{(X) \Delta, y : \mathbf{proof}(\phi \vee \psi) \vdash_{G_{3i}} \mathbf{OR-L}(y_1 : \delta_1, y_2 : \delta_2, y) : \mathbf{proof}(\chi)} \quad v\text{-OR-L}$$

where $\Delta_1 = \Delta, y : \mathbf{proof}(\phi \vee \psi), y_1 : \mathbf{proof}(\phi)$ and $\Delta_2 = \Delta, y : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\Delta_2)$.

$$\frac{(X) \Delta \vdash_{G_{3i}} \delta_i : \mathbf{proof}(\phi_i)}{(X) \Delta \vdash_{G_{3i}} \mathbf{OR-R}_i(\delta_i) : \mathbf{proof}(\phi_i)} \quad v\text{-OR-R}_i$$

$$\frac{(X) \Delta, y : \mathbf{proof}(\phi \supset \psi) \vdash_{G_{3i}} \delta_1 : \mathbf{proof}(\phi) \quad (X) \Delta_1 \vdash_{G_{3i}} \delta_2 : \mathbf{proof}(\chi)}{(X) \Delta, y : \mathbf{proof}(\phi \supset \psi) \vdash_{G_{3i}} \mathbf{IMP-L}(\delta_1, y_2 : \delta_2, y) : \mathbf{proof}(\chi)} \quad v\text{-IMP-L}$$

where $\Delta_1 = \Delta, y : \mathbf{proof}(\phi \supset \psi), y_2 : \mathbf{proof}(\psi)$.

$$\frac{(X) \Delta, y : \mathbf{proof}(\phi) \vdash_{G_{3i}} \delta : \mathbf{proof}(\psi)}{(X) \Delta \vdash_{G_{3i}} \mathbf{IMP-R}(y : \delta) : \mathbf{proof}(\phi \supset \psi)} \quad v\text{-IMP-R}$$

$$\frac{}{(X) \Delta \vdash_{G_{3i}} \mathbf{TOP} : \mathbf{proof}(\top)} \quad v\text{-TOP}$$

$$\frac{(X) \Delta, y : \mathbf{proof}(\forall x \phi), y_1 : \mathbf{proof}(\phi) \vdash_{G_{3i}} \delta : \mathbf{proof}(\chi)}{(X) \Delta, y : \mathbf{proof}(\forall x \phi) \vdash_{G_{3i}} \mathbf{FORALL-L}(y_1 : \delta, y) : \mathbf{proof}(\chi)} \quad v\text{-FORALL-L}$$

$$\frac{(X, x) \Delta \vdash_{G_{3i}} \delta : \mathbf{proof}(\phi)}{(X) \Delta \vdash_{G_{3i}} \mathbf{FORALL-R}(\delta) : \mathbf{proof}(\forall x \phi)} \quad v\text{-FORALL-R}$$

$$\frac{(X) \Delta, y : \mathbf{proof}(\exists x \phi), y_1 : \mathbf{proof}(\phi) \vdash_{G_{3i}} \delta : \mathbf{proof}(\chi)}{(X) \Delta, y : \mathbf{proof}(\exists x \phi) \vdash_{G_{3i}} \mathbf{EXISTS-L}(y_1 : \delta, y) : \mathbf{proof}(\chi)} \quad v\text{-EXISTS-L}$$

$$\frac{(X, x) \Delta \vdash_{G_{3i}} \delta : \mathbf{proof}(\phi)}{(X) \Delta \vdash_{G_{3i}} \mathbf{EXISTS-R}(\delta) : \mathbf{proof}(\exists x \phi)} \quad v\text{-EXISTS-R}$$

We can use the judgements-as-types correspondence to allow us to define a $\lambda\Pi$ -signature $\Sigma_{G_{3i}}$. This can be found in Appendix A

Lemma 12.3 (Representing G_{3i} in LF)

The judged proof system G_{3i} can be adequately represented in LF.

Proof (Sketch) Proved by the usual methods (*cf.* (Harper et al. 1993)) ■

12.2 Classical Logic

Turning our attention to classical logic, we recall that the sequents have to be multi-conclusioned. This allows the law of excluded middle, for example, to hold.

$$\frac{\frac{\frac{}{Ax}}{\phi \vdash \phi} \neg R}{\vdash \neg\phi, \phi} \vee R}{(\neg\phi) \vee \phi}$$

Again, we follow Pfenning and use the classical version of G_3 . The principal formula of each rule is now present in the premisses of each right rule as well as the left rules. This introduces a lot of symmetry between rules. We do not use the symmetric rule for $\vee r$ because when we use it for proof-search in § 13, we need to have a single formula ‘discharged’ on the right.

Definition 12.4 (G_{3c})

The judged proof system G_{3c} is given by the alphabet $A = (S, V, E, C, J)$ where

- $S = \{\iota, o\}$,
- $V = \{\iota\}$,
- $E = \{\top, \wedge, \vee, \supset, \neg, \forall, \exists\}$,
- $C = \{\wedge, \vee, \supset, \neg, \forall, \exists\}$,
- $J = \{\text{ant}, \text{suc}\}$.

Each connective is assigned an arity and a level. \top has arity o and level 0, \neg has arity $o \rightarrow o$ and level 1, \wedge, \vee, \supset have arity $(o, o) \rightarrow o$ and level 1 and \forall and \exists have arity $(\iota \rightarrow o) \rightarrow o$ and level 2. The judgements both have arity 0. Together with the rules:

$$\frac{}{\Delta, \text{ant}(\phi) \vdash_{G_{3c}} \text{suc}(\phi)\Theta} Ax$$

$$\frac{}{\Delta \vdash_{G_{3c}} \text{suc}(\top), \Theta} \top$$

$$\frac{\Delta, \text{ant}(\phi \wedge \psi), \text{ant}(\phi), \text{ant}(\psi) \vdash_{G_{3c}} \Theta}{\Delta, \text{ant}(\phi \wedge \psi) \vdash_{G_{3c}} \Theta} \wedge l$$

$$\frac{\Delta \vdash_{G_{3c}} \text{suc}(\phi), \text{suc}(\phi \wedge \psi), \Theta \quad \Delta \vdash_{G_{3c}} \text{suc}(\psi), \text{suc}(\phi \wedge \psi), \Theta}{\Delta \vdash_{G_{3c}} \text{suc}(\phi \wedge \psi), \Theta} \wedge r$$

$$\frac{\Delta, \text{ant}(\phi \vee \psi), \text{ant}(\phi) \vdash_{G_{3c}} \Theta \quad \Delta, \text{ant}(\phi \vee \psi), \text{ant}(\psi) \vdash_{G_{3c}} \Theta}{\Delta, \text{ant}(\phi \vee \psi) \vdash_{G_{3c}} \Theta} \vee l$$

$$\begin{array}{c}
\frac{\Delta \vdash_{G_{3c}} \text{suc}(\phi_i), \text{suc}(\phi_1 \vee \phi_2), \Theta}{\Delta \vdash_{G_{3c}} \text{suc}(\phi_1 \vee \phi_2), \Theta} \vee r_i \\
\frac{\Delta, \text{ant}(\phi \supset \psi) \vdash_{G_{3c}} \text{suc}(\phi), \Theta \quad \Delta, \text{ant}(\phi \supset \psi), \text{ant}(\psi) \vdash_{G_{3c}} \Theta}{\Delta, \text{ant}(\phi \supset \psi) \vdash_{G_{3c}} \Theta} \supset l \\
\frac{\Delta, \text{ant}(\phi) \vdash_{G_{3c}} \text{suc}(\psi), \text{suc}(\phi \supset \psi), \Theta}{\Delta \vdash_{G_{3c}} \text{suc}(\phi \supset \psi), \Theta} \supset r \\
\frac{\Delta, \text{ant}(\neg\phi) \vdash_{G_{3c}} \text{suc}(\phi), \Theta}{\Delta, \text{ant}(\neg\phi) \vdash_{G_{3c}} \Theta} \neg l \\
\frac{\Delta, \text{ant}(\phi) \vdash_{G_{3c}} \text{suc}(\neg\phi), \Theta}{\Delta \vdash_{G_{3c}} \text{suc}(\neg\phi), \Theta} \neg r \\
\frac{\Delta, \text{ant}(\forall x\phi), \text{ant}(\phi) \vdash_{G_{3c}} \Theta}{\Delta, \text{ant}(\forall x\phi) \vdash_{G_{3c}} \Theta} \forall l \\
\frac{\Delta \vdash_{G_{3c}} \text{suc}(\phi), \text{suc}(\forall x\phi), \Theta}{\Delta \vdash_{G_{3c}} \text{suc}(\forall x\phi), \Theta} \forall r \\
\frac{\Delta, \text{ant}(\exists x\phi), \text{ant}(\phi) \vdash_{G_{3c}} \Theta}{\Delta, \text{ant}(\exists x\phi) \vdash_{G_{3c}} \Theta} \exists l \\
\frac{\Delta \vdash_{G_{3c}} \text{suc}(\phi), \text{suc}(\exists x\phi), \Theta}{\Delta \vdash_{G_{3c}} \text{suc}(\exists x\phi), \Theta} \exists r
\end{array}$$

where the antecedent and succedents are multisets. ■

We show that G_{3c} is equivalent to LK , the ‘usual’ sequent presentation of intuitionistic logic. The sequent system LK can be found in Appendix C. We have the following corollary to Lemma 12.2.

Corollary 12.5 (Kleene (1952))

Let Δ and Θ be multisets, Δ' and Θ' be sets and $\text{ant}(\phi)$ and $\text{suc}(\psi)$ be formulæ in G_{3c} . We have that

$$\Delta, \text{ant}(\phi) \vdash_{G_{3c}} \text{suc}(\psi), \Theta \text{ if and only if } \Delta', \text{ant}(\phi) \vdash_{LK} \text{suc}(\psi), \Theta'$$

where Δ and Δ' contain the same formulæ and Θ and Θ' contain the same formulæ.

Proof We begin by showing the left to right direction. We only show the right rules since the left rules follow from Lemma 12.2. We write the rules of G_{3c} on the left and the corresponding derivation in LK on the right. We omit

judgements to keep the width of the derivation small.

$$\begin{array}{c}
\frac{\Delta \vdash_{G_{3c}} \phi, \phi \wedge \psi, \Theta \quad \Delta \vdash_{G_{3c}} \psi, \phi \wedge \psi, \Theta}{\Delta \vdash_{G_{3c}} \phi \wedge \psi, \Theta} \quad \frac{\Delta \vdash_{LK} \phi, \phi \wedge \psi, \Theta \quad \Delta \vdash_{LK} \psi, \phi \wedge \psi, \Theta}{\frac{\Delta \vdash_{LK} \phi \wedge \psi, \phi \wedge \psi, \Theta}{\Delta \vdash_{LK} \phi \wedge \psi}}
\\
\\
\frac{\Delta \vdash_{G_{3c}} \phi_i, \phi_1 \vee \phi_2, \Theta}{\Delta \vdash_{G_{3c}} \phi_1 \vee \phi_2, \Theta} \vee r_i \quad \frac{\frac{\Delta \vdash_{LK} \phi_1, \phi_1 \vee \phi_2, \Theta}{\Delta \vdash_{LK} \phi_1 \vee \phi_2, \phi_1 \vee \phi_2, \Theta} \vee r_i}{\Delta \vdash_{LK} \phi_1 \vee \phi_2, \Theta} \text{contraction}
\\
\\
\frac{\Delta, \phi \vdash_{G_{3c}} \psi, \phi \supset \psi, \Theta}{\Delta \vdash_{G_{3c}} \phi \supset \psi, \Theta} \supset r \quad \frac{\frac{\Delta, \phi \vdash_{LK} \psi, \phi \supset \psi, \Theta}{\Delta \vdash_{LK} \phi \supset \psi, \phi \supset \psi, \Theta} \supset r}{\Delta \vdash_{LK} \phi \supset \psi} \text{contraction}
\\
\\
\frac{\Delta, \phi \vdash_{G_{3c}} \neg \phi, \Theta}{\Delta \vdash_{G_{3c}} \neg \phi, \Theta} \neg r \quad \frac{\frac{\Delta, \phi \vdash_{LK} \neg \phi, \Theta}{\Delta \vdash_{LK} \neg \phi, \neg \phi, \Theta} \neg r}{\Delta \vdash_{LK} \neg \phi, \Theta} \text{contraction}
\\
\\
\frac{\Delta \vdash_{G_{3c}} \phi, \forall x \phi, \Theta}{\Delta \vdash_{G_{3c}} \forall x \phi, \Theta} \forall r \quad \frac{\frac{\Delta \vdash_{LK} \phi, \forall x \phi, \Theta}{\Delta \vdash_{LK} \forall x \phi, \forall x \phi, \Theta} \forall r}{\Delta \vdash_{LK} \forall x \phi, \Theta} \text{contraction}
\\
\\
\frac{\Delta \vdash_{G_{3c}} \phi, \exists x \phi, \Theta}{\Delta \vdash_{G_{3c}} \exists x \phi, \Theta} \exists r \quad \frac{\frac{\Delta \vdash_{LK} \phi, \exists x \phi, \Theta}{\Delta \vdash_{LK} \exists x \phi, \exists x \phi, \Theta} \exists r}{\Delta \vdash_{LK} \exists x \phi, \Theta} \text{contraction}
\end{array}$$

We have explicitly stated the use of contraction to highlight the way that the derivation in LK represents the rules of G_{3c} . We now turn to the right to left direction. We work with G_{3c}' which is G_{3c} plus contraction and weakening. We only deal with the right rules. We place the rules of LK on the left and the corresponding derivation in G_{3c}' on the right.

$$\begin{array}{c}
\frac{\Delta \vdash_{LK} \phi, \Theta \quad \Delta \vdash_{LK} \psi, \Theta}{\Delta \vdash_{LK} \phi \wedge \psi, \Theta} \quad \frac{\frac{\Delta \vdash_{G_{3c}'} \phi, \Theta}{\Delta \vdash_{G_{3c}'} \phi, \phi \wedge \psi, \Theta} \quad \frac{\Delta \vdash_{G_{3c}'} \psi, \Theta}{\Delta \vdash_{G_{3c}'} \psi, \phi \wedge \psi, \Theta}}{\Delta \vdash_{G_{3c}'} \phi \wedge \psi}
\\
\\
\frac{\Delta \vdash_{LK} \phi_i, \Theta}{\Delta \vdash_{LK} \phi_1 \vee \phi_2} \vee r_i \quad \frac{\frac{\Delta \vdash_{G_{3c}'} \phi_i, \Theta}{\Delta \vdash_{G_{3c}} \phi_i, \phi_1 \vee \phi_2} \text{weakening}}{\Delta \vdash_{G_{3c}'} \phi_1 \vee \phi_2, \Theta} \vee r_i
\end{array}$$

$$\begin{array}{c}
\frac{\Delta, \phi \vdash_{LK} \psi, \Theta}{\Delta \vdash_{LK} \phi \supset \psi} \supset r \quad \frac{\frac{\Delta, \phi \vdash_{G_3c'} \psi, \Theta}{\Delta, \phi \vdash_{G_3c'} \psi, \phi \supset \psi, \Theta} \text{weakening}}{\Delta \vdash_{G_3c'} \phi \supset \psi, \Theta} \supset r \\
\\
\frac{\Delta \vdash_{LK} \phi, \Theta}{\Delta \vdash_{LK} \forall x \phi, \Theta} \forall r \quad \frac{\frac{\Delta \vdash_{G_3c'} \phi, \Theta}{\Delta \vdash_{G_3c'} \phi, \forall x \phi, \Theta} \text{weakening}}{\Delta \vdash_{G_3c'} \forall x \phi, \Theta} \forall r \\
\\
\frac{\Delta \vdash_{LK} \phi, \Theta}{\Delta \vdash_{LK} \exists x \phi, \Theta} \exists r \quad \frac{\frac{\Delta \vdash_{G_3c'} \phi, \Theta}{\Delta \vdash_{G_3c'} \phi, \exists x \phi, \Theta} \text{weakening}}{\Delta \vdash_{G_3c'} \exists x \phi, \Theta} \exists r
\end{array}$$

It remains to show that contraction and weakening are admissible in G_3c . This is done by induction over the inference rules of G_3c . We only show the cases which involve right rules since the cases for the left rules follow from Lemma 12.2. We begin with contraction and just show the induction step when the last rule used is $\supset r$, the other rules being similar. There are three cases here, the first is when the formulæ being contracted are distinct from the formulæ used in the rule. The second is when the principal formula is one of the contracted formulæ. The third is when one of the contracted formulæ is one of the formulæ ‘used up’ by the rule. We begin with the first case. We have the proof-tree

$$\frac{\Delta, \chi, \chi \vdash_{G_3c} \tau, \tau, \phi, \phi \wedge \psi, \Theta \quad \Delta, \chi, \chi \vdash_{G_3c} \tau, \tau, \psi, \phi \wedge \psi, \Theta}{\Delta, \chi, \chi \vdash_{G_3c} \tau, \tau, \phi \wedge \psi, \Theta} \wedge r$$

and we apply the induction hypothesis to the premiss to obtain $\Delta, \chi \vdash_{G_3c} \tau, \phi, \phi \wedge \psi, \Theta$ and $\Delta, \chi \vdash_{G_3c} \tau, \psi, \phi \wedge \psi, \Theta$. We now apply $\wedge r$ to obtain

$$\frac{\Delta, \chi \vdash_{G_3c} \tau, \phi, \phi \wedge \psi, \Theta \quad \Delta, \chi \vdash_{G_3c} \tau, \psi, \phi \wedge \psi, \Theta}{\Delta, \chi \vdash_{G_3c} \tau, \phi \wedge \psi, \Theta} \wedge r$$

which completes this case.

We turn to the second case and assume that we have the proof-tree

$$\frac{\Delta, \chi, \chi \vdash_{G_3c} \phi \wedge \psi, \phi, \phi \wedge \psi, \Theta \quad \Delta, \chi, \chi \vdash_{G_3c} \phi \wedge \psi, \psi, \phi \wedge \psi, \Theta}{\Delta, \chi, \chi \vdash_{G_3c} \phi \wedge \psi, \phi \wedge \psi, \Theta} \wedge r$$

and we apply the induction hypothesis to the premiss to obtain $\Delta, \chi \vdash_{G_3c} \phi, \phi \wedge \psi, \Theta$ and $\Delta, \chi \vdash_{G_3c} \psi, \phi \wedge \psi, \Theta$. We now apply $\wedge r$ to obtain

$$\frac{\Delta, \chi \vdash_{G_3c} \phi, \phi \wedge \psi, \Theta \quad \Delta, \chi \vdash_{G_3c} \psi, \phi \wedge \psi, \Theta}{\Delta, \chi \vdash_{G_3c} \phi \wedge \psi, \Theta} \wedge r$$

and we are done. Finally, we assume that the contracted formulæ is ‘used up’ by the rule. Here the rule itself automatically does the contraction in the succedent; the contraction in the antecedent still needs to be dealt with by induction. This completes the proof that contraction is admissible. We now turn to weakening, which is similar. We just show the case for $\supset r$. We have the rule

$$\frac{\Delta, \phi \vdash_{G_{3i}} \psi}{\Delta \vdash_{G_{3i}} \phi \supset \psi} \supset r$$

and we apply the induction hypothesis to obtain $\Delta, \chi, \phi \vdash_{G_{3i}} \tau, \psi$. We apply $\supset r$ to obtain

$$\frac{\Delta, \chi, \phi \vdash_{G_{3i}} \tau, \psi}{\Delta, \chi \vdash_{G_{3i}} \tau, \phi \supset \psi} \supset r$$

which finishes the case. ■

Having shown that G_{3c} is a suitable presentation of classical logic, we turn to the rules for valid proof expressions. It is important to stress here that we have to switch to realizations to be able to do this. The proof-objects for intuitionistic logic and other single-conclusioned systems are directly associated to the formula being proven. This does not hold here since we have multiple formulæ in the succedent. We thus follow Pfenning and give the following collection of rules for valid proof expressions of G_{3c} . We add a set X of syntactic variables used in each formulæ used in the realizer and label each formula with a proof variable y_i .

$$\frac{}{\vdash_{G_{3c}} X\Delta, y : \mathbf{ant}(\phi) \xrightarrow{HYP(y,z)} z : \mathbf{suc}(\phi), \Theta} v-Ax$$

$$\frac{}{\vdash_{G_{3c}} \Delta \xrightarrow{TOP(z)} z : \mathbf{suc}(\top), \Theta} v-TOP$$

$$\frac{\vdash_{G_{3c}} X\Delta, y : \mathbf{ant}(\phi \wedge \psi), y_1 : \mathbf{ant}(\phi), y_2 : \mathbf{ant}(\psi) \xrightarrow{\delta} \Theta}{\vdash_{G_{3c}} X\Delta, y : \mathbf{ant}(\phi \wedge \psi) \xrightarrow{AND-L((y_1, y_2) : \delta, y)} \Theta} v-AND-L$$

$$\frac{\vdash_{G_{3c}} X\Delta \xrightarrow{\delta_1} \Theta_1 \quad \vdash_{G_{3c}} X\Delta \xrightarrow{\delta_2} \Theta_2}{\vdash_{G_{3c}} X\Delta \xrightarrow{AND-R(z_1 : \delta_1, z_2 : \delta_2, z)} \Theta} v-AND-R$$

where $\Theta_1 = z_1 : \mathbf{suc}(\phi), z : \mathbf{suc}(\phi \wedge \psi), \Theta$ and $\Theta_2 = z_2 : \mathbf{suc}(\psi), z : \mathbf{suc}(\phi \wedge \psi), \Theta$

$$\frac{\vdash_{G_{3c}} X\Delta_1 \xrightarrow{\delta_1} \Theta \quad \vdash_{G_{3c}} X\Delta_2 \xrightarrow{\delta_2} \Theta}{\vdash_{G_{3c}} X\Delta, y : \mathbf{ant}(\phi \vee \psi) \xrightarrow{OR-L(y_1 : \delta_1, y_2 : \delta_2, y)} \Theta} v-OR-L$$

where $\Delta_1 = \Delta, y : \text{ant}(\phi \vee \psi), y_1 : \text{ant}(\phi)$ and $\Delta_2 = \Delta, y : \text{ant}(\phi \vee \psi), y_2 : \text{ant}(\psi)$

$$\frac{\vdash_{G_{3c}} X\Delta \xrightarrow{\delta} z_1 : \text{suc}(\phi), z : \text{suc}(\phi \vee \psi)}{\vdash_{G_{3c}} X\Delta \xrightarrow{OR_i-L(z_1 : \delta, z)} z : \text{suc}(\phi \vee \psi)} v-OR-R_i$$

$$\frac{\vdash_{G_{3c}} X\Delta_1 \xrightarrow{\delta_1} z_1 : \text{suc}(\phi), \Theta \quad \vdash_{G_{3c}} X\Delta_2 \xrightarrow{\delta_2} \Theta}{\vdash_{G_{3c}} X\Delta, y : \text{ant}(\phi \supset \psi) \xrightarrow{IMP-L(z_1 : \delta_1, z_2 : \delta_2, y)} \Theta} v-IMP-L$$

where $\Delta_1 = \Delta, y : \text{ant}(\phi \supset \psi)$ and $\Delta_2 = \Delta, y : \text{ant}(\phi \supset \psi), y_1 : \text{ant}(\psi)$

$$\frac{\vdash_{G_{3c}} X\Delta, y_1 : \text{ant}(\phi) \xrightarrow{\delta} z_1 : \text{suc}(\psi), z : \text{suc}(\phi \supset \psi)}{\vdash_{G_{3c}} X\Delta \xrightarrow{IMP-R((y_1, y_2) : \delta, z)} z : \text{proof}(\phi \supset \psi)} v-IMP-R$$

$$\frac{\vdash_{G_{3c}} X\Delta, y : \text{ant}(\neg\phi) \xrightarrow{\delta} z_1 : \text{suc}(\phi), \Theta}{\vdash_{G_{3c}} X\Delta, y : \text{ant}(\neg\phi) \xrightarrow{NEG-R(z_1 : \delta, y)} \Theta} v-NEG-L$$

$$\frac{\vdash_{G_{3c}} X\Delta, y_1 : \text{ant}(\phi) \xrightarrow{\delta} z : \text{suc}(\neg\phi), \Theta}{\vdash_{G_{3c}} X\Delta \xrightarrow{NEG-R(y_1 : \delta, z)} z : \text{suc}(\neg\phi), \Theta} v-NEG-R$$

$$\frac{\vdash_{G_{3c}} X\Delta, y : \text{ant}(\forall x . \phi), y_1 : \text{ant}(\phi) \xrightarrow{\delta} \Theta}{\vdash_{G_{3c}} X\Delta, y : \text{ant}(\forall x . \phi) \xrightarrow{FORALL-L(y_1 : \delta, y)} \Theta} v-FORALL-L$$

$$\frac{\vdash_{G_{3c}} X, x\Delta \xrightarrow{\delta} z_1 : \text{ant}(\phi), z : \text{ant}(\forall x . \phi), \Theta}{\vdash_{G_{3c}} X\Delta \xrightarrow{FORALL-L(z_1 : \delta, z)} z : \text{ant}(\forall x . \phi), \Theta} v-FORALL-R$$

$$\frac{\vdash_{G_{3c}} (X, x) \Delta, y_1 : \text{ant}(\phi), y : \text{ant}(\exists x . \phi) \xrightarrow{\delta} \Theta}{\vdash_{G_{3c}} X\Delta, y : \text{ant}(\exists x . \phi) \xrightarrow{EXISTS-L(y_1 : \delta, y)} \Theta} v-EXISTS-L$$

$$\frac{\vdash_{G_{3c}} X\Delta \xrightarrow{\delta} z_1 : \text{suc}(\phi), z : \text{suc}(\exists x . \phi), \Theta}{\vdash_{G_{3c}} X\Delta \xrightarrow{EXISTS-R(z_1 : \delta, z)} z : \text{suc}(\exists x . \phi), \Theta} v-EXISTS-R$$

We now have to represent these valid proof expressions in LF. We do so by taking both the antecedent and succedents as contexts in $\lambda\Pi$ and introducing a new type $\#$ which is intended to be the type of every valid proof term (*cf.*, Pfenning). One can think of $\#$ as the empty type or contradiction if one were to carry out classical resolution (Robinson 1965) in this setting. We thus encode a realization

$$\vdash_{G_{3c}} (X) \Delta \xrightarrow{\delta} \Theta$$

as the assertion

$$\Gamma_X, \Gamma_\Delta, \Gamma_\Theta \vdash_{\Sigma_{G_3c}} M_\delta : \#$$

in LF. One might expect the context Γ_Θ to contain the negation of all the formulæ in Θ ; however, we have not negated any formulæ. The negation is hidden by the use of judgements. A careful examination of the inference rules shows that $\neg r$ and $\neg l$ classify a relationship between the judgements: the formulæ judged by **ant** are the negation of formulæ judged by **suc** and vice-versa. The $\lambda\Pi$ -signature for G_3c can be found in AppendixC. To help illustrate how $\#$ is used in LF, we take two classical proofs and give their representation in LF.

Example 12.6 (Tertium non Datur)

We have the following proof of the law of excluded middle in G_3c :

$$\frac{\frac{\frac{\text{Ax}}{\vdash_{G_3c} X \ y_1 : \mathbf{ant}(\phi) \xrightarrow{HYP(y_1, z_1)} z_1 : \mathbf{suc}(\phi), z_2 : \mathbf{suc}(\neg\phi), z_3 : \mathbf{suc}((\neg\phi) \vee \phi)}}{X \ \langle \rangle \xrightarrow{NEG-R(y_1 : HYP(y_1, z_1) z_2)} z_2 : \mathbf{suc}(\neg\phi), z_1 : \mathbf{suc}(\phi), z_3 : \mathbf{suc}((\neg\phi) \vee \phi)}}{X \ \langle \rangle \xrightarrow{OR-R((z_2, z_1) : NEG-R(y_1 : HYP(y_1, z_1), z_2), z_3)} z_3 : \mathbf{suc}((\neg\phi) \vee \phi)}}}{}$$

where we have used a different, but equivalent, rule for v - OR - R , which is encoded in LF as the following assertion:

$$\Gamma_X, z_3 : \mathbf{suc}(\neg\phi \vee \phi) \vdash_{\Sigma_{G_3c}} OR-R((z_2, z_1) : NEG-R(y_1 : HYP(y_1, z_1), z_2), z_3) z_3) : \mathbf{suc}(\neg\phi \vee \phi) : \#$$

We now provide the derivation of this assertion. We write the derivation in stages due to its size and we omit the instantiation of formulæ for clarity.

$$\frac{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_3c}} HYP : \mathbf{ant}(\phi) \rightarrow \mathbf{suc}(\phi) \rightarrow \# \quad \Gamma_X, \Gamma_1, y_1 : \mathbf{ant}(\phi) \vdash_{\Sigma_{G_3c}} y_1 : \mathbf{ant}(\phi)}{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_3c}} HYP(y_1) : \mathbf{suc}(\phi) \rightarrow \#}$$

where $\Gamma_2 = y_1 : \mathbf{ant}(\phi), z_1 : \mathbf{suc}(\phi), z_2 : \mathbf{suc}(\neg\phi), z_3 : \mathbf{suc}(\neg\phi \vee \phi)$ and $\Gamma_1 = z_1 : \mathbf{suc}(\phi), z_2 : \mathbf{suc}(\neg\phi), z_3 : \mathbf{suc}(\neg\phi \vee \phi)$.

$$\frac{\frac{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_3c}} HYP(y_1) : \mathbf{suc}(\phi) \rightarrow \# \quad \Gamma_X, \Gamma_3, z_1 : \mathbf{suc}(\phi) \vdash_{\Sigma_{G_3c}} z_1 : \mathbf{suc}(\phi)}{\Gamma_X, \Gamma_1, y_1 : \mathbf{ant}(\phi) \vdash_{\Sigma_{G_3c}} HYP(y_1) z_1 : \#}}{\Gamma_X, \Gamma_1 \vdash_{\Sigma_{G_3c}} \lambda y_1 : \mathbf{ant}(\phi) . HYP(y_1) z_1 : \mathbf{ant}(\phi) \rightarrow \#}$$

where $\Gamma_3 = y_1 : \text{ant}(\phi), z_2 : \text{suc}(\neg\phi), z_3 : \text{suc}(\neg\phi \vee \phi)$.

$$\frac{\Gamma_X, \Gamma_1 \vdash_{\Sigma_{G_3c}} \text{NEG-R} : (\text{ant}(\phi) \rightarrow \#) \rightarrow (\text{suc}(\neg\phi) \rightarrow \#) \quad \Gamma_X, \Gamma_1 \vdash N : \text{ant}(\phi)}{\Gamma_X, \Gamma_1 \vdash_{\Sigma_{G_3c}} \text{NEG-R}(N) : \text{suc}(\neg\phi) \rightarrow \#}$$

where $N = \lambda y_1 : \text{ant}(\phi) . \text{HYP}(y_1)z_1$.

$$\frac{\Gamma_X, \Gamma_1 \vdash_{\Sigma_{G_3c}} \text{NEG-R} : \text{suc}(\neg\phi) \rightarrow \# \quad \Gamma_X, \Gamma_4, z_2 : \text{suc}(\neg\phi) \vdash_{\Sigma_{G_3c}} z_4 : \text{suc}(\neg\phi)}{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_3c}} \text{NEG-R}(N)z_2 : \#}$$

where $\Gamma_4 = z_1 : \text{suc}(\phi), z_3 : \text{suc}(\phi \vee \neg\phi)$.

$$\frac{\Gamma_X, \Gamma_5, z_1 : \text{suc}(\phi) \vdash_{\Sigma_{G_3c}} \text{NEG-R}(N)z_2 : \#}{\Gamma_X, \Gamma_5 \vdash_{\Sigma_{G_3c}} \lambda z_1 : \text{suc}(\phi) . \text{NEG-R}(N)z_2 : \text{suc}(\phi) \rightarrow \#}$$

where $\Gamma_5 = z_3 : \text{suc}(\neg\phi \vee \phi), z_2 : \text{suc}(\neg\phi)$.

$$\frac{\Gamma_X, z_3 : \text{suc}(\neg\phi \vee \phi), z_2 : \text{suc}(\neg\phi) \vdash_{\Sigma_{G_3c}} M : \text{suc}(\phi) \rightarrow \#}{\Gamma_X, z_3 : \text{suc}(\neg\phi \vee \phi) \vdash_{\Sigma_{G_3c}} \lambda z_2 : \text{suc}(\phi) . M : \text{suc}(\neg\phi) \rightarrow \text{suc}(\phi) \rightarrow \#}$$

where $M = \lambda z_1 : \text{suc}(\phi) . \text{NEG-R}(N)z_2$.

$$\frac{\Gamma_X, z_3 : \text{suc}(\neg\phi \vee \phi) \vdash \text{OR-R} \quad \Gamma_X, z_3 : \text{suc}(\neg\phi \vee \phi) \vdash P : \text{suc}(\neg\phi) \rightarrow \text{suc}(\phi) \rightarrow \#}{\Gamma_X, z_3 : \text{suc}(\neg\phi \vee \phi) \vdash_{\Sigma_{G_3c}} \text{OR-R}(P) : \text{suc}(\neg\phi \vee \phi) \rightarrow \#}$$

where $P = \lambda z_2 : \text{suc}(\phi) . M$.

$$\frac{\Gamma' \vdash_{\Sigma_{G_3c}} \text{OR-R}(P) : \text{suc}(\neg\phi \vee \phi) \rightarrow \# \quad \Gamma' \vdash_{\Sigma_{G_3c}} z_3 : \text{suc}(\neg\phi \vee \phi)}{\Gamma_X, z_3 : \text{suc}(\neg\phi \vee \phi) \vdash_{\Sigma_{G_3c}} \text{OR-R}(P)z_3 : \#}$$

where $\Gamma' = \Gamma_X, z_3 : \text{suc}(\neg\phi \vee \phi)$, which completes the derivation. ■

For our second example we turn to a proof of DeMorgan's law for quantification, again this is an example of a classical proof.

Example 12.7 (DeMorgan)

We have the following proof in G_{3c}

$$\begin{array}{c}
\frac{}{\neg(\forall x\phi), \phi \vdash_{G_{3c}} \phi, \neg\phi, \forall x\phi, \exists x(\neg\phi)} Ax \\
\frac{}{\neg(\forall x\phi) \vdash_{G_{3c}} \phi, \neg\phi, \forall x\phi, \exists x(\neg\phi)} \neg r \\
\frac{}{\neg(\forall x\phi) \vdash_{G_{3c}} \phi, \forall x\phi, \exists x(\neg\phi)} \exists r \\
\frac{}{\neg(\forall x\phi) \vdash_{G_{3c}} \forall x\phi, \exists x(\neg\phi)} \forall r \\
\frac{}{\neg(\forall x\phi) \vdash_{G_{3c}} \exists x(\neg\phi)} \neg l
\end{array}$$

which is represented in LF by the assertion

$$\Gamma_X, y_1 : \mathbf{ant}(\phi(\neg(\forall(\lambda x : \iota. \phi)))), z_4 : \mathbf{suc}(\exists(\lambda x : \iota. (\neg\phi))) \vdash_{\Sigma_{G_{3c}}}$$

$$NEG-L(\lambda z_3 : \mathbf{suc}(\forall(\lambda x : \iota. \phi))). FORALL-R(\lambda z_1 : \mathbf{suc}(\phi)).$$

$$EXISTS-R(\lambda z_2 : \mathbf{suc}(\neg\phi)). NEG-R(\lambda z_2 : \mathbf{ant}(\phi)). HYP(y_2)z_1)z_2)z_4)z_3)y_1) : \#$$

We show the derivation of this assertion in stages due to its size.

$$\frac{\Gamma_X, \Gamma_1 \vdash_{\Sigma_{G_{3c}}} HYP \quad \Gamma_X, \Gamma_2, y_2 : \mathbf{ant}(\phi) \vdash_{\Sigma_{G_{3c}}} y_2 : \mathbf{ant}(\phi)}{\Gamma_X, \Gamma_1 \vdash_{\Sigma_{G_{3c}}} HYP(y_2) : \mathbf{suc}(\phi) \rightarrow \#}$$

where $\Gamma_1 = y_1 : \mathbf{ant}(\neg(\forall(\lambda x : \iota. \phi))), y_2 : \mathbf{ant}(\phi), z_1 : \mathbf{suc}(\phi), z_2 : \mathbf{suc}(\neg\phi), z_3 : \mathbf{suc}(\forall(\lambda x : \iota. \phi)), z_4 : \mathbf{suc}(\exists(\lambda x : \iota. (\neg\phi)))$ and $\Gamma_2 = y_1 : \mathbf{ant}(\neg(\forall(\lambda x : \iota. \phi))), z_1 : \mathbf{suc}(\phi), z_2 : \mathbf{suc}(\neg\phi), z_3 : \mathbf{suc}(\forall(\lambda x : \iota. \phi)), z_4 : \mathbf{suc}(\exists(\lambda x : \iota. (\neg\phi)))$.

$$\frac{\Gamma_X, \Gamma_1 \vdash_{\Sigma_{G_{3c}}} HYP(y_2) : \mathbf{suc}(\phi) \rightarrow \# \quad \Gamma_X, \Gamma', z_1 : \mathbf{suc}(\phi) \vdash_{\Sigma_{G_{3c}}} z_1 : \mathbf{suc}(\phi)}{\Gamma_X, \Gamma_1 \vdash_{\Sigma_{G_{3c}}} HYP(y_1)z_1 : \#}$$

where $\Gamma' = y_1 : \mathbf{ant}(\neg(\forall(\lambda x : \iota. \phi))), y_2 : \mathbf{ant}(\phi), z_2 : \mathbf{suc}(\neg\phi), z_3 : \mathbf{suc}(\forall(\lambda x : \iota. \phi)), z_4 : \mathbf{suc}(\exists(\lambda x : \iota. (\neg\phi)))$.

$$\frac{\Gamma_X, \Gamma_2, y_2 : \mathbf{ant}(\phi) \vdash_{\Sigma_{G_{3c}}} HYP(y_1)z_1 : \#}{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_{3c}}} \lambda y_2 : \mathbf{ant}(\phi). HYP(y_1)z_1 : \mathbf{ant}(\phi) \rightarrow \#}$$

$$\frac{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_{3c}}} NEG \quad \Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_{3c}}} \lambda y_2 : \mathbf{ant}(\phi). HYP(y_1)z_1 : \mathbf{ant}(\phi) \rightarrow \#}{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_{3c}}} NEG-R(\lambda y_2 : \mathbf{ant}(\phi). HYP(y_1)z_1) : \mathbf{suc}(\neg\phi) \rightarrow \#}$$

$$\frac{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_{3c}}} M : \mathbf{suc}(\neg\phi) \rightarrow \# \quad \Gamma_X, \Gamma_3, z_2 : \mathbf{suc}(\neg\phi) \vdash_{\Sigma_{G_{3c}}} z_2 : \mathbf{suc}(\neg\phi)}{\Gamma_X, \Gamma_2 \vdash_{\Sigma_{G_{3c}}} Mz_2 : \#}$$

where $\Gamma_3 = y_1 : \mathbf{ant}(\neg(\forall(\lambda x : \iota. \phi))), z_1 : \mathbf{suc}(\phi), z_3 : \mathbf{suc}(\forall(\lambda x : \iota. \phi)), z_4 : \mathbf{suc}(\exists(\lambda x :$

$\iota.(\neg\phi))$ and $M = NEG-R(\lambda y_2:\text{ant}(\phi). (HYP(y_1)z_1))$.

$$\frac{\Gamma_X, \Gamma_3, z_2:\text{succ}(\neg\phi) \vdash_{\Sigma_{G_{3c}}} Mz_2:\#}{\Gamma_X, \Gamma_3 \vdash_{\Sigma_{G_{3c}}} \lambda z_2:\text{succ}(\neg\phi). (Mz_2):\text{succ}(\neg\phi) \rightarrow \#}$$

$$\frac{\Gamma_X, \Gamma_3 \vdash_{\Sigma_{G_{3c}}} EXISTS-R \quad \Gamma_X, \Gamma_3 \vdash_{\Sigma_{G_{3c}}} \lambda z_2:\text{succ}(\neg\phi). (Mz_2):\text{succ}(\neg\phi) \rightarrow \#}{\Gamma_X, \Gamma_3 \vdash_{\Sigma_{G_{3c}}} EXISTS-R(\lambda z_2:\text{succ}(\neg\phi). (Mz_2)):\text{succ}(\exists(\lambda x:\iota. \phi)) \rightarrow \#}$$

$$\frac{\Gamma_X, \Gamma_3 \vdash_{\Sigma_{G_{3c}}} EXISTS-R(\lambda z_2:\text{succ}(\neg\phi). (Mz_2)) \quad \Gamma_X, \Gamma_4, z_4:\text{succ}(\exists(\lambda x:\iota. (\neg\phi)))}{\Gamma_X, \Gamma_3 \vdash_{\Sigma_{G_{3c}}} EXISTS-R(\lambda z_2:\text{succ}(\neg\phi). (Mx_2))z_4:\#}$$

$$\frac{\Gamma_X, \Gamma_4, z_1:\text{succ}(\phi) \vdash_{\Sigma_{G_{3c}}} N:\#}{\Gamma_X, \Gamma_4 \vdash_{\Sigma_{G_{3c}}} \lambda z:\text{succ}(\phi). N:\text{succ}(\phi) \rightarrow \#}$$

where $\Gamma_4 = y_1:\text{ant}(\neg(\forall(\lambda x:\iota. \phi)))$, $z_3:\text{succ}(\forall(\lambda x:\iota. \phi))$, $z_4:\text{succ}(\exists(\lambda x:\iota. (\neg\phi)))$.

$$\frac{\Gamma_X, \Gamma_4 \vdash_{\Sigma_{G_{3c}}} FORALL-R \quad \Gamma_X, \Gamma_4 \vdash_{\Sigma_{G_{3c}}} \lambda z_2:\text{succ}(\phi). N:\text{succ}(\phi) \rightarrow \#}{\Gamma_X, \Gamma_4 \vdash_{\Sigma_{G_{3c}}} FORALL-R(\lambda z_2:\text{succ}(\phi). N):\text{succ}(\forall(\lambda x:\iota. \phi)) \rightarrow \#}$$

$$\frac{\Gamma_X, \Gamma_4 \vdash_{\Sigma_{G_{3c}}} P:\text{succ}(\forall(\lambda x:\iota. \phi)) \rightarrow \# \quad \Gamma_X, \Gamma_4 \vdash_{\Sigma_{G_{3c}}} z_3:\text{succ}(\forall(\lambda x:\iota. \phi))}{\Gamma_X, \Gamma_4 \vdash_{\Sigma_{G_{3c}}} Pz_3:\#}$$

where $P = FORALL-R(\lambda z_2:\text{succ}(\phi). N)$.

$$\frac{\Gamma_X, \Gamma_4 \vdash_{\Sigma_{G_{3c}}} Pz_3:\#}{\Gamma_X, \Gamma_5 \vdash_{\Sigma_{G_{3c}}} \lambda z_3:\text{succ}(\forall(\lambda x:\iota. \phi)). Pz_3:\text{succ}(\forall(\lambda x:\iota. \phi)) \rightarrow \#}$$

where $\Gamma_5 = y_1:\text{ant}(\neg(\forall(\lambda x:\iota. \phi)))$, $z_4:\text{succ}(\exists(\lambda x:\iota. (\neg\phi)))$.

$$\frac{\Gamma_X, \Gamma_5 \vdash_{\Sigma_{G_{3c}}} NEG-L \quad \Gamma_X, \Gamma_5 \vdash_{\Sigma_{G_{3c}}} Q:\text{succ}(\forall(\lambda x:\iota. \phi)) \rightarrow \#}{\Gamma_X, \Gamma_5 \vdash_{\Sigma_{G_{3c}}} NEG-LQ:\text{ant}(\neg(\forall(\lambda x:\iota. \phi))) \rightarrow \#}$$

where $Q = \lambda z_3:\text{succ}(\forall(\lambda x:\iota. \phi)). Pz_3$.

$$\frac{\Gamma_X, \Gamma_5 \vdash_{\Sigma_{G_{3c}}} NEG-LQ \quad \Gamma_X, \Gamma_5 \vdash_{\Sigma_{G_{3c}}} y_1:\text{ant}(\neg(\forall(\lambda x:\iota. \phi)))}{\Gamma_X, \Gamma_5 \vdash_{\Sigma_{G_{3c}}} NEG-LQy_1:\#} \quad \blacksquare$$

The encoding of G_{3c} in LF is adequate using the judgements-as-types representation mechanism.

Lemma 12.8 (Representing G_{3c} in LF)

The judged proof system G_{3c} can be adequately represented in LF.

Proof (Sketch) Proved using the standard method (*cf.* (Harper et al. 1993)) ■

12.3 Encoding Higher-Order Intuitionistic Logic in LF

We begin by defining a sequent presentation of higher-order logic. There are many presentations of higher-order logic and we base ours on the one given by Church (1940). We do not present the whole of Church's type theory, we just present the logical connectives and the calculus of λ -conversion. We require that each sequent must have the principal formula in each premiss to make it the higher-order extension of G_3i . The system described below, as far as we know, does not appear in the literature.

Definition 12.9 (Higher-Order Intuitionistic Logic)

The judged proof system G_3HOIL is given by the alphabet $A = (S, V, E, C, J)$ where

$$\begin{aligned} S &= \{\iota, o, \Rightarrow\} \\ V &= \{\iota, o, \Rightarrow\} \\ E &= \{\top, \wedge, \vee, \supset, \forall_\sigma, \exists_\sigma, \Lambda_{\sigma,\tau}, \mathbf{ap}_{\sigma,\tau}, =_\sigma\} \\ C &= \{\wedge, \vee, \supset, \forall_\sigma, \exists_\sigma\} \\ J &= \{\mathbf{proof}\} \end{aligned}$$

and ι and o have arity one, \Rightarrow has arity 2, \top has arity o , \wedge , \vee and \supset have arity $o \Rightarrow o \Rightarrow o$, \forall_σ and \exists_σ have arity $(\sigma \Rightarrow o) \Rightarrow o$, where σ is any syntactic category, $\Lambda_{\sigma,\tau}$ has arity $(\sigma \rightarrow \tau) \rightarrow (\sigma \Rightarrow \tau)$, $\mathbf{ap}_{\sigma,\tau}$ has arity $(\sigma \Rightarrow \tau) \rightarrow \sigma \rightarrow \tau$ where σ and τ are arbitrary syntactic categories, $=_\sigma$ has arity $\sigma \Rightarrow \sigma \Rightarrow o$ and **proof** has arity o . Together with the following rules:

$$\begin{aligned} & \frac{}{\Delta, \mathbf{proof}(\phi) \vdash_{G_3HOIL} \mathbf{proof}(\phi)} Ax \\ & \frac{}{\Delta \vdash_{G_3HOIL} \mathbf{proof}(\top)} \top \\ & \frac{\Delta \vdash_{G_3HOIL} \mathbf{proof}(\phi) \quad \Delta \vdash_{G_3HOIL} \mathbf{proof}(\psi)}{\Delta \vdash_{G_3HOIL} \mathbf{proof}(\phi \wedge \psi)} \wedge R \\ & \frac{\Delta, \mathbf{proof}(\phi \wedge \psi), \mathbf{proof}(\phi), \mathbf{proof}(\psi) \vdash_{G_3HOIL} \mathbf{proof}(\chi)}{\Delta, \mathbf{proof}(\phi \wedge \psi) \vdash_{G_3HOIL} \mathbf{proof}(\chi)} \wedge L \\ & \frac{\Delta \vdash_{G_3HOIL} \mathbf{proof}(\phi)}{\Delta \vdash_{G_3HOIL} \mathbf{proof}(\phi \vee \psi)} \vee R1 \\ & \frac{\Delta \vdash_{G_3HOIL} \mathbf{proof}(\psi)}{\Delta \vdash_{G_3HOIL} \mathbf{proof}(\phi \vee \psi)} \vee R2 \end{aligned}$$

$$\frac{\Delta', \text{proof}(\phi) \vdash_{G_3HOIL} \text{proof}(\chi) \quad \Delta', \text{proof}(\psi) \vdash_{G_3HOIL} \text{proof}(\chi)}{\Delta' \vdash_{G_3HOIL} \text{proof}(\chi)} \vee L$$

where $\Delta' = \Delta, \text{proof}(\phi \vee \psi)$.

$$\frac{\Delta, \text{proof}(\phi) \vdash_{G_3HOIL} \text{proof}(\psi)}{\Delta \vdash_{G_3HOIL} \text{proof}(\phi \supset \psi)} \supset R$$

$$\frac{\Delta' \vdash_{G_3HOIL} \text{proof}(\phi) \quad \Delta', \text{proof}(\psi) \vdash_{G_3HOIL} \text{proof}(\chi)}{\Delta', \text{proof}(\phi \supset \psi) \vdash_{G_3HOIL} \text{proof}(\chi)} \supset L$$

where $\Delta' = \Delta, \text{proof}(\phi \supset \psi)$.

$$\frac{\Delta \vdash_{G_3HOIL} \text{proof}(\phi)}{\Delta \vdash_{G_3HOIL} \text{proof}(\forall_\sigma \phi)} \forall R$$

$$\frac{\Delta, \text{proof}(\forall_\sigma \phi), \text{proof}(\phi x) \vdash_{G_3HOIL} \text{proof}(\chi)}{\Delta, \text{proof}(\forall_\sigma \phi) \vdash_{G_3HOIL} \text{proof}(\chi)} \forall L$$

$$\frac{\Delta \vdash_{G_3HOIL} \text{proof}(\phi)}{\Delta \vdash_{G_3HOIL} \text{proof}(\exists_\sigma \phi)} \exists R$$

$$\frac{\Delta, \text{proof}(\exists_\sigma \phi), \text{proof}(\phi) \vdash_{G_3HOIL} \text{proof}(\chi)}{\Delta, \text{proof}(\exists_\sigma \phi) \vdash_{G_3HOIL} \text{proof}(\chi)} \exists L$$

$$\frac{\Delta \vdash_{G_3HOIL} \text{proof}(\phi) \quad \Delta \vdash_{G_3HOIL} \text{proof}(\phi =_o \psi)}{\Delta \vdash_{G_3HOIL} \text{proof}(\psi)} EQ$$

$$\frac{\Delta \vdash_{G_3HOIL} \text{proof}(\phi =_\tau \psi)}{\Delta \vdash_{G_3HOIL} \text{proof}(\Lambda_{\sigma,\tau} x . \phi =_{\sigma \Rightarrow \tau} \Lambda_{\sigma,\tau} x . \psi)} LAM$$

$$(\Lambda_{\sigma,\tau} x . \phi) \psi =_\tau \phi[\psi](\beta)$$

$$\Lambda_{\sigma,\tau} x . (\phi) =_{\sigma \Rightarrow \tau} \phi(\eta)$$

where each antecedent is a multiset. ■

There are a few points worth noting about the above definition. The first is that we have a syntactic category \Rightarrow of arity 2. This syntactic category is intended to be the \rightarrow in Church's type theory. We then get as syntactic categories all the types of Church's type theory. The syntactic categories of the form $\sigma \Rightarrow \tau$, where σ and τ are syntactic categories, are called functional types and they type functional expressions. Note that all the logical connectives have an arity of level 0. This is because they are given the syntactic category corresponding to their type in Church's type theory. Only $\Lambda_{\sigma,\tau}$ and $\mathbf{ap}_{\sigma,\tau}$ have an arity higher than 0. This is to allow them to form and 'destroy' functional types. We have used $\Lambda_{\sigma,\tau}$

rather than the λ_σ used by Church to distinguish it from the arity abstraction and λ in $\lambda\Pi$. We use the Barendregt convention (Barendregt 1991): no variable occurs both free and bound; and, distinct binders use distinct variable names. This ensures that we do not explicitly have to deal with α -conversion.

As with the other logics, we now give valid proof rules. There is one important difference here. The valid proofs of G_3HOIL are defined with respect to a proof context $((\mathcal{A}), \Delta)$ rather than a proof context (X, Δ) . (\mathcal{A}) is an assignment governing the free variables of the proof. The valid proof rules are as follows:

$$\frac{}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} HYP(y) : \mathbf{proof}(\phi)} v\text{-}HYP$$

$$\frac{}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} TOP : \mathbf{proof}(\top)} v\text{-}TOP$$

$$\frac{(\mathcal{A}), \Delta \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\phi) \quad (\mathcal{A}), \Delta \vdash_{G_3HOIL} \delta_2 : \mathbf{proof}(\psi)}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} AND\text{-}R(\delta_1, \delta_2) : \mathbf{proof}(\phi \wedge \psi)} v\text{-}AND\text{-}R$$

$$\frac{(\mathcal{A}), \Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi) \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\chi)}{(\mathcal{A}), \Delta, y : \mathbf{proof}(\phi \wedge \psi) \vdash_{G_3HOIL} AND\text{-}L((y_1, y_2) : \delta_2, y) : \mathbf{proof}(\chi)} v\text{-}AND\text{-}L$$

$$\frac{(\mathcal{A}), \Delta \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\phi)}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} OR\text{-}R1(\delta_1) : \mathbf{proof}(\phi \vee \psi)} v\text{-}OR\text{-}R1$$

$$\frac{(\mathcal{A}), \Delta \vdash_{G_3HOIL} \delta_2 : \mathbf{proof}(\psi)}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} OR\text{-}R2(\delta_2) : \mathbf{proof}(\phi \vee \psi)} v\text{-}OR\text{-}R2$$

$$\frac{\mathcal{A}, \Delta', y_1 : \mathbf{proof}(\phi) \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\chi) \quad \mathcal{A}\Delta', y_2 : \mathbf{proof}(\psi) \vdash_{G_3HOIL} \delta_2 : \mathbf{proof}(\chi)}{\mathcal{A}\Delta' \vdash_{G_3HOIL} OR\text{-}L(y_1 : \delta_1, y_2 : \delta_2, y) : \mathbf{proof}(\chi)}$$

where $\Delta' = \Delta, y : \mathbf{proof}(\phi \vee \psi)$.

$$\frac{(\mathcal{A}), \Delta, y_1 : \mathbf{proof}(\phi) \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\psi)}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} IMP\text{-}R(y_1 : \delta_1) : \mathbf{proof}(\phi \supset \psi)} v\text{-}IMP\text{-}R$$

$$\frac{\mathcal{A}\Delta' \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\phi) \quad \mathcal{A}\Delta', y_1 : \mathbf{proof}(\psi) \vdash_{G_3HOIL} \delta_2 : \mathbf{proof}(\chi)}{\mathcal{A}\Delta' \vdash_{G_3HOIL} IMP\text{-}L(\delta_1, y_1 : \delta_2, y) : \mathbf{proof}(\chi)}$$

where $\Delta' = \Delta, y : \mathbf{proof}(\phi \supset \psi)$.

$$\frac{(\mathcal{A}, x : \sigma), \Delta \vdash_{G_3HOIL} \delta : \mathbf{proof}(\phi)}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} FORALL\text{-}R(\delta) : \mathbf{proof}(\forall_\sigma \phi)} v\text{-}FORALL\text{-}R$$

$$\frac{(\mathcal{A}), \Delta, y : \mathbf{proof}(\forall_\sigma \phi), y_1 : \mathbf{proof}(\phi) \vdash_{G_3HOIL} \delta \mathbf{proof}(\chi)}{(\mathcal{A}), \Delta, y : \mathbf{proof}(\forall_\sigma \phi) \vdash_{G_3HOIL} FORALL\text{-}L(y_1 : \delta, y) : \mathbf{proof}(\chi)} v\text{-}FORALL\text{-}L$$

$$\begin{array}{c}
\frac{(\mathcal{A}), \Delta \vdash_{G_3HOIL} \delta : \mathbf{proof}(\phi)}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} \mathbf{EXISTS-R}(\delta) : \mathbf{proof}(\exists_\sigma \phi)} v\text{-EXISTS-R} \\
\frac{(\mathcal{A}, x : \sigma), \Delta, y : \mathbf{proof}(\exists_\sigma \phi), y_1 : \mathbf{proof}(\phi) \vdash_{G_3HOIL} \delta : \mathbf{proof}(\chi)}{(\mathcal{A}), \Delta, y : \mathbf{proof}(\exists_\sigma \phi) \vdash_{G_3HOIL} \mathbf{EXISTS-L}(y_1 : \delta, y) : \mathbf{proof}(\chi)} v\text{-EXISTS-L} \\
\frac{(\mathcal{A}), \Delta \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\phi) \quad (\mathcal{A}), \Delta \vdash_{G_3HOIL} \delta_2 : \mathbf{proof}(\phi =_o \psi)}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} \mathbf{EQ}(\delta_1, \delta_2) : \mathbf{proof}(\psi)} v\text{-EQ} \\
\frac{(\mathcal{A}, x : \sigma), \Delta \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\phi =_\tau \psi)}{(\mathcal{A}), \Delta \vdash_{G_3HOIL} \mathbf{LAM}(\delta_1) \mathbf{proof}(\Lambda_\sigma x \phi =_{\sigma \Rightarrow \tau} \Lambda_\sigma x \psi)} v\text{-LAM}
\end{array}$$

Since this presentation is non-standard we prove that it is equivalent to the sequent presentation of higher-order logic in (Miller et al. 1991), which we shall call *HOIL*. A slight adaptation of Miller's system can be found in Appendix C.

Lemma 12.10

Let $\Gamma \vdash_{G_3HOIL} \mathbf{proof}(\phi)$ be a provable assertion in *G₃HOIL* and $\Gamma \vdash_{HOIL} \mathbf{proof}(\phi)$ be a provable assertion in *HOIL*. Then $\Gamma \vdash_{G_3HOIL} \mathbf{proof}(\phi)$ if and only if $\Gamma \vdash_{HOIL} \mathbf{proof}(\phi)$.

Proof We begin by showing the left to right direction. We show that each inference rule in *G₃HOIL* can be translated to a proof-tree in *HOIL* with the same premisses and conclusion. The right rules are the same in both systems so we only consider the left rules. We also ignore the rules for \top and Ax for the same reason. We list the rules of *G₃HOIL* on the left and the corresponding proof-tree in *HOIL* on the right. We drop the judgements and subscripts for size reasons.

$$\begin{array}{c}
\frac{\Delta, \phi \wedge \psi, \phi, \psi \vdash_{G_3} \chi}{\Delta, \phi \wedge \psi \vdash_{G_3} \chi} \wedge L \quad \frac{\Delta, \phi \wedge \psi, \phi, \psi \vdash \chi}{\phi \wedge \psi, \phi \wedge \psi \vdash \chi} \wedge L \\
\frac{\Delta, \phi \vee \psi, \phi \vdash_{G_3} \chi \quad \Delta, \phi \vee \psi, \psi \vdash_{G_3} \chi}{\Delta, \phi \vee \psi \vdash_{G_3} \chi} \vee L \quad \frac{\Delta, \phi \vee \psi, \phi \vdash \chi \quad \Delta, \phi \vee \psi, \psi \vdash \chi}{\Delta, \phi \vee \psi, \phi \vee \psi \vdash \chi} \vee \\
\frac{\Delta, \phi \supset \psi \vdash_{G_3} \phi \quad \Delta, \phi \supset \psi, \psi \vdash_{G_3} \chi}{\Delta, \phi \supset \psi \vdash_{G_3} \chi} \supset L \quad \frac{\Delta, \phi \supset \psi \vdash \phi \quad \Delta, \phi \supset \psi, \psi \vdash \chi}{\Delta, \phi \supset \psi, \phi \supset \psi \vdash \chi} \supset \\
\frac{\Delta, \forall_\sigma \phi, \phi \vdash_{G_3} \chi}{\Delta, \forall_\sigma \phi \vdash_{G_3} \chi} \forall L \quad \frac{\Delta, \forall_\sigma \phi, \phi \vdash \chi}{\Delta, \forall_\sigma \phi, \forall_\sigma \phi \vdash \chi} \forall L \\
\frac{\Delta, \forall_\sigma \phi \vdash_{G_3} \chi}{\Delta, \forall_\sigma \phi \vdash_{G_3} \chi} \forall L \quad \frac{\Delta, \forall_\sigma \phi, \forall_\sigma \phi \vdash \chi}{\Delta, \forall_\sigma \phi \vdash \chi} \forall L
\end{array}$$

$$\frac{\Delta, \exists_{\sigma}\phi, \phi \vdash_{G_3} \chi}{\Delta, \exists_{\sigma}\phi \vdash_{G_3} \chi} \exists L \quad \frac{\Delta, \exists_{\sigma}\phi, \phi \vdash_{G_3} \chi}{\Delta, \exists_{\sigma}\phi, \exists_{\sigma}\phi \vdash_{G_3} \chi} \exists L \quad \frac{\Delta, \exists_{\sigma}\phi, \exists_{\sigma}\phi \vdash_{G_3} \chi}{\Delta, \exists_{\sigma}\phi \vdash_{G_3} \chi} c$$

We have explicitly stated the use of contraction above. This is to make it clear that *HOIL* absorbs the additional formula in the antecedent.

We just have to show that our theory of λ -conversion corresponds to Miller's λ rule. The equality relation defined in our theory of λ -conversion equates terms that can be obtained by α , β and η -reduction. Replacing a term in a sequent by an equal term can be achieved in *HOIL* by using the λ rule where arbitrary terms in the succedent and antecedent are replaced by terms which are $\alpha\beta\eta$ -convertible. Thus the λ rule can be used to mimic the equality used in *G₃HOIL*.

We now have to show the right to left direction. We show that each rule in *HOIL* is equivalent to a proof-tree in *G₃HOIL'*, with the same premisses and conclusion. *G₃HOIL'* is *G₃HOIL* plus weakening and contraction. The right rules are the same and so we concentrate on the left rules. We also ignore the rules for \top and *Ax* for the same reason. We list the rules of *HOIL* on the left and the corresponding proof-tree in *G₃HOIL'* on the right.

$$\frac{\Delta, \phi, \psi \vdash \chi}{\Delta, \phi \wedge \psi \vdash \chi} \wedge L \quad \frac{\Delta, \phi, \psi \vdash \chi}{\Delta, \phi \wedge \psi, \phi, \psi \vdash \chi} w \quad \frac{\Delta, \phi \wedge \psi, \phi, \psi \vdash \chi}{\Delta, \phi \wedge \psi \vdash \chi} \wedge L$$

$$\frac{\Delta, \phi \vdash \chi \quad \Delta, \psi \vdash \chi}{\Delta, \phi \vee \psi \vdash \chi} \vee L \quad \frac{\Delta, \phi \vdash \chi}{\Delta, \phi \vee \psi, \phi \vdash \chi} w \quad \frac{\Delta, \psi \vdash \chi}{\Delta, \phi \vee \psi, \psi \vdash \chi} w \quad \frac{\Delta, \phi \vee \psi, \phi \vdash \chi \quad \Delta, \phi \vee \psi, \psi \vdash \chi}{\Delta, \phi \vee \psi \vdash \chi} \vee L$$

$$\frac{\Delta \vdash \phi \quad \Delta, \psi \vdash \chi}{\Delta, \phi \supset \psi \vdash \chi} \supset L \quad \frac{\Delta \vdash_{G'_3} \phi}{\Delta, \phi \supset \psi \vdash_{G'_3} \phi} w \quad \frac{\Delta, \psi \vdash_{G'_3} \chi}{\Delta, \phi \supset \psi, \psi \vdash_{G'_3} \chi} w \quad \frac{\Delta, \phi \supset \psi \vdash_{G'_3} \phi \quad \Delta, \phi \supset \psi, \psi \vdash_{G'_3} \chi}{\Delta, \phi \supset \psi \vdash_{G'_3} \chi} \supset L$$

$$\frac{\Delta, \phi \vdash \chi}{\Delta, \forall x\phi \vdash \chi} \forall L \quad \frac{\Delta, \phi \vdash_{G'_3} \chi}{\Delta, \forall x\phi, \phi \vdash_{G'_3} \chi} w \quad \frac{\Delta, \forall x\phi, \phi \vdash_{G'_3} \chi}{\Delta, \forall x\phi \vdash_{G'_3} \chi} \forall L$$

$$\frac{\Delta, \phi \vdash \chi}{\Delta, \exists x\phi \vdash \chi} \exists L \quad \frac{\Delta, \phi \vdash_{G'_3} \chi}{\Delta, \exists x\phi, \phi \vdash_{G'_3} \chi} w \quad \frac{\Delta, \exists x\phi, \phi \vdash_{G'_3} \chi}{\Delta, \exists x\phi \vdash_{G'_3} \chi} \exists L$$

$$\frac{\Delta \vdash \phi}{\Delta, \psi \vdash \phi} \textit{weakening} \quad \frac{\Delta \vdash_{G'_3} \phi}{\Delta, \psi \vdash_{G'_3} \phi} \textit{weakening}$$

$$\frac{\Delta, \phi, \phi \vdash \psi}{\Delta, \phi \vdash \psi} \text{contraction} \quad \frac{\Delta, \phi, \phi \vdash_{G'_3} \psi}{\Delta, \phi \vdash_{G'_3} \psi} \text{contraction}$$

We have explicitly shown the cases for weakening and contraction even though contraction is guaranteed in *HOIL* since it takes the antecedent to be a set.

The λ rule is captured by the equality relation of *G₃HOIL'*.

To complete the proof, we need to show that weakening and contraction are admissible in *G₃HOIL*. These are proven by induction over the depth of the proof-tree. We begin by proving contraction.

We start with the base case, *i.e.* we have a proof-tree of depth 1:

$$\frac{}{\Delta, \phi, \phi \vdash_{G_3HOIL} \chi}$$

where either χ is atomic or χ is \top . If χ is \top then we can use the rule \top ,

$$\frac{}{\Delta, \phi \vdash_{G_3HOIL} \top} \top$$

and we are done. If χ is atomic then either $\chi \in \Gamma$ or $\chi = \phi$. If $\chi = \phi$ then we also have the axiom

$$\frac{}{\Delta, \phi \vdash_{G_3HOIL} \phi} Ax$$

and we are done. If $\chi \in \Delta$ then we have the axiom

$$\frac{}{\Delta, \phi \vdash_{G_3HOIL} \chi} Ax$$

since χ is still in Δ and we complete the base case.

We prove $\wedge R$ and $\wedge L$ to illustrate how the induction step works. The other cases are similar.

We begin with $\wedge R$. We have the rule

$$\frac{\Delta, \chi, \chi \vdash_{G_3HOIL} \phi \quad \Delta, \chi, \chi \vdash_{G_3HOIL} \psi}{\Delta, \chi, \chi \vdash_{G_3HOIL} \phi \wedge \psi} \wedge R$$

and we can apply the induction hypothesis to the premisses. We obtain $\Delta, \chi \vdash_{G_3HOIL} \phi$ and $\Delta, \chi \vdash_{G_3HOIL} \psi$. We now apply $\wedge R$ to obtain

$$\frac{\Delta, \chi \vdash_{G_3HOIL} \phi \quad \Delta, \chi \vdash_{G_3HOIL} \psi}{\Delta, \chi \vdash_{G_3HOIL} \phi \wedge \psi} \wedge R$$

which completes this case.

We now look at $\wedge L$. There are three cases here, the first is when the formulæ being contracted are distinct from the principal formula. The second when the principal formula is one of the contracted formulæ and the third is when one of

the contracted formulæ is one of the formulæ ‘used up’ by the rule. We begin with the first case. We have the rule

$$\frac{\Delta, \chi, \chi, \phi \wedge \psi, \phi, \psi \vdash_{G_3HOIL} \tau}{\Delta, \chi, \chi, \phi \wedge \psi \vdash_{G_3HOIL} \tau} \wedge L$$

We now apply the induction hypothesis to the premiss to obtain $\Delta, \chi, \phi \wedge \psi, \phi, \psi \vdash_{G_3HOIL} \tau$. We apply $\wedge L$ to obtain

$$\frac{\Delta, \chi, \phi \wedge \psi, \phi, \psi \vdash_{G_3HOIL} \tau}{\Delta, \chi \phi \wedge \psi \vdash_{G_3HOIL} \tau} \wedge L$$

completing the case.

We now assume that we have the rule

$$\frac{\Delta, \phi \wedge \psi, \phi \wedge \psi, \phi, \psi \vdash_{G_3HOIL} \tau}{\Delta, \phi \wedge \psi, \phi \wedge \psi \vdash_{G_3HOIL} \tau} \wedge L$$

We apply the induction hypothesis to the premiss to obtain $\Delta, \phi \wedge \psi, \phi, \psi \vdash_{G_3HOIL} \tau$ and we can now apply $\wedge L$ to obtain

$$\frac{\Delta, \phi \wedge \psi, \phi, \psi \vdash_{G_3HOIL} \tau}{\Delta, \phi \wedge \psi \vdash_{G_3HOIL} \tau} \wedge L$$

which completes this case.

When the contracted formula is the formula ‘used up’ by the rule, the rule automatically does the contraction.

$$\frac{\Delta, \phi \wedge \psi, \phi, \phi \psi \vdash_{G_3HOIL} \tau}{\Delta, \phi \wedge \psi, \phi \vdash_{G_3HOIL} \tau} \wedge L$$

We now look at weakening. For the base case we have the the Ax rule and \top . If we have a proof-tree

$$\frac{}{\Delta \vdash_{G_3HOIL} \chi} Ax$$

then we also have a proof-tree

$$\frac{}{\Delta, \phi \vdash_{G_3HOIL} \chi} Ax$$

Similarly for \top , if we have a proof-tree

$$\frac{}{\Delta, \vdash_{G_3HOIL} \top} \top$$

then we also have

$$\frac{}{\Delta, \phi \vdash_{G_3HOIL} \top} \top$$

The induction steps are straightforward, we illustrate this with $\supset R$ and $\supset L$. We begin with $\supset R$. We have the rule

$$\frac{\Delta, \phi \vdash_{G_3HOIL} \psi}{\Delta \vdash_{G_3HOIL} \phi \supset \psi} \supset R$$

We apply the induction hypothesis to the premiss to obtain $\Delta, \chi, \phi \vdash_{G_3HOIL} \psi$. We can now use $\supset R$ to obtain

$$\frac{\Delta, \chi, \phi \vdash_{G_3HOIL} \psi}{\Delta, \chi \vdash_{G_3HOIL} \phi \supset \psi} \supset R$$

which finishes the case.

We now prove $\supset L$. We have the rule

$$\frac{\Delta, \phi \supset \psi \vdash_{G_3HOIL} \phi \quad \Delta, \phi \supset \psi, \psi \vdash_{G_3HOIL} \chi}{\Delta, \phi \supset \psi \vdash_{G_3HOIL} \chi} \supset L$$

We apply the induction hypothesis to the premiss to obtain $\Delta, \phi \supset \psi, \tau \vdash_{G_3HOIL} \phi$ and $\Delta, \phi \supset \psi, \psi, \tau \vdash_{G_3HOIL} \chi$. Applying $\supset L$ gives the following proof-tree

$$\frac{\Delta, \phi \supset \psi, \tau \vdash_{G_3HOIL} \phi \quad \Delta, \phi \supset \psi, \psi, \tau \vdash_{G_3HOIL} \chi}{\Delta, \phi \supset \psi, \tau \vdash_{G_3HOIL} \chi} \supset L$$

which completes this case.

Finally, since antecedents in G_3HOIL are multisets, we can permute formulæ in G_3HOIL whenever they are permuted in $HOIL$. ■

We now show how to encode G_3HOIL in LF. It is encoded using the same method used to encode G_3i . The signature Σ_{G_3HOIL} can be found in Appendix A. We have followed Harper et al. (1993) by using the ‘externalization’ of the equality constant as \approx . This is only done for notationally expediency.

Lemma 12.11

G_3HOIL can be adequately represented in LF.

Proof Similar to the proof in (Harper et al. 1993). ■

We can also encode higher-order classical logic. We just repeat the above analysis using the sequent rules for G_3c instead of those for G_3i and encode as per G_3c .

Chapter 13

Representing Abstract Logic Programming Languages in LF

Having shown how to represent sequent systems in a logical framework, we turn to uniform proofs. Due to time constraints, this chapter is an extended example. We take a definition of an abstract logic programming language in terms of uniform proof uniform proof and show that it is possible to classify the LF proof-terms which represent them. In this chapter we work with a system **NR** introduced in (Pym 1990), rather than a more general system for simplicity. **NR** is the calculus **N** defined in § 2.1 with the rules (2.16) and (2.22) replaced by the resolution rule

$$\frac{\Gamma \vdash_{\Sigma} M_1 : A_1 \dots \Gamma \vdash_{\Sigma} M_m : A_m \quad \Gamma \vdash_{\Sigma} N_1 : D_1 \dots \Gamma \vdash_{\Sigma} N_n : D_n}{\Gamma \vdash_{\Sigma} M_1 \dots M_m N_1 \dots N_n : E} \text{Res}$$

where

$$\textcircled{\@} : \Pi x_1 : E_1 \dots \Pi E_m : B_1 \rightarrow (B_2 \dots \rightarrow (B_n \rightarrow C) \dots) \in \Sigma \cup \Gamma,$$

$$A_i[M_1/x_1, \dots, M_{i-1}/x_{i-1}] =_{\beta\eta} E_i \text{ for } 1 \leq i \leq m,$$

$$B_i[M_1/x_1, \dots, M_m/x_m] =_{\beta\eta} D_i \text{ for } 1 \leq i \leq n,$$

$$\text{and } C[M_1/x_1, \dots, M_m/x_m] =_{\beta\eta} E.$$

We also require that $\Gamma, x : E \vdash_{\Sigma} x : E$, where $x \notin \text{Dom}(\Gamma)$ be a premise of the resolution rule, with this premise omitted for clarity of presentation.

This rule is syntax-directed and essentially deals with an application of a term $\textcircled{\@}$ in clausal form. The rule is syntax-directed in two senses; (i) the left rule is driven by a choice of clausal type, the principal formula of the rule, from the context (corresponding, in logic programming terms, to the choice of program clause, for which it is desirable that the rule builds in a contraction of the principal formula, thereby permitting the re-use of the clause); and, (ii) the rightmost

(atomic) type matches the succedent of the sequent. The following theorem shows the relationship between NR and N.

Theorem 13.1 (Soundness and Completeness (Pym 1995))

If NR proves $\Gamma \vdash_{\Sigma} M : A$ then N proves $\Gamma \vdash_{\Sigma} M : A$. Let M be in β -long normal form. If N proves $\Gamma \vdash_{\Sigma} M : A$ then NR proves $\Gamma \vdash_{\Sigma} M : A$. ■

This chapter forms the first step on the road to a general theory of computation based on logical frameworks. The results in this chapter are all original.

It is advantageous to be able to classify abstractly logics or fragments of logics which can be used as a suitable basis for logic programming languages. One example of such an abstract characterization is given in Miller et al. (1991). Here the abstract characterization is based on a notion of uniform provability. Miller *et al.*'s definition of a logic programming language says that any logic or fragment of a logic which is closed under uniform provability is an abstract logic programming language. Uniform provability characterizes a (well-motivated) set of desiderata that Miller *et al.* claim should be satisfied by any logic programming language. This set of desiderata essentially says that a logic programming language should implement the following search instructions:

- SUCCESS – search has been successful, finish;
- AND – search two separate paths for success;
- OR – search two separate paths but only one success required ;
- INSTANCE – search all possible paths to find a single success;
- AUGMENT – add a new clause and continue search; and,
- GENERIC – introduce a new parameter and attempt to find success for new goal.

The logical connectives \top , \wedge , \vee , \exists_{σ} , \supset and \forall_{σ} are then associated with each of these search instructions respectively. \top is intended to signify a successfully completed search; \wedge and \vee provide the specification of non-deterministic AND and OR nodes in the interpreter's search space; \exists_{σ} specifies an infinite non-deterministic OR branch; where the disjuncts are parameterized by the set of all terms; \supset tells the interpreter to augment its program; and, \forall_{σ} instructs the interpreter to introduce a new parameter and to try to prove the resulting generic instance of the goal. The reason this set of desiderata is a set of search instructions is because computation in logic programming involves goal-directed search, *cf.*, § 10. We now define the notion of a uniform proof.

Definition 13.2 (Uniform Proof (Miller et al. 1991))

Let \vdash_I be the consequence relation of LJ. A *uniform proof* in LJ is a proof $\Gamma \vdash_I G$ which satisfies the following conditions:

- if G is \top then that assertion is an axiom;
- if G is $\phi \wedge \psi$ then that assertion is inferred by $\wedge r$ from $\Gamma \vdash_I \phi$ and $\Gamma \vdash_I \psi$;
- if G is $\phi \vee \psi$ then that assertion is inferred by $\vee r$ from $\Gamma \vdash_I \phi$ or $\Gamma \vdash_I \psi$;
- if G is $\exists x\phi$ then that assertion is inferred by $\exists r$ from $\Gamma \vdash_I \phi[t/x]$;
- if G is $\phi \supset \psi$ then that assertion is inferred by $\supset r$ from $\Gamma, \phi \vdash_I \psi$;
- if G is $\forall x\phi$ then that assertion is inferred by $\phi[c/x]$, where c is a parameter that does not occur in the given assertion. ■

A uniform proof reflects the desiderata of Miller *et al.* We can think of a uniform proof as characterized by always preferring right rules over left rules. Uniform proofs are intuitionistic. We define the consequence relation \vdash_O , which characterizes uniform provability, as follows:

Definition 13.3 (Uniform Provability (Miller et al. 1991))

Let \mathcal{P} be a set of well-formed formulæ. We say that $\mathcal{P} \vdash_O G$ if and only if there is a uniform proof of G from \mathcal{P} . ■

Miller *et al.*'s definition is subtly different from ours. They take uniform proof to be an operational condition and thus define \vdash_O semantically. We have been more concrete and defined uniform proof in terms of a sequent proof system.

The intended meaning of $\mathcal{P} \vdash_O G$ is that the interpreter succeeds on the goal G given the program \mathcal{P} . We can now define an abstract logic programming language to be any logic or fragment of a logic closed under \vdash_O .

Definition 13.4 (Abstract Logic Programming Language (Miller et al. 1991))

Let \mathcal{G} and \mathcal{D} be well-formed sets of formulæ. An *abstract logic programming language* is a triple $\langle \mathcal{D}, \mathcal{G}, \vdash \rangle$ such that for all finite subsets \mathcal{P} of \mathcal{D} and all formulæ G of \mathcal{G} , $\mathcal{P} \vdash G$ if and only if $\mathcal{P} \vdash_O G$. ■

All the sequent systems we defined in § 12 all have fragments that are abstract logic programming languages. The proofs that these fragments are indeed abstract logic programming languages can be found in (Miller et al. 1991).

Example 13.5 (The Logic Programming Fragment of G_{3i})

Let A be an atomic formula of G_{3i} . Let \mathcal{G}_3 and \mathcal{D}_3 be the sets of all first-order G - and D -formulæ defined by the grammars:

$$G := \top \mid A \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid \forall xG \mid \exists xG \mid D \supset G$$

$$D := A \mid G \supset A \mid \forall xD \mid D_1 \wedge D_2$$

The formulæ in \mathcal{D}_3 are called (*first-order*) *hereditary Harrop formulæ*. The triple $\langle \mathcal{D}_3, \mathcal{G}_3, \vdash_{G_{3i}} \rangle$, which we call fohh, is an abstract logic programming language. ■

Example 13.6 (The Logic Programming Fragment of G_3HOIL)

Let \mathcal{H}_2 be the set of expressions of G_3HOIL that are in β -normal form that do not contain the logical connective \supset . An atomic formula A in \mathcal{H}_2 is said to be *rigid* if A is of the form $Pt_1 \dots t_n$, where P is a non-logical constant. Let A and A_r be atomic and rigid formulæ in \mathcal{H}_2 respectively. Let \mathcal{G}_4 and \mathcal{D}_4 be the sets of G - and D -formulæ that are defined by the grammars:

$$G := \top \mid A \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid \forall_\sigma G \mid \exists_\sigma G \mid D \supset G$$

$$D := A_r \mid G \supset A_r \mid \forall_\sigma D \mid D_1 \wedge D_2$$

The formulæ in \mathcal{D}_4 are called *higher-order hereditary Harrop formulæ*. The triple $\langle \mathcal{D}_4, \mathcal{G}_4, \vdash_{G_3HOIL} \rangle$, which we call hohh, is an abstract logic programming language. ■

Example 13.7 (The Logic Programming Fragment of G_{3c})

Let A be an atomic formula and \mathcal{G}_1 and \mathcal{D}_1 be the sets of all first-order G - and D -formulæ defined by the grammars:

$$G := \top \mid A \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid \exists x G$$

$$D := A \mid G \supset A \mid D_1 \wedge D_2 \mid \forall x D$$

The formulæ contained in \mathcal{D} are the (*first-order*) *Horn clauses*. The triple $\langle \mathcal{D}_1, \mathcal{G}_1, \vdash_{G_{3c}} \rangle$, which we call fohc, is an abstract logic programming language. ■

Example 13.8 (The logic programming fragment of G_{3HOCL})

Let \mathcal{H}_1 be the set of expressions of G_{3HOCL} that are in β -normal form that do not contain the logical connectives \supset and \forall . Let A and A_r be atomic and atomic rigid formulæ in \mathcal{H}_1 respectively. Let \mathcal{G}_2 and \mathcal{D}_2 be the sets of all higher-order G - and D -formulæ defined by the grammars:

$$G := \top \mid A \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid \exists_\sigma G$$

$$D := A_r \mid G \supset A_r \mid D_1 \wedge D_2 \mid \forall_\sigma D$$

The formulæ in \mathcal{D}_2 are called *higher-order Horn clauses*. The triple $\langle \mathcal{D}_2, \mathcal{G}_2, \vdash_{G_{3HOCL}} \rangle$, which we call hohc, is an abstract logic programming language. ■

We did not define G_{3HOCL} previously but it can be obtained by replacing all the single-conclusioned rules of G_3HOIL with the appropriate multi-conclusioned rules in an analogous fashion to the way that G_{3c} is obtained from G_3i .

We now look at the representations of G_{3i} and G_3HOIL in LF and explore how LF represents uniform proof. We do not consider G_{3c} and G_{3HOCL} for the moment because they required a different method of encoding and have to be

treated separately. We begin by showing how these systems are represented in the logical framework obtained when we take the language to be NR instead of the $\lambda\Pi$ -calculus. We also call this logical framework LF because the proof-terms are the same as the logical framework whose language is the $\lambda\Pi$ -calculus. Although, strictly speaking, it is a different logical framework.

13.1 Representing Uniform Proofs in G_3i in LF

We have already shown that G_3i can be adequately represented in LF in § 12.1. We begin by analysing how the constants which represent valid proof rules in G_3i are handled by the resolution rule of NR. We observe that all the constants representing valid proof rules in G_3i are in clausal form.

We begin by looking at the resolution rule when the clause @ is taken to be the constant

$$\begin{aligned} AND-L: \Pi p: o. \Pi q: o. \Pi r: o. (\text{proof}(p) \rightarrow \text{proof}(q) \rightarrow \text{proof}(r)) \rightarrow (\text{proof}(p \wedge q) \\ \rightarrow \text{proof}(r)) \end{aligned}$$

We omit the assertions $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \phi: o$, $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \psi: o$ and $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \chi: o$ and the judgements to keep the size of the derivations down.

$$\frac{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \lambda y_1: \phi. \lambda y_2: \psi. M_\delta: \phi \rightarrow \psi \rightarrow \chi \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} y: \phi \wedge \psi}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} AND-L(\lambda y_1: \phi. \lambda y_2: \psi. M_\delta)y: \chi} Res$$

where $y: \phi \wedge \psi \in \Gamma_\Delta$. We observe that we have the generalized elimination rule for \wedge given by Prawitz (1978).

This observation holds for the other left rule of G_3i , as we shall see. We claim further that this observation is more general and that the resolution rule usually encodes left and right rules of a logic as generalized elimination and introduction rules respectively.

We now look at the case where @ is taken to be the constant

$$\begin{aligned} OR-L: \Pi p, q, r: o. (\text{proof}(p) \rightarrow \text{proof}(r) \rightarrow \text{proof}(q) \rightarrow \text{proof}(r)) \\ \rightarrow (\text{proof}(p \vee q) \rightarrow \text{proof}(r)) \end{aligned}$$

This is represented by resolution as

$$\frac{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma} \lambda y_1. M_{\delta_1}: \phi \rightarrow \chi \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma} \lambda y_2. M_{\delta_2}: \psi \rightarrow \chi \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma} y: \phi \vee \psi}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} OR-L(\lambda y_2: \phi. M_{\delta_1})(\lambda y_2: \psi. M_{\delta_2})y: \chi}$$

where $y: \phi \vee \psi \in \Delta$, which is in the form of a generalized elimination rule.

We look at the resolution rule when the clause @ is taken to be the constant

$$\begin{aligned} \text{IMP-L} : \Pi p : o . \Pi q : o . \Pi r : o . \text{proof}(p) \rightarrow (\text{proof}(q) \rightarrow \text{proof}(r)) \rightarrow \\ (\text{proof}(p \supset q) \rightarrow \text{proof}(r)). \end{aligned}$$

We have

$$\frac{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} M_{\delta_1} : \phi \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} \lambda y_1 : \psi . M_{\delta_2} : \psi \rightarrow \chi \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} y : \phi \supset \psi}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} \text{IMP-L}(M_{\delta_1})(\lambda y_1 : \phi . M_{\delta_2})y : \chi}$$

where $y : \text{proof}(\phi \supset \psi) \in \Gamma_\Delta$, which is in the same form as a generalized elimination rule.

We now take the clause @ to be the constant

$$\begin{aligned} \text{FORALL-L} : \Pi F : \iota \rightarrow o . \Pi r : o . \Pi x : \iota . (\text{proof}(Fx) \rightarrow \text{proof}(r)) \rightarrow (\text{proof}(\forall \\ (\lambda x : \iota . Fx)) \rightarrow \text{proof}(r)) \end{aligned}$$

We have

$$\frac{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} \lambda y_1 : \phi x . M_\delta : \phi x \rightarrow \chi \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} y : \forall(\lambda x : \iota . \phi x)}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} \text{FORALL-L}(\lambda y_1 : \phi x . M_\delta)y : \chi} \text{Res}$$

where $y : \text{proof}(\forall(\lambda x : \iota . \phi x)) \in \Gamma_\Delta$, which is in the form of a generalized elimination rule.

Finally, we take the clause @ to be the constant

$$\begin{aligned} \text{EXISTS-L} : \Pi F : \iota \rightarrow o . \Pi r : o . (\Pi x : \iota . \text{proof}(Fx) \rightarrow \text{proof}(r)) \\ \rightarrow (\text{proof}(\exists(\lambda x : \iota . Fx)) \rightarrow \text{proof}(r)) \end{aligned}$$

which is represented as

$$\frac{\Gamma_X, \Gamma \vdash_{\Sigma_{G_3^i}} \lambda y_1 : \chi . \lambda x : \iota . M_\delta : \phi x \rightarrow \chi \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} y : \exists(\lambda x : \iota . \phi x)}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3^i}} \text{EXISTS-R}(\lambda y_1 : \chi . \lambda x : \iota . M_\delta)y : \chi} \text{Res}$$

where $y : \exists(\lambda x : \iota . \phi x) \in \Gamma_\Delta$, which is in the form of a generalized elimination rule.

We now show how the constants representing the right rules are handled by the resolution rule. We begin with the constant

$$\text{AND-R} : \Pi p : o . \Pi q : o . \text{proof}(p) \rightarrow \text{proof}(q) \rightarrow \text{proof}(p \wedge q)$$

We have

$$\frac{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} M_{\delta_1} : \mathbf{proof}(\phi) \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} M_{\delta_2} : \mathbf{proof}(\psi)}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \mathbf{AND-R}(M_{\delta_1})(M_{\delta_2}) : \mathbf{proof}(\phi \wedge \psi)} \text{Res}$$

which is just an introduction rule. In fact all the right rules encode as introduction rules.

Let @ be the constant

$$\mathbf{OR-R1} : \Pi p : o . \Pi q : o . \mathbf{proof}(p) \rightarrow \mathbf{proof}(p \vee q)$$

We have

$$\frac{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} M_\delta : \mathbf{proof}(\phi)}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \mathbf{OR-R}(M_\delta) : \mathbf{proof}(\phi \vee \psi)} \text{Res}$$

with the rule *OR-R2* being similar.

We take the clause @ to be the constant

$$\mathbf{IMP-R} : \Pi p : o . \Pi q : o . (\mathbf{proof}(p) \rightarrow \mathbf{proof}(q)) \rightarrow \mathbf{proof}(p \supset q)$$

$$\frac{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \lambda y_1 : \mathbf{proof}(\phi) . M_\delta : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi)}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \mathbf{IMP-R}(\lambda y_1 : \mathbf{proof}(\phi) . M_\delta) : \mathbf{proof}(\phi \supset \psi)} \text{Res}$$

which is not quite the introduction rule, but we observe that the derivation $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \lambda y_1 : \mathbf{proof}(\phi) . M_\delta : (\mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi))$ is obtained from $\Gamma_X, \Gamma_\Delta, y_1 : \mathbf{proof}(\phi) \vdash_{\Sigma_{G_3i}} M_\delta : \mathbf{proof}(\psi)$ and this is the premiss of an introduction rule.

Finally we take @ to be the constant

$$\mathbf{FORALL-R} : \Pi F : \iota \rightarrow o . (\Pi x : \iota . \mathbf{proof}(Fx)) \rightarrow \mathbf{proof}(\forall(\lambda x : \iota . Fx))$$

We have

$$\frac{\Gamma_X \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \lambda x : \iota . M_\delta : \Pi x : \iota . \mathbf{proof}(\phi x)}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} \mathbf{FORALL-R}(\lambda x : \iota . M_\delta) : \mathbf{proof}(\lambda x : \iota . \phi x)} \text{Res}$$

which is the introduction rule for \forall .

We now look at the representation of the uniform proofs of G_3i in **NR**. The first observation is that all proofs in G_3i are represented by terms of **NR** in long $\beta\eta$ -normal form. We thus require an extra condition on the terms of **NR** in long $\beta\eta$ -normal form such that if terms satisfy this condition they represent uniform proofs in G_3i . The second observation is that each term produced by resolution is of the form $P_1 \dots P_n$, where P_1 is a constant corresponding to a valid proof rule of G_3i . The type of this term corresponds to the judged formula in

- N never contains $NEG-L$;
- if N has type $\mathbf{proof}(\top)$ then $N = TOP$;
- if N has type $\mathbf{proof}(\phi \wedge \psi)$ then $N = P_1 \dots P_n$ where $P_1 = AND-R$;
- if N has type $\mathbf{proof}(\phi \vee \psi)$ then $N = P_1 \dots P_n$ where $P_1 = OR-R$;
- if N has type $\mathbf{proof}(\phi \supset \psi)$ then $N = P_1 \dots P_n$ where $P_1 = IMP-R$;
- if N has type $\mathbf{proof}(\forall x\phi)$ then $N = P_1 \dots P_n$ where $P_1 = FORALL-R$;
- if N has type $\mathbf{proof}(\exists x\phi)$ then $N = P_1 \dots P_n$ where $P_1 = EXISTS-R$. ■

The first example satisfies this definition while the other fails it. The first two conditions of the definition just ensure that we do not have \neg involved in the proof. We now provide a proof that this definition defines $\lambda\Pi$ -proof terms which represent uniform proofs in G_3i .

Lemma 13.12 (Uniform Proof-terms Represent Uniform Proofs)

Let $X\Delta \vdash_{G_3i} \delta : \mathbf{proof}(\phi)$ be provable in G_3i and let $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_3i}} M_\delta : \mathbf{proof}(\phi)$ be its representation in LF . δ is in long $\beta\eta$ normal form if and only if M_δ is in long $\beta\eta$ normal form and a uniform proof-term.

Proof Since the representation of any valid assertion in G_3i in \mathbf{NR} is adequate, we know that M_δ is always in long $\beta\eta$ normal form. We prove the left to right direction of the implication first. Let δ be in long $\beta\eta$ normal form.

Since δ is in long $\beta\eta$ -normal form, any left rules used in δ must have been applied to a sequent with an atomic formula in the succedent. This means that all subterms of M_δ of the form $\#-LP_1 \dots P_n$ have type $\mathbf{proof}(\chi)$, where χ is atomic. This is enough to ensure that M_δ is a uniform proof-term.

Suppose that M_δ is a uniform proof-term, then any subterm N of the form $\#-L$ must have a type $\mathbf{proof}(\chi)$ where χ is an atomic formula. Hence any left rule in G_3i must have been applied when the succedent was atomic. Thus δ is in long $\beta\eta$ -normal form. ■

Having found the terms in \mathbf{NR} which represent uniform proofs in G_3i , we turn to the representation of hereditary Harrop formulæ in \mathbf{NR} . We are interested in the hereditary Harrop formulæ because every hereditary Harrop sequent has a uniform proof in G_3i , cf. (Miller et al. 1991). We show that an analogous result holds for their representation in \mathbf{NR} . We begin by defining hereditary Harrop formulæ and sequents.

Definition 13.13 (Hereditary Harrop Sequent)

Goal formulæ G and hereditary Harrop formulæ D are defined by the following grammars:

$$D := A \mid G \supset A \mid \forall xD \mid D_1 \wedge D_2$$

which tells us that there is a uniform proof-term

$$M = \text{AND-R}(y_1)(\text{IMP-R}(\lambda y_3 : \text{proof}(\psi) . \text{AND-R}(y_2)(\text{FORALL-L}(\lambda y_5 : \text{proof}(\chi) . y_5)y_4)))$$

such that $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_{3^i}}} M : \text{proof}(\phi \wedge (\psi \supset (\tau \wedge \chi)))$ is provable in $\text{NR}_{\Sigma_{G_{3^i}}}$, where $\Gamma_\Delta = y_1 : \text{proof}(\phi), y_2 : \text{proof}(\tau), y_4 : \text{proof}(\forall x\chi)$. We have the following derivation for this assertion. We write the derivation in stages due to its size.

$$\frac{\Gamma_1 \vdash_{\Sigma_{G_{3^i}}} y_2 : \text{proof}(\tau) \quad \Gamma_1 \vdash_{\Sigma_{G_{3^i}}} y_5 : \text{proof}(\chi)}{\Gamma_1 \vdash_{\Sigma_{G_{3^i}}} \text{AND-R}(y_2)y_5 : \text{proof}(\tau \wedge \chi)} \text{Res}$$

where $\Gamma_1 = \Gamma_X, \Gamma_\Delta, y_5 : \text{proof}(\chi), y_3 : \text{proof}(\psi)$

$$\frac{\Gamma_1 \vdash_{\Sigma_{G_{3^i}}} \text{AND-R}(y_2)y_5 : \text{proof}(\tau \wedge \chi)}{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} \lambda y_3 : \text{proof}(\psi) . \text{AND-R}(y_2)y_5 : \text{proof}(\psi) \rightarrow \text{proof}(\tau \wedge \chi)} \rightarrow I$$

where $\Gamma_2 = \Gamma_X, \Gamma_\Delta, y_5 : \text{proof}(\chi)$.

$$\frac{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} \lambda y_3 : \text{proof}(\psi) . \text{AND-R}(y_2)y_5 : \text{proof}(\psi) \rightarrow \text{proof}(\tau \wedge \chi)}{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} \text{IMP-R}(\lambda y_3 : \text{proof}(\psi) . \text{AND-R}(y_2)y_5) : \text{proof}(\psi \supset (\tau \wedge \chi))} \text{Res}$$

$$\frac{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} N : \text{proof}(\psi \supset (\tau \wedge \chi))}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_{3^i}}} \lambda y_5 : \text{proof}(\chi) . N : \text{proof}(\chi) \rightarrow \text{proof}(\psi \supset (\tau \wedge \chi))} \rightarrow I$$

where $N = \text{IMP-R}(\lambda y_3 : \text{proof}(\psi) . \text{AND-R}(y_2)y_5)$.

$$\frac{\Gamma_X, \Gamma_\Delta \vdash \lambda y_5 . N : \text{proof}(\chi) \rightarrow \text{proof}(\psi \supset (\tau \wedge \chi)) \quad \Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_{3^i}}} y_4 : \text{proof}(\forall x\chi)}{\Gamma_X, \Gamma_\Delta \vdash \text{FORALL-L}(\lambda y_5 : \text{proof}(\chi) . N)y_4 : \text{proof}(\psi \supset (\tau \wedge \chi))} \text{Res}$$

$$\frac{\Gamma_X, \Gamma_\Delta \vdash y_3 : \text{proof}(\phi) \quad \Gamma_X, \Gamma_\Delta \vdash P : \text{proof}(\psi \supset (\tau \wedge \chi))}{\Gamma_X, \Gamma_\Delta \vdash \text{AND-R}(y_1)(P) : \text{proof}(\phi \wedge (\psi \supset (\tau \wedge \chi)))} \text{Res}$$

where $P = \text{FORALL-L}(\lambda y_5 : \text{proof}(\chi) . N)y_4$. We observe that if we move the resolution step which resolves the constant *FORALL-L* upwards then, we also need to move the $\rightarrow I$ rule upwards. It is easier if we consider the movement of the $\rightarrow I$ rule. This can be moved towards the top of the tree provided the variable that it takes across the turnstile is still on the left. In this case, we can move the $\rightarrow I$ right to the leaf and then apply resolution. This gives us the

proof-tree:

$$\frac{\Gamma_1 \vdash_{\Sigma_{G_{3i}}} y_5 : \mathbf{proof}(\chi)}{\Gamma_2 \vdash_{\Sigma_{G_{3i}}} \lambda y_5 : \mathbf{proof}(\chi) . y_5 : \mathbf{proof}(\chi) \rightarrow \mathbf{proof}(\chi)} \rightarrow I$$

where $\Gamma_1 = \Gamma_X, \Gamma_\Delta, y_3 : \mathbf{proof}(\psi), y_5 : \mathbf{proof}(\chi)$ and $\Gamma_2 = \Gamma_X, \Gamma_\Delta, y_3 : \mathbf{proof}(\psi)$.

$$\frac{\Gamma_2 \vdash_{\Sigma_{G_{3i}}} \lambda y_5 : \mathbf{proof}(\chi) . y_5 : \mathbf{proof}(\chi) \rightarrow \mathbf{proof}(\chi) \quad \Gamma_2 \vdash_{\Sigma_{G_{3i}}} y_4 : \mathbf{proof}(\forall x \chi)}{\Gamma_2 \vdash_{\Sigma_{G_{3i}}} \mathbf{FORALL-L}(\lambda y_5 : \mathbf{proof}(\chi) . y_5) y_4 : \mathbf{proof}(\chi)} \text{Res}$$

$$\frac{\Gamma_2 \vdash_{\Sigma_{G_{3i}}} y_2 : \mathbf{proof}(\tau) \quad \Gamma_2 \vdash_{\Sigma_{G_{3i}}} \mathbf{FORALL-L}(\lambda y_5 : \mathbf{proof}(\chi) . y_5) y_4 : \mathbf{proof}(\chi)}{\Gamma_2 \vdash_{\Sigma_{G_{3i}}} \mathbf{AND-R}(y_2)(\mathbf{FORALL-L}(\lambda y_5 : \mathbf{proof}(\chi) . y_5) y_4) : \mathbf{proof}(\tau \wedge \chi)} \text{Res}$$

$$\frac{\Gamma_2 \vdash_{\Sigma_{G_{3i}}} N}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_{3i}}} \lambda y_3 : \mathbf{proof}(\psi) . N : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\tau \wedge \chi)} \rightarrow I$$

where $N = \mathbf{AND-R}(y_2)(\mathbf{FORALL-L}(\lambda y_5 : \mathbf{proof}(\chi) . y_5) y_4) : \mathbf{proof}(\tau \wedge \chi)$.

$$\frac{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_{3i}}} \lambda y_3 : \mathbf{proof}(\psi) . N : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\tau \wedge \chi)}{\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_{3i}}} \mathbf{IMP-R}(\lambda y_3 : \mathbf{proof}(\psi) . N) : \mathbf{proof}(\psi \supset (\tau \wedge \chi))} \text{Res}$$

$$\frac{\Gamma_X, \Gamma_\Delta \vdash y_1 : \mathbf{proof}(\phi) \quad \Gamma_X, \Gamma_\Delta \vdash \mathbf{IMP-R}(\lambda y_3 : \mathbf{proof}(\psi) . N) : \mathbf{proof}(\psi \supset (\tau \wedge \chi))}{\Gamma_X, \Gamma_\Delta \vdash \mathbf{AND-R}(y_1)(\mathbf{IMP-R}(\lambda y_3 : \mathbf{proof}(\psi) . N)) : \mathbf{proof}(\phi \wedge (\psi \supset (\tau \wedge \chi)))}$$

which we see has the same proof-term as Example 13.9. ■

In the above example we saw that the permutation of rules in G_{3i} which turned a non-uniform proof of a hereditary Harrop sequent into a uniform proof are mirrored in **NR**. We claim that this is possible for any hereditary Harrop sequent. The basis of this claim is that the permutation of a non-uniform proof of a hereditary Harrop sequent into a uniform proof only involves permuting left rules up the proof tree. Permuting a left rule upwards is possible until it encounters another left rule or a leaf. To avoid a clash of left rules during permutation, we start with the uppermost rule and work downwards. A permutation in G_{3i} corresponds to the permutation of a resolution rule in **NR**. We have to be careful because in **NR** we have to permute any abstraction needed for resolutions upwards as well. These rules can always be permuted upwards providing the variables in the context being abstracted are present.

We formalise the above discussion in the following lemma:

Lemma 13.16

Let $X, \Delta \vdash_{G_{3i}} \delta : \mathbf{proof}(\phi)$ be a provable hereditary Harrop sequent in G_{3i} and $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G_{3i}}} M_\delta : \mathbf{proof}(\phi)$ be its representation in $\mathbf{NR}_{\Sigma_{G_{3i}}}$. If δ is not in long $\beta\eta$ -normal form then the permutation which turns δ into long $\beta\eta$ -normal form

is mimicked in $\text{NR}_{\Sigma_{G_3i}}$ by a permutation of M_δ into a uniform proof-term. Similarly the permutation of M_δ into a uniform proof-term is mimicked in G_3i by a permutation of δ into long $\beta\eta$ -normal form.

Proof (Sketch) Let δ not be in long $\beta\eta$ -normal form. There exists a permutation which turns δ into long $\beta\eta$ -normal form because $(X) \Delta \vdash_{G_3i} \delta : \text{proof}(\phi)$ is a hereditary Harrop sequent. The permutation is on the proof-tree of δ and involves permuting a left rule upwards until the sequent in its conclusion has an atomic formula on the right. It is sufficient to show that any permutation of a left rule past a right rule has a corresponding permutation in $\text{NR}_{\Sigma_{G_3i}}$. We only prove a few cases, the rest are similar.

We begin with *v-AND-L* and *v-AND-R*. We have *v-AND-L* beneath *v-AND-R* thus:

$$\frac{(X) \Delta' \vdash \delta_1 : \text{proof}(\phi) \quad (X) \Delta' \vdash \delta_2 : \text{proof}(\psi)}{(X) \Delta, y : \text{proof}(\phi \wedge \psi), y_1 : \text{proof}(\phi), y_1 : \text{proof}(\psi) \vdash \text{AND-R}(\delta_1, \delta_2) : \text{proof}(\phi \wedge \psi)} \\ \frac{}{(X) \Delta, y : \text{proof}(\phi \wedge \psi) \vdash \text{AND-L}((y_1, y_1) : \text{AND-R}(\delta_1, \delta_2), y) : \text{proof}(\phi \wedge \psi)}$$

where $\Delta' = \Delta, y : \text{proof}(\phi \wedge \psi), y_1 : \text{proof}(\phi), y_1 : \text{proof}(\psi)$. Permuting *v-AND-L* yields

$$\frac{(X) \Delta, y : \text{proof}(\phi \wedge \psi), y_1 : \text{proof}(\phi), y_2 : \text{proof}(\psi) \vdash_{G_3i} \delta_1 : \text{proof}(\phi)}{(X) \Delta, y : \text{proof}(\phi \wedge \psi) \vdash_{G_3i} \text{AND-L}((y_1, y_2) : \delta_1, y) : \text{proof}(\phi)} \text{v-AND-L} \\ \frac{(X) \Delta, y : \text{proof}(\phi \wedge \psi), y_1 : \text{proof}(\phi), y_2 : \text{proof}(\psi) \vdash_{G_3i} \delta_2 : \text{proof}(\psi)}{(X) \Delta, y : \text{proof}(\phi \wedge \psi) \vdash_{G_3i} \text{AND-L}((y_1, y_2) : \delta_2, y) : \text{proof}(\psi)} \text{v-AND-L} \\ \frac{(X) \Delta, y : \text{proof}(\phi \wedge \psi) \vdash_{G_3i} P : \text{proof}(\psi) \quad (X) \Delta, y : \text{proof}(\phi \wedge \psi) \vdash_{G_3i} Q : \text{proof}(\phi)}{(X) \Delta, y : \text{proof}(\phi \wedge \psi) \vdash_{G_3i} \text{AND-R}(P)(Q) : \text{proof}(\phi \wedge \psi)}$$

where $P = \text{AND-L}((y_1, y_2) : \delta_2, y)$ and $Q = \text{AND-L}((y_1, y_2) : \delta_1, y)$. We can represent the first proof-tree above by the following derivation in $\text{NR}_{\Sigma_{G_3i}}$:

$$\frac{\frac{\Gamma_1 \vdash_{\Sigma_{G_3i}} M_{\delta_1} : \text{proof}(\phi) \quad \Gamma_1 \vdash_{\Sigma_{G_3i}} M_{\delta_2} : \text{proof}(\psi)}{\Gamma_1 \vdash_{\Sigma_{G_3i}} \text{AND-R}(M_{\delta_1})(M_{\delta_2}) : \text{proof}(\phi \wedge \psi)} \text{Res}}{\Gamma_1 \vdash_{\Sigma_{G_3i}} \text{AND-R}(M_{\delta_1})(M_{\delta_2}) : \text{proof}(\phi \wedge \psi)} \\ \frac{\Gamma_1 \vdash_{\Sigma_{G_3i}} \text{AND-R}(M_{\delta_1})(M_{\delta_2}) : \text{proof}(\phi \wedge \psi)}{\Gamma_2 \vdash_{\Sigma_{G_3i}} \lambda y_2 : \text{proof}(\psi) . \text{AND-R}(M_{\delta_1})(M_{\delta_2}) : \text{proof}(\psi) \rightarrow \text{proof}(\phi \wedge \psi)} \rightarrow I \\ \frac{\Gamma_2 \vdash_{\Sigma_{G_3i}} N : \text{proof}(\psi) \rightarrow \text{proof}(\phi \wedge \psi)}{\Gamma_3 \vdash_{\Sigma_{G_3i}} \lambda y_2 : \text{proof}(\phi) . N : \text{proof}(\phi) \rightarrow \text{proof}(\psi) \rightarrow \text{proof}(\phi \wedge \psi)}$$

$$\frac{\Gamma_3 \vdash \lambda y_2 : \mathbf{proof}(\phi) . N : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\phi \wedge \psi) \quad \Gamma_3 \vdash y : \mathbf{proof}(\phi \wedge \psi)}{\Gamma_3 \vdash \mathbf{AND-L}(\lambda y_2 : \mathbf{proof}(\phi) . N)y : \mathbf{proof}(\phi \wedge \psi)}$$

where $N = \lambda y_2 : \mathbf{proof}(\psi) . \mathbf{AND-R}(M_{\delta_1})(M_{\delta_2})$, $\Gamma_1 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi)$, $\Gamma_2 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi)$ and $\Gamma_3 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi)$. We have left out the judgements of the form $\Gamma \vdash \epsilon_s(\phi) : o$ to keep the derivation small.

We present the representation of the second proof-tree in three stages:

$$\frac{\frac{\frac{\Gamma_1 \vdash_{\Sigma_{G_{3^i}}} M_{\delta_1} : \mathbf{proof}(\phi)}{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} \lambda y_2 : \mathbf{proof}(\psi) . M_{\delta_1} : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\phi)}{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} \lambda y_2 : \mathbf{proof}(\psi) . M_{\delta_1} : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\phi)}}{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} \lambda y_1 : \mathbf{proof}(\phi) . \lambda y_2 : \mathbf{proof}(\psi) . M_{\delta_1} : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\psi)}}{\Gamma_3 \vdash N : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\psi) \quad \Gamma_3 \vdash y : \mathbf{proof}(\phi \wedge \psi)}{\Gamma_3 \vdash \mathbf{AND-L}(N)y : \mathbf{proof}(\phi)}$$

$$\frac{\frac{\frac{\Gamma_1 \vdash_{\Sigma_{G_{3^i}}} M_{\delta_2} : \mathbf{proof}(\psi)}{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} \lambda y_2 : M_{\delta_2} : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\psi)}{\Gamma_2 \vdash_{\Sigma_{G_{3^i}}} \lambda y_2 : M_{\delta_2} : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\psi)}}{\Gamma_3 \vdash_{\Sigma_{G_{3^i}}} \lambda y_1 : \mathbf{proof}(\phi) . \mathbf{proof}(\psi) . M_{\delta_2} : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\psi)}}{\Gamma_3 \vdash P : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\psi) \quad \Gamma_3 \vdash y : \mathbf{proof}(\phi \wedge \psi)}{\Gamma_3 \vdash \mathbf{AND-L}(P)y : \mathbf{proof}(\psi)}$$

$$\frac{\Gamma_3 \vdash M_{\delta_3} : \mathbf{proof}(\phi) \quad \Gamma_3 \vdash M_{\delta_3} : \mathbf{proof}(\psi)}{\Gamma_3 \vdash \mathbf{AND-R}(M_{\delta_3})(M_{\delta_4}) : \mathbf{proof}(\phi \wedge \psi)} \text{Res}$$

where $M_{\delta_3} = \mathbf{AND-L}(\lambda y_1 : \mathbf{proof}(\phi) . \lambda y_2 : \mathbf{proof}(\psi) . M_{\delta_1})y$, $M_{\delta_4} = \mathbf{AND-L}(\lambda y_1 : \mathbf{proof}(\phi) . y_2 : \mathbf{proof}(\psi) . M_{\delta_2})y$, $N = \lambda y_1 : \mathbf{proof}(\phi) . \lambda y_2 : \mathbf{proof}(\psi) . M_{\delta_1}$, $P = \lambda y_1 : \mathbf{proof}(\phi) . \mathbf{proof}(\psi) . M_{\delta_2}$, $\Gamma_1 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi)$, $\Gamma_2 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi)$ and $\Gamma_3 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi)$. We observe that the second $\mathbf{NR}_{\Sigma_{G_{3^i}}}$ derivation can be obtained from the first. We permute the two $\rightarrow R$ rules and the resolution step resolving $\mathbf{AND-L}$ past the resolution step resolving $\mathbf{AND-R}$.

We now consider the permutation of $v\text{-IMP-L}$ past $v\text{-AND-R}$

$$\frac{(X) \Delta, y_1 : \mathbf{proof}(\psi) \vdash \delta_2 : \mathbf{proof}(\chi) \quad (X) \Delta, y_1 : \mathbf{proof}(\psi) \vdash \delta_3 : \mathbf{proof}(\tau)}{(X) \Delta, y_1 : \mathbf{proof}(\psi) \vdash \mathbf{AND-R}(\delta_2, \delta_3) : \mathbf{proof}(\chi \wedge \tau)}$$

$$\frac{X\Delta \vdash \delta_1 : \mathbf{proof}(\chi) \quad X\Delta, y_1 : \mathbf{proof}(\psi) \vdash N : \mathbf{proof}(\chi \wedge \tau) \quad X\Delta \vdash y : \mathbf{proof}(\phi \supset \psi)}{(X) \Delta \vdash \mathit{IMP-L}(\delta_1, y_1 : \mathit{AND-R}(\delta_2, \delta_3), y) : \mathbf{proof}(\chi \wedge \tau)}$$

where $P = \mathit{AND-R}(\delta_2, \delta_2)$. We permute $v\text{-IMP-L}$ upwards to obtain

$$\frac{(X) \Delta \vdash \delta_1 : \mathbf{proof}(\phi) \quad (X) \Delta, y_1 : \mathbf{proof}(\psi) \vdash \delta_2 : \mathbf{proof}(\chi) \quad (X) \Delta \vdash y : \mathbf{proof}(\phi \supset \psi)}{(X) \Delta \vdash \mathit{IMP-L}(\delta_1, y_1 : \delta_2, y) : \mathbf{proof}(\chi)}$$

$$\frac{(X) \Delta \vdash \delta_1 : \mathbf{proof}(\psi) \quad (X) \Delta, y_1 : \mathbf{proof}(\psi) \vdash \delta_3 : \mathbf{proof}(\tau) \quad (X) \Delta \vdash y : \mathbf{proof}(\phi \supset \psi)}{(X) \Delta \vdash \mathit{IMP-L}(\delta_1, y_1 : \delta_3, y) : \mathbf{proof}(\tau)}$$

$$\frac{(X) \Delta \vdash N : \mathbf{proof}(\chi) \quad (X) \Delta \vdash P : \mathbf{proof}(\tau)}{(X) \Delta \vdash \mathit{AND-R}(N, P) : \mathbf{proof}(\chi \wedge \tau)}$$

where $N = \mathit{IMP-L}(\delta_1, y_1 : \delta_2, y) : \mathbf{proof}(\chi)$ and $P = \mathit{IMP-L}(\delta_1, y_1 : \delta_3, y) : \mathbf{proof}(\tau)$.

The first proof-tree is represented in $\mathbf{NR}_{\Sigma_{G_{3i}}}$ by the following derivation:

$$\frac{\frac{\frac{\Gamma_2 \vdash M_{\delta_2} : \mathbf{proof}(\chi) \quad \Gamma_2 \vdash M_{\delta_3} : \mathbf{proof}(\tau)}{\Gamma_2 \vdash \mathit{AND-R}(M_{\delta_2})(M_{\delta_3}) : \mathbf{proof}(\chi \wedge \tau)} \text{Res}}{\Gamma_2 \vdash \mathit{AND-R}(M_{\delta_2})(M_{\delta_3}) : \mathbf{proof}(\chi \wedge \tau)}}{\Gamma_1 \vdash \lambda y_1 : \mathbf{proof}(\psi) . \mathit{AND-R}(M_{\delta_2})(M_{\delta_3}) : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\chi \wedge \tau)}$$

$$\frac{\Gamma_1 \vdash M_{\delta_1} : \mathbf{proof}(\psi) \quad \Gamma_1 \vdash N : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\chi \wedge \tau) \quad \Gamma_1 \vdash y : \mathbf{proof}(\phi \supset \psi)}{\Gamma_1 \vdash \mathit{IMP-L}(M_{\delta_1})(N)y : \mathbf{proof}(\chi \wedge \tau)}$$

where $N = \lambda y_1 : \mathbf{proof}(\psi) . \mathit{AND-R}(M_{\delta_2})(M_{\delta_3})$.

We represent the second proof-tree in stages due to its size.

$$\frac{\Gamma_2 \vdash M_{\delta_2} : \mathbf{proof}(\chi)}{\Gamma_1 \vdash \lambda y_1 : \mathbf{proof}(\psi) . M_{\delta_2} : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\chi)}$$

$$\frac{\Gamma_1 \vdash M_{\delta_1} : \mathbf{proof}(\phi) \quad \Gamma_1 \vdash N : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\chi) \quad \Gamma_1 \vdash y : \mathbf{proof}(\phi \supset \psi)}{\Gamma_1 \vdash \mathit{IMP-L}(M_{\delta_1})Ny : \mathbf{proof}(\chi)}$$

$$\frac{\Gamma_2 \vdash M_{\delta_3} : \mathbf{proof}(\tau)}{\Gamma_1 \vdash \lambda y_1 : \mathbf{proof}(\psi) . M_{\delta_3} : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\tau)}$$

$$\frac{\Gamma_1 \vdash M_{\delta_1} : \mathbf{proof}(\chi) \quad \Gamma_1 \vdash P : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\tau) \quad \Gamma_1 \vdash y : \mathbf{proof}(\phi \supset \psi)}{\Gamma_1 \vdash \mathit{IMP-L}(M_{\delta_1})Py : \mathbf{proof}(\tau)}$$

$$\frac{\Gamma_1 \vdash M_{\delta_4} : \mathbf{proof}(\chi) \quad \Gamma_1 \vdash M_{\delta_5} : \mathbf{proof}(\tau)}{\Gamma_1 \vdash \mathit{AND-R}(M_{\delta_4})(M_{\delta_5}) : \mathbf{proof}(\chi \wedge \tau)}$$

where $\Gamma_1 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \supset \psi)$, $\Gamma_2 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \supset \psi), y_1 : \mathbf{proof}(\psi)$,

$M_{\delta_4} = IMP-L(M_{\delta_1})Py$, $M_{\delta_5} = IMP-L(M_{\delta_1})(\lambda y_1 : \mathbf{proof}(\psi) . M_{\delta_3})y$, $N = \lambda y_1 : \mathbf{proof}(\psi) . M_{\delta_2}$ and $P = \lambda y_1 : \mathbf{proof}(\psi) . M_{\delta_3}$.

Finally, we show $v-AND-L$ permuting up past $v-FORALL-R$. We initially have

$$\frac{\frac{X, x \Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi) \vdash \delta : \mathbf{proof}(\phi[x])}{(X) \Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi) \vdash FORALL-R(\delta) : \mathbf{proof}(\forall x . \phi)}}{(X) \Delta, y : \mathbf{proof}(\phi \wedge \psi) \vdash AND-L((y_1, y_2) : FORALL-R(\delta), y) : \mathbf{proof}(\forall x \phi)}$$

We now permute $v-AND-L$ upwards to obtain

$$\frac{\frac{X, x \Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi) \vdash_{G_{3i}} \delta : \mathbf{proof}(\phi[x])}{X.x \Delta, y : \mathbf{proof}(\phi \wedge \psi) \vdash_{G_{3i}} AND-L((y_1, y_2) : \delta, y) : \mathbf{proof}(\phi[x])}}{(X) \Delta, y : \mathbf{proof}(\phi \wedge \psi) \vdash_{G_{3i}} FORALL-R(AND-L((y_1, y_2) : \delta, y)) : \mathbf{proof}(\forall x \phi)}$$

We encode the first proof-tree in $NR_{\Sigma_{G_{3i}}}$ to obtain

$$\frac{\frac{\frac{\Gamma_1 \vdash \delta : \mathbf{proof}(\phi x)}{\Gamma_2 \vdash \lambda x : \iota . \delta : \Pi x : \iota . \mathbf{proof}(\phi x)}{\Gamma_2 \vdash \lambda x : \iota . \delta : \Pi x : \iota . \mathbf{proof}(\phi x)}}{\Gamma_2 \vdash FORALL-R(\lambda x : \iota . \delta) : \mathbf{proof}(\forall(\lambda x : \iota . \phi x))}}{\Gamma_2 \vdash FORALL-R(\lambda x : \iota . \delta) : \mathbf{proof}(\forall(\lambda x : \iota . \phi x))}}{\Gamma_3 \vdash \lambda y_2 : \mathbf{proof}(\psi) . FORALL-R(\lambda x : \iota . \delta) : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\forall(\lambda x : \iota . \phi x))}}{\Gamma_3 \vdash N : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\forall(\lambda x : \iota . \phi x))}}{\Gamma_4 \vdash \lambda y_1 : \mathbf{proof}(\phi) . N : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\forall(\lambda x : \iota . \phi x))}}{\Gamma_4 \vdash \lambda y_1 : \mathbf{proof}(\phi) . N : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\forall(\lambda x : \iota . \phi x))}}{\Gamma_4 \vdash AND-L(\lambda y_1 : \mathbf{proof}(\phi) . N)y : \mathbf{proof}(\forall(\lambda x : \iota . \phi x))}}$$

where $\Gamma_1 = \Gamma_X, x : \iota, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi)$, $\Gamma_2 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi)$, $\Gamma_3 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi)$, $\Gamma_4 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi)$, $N = \lambda y_2 : \mathbf{proof}(\psi) . FORALL-R(\lambda x : \iota . \delta)$, We encode the permuted proof-tree in $NR_{\Sigma_{G_{3i}}}$ as

$$\frac{\frac{\frac{\Gamma_1 \vdash M_\delta : \mathbf{proof}(\phi x)}{\Gamma_2 \vdash \lambda y_2 : \mathbf{proof}(\psi) . M_\delta : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\phi x)}{\Gamma_2 \vdash \lambda y_2 : \mathbf{proof}(\psi) . M_\delta : \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\phi x)}}{\Gamma_3 \vdash \lambda y_1 : \mathbf{proof}(\phi) . \lambda y_2 : \mathbf{proof}(\psi) . M_\delta : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\phi x)}}$$

$$\begin{array}{c}
\Gamma_3 \vdash N : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\phi x) \\
\hline
\Gamma_3 \vdash \mathbf{AND-L}(N)y : \mathbf{proof}(\phi x) \\
\Gamma_3 \vdash \mathbf{AND-L}(N)y : \mathbf{proof}(\phi x) \\
\hline
\Gamma_4 \vdash \lambda x : \iota . \mathbf{AND-L}(N)y : \Pi x : \iota . \mathbf{proof}(\phi x) \\
\Gamma_4 \vdash \lambda x : \iota . \mathbf{AND-L}(N)y : \Pi x : \iota . \mathbf{proof}(\phi x) \\
\hline
\Gamma_4 \vdash \mathbf{FORALL-R}(\lambda x : \iota . \mathbf{AND-L}(N)y) : \mathbf{proof}(\forall(\lambda x : \iota . \phi x))
\end{array}$$

where $\Gamma_1 = \Gamma_X, x : \iota, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi)$, $\Gamma_2 = \Gamma_X, x : \iota, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi)$, $\Gamma_3 = \Gamma_X, x : \iota, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi)$, $\Gamma_4 = \Gamma_X, \Gamma_\Delta, y : \mathbf{proof}(\phi \wedge \psi)$ and $N = \lambda y_1 : \mathbf{proof}(\phi) . \lambda y_2 : \mathbf{proof}(\psi) . M_\delta$. This is also a permutation of the resolution step resolving *FORALL-L* together with the two $\rightarrow R$ rules upwards.

We have shown a few of the many interesting cases, the rest are similar. In each case, the permutation in G_3i corresponds directly to a permutation in $\text{NR}_{\Sigma_{G_3i}}$. The other direction is similar. \blacksquare

13.2 Representing Uniform Proofs in G_3HOIL in LF

In § 12.3, we showed that G_3HOIL could be adequately represented in LF. We begin by examining how the resolution rule in NR represents valid proofs of G_3HOIL . We begin with the right rules which are encoded as generalized introduction rules by *Res*. We write each right rule followed by its representation by *Res*. We ignore the **ap** used to form expressions in judgements and write the usual logical expression for clarity.

$$\begin{array}{c}
\frac{(\mathcal{A}) \Delta \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\phi) \quad (\mathcal{A}) \Delta \vdash_{G_3HOIL} \delta_2 : \mathbf{proof}(\psi)}{(\mathcal{A}) \Delta \vdash_{G_3HOIL} \mathbf{AND-R}(\delta_1, \delta_2) : \mathbf{proof}(\phi \wedge \psi)} \quad v\text{-AND-R} \\
\\
\frac{\Gamma_{\mathcal{A}}, \Gamma_\Delta \vdash_{\Sigma_{G_3HOIL}} M_{\delta_1} : \mathbf{proof}(\phi) \quad \Gamma_{\mathcal{A}}, \Gamma_\Delta \vdash_{\Sigma_{G_3HOIL}} M_{\delta_2} : \mathbf{proof}(\psi)}{\Gamma_{\mathcal{L}}, \Gamma_\Delta \vdash_{\Sigma_{G_3HOIL}} \mathbf{AND-R}\epsilon(\phi)\epsilon(\psi)M_{\delta_1}M_{\delta_2} : \mathbf{proof}(\phi \wedge \psi)} \quad Res \\
\\
\frac{(\mathcal{A}) \Delta \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\phi)}{(\mathcal{A}) \Delta \vdash_{G_3HOIL} \mathbf{OR-R1}(\delta_1) : \mathbf{proof}(\phi \vee \psi)} \quad v\text{-OR-R1} \\
\\
\frac{\Gamma_{\mathcal{A}}, \Gamma_\Delta \vdash_{\Sigma_{G_3HOIL}} M_{\delta_1} : \mathbf{proof}(\phi)}{\Gamma_{\mathcal{A}}, \Gamma_\Delta \vdash_{\Sigma_{G_3HOIL}} \mathbf{OR-R1}\epsilon(\phi)\epsilon(\psi)M_{\delta_1} : \mathbf{proof}(\phi \vee \psi)} \quad Res \\
\\
\frac{(\mathcal{A}) \Delta \vdash_{G_3HOIL} \delta_2 : \mathbf{proof}(\psi)}{(\mathcal{A}) \Delta \vdash_{G_3HOIL} \mathbf{OR-R2}(\delta_2) : \mathbf{proof}(\phi \vee \psi)} \quad v\text{-OR-R2}
\end{array}$$

$$\begin{array}{c}
\frac{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} M_{\delta_2} : \mathbf{proof}(\psi)}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} OR\text{-}R2\epsilon(\phi)\epsilon(\psi)M_{\delta_1} : \mathbf{proof}(\phi \vee \psi)} \text{Res} \\
\frac{(\mathcal{A}) \Delta, y_1 : \mathbf{proof}(\phi) \vdash_{G_3HOIL} \delta_1 : \mathbf{proof}(\psi)}{(\mathcal{A}) \Delta \vdash_{G_3HOIL} IMP\text{-}R(y_1 : \delta_1) : \mathbf{proof}(\phi \supset \psi)} \text{v-IMP-R} \\
\frac{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} \lambda y_1 : \mathbf{proof}(\phi) . M_{\delta_1} : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi)}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} IMP\text{-}R\epsilon(\phi)\epsilon(\psi)(\lambda y_1 : \mathbf{proof}(\psi) . M_{\delta_1}) : \mathbf{proof}(\phi \supset \psi)} \text{Res} \\
\frac{(\mathcal{A}, x : \sigma) \Delta \vdash_{G_3HOIL} \delta : \mathbf{proof}(\phi x)}{(\mathcal{A}) \Delta \vdash_{G_3HOIL} FORALL\text{-}R(\delta) : \mathbf{proof}(\forall_{\sigma} \phi)} \text{v-FORALL-R} \\
\frac{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} \lambda x : \mathbf{obj}(\sigma) . M_{\delta} : \Pi x : \sigma . \mathbf{proof}(\phi x)}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} FORALL\text{-}L\epsilon(\sigma)\epsilon(\phi)(\lambda x : \mathbf{obj}(\sigma) . M_{\delta}) : \mathbf{proof}(\forall_{\sigma} \phi)} \text{Res} \\
\frac{(\mathcal{A}) \Delta \vdash_{G_3HOIL} \delta : \mathbf{proof}(\phi x)}{(\mathcal{A}) \Delta \vdash_{G_3HOIL} EXISTS\text{-}R(\delta) : \mathbf{proof}(\exists_{\sigma} \phi)} \text{v-EXISTS-R} \\
\frac{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} M_{\delta} : \mathbf{proof}(\phi x)}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} EXISTS\text{-}R\epsilon(\sigma)\epsilon(\phi)\epsilon(x)M_{\delta} : \mathbf{proof}(\forall_{\sigma} \phi)} \text{Res}
\end{array}$$

We observe that each right rule is represented as a generalized introduction rule. We now represent the left rules using *Res* and observe that each left rule is represented as a generalized elimination rule.

$$\frac{(\mathcal{A}) \Delta, y : \mathbf{proof}(\phi \wedge \psi), y_1 : \mathbf{proof}(\phi), y_2 : \mathbf{proof}(\psi) \vdash_{G_3HOIL} \delta : \mathbf{proof}(\chi)}{(\mathcal{A}) \Delta, y : \mathbf{proof}(\phi \wedge \psi) \vdash_{G_3HOIL} AND\text{-}L((y_1, y_2) : \delta, y) : \mathbf{proof}(\chi)} \text{v-AND-L} \\
\frac{\Gamma \vdash N : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\psi) \rightarrow \mathbf{proof}(\chi) \quad \Gamma \vdash y : \mathbf{proof}(\phi \wedge \psi)}{\Gamma \vdash AND\text{-}L\epsilon(\phi)\epsilon(\psi)(N)y : \mathbf{proof}(\chi)}$$

where $\Gamma = \Gamma_{\mathcal{A}}, \Gamma_{\Delta}, y : \mathbf{proof}(\phi \vee \psi)$ and $N = \lambda y_1 : \mathbf{proof}(\phi) . \lambda y_2 : \mathbf{proof}(\psi) . M_{\delta}$.

$$\frac{(\mathcal{A}) \Delta, y_1 : \mathbf{proof}(\phi) \vdash \delta_1 : \mathbf{proof}(\chi) \quad (\mathcal{A}) \Delta, y_2 : \mathbf{proof}(\psi) \vdash \delta_2 : \mathbf{proof}(\chi)}{(\mathcal{A}) \Delta, y : \mathbf{proof}(\phi \vee \psi) \vdash OR\text{-}L(y_1 : \delta_1, y_2 : \delta_2, y) : \mathbf{proof}(\chi)} \text{v-OR-L}$$

where $\Delta = (\mathcal{A}) \Delta, y : \mathbf{proof}(\phi \vee \psi)$.

$$\frac{\Gamma \vdash N : \mathbf{proof}(\phi) \rightarrow \mathbf{proof}(\chi) \quad \Gamma \vdash P : \mathbf{proof}(\chi) \quad \Gamma \vdash y : \mathbf{proof}(\phi \vee \psi)}{\Gamma \vdash OR\text{-}L\epsilon(\phi)\epsilon(\psi)(\lambda y_1 : \mathbf{proof}(\phi) . M_{\delta_1})(\lambda y_2 : \mathbf{proof}(\psi) . M_{\delta_2})y : \mathbf{proof}(\chi)}$$

where $\Gamma = \Gamma_{\mathcal{A}}, \Gamma_{\Delta}, y : \text{proof}(\phi \vee \psi)$, $N = \lambda y_1 : \text{proof}(\phi) . M_{\delta_1}$, $P = \lambda y_2 : \text{proof}(\psi) . M_{\delta_2}$ and $y : \text{proof}(\phi \vee \psi) \in \Gamma_{\Delta}$.

$$\frac{\Delta \vdash \delta_1 : \text{proof}(\phi) \quad \Delta, y_1 : \text{proof}(\psi) \vdash \delta_2 : \text{proof}(\chi)}{(\mathcal{A}) \quad \Delta, y : \text{proof}(\phi \supset \psi) \vdash \text{IMP-R}(\delta_1, y_1 : \delta_2, y) : \text{proof}(\chi)} \text{v-IMP-L}$$

where $\Delta = (\mathcal{A}) \quad \Delta, y : \text{proof}(\phi \supset \psi)$.

$$\frac{\Gamma \vdash M_{\delta_1} : \text{proof}(\phi) \quad \Gamma \vdash \lambda y_1 : \text{proof}(\psi) . M_{\delta_2} : \text{proof}(\chi) \quad \Gamma \vdash y : \text{proof}(\phi \supset \psi)}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash \text{IMP-R}\epsilon(\phi)\epsilon(\psi)M_{\delta_1}(\lambda y_1 : \text{proof}(\psi) . M_{\delta_2})y : \text{proof}(\chi)}$$

where $\Gamma = \Gamma_{\mathcal{A}}, \Gamma_{\Delta}, y : \text{proof}(\phi \supset \psi)$.

$$\frac{(\mathcal{A}) \quad \Delta, y : \text{proof}(\forall_{\sigma}\phi), y_1 : \text{proof}(\phi x) \vdash_{G_3\text{HOIL}} \delta : \text{proof}(\chi)}{(\mathcal{A}) \quad \Delta, y : \text{proof}(\forall_{\sigma}\phi) \vdash_{G_3\text{HOIL}} \text{FORALL-L}(y_1 : \delta, y) : \text{proof}(\chi)} \text{v-FORALL-L}$$

$$\frac{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash \lambda y_1 : \text{proof}(\phi x) . M_{\delta} : \text{proof}(\phi x) \rightarrow \text{proof}(\chi) \quad \Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash y : \text{proof}(\forall_{\sigma}\phi)}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash \text{FORALL-L}\epsilon(\sigma)\epsilon(\phi)\epsilon(\chi)\epsilon(x)(\lambda y_1 : \text{proof}(\phi x) . M_{\delta})y : \text{proof}(\chi)}$$

where $y : \text{proof}(\forall_{\sigma}\phi) \in \Gamma_{\Delta}$.

$$\frac{(\mathcal{A}), x : \sigma \quad \Delta, y : \text{proof}(\exists_{\sigma}\phi), y_1 : \text{proof}(\phi x) \vdash_{G_3\text{HOIL}} \delta : \text{proof}(\chi)}{(\mathcal{A}) \quad \Delta, y : \text{proof}(\exists_{\sigma}\phi) \vdash_{G_3\text{HOIL}} \text{EXISTS-L}(\delta, y) : \text{proof}(\chi)} \text{v-EXISTS-R}$$

$$\frac{\Gamma \vdash N : (\Pi x : \sigma . \text{proof}(\phi x)) \rightarrow \text{proof}(\chi) \quad \Gamma \vdash y : \text{proof}(\exists_{\sigma}\phi x)}{\Gamma \vdash \text{EXISTS-L}\epsilon(\sigma)\epsilon(\phi)\epsilon(\chi)(\lambda x : \text{obj}(\sigma) . \lambda y_1 : \text{proof}(\phi x) . M_{\delta})y : \text{proof}(\chi)}$$

where $\Gamma = \Gamma_{\mathcal{A}}, \Gamma_{\Delta}$ and $N = \lambda x : \text{obj}(\sigma) . \lambda y_1 : \text{proof}(\phi x) . M_{\delta}$. We now just show how the remaining rules are represented by *Res*.

$$\frac{(\mathcal{A}) \quad \Delta \vdash_{G_3\text{HOIL}} \delta_1 : \text{proof}(\phi) \quad (\mathcal{A}) \quad \Delta \vdash_{G_3\text{HOIL}} \delta_2 : \text{proof}(\phi =_o \psi)}{(\mathcal{A}) \quad \Delta \vdash_{G_3\text{HOIL}} \text{EQ}(\delta_1, \delta_2) : \text{proof}(\psi)} \text{v-EQ}$$

$$\frac{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma G_3\text{HOIL}} M_{\delta_1} : \text{proof}(\phi) \quad \Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma G_3\text{HOIL}} M_{\delta_2} : \text{proof}(\phi \approx_o \psi)}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma G_3\text{HOIL}} \text{EQ}\epsilon(\phi)\epsilon(\psi)M_{\delta_1}M_{\delta_2} : \text{proof}(\psi)} \text{Res}$$

$$\frac{(\mathcal{A}), x : \sigma \quad \Delta \vdash_{G_3\text{HOIL}} \delta : \text{proof}(\phi =_{\tau} \psi)}{(\mathcal{A}) \quad \Delta \vdash_{G_3\text{HOIL}} \text{LAM}(\delta) : \text{proof}(\Lambda_{\sigma} x \phi =_{\sigma \Rightarrow \tau} \Lambda_{\sigma} x \psi)} \text{v-LAM}$$

$$\frac{\Gamma \vdash \lambda x : \text{obj}(\sigma) . M_{\delta} : \Pi x : \text{obj}(\sigma) . \text{proof}(\phi =_{\tau} \psi)}{\Gamma \vdash \text{LAM}\epsilon(\sigma)\epsilon(\tau)\epsilon(\phi)\epsilon(\psi)\lambda x : \text{obj}(\sigma) . M_{\delta} : \text{proof}(\Phi)}$$

where $\Gamma = \Gamma_{\mathcal{A}}, \Gamma_{\Delta}$ and $\Phi = \Lambda_{\sigma, \tau} \lambda x : \text{obj}(\sigma) . \phi x \approx_{\sigma \Rightarrow \tau} \Lambda_{\sigma, \tau} \lambda x : \text{obj}(\sigma) . \psi x$.

We give an example of a uniform proof and its encoding to allow us to begin

to understand the relationship between resolution in G_3HOIL and resolution in LF.

Example 13.17

The following proof is uniform.

$$\begin{array}{c}
\frac{}{(\mathcal{A}) \Delta_1 \vdash TOP : \mathbf{proof}(\top)} \top \\
\frac{}{(\mathcal{A}) \Delta_2 \vdash y_3 : \mathbf{proof}(Pa)} v\text{-}HYP \\
\frac{(\mathcal{A}) \Delta_1 \vdash TOP : \mathbf{proof}(\top) \quad (\mathcal{A}) \Delta_2 \vdash y_3 : \mathbf{proof}(Pa)}{(\mathcal{A}) \Delta_1 \vdash IMP\text{-}L(TOP, y_3 : y_3, y_2) : \mathbf{proof}(Pa)} v\text{-}IMP\text{-}L \\
\frac{(\mathcal{A}) \Delta_1 \vdash IMP\text{-}L(TOP, y_3 : y_3, y_2) : \mathbf{proof}(Pa)}{(\mathcal{A}) \Delta_3 \vdash FORALL\text{-}L(y_2 : IMP\text{-}L(TOP, y_3 : y_3, y_2), y_1) : \mathbf{proof}(Pa)} v\text{-}FORALL \\
\frac{(\mathcal{A}) \Delta_3 \vdash FORALL\text{-}L(y_2 : IMP\text{-}L(TOP, y_3 : y_3, y_2), y_1) : \mathbf{proof}(Pa)}{(\mathcal{A}) \Delta_3 \vdash EXISTS\text{-}R(FOR\text{-}L(y_2 : IMP\text{-}L(TOP, y_3 : y_3, y_2), y_1)) : \mathbf{proof}(\exists_\sigma y Py)}
\end{array}$$

where $\Delta_1 = y_1 : \mathbf{proof}(\forall_o(x \supset Pa)), y_2 : \mathbf{proof}(\top \supset Pa)$, $\Delta_2 = y_1 : \mathbf{proof}(\forall_o(x \supset Pa)), y_2 : \mathbf{proof}(\top \supset Pa), y_3 : \mathbf{proof}(Pa)$, $\Delta_3 = y_1 : \mathbf{proof}(\forall_o(x \supset Pa))$ and P has arity $\iota \rightarrow o$.

This proof is represented in NR by the following derivation. We complete the derivation in stages due to its size.

$$\begin{array}{c}
\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_2} \vdash y_3 : \mathbf{proof}(Pa)} \\
\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_1} \vdash \lambda y_3 . y_3 : \mathbf{proof}(Pa) \rightarrow \mathbf{proof}(Pa)} \\
\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_1} \vdash TOP : \mathbf{proof}(\top) \quad \Gamma \vdash N : \mathbf{proof}(Pa) \rightarrow \mathbf{proof}(Pa) \quad \Gamma \vdash y_2 : \mathbf{proof}(\top \supset Pa)} \\
\frac{}{\Gamma \vdash IMP\text{-}L\epsilon(\top)\epsilon(Pa)TOP(\lambda y_3 : \mathbf{proof}(Pa) . y_3)y_2 : \mathbf{proof}(Pa)} \\
\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_1} \vdash M_{\delta_1} : \mathbf{proof}(Pa)} \\
\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_3} \vdash \lambda y_2 : \mathbf{proof}(\top \supset Pa) . M_{\delta_1} : \mathbf{proof}(\top \supset Pa) \rightarrow \mathbf{proof}(Pa)} \\
\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_3} Q : \mathbf{proof}(\top \supset Pa) \rightarrow \mathbf{proof}(Pa) \quad \Gamma_{\mathcal{A}}, \Gamma_{\Delta_3} \vdash y_1 : \mathbf{proof}(\forall_\sigma(x \supset Pa)))} \\
\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_3} \vdash FORALL\text{-}L\epsilon(\sigma)\epsilon(x \supset Pa)\epsilon(x)(Q)y_1 : \mathbf{proof}(Pa)}
\end{array}$$

where $\Gamma = \Gamma_{\mathcal{A}}, \Gamma_{\Delta_1}$, $N = \lambda y_3 . y_3$, $Q = \lambda y_2 : \mathbf{proof}(\top \supset Pa) . M_{\delta_1}$ and $M_{\delta_1} = IMP\text{-}L\epsilon(\top)\epsilon(Pa)TOP(\lambda y_3 : \mathbf{proof}(Pa) . y_3)y_2$.

$$\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_3} \vdash M_{\delta_2} : \mathbf{proof}(Pa)} \\
\frac{}{\Gamma_{\mathcal{A}}, \Gamma_{\Delta_3} \vdash EXISTS\text{-}R\epsilon(\tau)\epsilon(Qy)\epsilon(y)(M_{\delta_2}) : \mathbf{proof}(\exists_\tau Py)}$$

where $M_{\delta_2} = \text{FORALL-L}\epsilon(o)\epsilon(x \supset Pa)\epsilon(x)(\lambda y_2 : \text{proof}(\top \supset Pa) . M_{\delta_1})y_1$. ■

The same observations about how uniform proofs are represented in LF for G_3i also hold for G_3HOIL . We now provide a definition of uniform proof-terms, which we will show are the terms in \mathbf{NR} which represent uniform proofs in G_3HOIL .

Definition 13.18 (Uniform Proof-Terms in $\mathbf{NR}_{\Sigma_{G_3HOIL}}$)

A *uniform proof* in $\mathbf{NR}_{\Sigma_{G_3HOIL}}$ is a term M in long $\beta\eta$ -normal form of type $\text{proof}(\phi)$ in which all subterms N , which are not variables, satisfy the following conditions:

- N never has type $\text{proof}(\neg\phi)$;
- N never contains the term $NEG-L$;
- if N has type $\text{proof}(\top)$ then $N = TOP$;
- if N has type $\text{proof}(\phi \wedge \psi)$ then $N = P_1 \cdots P_n$ where $P_1 = AND-R$;
- if N has type $\text{proof}(\phi \vee \psi)$ then $N = P_1 \cdots P_n$ where $P_1 = OR-R$;
- if N has type $\text{proof}(\phi \supset \psi)$ then $N = P_1 \cdots P_n$ where $P_1 = IMP-R$;
- if N has type $\text{proof}(\forall_\sigma \phi x)$ then $N = P_1 \cdots P_n$ where $P_1 = FORALL-R$;
- if N has type $\text{proof}(\exists_\sigma \phi x)$ then $N = P_1 \cdots P_n$ where $P_1 = EXISTS-R$. ■

We have the following lemma.

Lemma 13.19 (Uniform Proof-terms Represent Uniform Proofs)

Let $(\mathcal{A}) \Delta \vdash_{G_3HOIL} \delta : \text{proof}(\phi)$ be a provable assertion in G_3HOIL and $\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{G_3HOIL}} M_\delta : \text{proof}(\phi)$ be its representation in $\mathbf{NR}_{\Sigma_{G_3HOIL}}$. δ is in long $\beta\eta$ -normal form if and only if M_δ is a uniform proof-term. ■

The proof is identical to Lemma 13.12. In Example 13.17, the proof-object is in long $\beta\eta$ -normal form and it is encoded in $\mathbf{NR}_{\Sigma_{G_3HOIL}}$ as a uniform proof-term.

There is a particular class of sequents which always have a uniform proof. These sequents are the higher-order hereditary Harrop sequents and we define them below:

Definition 13.20 (Hereditary Harrop Sequents (Miller et al. 1991))

Let \mathcal{H} be the set of all expressions of G_3HOIL in β -normal form which do not contain \supset . An atomic formula A in \mathcal{H} is said to be *rigid* if A is of the form $Pt_1 \dots t_n$ where P is a non-logical constant. We denote rigid formulæ by A_r .

We say that a sequent $\Delta \vdash_{G_3HOIL} \mathbf{proof}(\phi)$ in G_3HOIL is a *higher-order hereditary Harrop sequent* if ϕ is a G -formula and all the formulæ in Δ are D -formulæ defined by the grammars:

$$G := \top \mid A \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid \forall_\sigma G \mid \exists_\sigma G \mid D \supset G$$

$$D := A_r \mid G \supset A_r \mid D_1 \wedge D_2 \mid \forall_\sigma D$$

The D -formulæ defined by this grammar are called *higher-order hereditary Harrop clauses*. ■

The following corollary shows that higher-order hereditary Harrop sequents have the property we want.

Corollary 13.21

Every higher-order hereditary Harrop sequent has a uniform proof.

Proof (Sketch) We invoke Lemma 12.10 to show that every proof in G_3HOIL corresponds to a proof in G_{HOIL} . Miller et al. (1991) prove that every higher-order hereditary Harrop sequent in G_{HOIL} has a uniform proof. We invoke Lemma 12.10 to obtain a uniform proof in G_3HOIL . ■

We conclude this section with the following result which shows the relationship between permutations of proofs of higher-order hereditary Harrop sequents and permutations of derivations of their encodings.

Lemma 13.22

Let $(\mathcal{A}) \Delta \vdash_{G_3HOIL} \delta : \mathbf{proof}(\phi)$ be a higher-order hereditary Harrop sequent in G_3HOIL . If δ is not in long $\beta\eta$ -normal form then there exists δ' which is in long $\beta\eta$ -normal form such that $(\mathcal{A}) \Delta \vdash_{G_3HOIL} \delta' : \mathbf{proof}(\phi)$ and a permutation from δ to δ' . Let $M_{\delta'}$ be the uniform proof-term which represents δ' . There is a permutation M_δ to $M_{\delta'}$ corresponding to the permutation δ to δ' . Similarly, if M_δ is not a uniform proof-term, then there exists a uniform proof-term $M_{\delta'}$ together with a permutation from M_δ to $M_{\delta'}$. This permutation represents a permutation δ to δ' in G_3HOIL , where δ' is in long $\beta\eta$ -normal form.

Proof (Sketch) The proof is analogous to that of Lemma 13.16. ■

13.3 Representing ALPLs in LF, the Story so Far

We summarize the results on G_3i and G_3HOIL before moving on to discuss multi-conclusioned sequent systems. The reason for this is that the multi-conclusioned systems have a substantially different encoding which requires a more complex definition. We begin with a slightly more general definition of a uniform proof-term.

Definition 13.23 (Uniform Proof-Terms for Single-Concluded Systems)

Let \mathcal{L} be a judged proof system with single-concluded sequent rules, which can be adequately represented in LF. Let M be a term of the $\lambda\Pi_{\Sigma_{\mathcal{L}}}$ calculus in long $\beta\eta$ -normal form. We say that M is a *uniform proof-term for a single-concluded system* if all subterms N , which are not variables, satisfy the following conditions:

- N never has type $j(\#(\phi_1, \dots, \phi_n))$ where $\# \notin \{\top, \wedge, \vee, \supset, \forall_{\sigma}, \exists_{\sigma}\}$;
- N never contains the term *NEG-L*;
- if N has type $j(\top)$ then $N = TOP$;
- if N has type $j(\phi \wedge \psi)$ then $N = P_1 \cdots P_n$ where $P_1 = AND-R$;
- if N has type $j(\phi \vee \psi)$ then $N = P_1 \cdots P_n$ where $P_1 = OR-R$;
- if N has type $j(\phi \supset \psi)$ then $N = P_1 \cdots P_n$ where $P_1 = IMP-R$;
- if N has type $j(\forall_{\sigma}\phi x)$ then $N = P_1 \cdots P_n$ where $P_1 = FORALL-R$;
- if N has type $j(\exists_{\sigma}\phi x)$ then $N = P_1 \cdots P_n$ where $P_1 = EXISTS-R$. ■

This definition assumes that \forall and \exists are special cases of \forall_{σ} and \exists_{σ} . We note that we have made the requirement that M be in long $\beta\eta$ -normal form part of the definition of a uniform proof-term. This definition will not work for multi-concluded sequents. It relies on the type of the (sub)terms being the judged formula in the succedent. This is not true for the encoding of multi-concluded sequents. We have the following lemma.

Lemma 13.24 (Abstract Single-Concluded Logic Programming Language)

Let \mathcal{L} be a judged proof system with single-concluded sequent rules, which can be adequately represented in LF. If for all derivations $\Gamma_{\mathcal{A}}, \Gamma_{\Delta} \vdash_{\Sigma_{\mathcal{L}}} M : j(\phi)$, M is a uniform proof-term or M can be transformed into a uniform proof-term, then \mathcal{L} is an abstract logic programming language.

Proof The definition of a uniform proof-term restricts us to fragments of intuitionistic, minimal and higher-order intuitionistic logics. The result follows from Lemma 13.12 and 13.19. Minimal is treated as a special case of intuitionistic logic. ■

13.4 Representing Uniform Proofs in G_3c in LF

We repeat the analysis of § 13.1 and 13.2 for G_3c . We emphasize that the way G_3c is encoded in LF is different from G_3i and G_3HOIL even though a lot of similar observations hold. We begin by seeing how *Res* represents the valid proof rules of G_3c . Again we ignore the instantiation of variables for clarity. As was

the case for G_3i , we observe that all the left rules are represented by Res as generalized elimination rules. The right rules are not represented as introduction rules as we shall see shortly. They are represented by the elimination rule of their dual connective except for $\supset r$ and $\neg r$. For $\vee r$, we use the version of the rule which is suitable for proof-search; breaking the symmetry in this case.

We begin with $HYP: \mathbf{ant}(\phi) \rightarrow \mathbf{suc}(\phi) \rightarrow \#$ for which we have the resolution step:

$$\frac{\Gamma_X, \Gamma_\Delta, \Gamma_\Theta \vdash_{\Sigma_{G_3c}} y_1 : \mathbf{ant}(\phi) \quad \Gamma_X, \Gamma_\Delta, \Gamma_\Theta \vdash_{\Sigma_{G_3c}} z_1 : \mathbf{ant}(\psi)}{\Gamma_X, \Gamma_\Delta, \Gamma_\Theta \vdash_{\Sigma_{G_3c}} HYP(y_1)z_2 : \#} Res$$

where $y_1 : \mathbf{ant}(\phi) \in \Gamma_\Delta$ and $z_1 : \mathbf{suc}(\psi) \in \Gamma_\Theta$.

We now proceed to examine all of the constants which encode left rules beginning with $AND-L: (\mathbf{ant}(\phi) \rightarrow \mathbf{ant}(\psi) \rightarrow \#) \rightarrow (\mathbf{ant}(\phi \wedge \psi) \rightarrow \#)$. This has the following resolution step:

$$\frac{\Gamma \vdash \lambda y_1 : \mathbf{ant}(\phi) . \lambda y_2 : \mathbf{ant}(\psi) . M_\delta : \mathbf{ant}(\phi) \rightarrow \mathbf{ant}(\psi) \rightarrow \# \quad \Gamma \vdash y : \mathbf{ant}(\phi \wedge \psi)}{\Gamma \vdash AND-L(\lambda y_1 : \mathbf{ant}(\phi) . \lambda y_2 : \mathbf{ant}(\psi) . M_\delta)y : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, y : \mathbf{ant}(\phi \wedge \psi)$. We observe that we have the generalized elimination rule for \wedge .

For the constant $OR-L: (\mathbf{ant}(\phi) \rightarrow \#) \rightarrow (\mathbf{ant}(\psi) \rightarrow \#) \rightarrow (\mathbf{ant}(\phi \vee \psi) \rightarrow \#)$ we have the resolution step:

$$\frac{\Gamma \vdash M : \mathbf{ant}(\phi) \rightarrow \# \quad \Gamma \vdash N : \mathbf{ant}(\psi) \rightarrow \# \quad \Gamma \vdash y : \mathbf{ant}(\phi \vee \psi)}{\Gamma \vdash OR-L(\lambda y_1 : \mathbf{ant}(\phi) . M_{\delta_2})(\lambda y_2 : \mathbf{ant}(\psi) . M_{\delta_2})y : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, y : \mathbf{ant}(\phi \vee \psi)$, $M = \lambda y_1 : \mathbf{ant}(\phi) . M_{\delta_1}$ and $N = \lambda y_2 : \mathbf{ant}(\psi) . M_{\delta_2}$. Again we see that we have the generalized elimination rule for \vee .

For the constant $IMP-L: (\mathbf{suc}(\phi) \rightarrow \#) \rightarrow (\mathbf{ant}(\psi) \rightarrow \#) \rightarrow (\mathbf{ant}(\phi \supset \psi) \rightarrow \#)$ we have the resolution step:

$$\frac{\Gamma \vdash M : \mathbf{suc}(\phi) \rightarrow \# \quad \Gamma \vdash N : \mathbf{ant}(\psi) \rightarrow \# \quad \Gamma \vdash y : \mathbf{ant}(\phi \supset \psi)}{\Gamma \vdash IMP-L(\lambda z_1 : \mathbf{suc}(\phi) . M_{\delta_1})(\lambda y_1 : \mathbf{ant}(\psi) . M_{\delta_2})y : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, y : \mathbf{ant}(\phi \supset \psi)$, $M = \lambda z_1 : \mathbf{suc}(\phi) . M_{\delta_1}$ and $N = \lambda y_1 : \mathbf{ant}(\psi) . M_{\delta_2}$. For the constant $NEG-L: (\mathbf{suc}(\phi) \rightarrow \#) \rightarrow (\mathbf{ant}(\neg\phi) \rightarrow \#)$ we have the resolution step:

$$\frac{\Gamma \vdash \lambda z_1 : \mathbf{suc}(\phi) . M_\delta : \mathbf{suc}(\phi) \rightarrow \# \quad \Gamma \vdash y : \mathbf{ant}(\neg\phi)}{\Gamma \vdash NEG-L(\lambda z_1 : \mathbf{suc}(\phi) . M_\delta)y : \#} Res$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, y : \mathbf{ant}(\neg\phi)$, which is the generalized elimination rule for \neg .

For the constant $FORALL-L: (\text{ant}(\phi x) \rightarrow \#) \rightarrow (\text{ant}(\forall(\lambda x:\iota. \phi x)) \rightarrow \#)$ we have the resolution step:

$$\frac{\Gamma \vdash \lambda y_1 : \text{ant}(\phi x) . M_\delta : \text{ant}(\phi x) \rightarrow \# \quad \Gamma \vdash y : \text{ant}(\forall(\lambda x:\iota. \phi x))}{\Gamma \vdash FORALL-L(\lambda y_1 : \text{ant}(\phi x) . M_\delta)y : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, y : \text{ant}(\forall(\lambda x:\iota. \phi x))$, which is the generalized elimination rule for \forall .

We have the last constant encoding a left rule of G_{3c} , $EXISTS-L: (\Pi x : \iota. \text{ant}(\phi x) \rightarrow \#) \rightarrow (\text{ant}(\forall(\lambda x:\iota. \phi x)) \rightarrow \#)$. We have the resolution step:

$$\frac{\Gamma \vdash \lambda x:\iota. \lambda y_1 : \text{ant}(\phi x) . M_\delta : \Pi x:\iota. \text{ant}(\phi x) \rightarrow \# \quad \Gamma \vdash_{\Sigma_{G_{3c}}} y : \text{ant}(\exists(\lambda x:\iota. \phi x))}{\Gamma \vdash EXISTS-L(\lambda x:\iota. \lambda y_1 : \text{ant}(\phi x) . M_\delta)y : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, y : \text{ant}(\exists(\lambda x:\iota. \phi x))$. The resolution rule is also in the form of the generalized elimination rule for \exists .

We now turn to the representation of the right rules beginning with $AND-R: (\text{suc}(\phi) \rightarrow \#) \rightarrow (\text{suc}(\phi) \rightarrow \#) \rightarrow (\text{suc}(\phi \wedge \psi) \rightarrow \#)$. We have the resolution step:

$$\frac{\Gamma \vdash N : \text{suc}(\phi) \rightarrow \# \quad \Gamma \vdash P : \text{suc}(\psi) \rightarrow \# \quad \Gamma \vdash z : \text{suc}(\phi \wedge \psi)}{\Gamma \vdash AND-R(\lambda z_1 : \text{suc}(\phi) . M_{\delta_1})(\lambda z_2 : \text{suc}(\psi) . M_{\delta_2})z : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, z : \text{suc}(\phi \wedge \psi)$, $N = \lambda z_1 : \text{suc}(\phi) . M_{\delta_1}$ and $P = \lambda z_2 : \text{suc}(\psi) . M_{\delta_2}$. Here the rule is a generalized elimination rule but for \vee . To see this, replace ϕ by $\neg\phi$ and ψ by $\neg\psi$ and $\phi \wedge \psi$ by $\neg(\phi \wedge \psi) = \neg\phi \vee \neg\psi$. This is to be expected because of the symmetry present in G_{3c} .

For the constant $OR-R_i: (\text{suc}(\phi_i) \rightarrow \#) \rightarrow (\text{suc}(\phi_1 \vee \phi_2) \rightarrow \#)$ we have the resolution step

$$\frac{\Gamma \vdash \lambda z_i : \text{suc}(\phi_i) . M_{\delta_i} : \text{suc}(\phi_i) \rightarrow \# \quad \Gamma \vdash z : \text{suc}(\phi_1 \vee \phi_2)}{\Gamma \vdash OR-R_i(\lambda z_i : \text{suc}(\phi_i))z : \#}$$

where $\Gamma_X, \Gamma_\Delta, \Gamma_\Theta, z : \text{suc}(\phi_1 \vee \phi_2)$, which is not a generalized elimination rule. If we were using the symmetric version of this rule then it would be the generalized elimination rule for \wedge provided we replace each formula by its negation.

For the constant $IMP-R: (\text{suc}(\phi) \rightarrow \text{suc}(\psi) \rightarrow \#) \rightarrow (\text{suc}(\phi \supset \psi) \rightarrow \#)$ we have the resolution step:

$$\frac{\Gamma \vdash \lambda z_1 : \text{suc}(\phi) . \lambda z_2 : \text{suc}(\psi) . M_\delta : \text{suc}(\phi) \rightarrow \text{suc}(\psi) \rightarrow \# \quad \Gamma \vdash z : \text{suc}(\phi \supset \psi)}{\Gamma \vdash IMP-R(\lambda z_1 : \text{suc}(\phi) . \lambda z_2 : \text{suc}(\psi) . M_\delta)z : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, z : \text{suc}(\phi \supset \psi)$. We can use weakening to obtain a version of the rule which is in the form of a generalized elimination rule.

For the constant $NEG-R : (\text{ant}(\phi) \rightarrow \#) \rightarrow (\text{suc}(\neg\phi) \rightarrow \#)$ we have the resolution step:

$$\frac{\Gamma \vdash \lambda y_1 : \text{ant}(\phi) . M_\delta : \text{ant}(\phi) \rightarrow \# \quad \Gamma \vdash z : \text{suc}(\neg\phi)}{\Gamma \vdash NEG-R(\lambda y_1 : \text{ant}(\phi) . M_\delta)z : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, z : \text{suc}(\neg\phi)$. Here we have the generalized elimination rule for \neg .

For the constant $FORALL-R : (\Pi x : \iota . \text{suc}(\phi x) \rightarrow \#) \rightarrow (\text{suc}(\forall(\lambda x : \iota . \phi x)) \rightarrow \#)$ we have the resolution step:

$$\frac{\Gamma \vdash \lambda x : \iota . \lambda z_1 : \text{suc}(\phi x) . M_\delta : \Pi x : \iota . \text{suc}(\phi) \rightarrow \# \quad \Gamma \vdash z : \text{suc}(\forall(\lambda x : \iota . \phi x))}{\Gamma \vdash FORALL-R(\lambda x : \iota . \lambda z_1 : \text{suc}(\phi x) . M_\delta)z : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, z : \text{suc}(\forall(\lambda x : \iota . \phi x))$. We have the generalized elimination rule for \exists provided we replace each formula by its negation.

For the constant $EXISTS-R : (\text{suc}(\phi x) \rightarrow \#) \rightarrow (\text{suc}(\exists(\lambda x : \iota . \phi x)) \rightarrow \#)$ we have the resolution step:

$$\frac{\Gamma \vdash \lambda z_1 : \text{suc}(\phi x) . M_\delta : \text{suc}(\phi x) \rightarrow \# \quad \Gamma \vdash z : \text{suc}(\exists(\lambda x : \iota . \phi x))}{\Gamma \vdash EXISTS-R(\lambda z_1 : \text{suc}(\phi x) . M_\delta)z : \#}$$

where $\Gamma = \Gamma_X, \Gamma_\Delta, \Gamma_\Theta, z : \text{suc}(\exists(\lambda x : \iota . \phi x))$. Here we have the generalized elimination rule for \forall provided we replace each formula by its negation.

We recall Examples 12.6 and 12.7 and show how the G_{3c} proofs are represented in NR.

Example 13.25 (Tertium Non Datur)

The proof expression

$$\vdash_{G_{3c}} \langle \rangle \xrightarrow{OR-R((z_2, z_1) : NEG-R(y_1 : HYP(y_1, z_1), z_2), z_3)} z_3 : \text{suc}(\neg\phi \vee \phi)$$

is represented in NR by the assertion

$$\Gamma_X, z_3 : \text{suc}(\neg\phi \vee \neg\phi) \vdash_{\Sigma_{G_{3c}}} OR-R(\lambda z_2 : \text{suc}(\neg\phi) . \lambda z_1 : \text{suc}(\phi) . NEG-R(\lambda y_1 : \text{ant}(\phi) . HYP(y_1)z_1)z_2)z_3 : \#$$

This is derived as follows:

$$\frac{\Gamma_1 \vdash y_1 \text{ant}(\phi) \quad \Gamma_1 \vdash z_1 : \text{suc}(\phi)}{\Gamma_1 \vdash HYP(y_1)z_1 : \#} Res$$

$$\begin{array}{c}
\Gamma_1 \vdash HYP(y_1)z_1 : \# \\
\hline
\Gamma_2 \vdash \lambda y_1 : \text{ant}(\phi) . HYP(y_1)z_1 : \text{ant}(\phi) \rightarrow \# \\
\Gamma_2 \vdash \lambda y_1 : \text{ant}(\phi) . HYP(y_1)z_1 : \text{ant}(\phi) \rightarrow \# \quad \Gamma_2 \vdash z_2 : \text{suc}(\neg\phi) \\
\hline
\Gamma_2 \vdash NEG-R(\lambda y_1 : \text{ant}(\phi) . \text{ant}(\phi) . HYP(y_1)z_1)z_2 : \# \\
\Gamma_2 \vdash NEG-R(\lambda y_1 : \text{ant}(\phi) . \text{ant}(\phi) . HYP(y_1)z_1)z_2 : \# \\
\hline
\Gamma_3 \vdash \lambda z_1 : \text{suc}(\phi) . NEG-R(\lambda y_1 : \text{ant}(\phi) . \text{ant}(\phi) . HYP(y_1)z_1)z_2 : \text{suc}(\phi) \rightarrow \# \\
\Gamma_3 \vdash N : \text{suc}(\phi) \rightarrow \# \\
\hline
\Gamma_4 \vdash \lambda z_2 : \text{suc}(\neg\phi) . N : \text{suc}(\neg\phi) \rightarrow \text{suc}(\phi) \rightarrow \# \\
\Gamma_4 \vdash \lambda z_2 : \text{suc}(\neg\phi) . N : \text{suc}(\neg\phi) \rightarrow \text{suc}(\phi) \rightarrow \# \quad \Gamma_4 \vdash z_3 : \text{suc}(\neg\phi \vee \phi) \\
\hline
\Gamma_4 \vdash OR-R(\lambda z_2 : \text{suc}(\neg\phi) . N)z_3 : \#
\end{array}$$

where $\Gamma_1 = \Gamma_X, y_1 : \text{ant}(\phi), z_1 : \text{suc}(\phi), z_2 : \text{suc}(\neg\phi), z_3 : \text{suc}(\neg\phi \vee \phi), \Gamma_2 = \Gamma_X, z_1 : \text{suc}(\phi), z_2 : \text{suc}(\neg\phi), z_3 : \text{suc}(\neg\phi \vee \phi), \Gamma_3 = \Gamma_X, z_2 : \text{suc}(\neg\phi), z_3 : \text{suc}(\neg\phi \vee \phi)$ and $\Gamma_4 = \Gamma_X, z_3 : \text{suc}(\neg\phi \vee \phi), N = \lambda z_1 : \text{suc}(\phi) . NEG-R(\lambda y_1 : \text{ant}(\phi) . \text{ant}(\phi) . HYP(y_1)z_1)z_2$. ■

Example 13.26 (DeMorgan)

The proof expression

$$\vdash_{G_{3c}} y_1 : \text{ant}(\neg(\forall x . \phi x)) \xrightarrow{\delta} z_4 : \text{suc}(\exists x . (\neg\phi x))$$

where $\delta = NEG-L(z_3 : FORALL-L(z_1 : EXISTS-R(z_2 : NEG-R(y_2 : HYP(y_2, z_1)z_2), z_4), z_3), y_1)$ is encoded in NR as the assertion

$$\begin{array}{c}
\Gamma_X, y_1 : \text{ant}(\neg(\forall(\lambda x : \iota . \phi x))), z_4 : \text{suc}(\exists(\lambda x : \iota . (\neg\phi x))) \vdash_{\Sigma_{G_{3c}}} NEG-L\epsilon_s(\forall x . \phi x) \\
(\lambda z_3 : \text{suc}(\forall(\lambda x : \iota . \phi x)) . FORALL-R\epsilon_s(\phi x)\epsilon_s(x)(\lambda z_1 : \text{suc}(\phi x) . EXISTS-R\epsilon_s \\
(\neg\phi x)\epsilon_s(x)(\lambda z_2 : \text{suc}(\neg\phi x) . NEG-R\epsilon_s(\phi x)(\lambda y_2 : \text{ant}(\phi x)HYP\epsilon_s(\phi x)(y_2)z_1)z_2)z_4) \\
z_3)y_1 : \#
\end{array}$$

This is derived as follows:

$$\begin{array}{c}
\Gamma_1 \vdash y_2 : \text{ant}(\phi x) \quad \Gamma_1 \vdash z_1 : \text{suc}(\phi x) \\
\hline
\Gamma_1 \vdash HYP(y_2)z_1 : \# \\
\Gamma_1 \vdash HYP(y_2)z_1 : \# \\
\hline
\Gamma_2 \vdash \lambda y_2 : \text{ant}(\phi x) . HYP(y_2)z_1 : \text{ant}(\phi x) \rightarrow \# \\
\Gamma_2 \vdash \lambda y_2 : \text{ant}(\phi x) . HYP(y_2)z_1 : \text{ant}(\phi x) \rightarrow \# \quad \Gamma_2 \vdash z_2 : \text{suc}(\neg\phi x) \\
\hline
\Gamma_2 \vdash NEG-R(\lambda y_2 : \text{ant}(\phi x) . HYP(y_2)z_1)z_2 : \#
\end{array}$$

$$\begin{array}{c}
\Gamma_2 \vdash M_{\delta_1} : \# \\
\hline
\Gamma_3 \vdash \lambda z_2 : \text{succ}(\neg\phi x) . M_{\delta_1} : \text{succ}(\neg\phi x) \rightarrow \# \\
\Gamma_3 \vdash \lambda z_2 : \text{succ}(\neg\phi x) . M_{\delta_1} : \text{succ}(\neg\phi x) \rightarrow \# \quad \Gamma_3 \vdash z_4 : \text{succ}(\exists(\lambda x : \iota . \phi x)) \\
\hline
\Gamma_3 \vdash \text{EXISTS-R}(\lambda z_2 : \text{succ}(\neg\phi x) . M_{\delta_1}) z_4 : \# \\
\Gamma_3 \vdash \text{EXISTS-R}(\lambda z_2 : \text{succ}(\neg\phi x) . M_{\delta_1}) z_4 : \# \\
\hline
\Gamma_4 \vdash \lambda z_1 : \text{succ}(\phi x) . \text{EXISTS-R}(\lambda z_2 : \text{succ}(\neg\phi x) . M_{\delta_1}) z_4 : \text{succ}(\phi x) \rightarrow \# \\
\Gamma_4 \vdash M_{\delta_2} : \text{succ}(\phi x) \rightarrow \# \quad \Gamma_4 \vdash z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x)) \\
\hline
\Gamma_4 \vdash \text{FORALL-R}(M_{\delta_2}) z_3 : \# \\
\Gamma_4 \vdash \text{FORALL-R}(M_{\delta_2}) z_3 \\
\hline
\Gamma_5 \vdash \lambda z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x)) . \text{FORALL-R}(M_{\delta_2}) z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x)) \rightarrow \# \\
\Gamma_5 \vdash M_{\delta_3} : \text{succ}(\forall(\lambda x : \iota . \phi x)) \rightarrow \# \quad \Gamma_5 \vdash y_1 : \text{ant}(\neg(\forall(\lambda x : \iota . \phi x))) \\
\hline
\Gamma_5 \vdash \text{NEG-L}(M_{\delta_3}) y_1
\end{array}$$

where $\Gamma_1 = \Gamma_X, y_1 : \text{ant}(\neg(\forall(\lambda x : \iota . \phi x)))$, $y_2 : \text{ant}(\phi x)$, $z_1 : \text{succ}(\phi x)$, $z_2 : \text{succ}(\neg\phi x)$, $z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x))$, $z_4 : \text{succ}(\exists(\forall(\lambda x : \iota . (\neg\phi x))))$, $\Gamma_2 = \Gamma_X, y_1 : \text{ant}(\neg(\forall(\lambda x : \iota . \phi x)))$, $z_1 : \text{succ}(\phi x)$, $z_2 : \text{succ}(\neg\phi x)$, $z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x))$, $z_4 : \text{succ}(\exists(\forall(\lambda x : \iota . (\neg\phi x))))$, $\Gamma_3 = \Gamma_X, y_1 : \text{ant}(\neg(\forall(\lambda x : \iota . \phi x)))$, $z_1 : \text{succ}(\phi x)$, $z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x))$, $z_4 : \text{succ}(\exists(\forall(\lambda x : \iota . (\neg\phi x))))$, $\Gamma_4 = \Gamma_X, y_1 : \text{ant}(\neg(\forall(\lambda x : \iota . \phi x)))$, $z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x))$, $z_4 : \text{succ}(\exists(\forall(\lambda x : \iota . (\neg\phi x))))$, $\Gamma_5 = \Gamma_X, y_1 : \text{ant}(\neg(\forall(\lambda x : \iota . \phi x)))$, $z_4 : \text{succ}(\exists(\forall(\lambda x : \iota . (\neg\phi x))))$, $M_{\delta_1} = \text{NEG-R}(\lambda y_2 : \text{ant}(\phi x) . \text{HYP}(y_2) z_1) z_2$, $M_{\delta_2} = \lambda z_1 : \text{succ}(\phi x) . \text{EXISTS-R}(\lambda z_2 : \text{succ}(\neg\phi x) . M_{\delta_1}) z_4$ and $M_{\delta_3} = \lambda z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x)) . \text{FORALL-R}(M_{\delta_2}) z_3$. \blacksquare

We now look for a condition on the terms of $\text{NR}_{\Sigma_{G_3c}}$ in long $\beta\eta$ -normal form which ensures that they represent uniform proofs in G_3c . We have to be careful here with the notion of uniform proof because we are using the logic G_3c where every right rule has multiple-conclusioned premisses. The definition of uniform proof given by Miller et al. (1991) requires us to work with single-conclusions. A careful analysis of the way the extra formulæ are used in G_3c allows us to reconstruct their notion of uniform proof for our setting. The rules of G_3c require the principal formula to be present in the premisses. We ignore any formulæ which are not required by the rule and any principal formulæ in the premisses and then use the same notion of uniform proof as Miller *et al.* Only sequents of the form $\Gamma \rightarrow \phi$ can have uniform proofs. This condition is the same as Miller *et al.* We now need to adapt the conditions on the right rules so that they take into account the extra formulæ. The following idea is crucial here, working from the bottom up: we begin with a single-conclusioned sequent and if the formulæ on the right contain any logical connectives it must be the conclusion of the right rule of the principal connective, otherwise we apply a suitable left rule. We apply the same

analysis to the premisses of the rule but only to the new formulæ. We show how to derive a uniform proof for the sequent

$$\forall x\phi \rightarrow \phi \vee \psi$$

This has to be the conclusion of $\vee r$.

$$\frac{\forall x\phi \rightarrow \phi, \phi \vee \psi}{\forall x\phi \rightarrow \phi \vee \psi} \vee r$$

where ϕ is the new formula in the succedent. ϕ is atomic so we have to apply a left rule. We conclude it must be the conclusion of $\forall l$.

$$\frac{\forall x\phi, \phi \rightarrow \phi, \phi \vee \psi}{\forall x\phi \rightarrow \phi, \phi \vee \psi} \forall r$$

where ϕ is the new formula introduced on the left. We now have an axiom, so we are finished. The uniform proof is

$$\frac{\frac{\frac{}{Ax}}{\forall x\phi, \phi \rightarrow \phi, \phi \vee \psi} \forall r}{\forall x\phi \rightarrow \phi, \phi \vee \psi} \forall r}{\forall x\phi \rightarrow \phi \vee \psi} \vee r$$

The key here is that we only concentrate on new formulæ on the right of the sequent. This means that we are treating G_3c as if it were the sequent system LK , (Gentzen 1934). Hence the notion of uniform proof can be used in G_3c . We only require that the notion of uniform proof in Miller *et al.* hold for the principal formula on the right. Any other formulæ on the right are occurrences of formulæ introduced by right rules, working from the bottom upwards. We can rewrite the rules of Miller *et al.* for our setting and define uniform proof for G_3c thus:

Definition 13.27 (Uniform Proof for G_3c)

A proof of a sequent $\Delta \rightarrow \phi$ in G_3c is *uniform* if every occurrence of a sequent of the form $\Delta \rightarrow G, \Theta$ where G is the principal formula, satisfies the following conditions:

- If G is \top then the sequent is initial.
- If G is $\phi \wedge \psi$ then that sequent is inferred by $\wedge R$ from $\Delta \rightarrow \phi, \phi \wedge \psi, \Theta$ and $\Delta \rightarrow \psi, \phi \wedge \psi, \Theta$.
- If G is $\phi \vee \psi$ then that sequent is inferred by $\vee R$ from either $\Delta \rightarrow \phi, \phi \vee \psi, \Theta$ or $\Delta \rightarrow \psi, \phi \vee \psi, \Theta$.
- If G is $\phi \supset \psi$ then the sequent is inferred by $\supset R$ from $\Delta, \phi \rightarrow \psi, \phi \supset \psi, \Theta$.

- If G is $\forall x\phi$ then the sequent is inferred by $\forall R$ from $\Delta \rightarrow [c/x]\phi, \forall x\phi, \Theta$, where c is a parameter that does not occur in the given sequent.
- If G is $\exists x\phi$ then the sequent is inferred by $\exists R$ from $\Delta \rightarrow [t/x]\phi, \exists\phi, \Theta$ for some term t .
- G is never $\neg\phi$.
- No $\neg l$ rules are used.
- Each formula in Δ is used in at most one left rule. ■

Since we are starting from a sequent of the form $\Delta \vdash \phi$, there is at most one principal formula on the right all through the proof, so the above definition is well-defined. If we had not replaced $\forall R$ by the two equivalent rules we would have had two principal formulæ which would have meant the above definition was not well-defined. The last condition ensures that left rules are used correctly. In G_3c the formula introduced by the left rule are present throughout the rule and we do not want them reused.

We begin by looking at some examples of uniform and non-uniform proofs in G_3c .

Example 13.28 (Tertium Non Datur)

We recall Examples 12.6 and 13.25. The proof of the law of excluded middle is uniform since it does not involve any left rules. We have to ensure that any condition we put on the terms of $\mathbf{NR}_{\Sigma_{G_3c}}$ in long $\beta\eta$ normal form covers this special case. ■

Example 13.29 (DeMorgan)

We recall Examples 12.7 and 13.26. The proof used in these examples is non-uniform since it involves negation. The term representing this non-uniform proof is

$$\begin{aligned} & \text{NEG-L}\epsilon_s(\forall x . \phi x)(\lambda z_3 : \text{succ}(\forall(\lambda x : \iota . \phi x))) . \text{FORALL-R} \\ & \epsilon_s(\phi x)\epsilon_s(x)(\lambda z_1 : \text{succ}(\phi x)) . \text{EXISTS-R}\epsilon_s(\neg\phi x)\epsilon_s(x)(\lambda z_2 : \text{succ}(\neg\phi x)) . \\ & \text{NEG-R}\epsilon_s(\phi x)(\lambda y_2 : \text{ant}(\phi x)\text{HYP}\epsilon_s(\phi x)(y_2)z_1)z_2)z_4)z_3)y_1 \end{aligned} \quad \blacksquare$$

Example 13.30

The following proof is uniform.

$$\frac{\frac{\frac{\frac{}{Ax}}{(\forall x\phi x) \supset \psi, \phi x \rightarrow \phi x, \forall x\phi x, \psi \vee \tau}}{\phi x, (\forall x\phi x) \supset \psi \rightarrow \forall x\phi x, \psi \vee \tau}}{\phi x, (\forall x\phi x) \supset \psi \rightarrow \psi, \psi \vee \tau} \forall r}{\phi x, (\forall x\phi x) \supset \psi \rightarrow \psi, \psi \vee \tau} \supset l}{\frac{\phi x, (\forall x\phi x) \supset \psi \rightarrow \psi, \psi \vee \tau}{\phi x, (\forall x\phi x) \supset \psi \rightarrow \psi \vee \tau} \vee r} \supset l$$

which has valid proof expression

$$Xy_1 : \mathbf{ant}(\phi x), y_3 : \mathbf{ant}((\forall x \phi x) \supset \psi)$$

$$\frac{OR-R1(z_3 : IMP-L(z_2 : FORALL-R(z_1 : HYP(y_1, z_1), z_2), y_2 : HYP(y_2, z_3), y_3), z_4))}{}$$

$$z_4 : \mathbf{suc}(\psi \vee \tau)$$

where $y_2 : \mathbf{ant}(\psi)$, $z_1 : \mathbf{suc}(\phi x)$, $z_2 : \mathbf{suc}(\forall x \phi x)$, $z_3 : \mathbf{suc}(\psi)$. This assertion is represented in $\mathbf{NR}_{\Sigma G_{3c}}$ by the assertion

$$\Gamma_X, y_1 : \mathbf{ant}(\phi x), y_3 : \mathbf{ant}((\forall x \phi x) \supset \psi) \vdash_{\Sigma G_{3c}} OR-R\epsilon_s(\psi)\epsilon_s(\tau)(\lambda z_3 : \mathbf{suc}(\psi) .$$

$$IMP-L\epsilon_s(\forall x \phi x)\epsilon_s(\psi)(\lambda z_2 : \mathbf{suc}((\forall x (\lambda x : \iota . \phi x)) \supset \psi) . FORALL-R\epsilon_s(x)\epsilon_s(\phi x)$$

$$(\lambda z_1 : \mathbf{suc}(\phi x) . HYP(y_1)z_1)z_2)(\lambda y_2 : \mathbf{ant}(\psi) . HYP(y_2)z_3)y_3)z_4 : \# \quad \blacksquare$$

We now define a condition on the terms of $\mathbf{NR}_{\Sigma G_{3c}}$ in long $\beta\eta$ -normal form so that they represent uniform proofs in G_{3c} .

Definition 13.31 (Uniform Proof-terms in $\mathbf{NR}_{\Sigma G_{3c}}$)

Let $\Gamma_X, \Gamma_\Delta, z : \mathbf{suc}(\phi) \vdash_{\Sigma G_{3c}} M_\delta : \#$, where M_δ is in long $\beta\eta$ -normal form, be a derivable assertion in $\mathbf{NR}_{\Sigma G_{3c}}$. We say that M_δ is a *uniform proof-term* if all subterms of the form $My : \#$ typed in a context $\Gamma_X, \Gamma_\Delta, \Gamma_\Theta$ satisfy the following conditions:

- if $y : \mathbf{ant}(@(\phi_1, \dots, \phi_n))$, where $@ \in \{\top, \wedge, \vee, \supset, \forall_\sigma, \exists_\sigma\}$, then Γ_Θ contains $z : \mathbf{suc}(\psi)$, where ψ is atomic;
- if $y : \mathbf{suc}(\phi)$, then $\phi = @(\phi_1, \dots, \phi_n)$ where $@ \in \{\wedge, \vee, \supset, \forall, \exists\}$. \blacksquare

The above definition is very different from one given for $\mathbf{NR}_{\Sigma G_{3i}}$. That encoding relied on the type of each subterm corresponding to the judged formula in the succedent. Here the succedent in sequents in G_{3c} is encoded as a context. We thus have to examine the variables used in a term to determine when a left rule has been applied and then determine if the succedent was atomic. The second condition ensures that all the right rules were applied to non-atomic formulæ.

Lemma 13.32 (Uniform Proof-terms Represent Uniform Proofs)

Let $\vdash_{G_{3c}} (X) \Delta \xrightarrow{\delta} z : \mathbf{suc}(\phi)$ be a provable assertion in G_{3c} and $\Gamma_X, \Gamma_\Delta, z : \mathbf{suc}(\phi) \vdash_{\Sigma G_{3c}} M_\delta : \#$, where M_δ is in long $\beta\eta$ normal form, be its representation in $\mathbf{NR}_{\Sigma G_{3c}}$. δ is in long $\beta\eta$ normal form if and only if M_δ is a uniform proof-term.

Proof We show the right to left direction first. Let $(X) \Delta \xrightarrow{\delta} \mathbf{suc}(\phi)$ be a provable assertion in G_{3c} and δ be a uniform proof. Since δ is a uniform proof-term, working from the bottom upwards, any left rule is only applied when the

principal formula in the succedent is atomic. Thus any use of a left rule in a proof-object δ corresponds to a subterm of the form My where $y : \mathbf{ant}(\phi)$, provided ϕ is not equal to $\neg\tau$. This is typed in the context $\Gamma_X, \Gamma_\Delta, \Gamma_\Theta$, where Γ_Θ contains a variable $z : \mathbf{suc}(\psi)$, where ψ is atomic.

Working from right to left, we observe that the condition that $z : \mathbf{suc}(\psi) \in \Gamma_\Theta$, where ψ is atomic is enough to ensure that the principal formula of the succedent is atomic. Hence, working upwards, left rules are only applied in δ when the principal formula in the succedent is atomic and thus δ is in long $\beta\eta$ -normal form. ■

With G_3i we saw that every hereditary Harrop sequent had a uniform proof. There is a similar class of sequents for G_3c , the Horn sequents. We have the following definition.

Definition 13.33 (Horn Sequent (Miller et al. 1991))

A provable sequent $\Delta \rightarrow G$ in G_3c is a *Horn sequent* if $\Delta = D_1, \dots, D_n$ and D_i and G are defined by the following grammars:

$$\begin{aligned} D & := A \mid G \supset A \mid D_1 \wedge D_2 \mid \forall x D \\ G & := A \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid \exists x D \end{aligned}$$

where A is an atomic formula. Formulæ defined by D are called *Horn clauses*. ■

The following lemma clarifies the relationship between uniform proof-terms and the assertions of $\mathbf{NR}_{\Sigma_{G_3c}}$ which represent Horn sequents.

Lemma 13.34 (Representations of Horn Sequents have a Uniform Proof Term)

Let $\Delta \xrightarrow{G}$ be a provable Horn sequent in G_3c . Let $\vdash_{G_3c} (X) \Delta \xrightarrow{\delta} z : \mathbf{suc}(G)$ be its valid proof expression. There exists a uniform proof-term M such that $\Gamma_X, \Gamma_\Delta, z : \mathbf{suc}(\phi) \vdash M : \#$.

Proof The proof that every provable Horn sequent in G_3c has a uniform proof can be found in (Miller et al. 1991). It is important to realise that it must be adapted slightly to G_3c because G_3c does not allow introduction of formulæ on the right, only their removal. The trick lies in keeping track of the principal formula in each rule. We obtain the uniform proof-term M'_δ as follows. We take δ' which is in long $\beta\eta$ normal form and has valid proof expression $\vdash_{G_3c} (X) \Delta \xrightarrow{\delta'} z : \mathbf{suc}(G)$ and represent it in $\mathbf{NR}_{\Sigma_{G_3c}}$ as $\Gamma_X, \Gamma_\Delta, z : \mathbf{suc}(G) \vdash_{\Sigma_{G_3c}} M'_\delta : \#$ where M'_δ is a uniform proof-term. ■

We provide an example of a Horn sequent and show that the proof permutation in G_3c corresponds to a permutation in $\mathbf{NR}_{\Sigma_{G_3c}}$.

Example 13.35

The following proof is not uniform:

$$\begin{array}{c}
\frac{}{(\exists x\psi x) \supset \phi, \psi x, \phi \rightarrow \phi, \phi \wedge \psi x, (\phi \wedge \psi x) \vee \tau} Ax \\
\\
\frac{}{(\exists x\psi x) \supset \phi, \phi, \psi x \rightarrow \psi x, \phi \wedge \psi x, \phi \wedge \psi x, \phi \wedge \psi x, (\phi \wedge \psi x) \vee \tau} Ax \\
\frac{M \quad N}{(\exists x\psi x) \supset \phi, \phi, \psi x \rightarrow \phi \wedge \psi x, (\phi \wedge \psi x) \vee \tau} \wedge r \\
\frac{(\exists x\psi x) \supset \phi, \phi, \psi x \rightarrow \phi \wedge \psi x, (\phi \wedge \psi x) \vee \tau}{(\exists x\psi x) \supset \phi, \psi x, \phi \rightarrow (\phi \wedge \psi x) \vee \tau} \vee r_1 \\
\frac{}{(\exists x\psi x) \supset \phi, \psi x \rightarrow \psi x, \exists x\phi x, (\phi \wedge \psi x) \vee \tau} Ax \\
\frac{(\exists x\psi x) \supset \phi, \psi x \rightarrow \psi x, \exists x\phi x, (\phi \wedge \psi x) \vee \tau}{(\exists x\psi x) \supset \phi, \psi x \rightarrow \exists x\phi x, (\phi \wedge \psi x) \vee \tau} \exists r \\
\frac{(\exists x\psi x) \supset \phi, \psi x \rightarrow \exists x\phi x, (\phi \wedge \psi x) \vee \tau \quad P}{(\exists x\psi x) \supset \phi, \psi x \rightarrow (\phi \wedge \psi x) \vee \tau} \supset l
\end{array}$$

where $M = (\exists x\psi x) \supset \phi, \psi x, \phi \rightarrow \phi, \phi \wedge \psi x, (\phi \wedge \psi x) \vee \tau$, $N = (\exists x\psi x) \supset \phi, \phi, \psi x \rightarrow \psi x, \phi \wedge \psi x, \phi \wedge \psi x, \phi \wedge \psi x, (\phi \wedge \psi x) \vee \tau$ and $P = (\exists x\psi x) \supset \phi, \phi, \psi x \rightarrow \phi \wedge \psi x, (\phi \wedge \psi x) \vee \tau$. It has the valid proof expression:

$$\vdash_{G_{3c}} (X) y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}((\exists x\psi x) \supset \phi) \xrightarrow{\delta} z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau)$$

where $y_2 : \mathbf{ant}(\phi)$, $z_1 : \mathbf{suc}(\psi x)$, $z_2 : \mathbf{suc}(\phi)$, $z_3 : \mathbf{suc}(\exists x\psi x)$, $z_4 : \mathbf{suc}(\phi \wedge \psi x)$ and $\delta = IMP-L(z_3 : EXISTS-R(z_1 : HYP(y_1, z_1), z_3), y_2 : OR-R1(z_4 : AND-R(z_2 : HYP(y_2, z_2), z_1 : HYP(y_1, z_1), z_4), z_5), y_3))$. This is represented in $\mathbf{NR}_{\Sigma_{G_{3c}}}$ by the assertion

$$\begin{aligned}
& \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x) \supset \phi), z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau) \vdash_{\Sigma_{G_{3c}}} IMP-L\epsilon_s \\
& (\exists(\lambda x : \iota. \psi x)\epsilon_s(\phi)(\lambda z_3 : \mathbf{suc}(\exists(\lambda x : \iota. \psi x) . EXISTS-R\epsilon_s(\psi x)(\lambda x : \iota. \lambda z_1 : \mathbf{suc}(\psi x) . \\
& HYP\epsilon_s(\psi x)(y_1)z_1)z_3)(\lambda y_2 : \mathbf{ant}(\phi) . OR-R1\epsilon_s(\phi \wedge \psi x)\epsilon_s(\tau)(\lambda z_4 : \mathbf{suc}(\phi \wedge \psi x) . \\
& AND-R\epsilon_s(\phi)\epsilon_s(\psi x)(\lambda z_2 : \mathbf{suc}(\phi) . HYP\epsilon_s(\phi)(y_2)z_2)(\lambda z_1 : \mathbf{suc}(\psi x) . HYP\epsilon_s(\psi x) \\
& (y_1)z_1)z_4)z_5)y_3 : \#
\end{aligned}$$

which has the following derivation. We do the derivation in stages due to the

size.

$$\begin{array}{c}
\frac{\Gamma_1 \vdash y_1 : \mathbf{ant}(\psi x) \quad \Gamma_1 \vdash z_2 : \mathbf{suc}(\psi x)}{\Gamma_1 \vdash HYP(y_1)z_1 : \#} \text{Res} \\
\Gamma_1 \vdash HYP(y_1)z_1 : \# \\
\hline
\Gamma_2 \vdash \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1 : \mathbf{suc}(\psi x) \rightarrow \# \\
\Gamma_2 \vdash \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1 : \mathbf{suc}(\psi x) \rightarrow \# \\
\hline
\Gamma_3 \vdash \lambda x : \iota . \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1 : \Pi x : \iota . \mathbf{suc}(\psi x) \rightarrow \# \\
\Gamma_3 \vdash M : \Pi x : \iota . \mathbf{suc}(\psi x) \rightarrow \# \quad \Gamma_3 \vdash z_3 : \mathbf{suc}(\exists(\lambda x : \iota . \psi x)) \\
\hline
\Gamma_3 \vdash EXISTS-R(M)z_3 : \#
\end{array}$$

where $\Gamma_1 = \Gamma_X, x : \iota, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}(\exists(\lambda x : \iota . \psi x)), z_1 : \mathbf{suc}(\psi x), z_3 : \mathbf{suc}(\exists(\lambda x : \iota . \psi x)), z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau)$, $\Gamma_2 = \Gamma_X, x : \iota, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}(\exists(\lambda x : \iota . \psi x)), z_3 : \mathbf{suc}(\exists(\lambda x : \iota . \psi x)), z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau)$ and $\Gamma_3 = \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}(\exists(\lambda x : \iota . \psi x)), z_3 : \mathbf{suc}(\exists(\lambda x : \iota . \psi x)), z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau)$ and $M = \lambda x : \iota . \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1$.

$$\begin{array}{c}
\frac{\Gamma_4 \vdash y_2 : \mathbf{ant}(\phi) \quad \Gamma_4 \vdash z_2 : \mathbf{suc}(\phi)}{\Gamma_4 \vdash HYP(y_2)z_2 : \#} \\
\Gamma_4 \vdash HYP(y_2)z_2 : \# \\
\hline
\Gamma_5 \vdash \lambda z_2 : \mathbf{suc}(\phi) . HYP(y_2)z_2 : \mathbf{suc}(\phi) \rightarrow \# \\
\frac{\Gamma_8 \vdash y_1 : \mathbf{ant}(\psi x) \quad \Gamma_8 \vdash z_1 : \mathbf{suc}(\psi x)}{\Gamma_8 \vdash HYP(y_1)z_1 : \#} \\
\Gamma_8 \vdash HYP(y_1)z_1 : \# \\
\hline
\Gamma_5 \vdash \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1 : \mathbf{suc}(\psi x) \rightarrow \# \\
\Gamma_5 \vdash P : \mathbf{suc}(\phi) \rightarrow \# \quad \Gamma_5 \vdash Q : \mathbf{suc}(\psi x) \rightarrow \# \quad \Gamma_5 \vdash z_4 : \mathbf{suc}(\phi \wedge \psi x) \\
\hline
\Gamma_5 \vdash AND-R(P)(Q)z_4 : \#
\end{array}$$

where $\Gamma_4 = \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}(\psi x), z_2 : \mathbf{suc}(\phi), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau)$, $\Gamma_5 = \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}(\psi x), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau)$, $\Gamma_8 = \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}(\psi x), z_1 : \mathbf{suc}(\psi x), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau)$, $P = \lambda z_2 : \mathbf{suc}(\phi) . HYP(y_2)z_2$ and $Q = \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1$.

$$\begin{array}{c}
\Gamma_5 \vdash M_{\delta_2} : \# \\
\hline
\Gamma_6 \vdash \lambda z_4 : \mathbf{suc}(\phi \wedge \psi x) . M_{\delta_2} : \mathbf{suc}(\phi \wedge \psi x) \rightarrow \# \\
\Gamma_6 \vdash \lambda z_4 : \mathbf{suc}(\phi \wedge \psi x) . M_{\delta_2} : \mathbf{suc}(\phi \wedge \psi x) \rightarrow \# \quad \Gamma_6 \vdash z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau) \\
\hline
\Gamma_6 \vdash OR-R1(\lambda z_4 : \mathbf{suc}(\phi \wedge \psi x) . M_{\delta_2})z_5 : \#
\end{array}$$

where $M_{\delta_2} = AND-R\epsilon_s(\phi)\epsilon_s(\psi x)(\lambda z_2 : \mathbf{suc}(\phi) . HYP\epsilon_s(\phi)(y_2)z_2)(\lambda z_1 : \mathbf{suc}(\psi x) .$

where $y_2 : \mathbf{ant}(\phi)$, $z_1 : \mathbf{suc}(\psi x)$, $z_2 : \mathbf{suc}(\phi)$, $z_3 : \mathbf{suc}(\exists x \psi x)$, $z_4 : \mathbf{suc}(\phi \wedge \psi x)$ and $\delta = OR-R1(z_4 : AND-R(z_2 : IMP-L(z_3 : EXISTS-R(z_1 : HYP(y_1, z_1), z_3), y_2 : HYP(y_2, z_2), y_3), z_1 : HYP(y_1, z_1), z_4), z_5))$. This is represented in $\mathbf{NR}_{\Sigma_{G_3c}}$ by the assertion

$$\begin{aligned} & \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x) \supset \phi), z_5 : \mathbf{suc}((\phi \wedge \psi x) \supset \tau) \vdash_{\Sigma_{G_3c}} OR-R1 \\ & \quad \epsilon_s(\phi \wedge \psi x)\epsilon_s(\tau)(\lambda z_4 : \mathbf{suc}(\phi \wedge \psi x) . AND-R\epsilon_s(\phi)\epsilon_s(\psi x)(\lambda z_2 : \mathbf{suc}(\phi) . IMP-L \\ & \quad \quad \epsilon_s(\exists(\lambda x : \iota. \psi x)\epsilon_s(\phi)(\lambda z_3 : \mathbf{suc}(\exists(\lambda x : \iota. \psi x) . EXISTS-R\epsilon_s(\psi x)(\lambda x : \iota. \\ & \quad \lambda z_1 : HYP\epsilon_s(\psi x)(y_1)z_1)z_3)(\lambda y_2 : \mathbf{ant}(\phi) . HYP\epsilon_s(\phi)(y_2)z_2)y_3)(\lambda z_1 : HYP\epsilon_s(\psi x) \\ & \quad \quad (y_1)z_1)z_4)z_5 \end{aligned}$$

which has a uniform proof-term. This assertion has the following derivation. Again we do it in parts due to the size.

$$\begin{aligned} & \frac{\Gamma \vdash y_1 : \mathbf{ant}(\psi x) \quad \Gamma_1 \vdash z_1 : \mathbf{suc}(\psi x)}{\Gamma_1 \vdash HYP(y_1)z_1 : \#} \\ & \frac{\Gamma_1 \vdash HYP(y_1)z_1 : \#}{\Gamma_2 \vdash \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1 : \mathbf{suc}(\psi x) \rightarrow \#} \\ & \frac{\Gamma_2 \vdash \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1 : \mathbf{suc}(\psi x) \rightarrow \#}{\Gamma_3 \vdash \lambda x : \iota. \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1 : \prod x : \iota. \mathbf{suc}(\psi x) \rightarrow \#} \\ & \frac{\Gamma_3 \vdash P : \mathbf{suc}(\psi x) \rightarrow \# \quad \Gamma_3 \vdash z_3 : \mathbf{suc}(\exists(\lambda x : \iota. \psi x))}{\Gamma_3 \vdash EXISTS-R(P)z_3 : \#} \end{aligned}$$

where $\Gamma_1 = \Gamma_X, x : \iota, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x)) \supset \phi), z_1 : \mathbf{suc}(\psi x), z_2 : \mathbf{suc}(\phi), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi) \vee \tau)$, $\Gamma_2 = \Gamma_X, x : \iota, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x)) \supset \phi), z_2 : \mathbf{suc}(\phi), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi) \vee \tau)$, $\Gamma_3 = \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x)) \supset \phi), z_2 : \mathbf{suc}(\phi), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi) \vee \tau)$ and $P = \lambda x : \iota. \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1$.

$$\begin{aligned} & \frac{\Gamma_4 \vdash \lambda y_2 : \mathbf{ant}(\phi) \quad \Gamma_4 \vdash z_2 : \mathbf{suc}(\phi)}{\Gamma_4 \vdash HYP(y_2)z_2 : \#} \\ & \frac{\Gamma_4 \vdash HYP(y_2)z_2 : \#}{\Gamma_3 \vdash \lambda y_2 : \mathbf{ant}(\phi) . HYP(y_2)z_2 : \mathbf{ant}(\phi) \rightarrow \#} \\ & \frac{\Gamma_3 \vdash M_{\delta_1} : \mathbf{suc}(\exists(\lambda x : \iota. \psi x)) \rightarrow \# \quad \Gamma_3 \vdash Q \quad \Gamma_3 \vdash y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x)) \supset \phi)}{\Gamma_3 \vdash IMP-L(M_{\delta_1})(Q)y_3 : \#} \end{aligned}$$

where $M_{\delta_1} = EXISTS-R\epsilon_s(\psi x)(\lambda x : \iota. \lambda z_2 : \mathbf{suc}(\psi x) . HYP\epsilon_s(\psi x)(y_1)z_1)z_3$ and

$\Gamma_4 = \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_1 : \mathbf{ant}(\phi), y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x)) \supset \phi), z_2 : \mathbf{suc}(\phi), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi) \vee \tau)$ and $Q = \lambda y_2 : \mathbf{ant}(\phi) . HYP(y_2)z_2$.

$$\begin{array}{c}
\Gamma_3 \vdash M_{\delta_2} : \# \\
\hline
\Gamma_5 \vdash \lambda z_1 : \mathbf{suc}(\phi) . M_{\delta_2} : \mathbf{suc}(\phi) \rightarrow \# \\
\Gamma_6 \vdash y_1 : \mathbf{ant}(\psi x) \quad \Gamma_6 \vdash z_1 : \mathbf{suc}(\psi x) \\
\hline
\Gamma_6 \vdash HYP(y_1)z_1 : \# \\
\Gamma_6 \vdash HYP(y_1)z_1 : \# \\
\hline
\Gamma_5 \vdash \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1 : \mathbf{suc}(\psi x) \rightarrow \# \\
\Gamma_5 \vdash \lambda z_1 : \mathbf{suc}(\phi) . M_{\delta_2} : \mathbf{suc}(\phi) \rightarrow \# \quad \Gamma_5 \vdash R : \mathbf{suc}(\psi x) \rightarrow \# \quad \Gamma_5 \vdash z_4 : \mathbf{suc}(\phi \wedge \psi x) \\
\hline
\Gamma_5 \vdash AND-R(\lambda z_1 : \mathbf{suc}(\phi) . M_{\delta_2})(R)z_4
\end{array}$$

where $M_{\delta_2} = IMP-L\epsilon_s(\exists(\lambda x : \iota. \psi x))\epsilon_s(\phi)(M_{\delta_1})(\lambda y_2 : \mathbf{ant}(\phi) . HYP\epsilon_s(\phi)(y_2)z_2)0y_3$, $\Gamma_5 = \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x)) \supset \phi), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi) \vee \tau)$, $\Gamma_6 = \Gamma_X, y_1 : \mathbf{ant}(\psi x), y_3 : \mathbf{ant}((\exists(\lambda x : \iota. \psi x)) \supset \phi), z_1 : \mathbf{suc}(\psi x), z_4 : \mathbf{suc}(\phi \wedge \psi x), z_5 : \mathbf{suc}((\phi \wedge \psi) \vee \tau)$ and $R = \lambda z_1 : \mathbf{suc}(\psi x) . HYP(y_1)z_1$.

$$\begin{array}{c}
\Gamma_5 \vdash M_{\delta_3} : \# \\
\hline
\Gamma_7 \vdash \lambda z_4 : \mathbf{suc}(\phi \wedge \psi x) . M_{\delta_3} : \mathbf{suc}(\phi \wedge \psi x) \rightarrow \# \\
\Gamma_7 \vdash \lambda z_4 : \mathbf{suc}(\phi \wedge \psi x) . M_{\delta_3} : \mathbf{suc}(\phi \wedge \psi x) \rightarrow \# \quad \Gamma_7 \vdash z_5 : \mathbf{suc}((\phi \wedge \psi x) \vee \tau) \\
\hline
\Gamma_7 \vdash OR-R(\lambda z_4 : \mathbf{suc}(\phi \wedge \psi x) . M_{\delta_3})z_5 : \#
\end{array}$$

It is possible to obtain this derivation from the first one by moving the resolution step representing $v-IMP-I$ up the derivation providing we are careful. It is important that we move the $\rightarrow R$ rule up the tree with the resolution step. We can only move the resolution step up as far as the occurrence of the variables that $\rightarrow I$ abstracts exist. \blacksquare

The analysis we did for G_3i holds here. We can permute a resolution step resolving a constant encoding a left rule, together with a $\rightarrow R$ rule; and, possibly a ΠI rule up the proof-tree until the first occurrence of the variables which the $\rightarrow I$ or ΠI abstracts, in the context. The following lemma formalizes this remark.

Lemma 13.36

Let $\vdash_{G_{3c}}(X) \Delta \xrightarrow{\delta_1} z : \mathbf{suc}(\phi)$ be a provable Horn sequent in G_{3c} and $\Gamma_X, \Gamma_\Delta, z : \mathbf{suc}(\phi) \vdash_{\Sigma_{G_{3c}}} M_{\delta_1} : \#$ be its representation in $\mathbf{NR}_{\Sigma_{G_{3c}}}$. If δ is not in long $\beta\eta$ -normal form, then there exists a proof-object δ' in long $\beta\eta$ -normal form together with a permutation from δ to δ' . This permutation has a corresponding permutation taking M_δ to $M_{\delta'}$. Similarly, if M_δ is not a uniform proof-term, then there exists a uniform proof-term $M_{\delta'}$ together with a permutation from M_δ to $M_{\delta'}$. This

transformation corresponds to a permutation from δ to δ' , where δ' is a uniform proof-term.

Proof It is sufficient to show that any permutation in G_3c corresponds to a permutation in $\text{NR}_{\Sigma_{G_3c}}$. Since we are transforming a proof into long $\beta\eta$ -normal form, we are only going to be permutating left rules past right rules. We show some of the more interesting cases, the rest are similar.

We begin with $v\text{-OR-L}$ being permutated past a $v\text{-AND-R}$. We begin with the proof-tree:

$$\frac{\frac{\frac{\frac{\frac{\vdash (X) \Delta_1 \xrightarrow{\delta_1} z_1 : \text{succ}(\phi), z : \text{succ}(\phi \wedge \psi), \Theta \vdash (X) \Delta_1 \xrightarrow{\delta_2} z_2 : \text{succ}(\psi), z : \text{succ}(\phi \wedge \psi), \Theta}{\vdash (X) \Delta_1 \xrightarrow{\text{AND-R}(z_1 : \delta_1, z_2 : \delta_2, z)} z : \text{succ}(\phi \wedge \psi), \Theta}}{\vdash (X) \Delta_1 \xrightarrow{\text{AND-R}(z_1 : \delta_1, z_2 : \delta_2, z)} \Theta_1} \quad \vdash (X) \Delta_2 \xrightarrow{\delta_3} \Theta_1}{\vdash (X) \Delta, y : \text{ant}(\chi \vee \tau) \xrightarrow{\text{OR-L}(y_1 : \text{AND-R}(z_1 : \delta_1, z_2 : \delta_2, z), y_2 : \delta_3, y)} z : \text{succ}(\phi \wedge \tau), \Theta}}$$

where $\Delta_1 = \Delta, y_1 : \text{ant}(\chi), y : \text{ant}(\chi \vee \tau)$, $\Delta_2 = \Delta, y_2 : \text{ant}(\tau), y : \text{succ}(\chi \vee \tau)$ and $\Theta_1 = z : \text{succ}(\phi \vee \psi), \Theta$, which is permutated to

$$\frac{\frac{\frac{\frac{\frac{\frac{\vdash (X) \Delta_1 \xrightarrow{\delta_1} \Theta_1 \quad \vdash (X) \Delta \xrightarrow{\delta'_3} \Theta_1}{\vdash (X) \Delta_3 \xrightarrow{\text{OR-L}(y_1 : \delta_1, y_2 : \delta'_3, y)} \Theta_1}}{\vdash (X) \Delta_1 \xrightarrow{\delta_2} \Theta_2 \quad \vdash (X) \Delta_2 \xrightarrow{\delta''_3} \Theta_2}{\vdash (X) \Delta_3 \xrightarrow{\text{OR-L}(y_1 : \delta_2, y_2 : \delta''_3, y)} \Theta_2}}{\vdash (X) \Delta_3 \xrightarrow{\text{OR-L}(y_1 : \delta_1, y_2 : \delta'_3, y)} \Theta_1 \quad \vdash (X) \Delta_3 \xrightarrow{\text{OR-L}(y_1 : \delta_2, y_2 : \delta''_3, y)} \Theta_2}}{\vdash (X) \Delta_3 \xrightarrow{\text{AND-R}(z_1 : \text{OR-L}(y_1 : \delta_1, y_2 : \delta'_3, y), z_2 : \text{OR-L}(y_1 : \delta_2, y_2 : \delta''_3, y), z)} \Theta_3}}$$

where $\Delta_1 = \Delta, y_1 : \text{ant}(\chi), y : \text{ant}(\chi \vee \tau)$, $\Delta_2 = \Delta, y_2 : \text{ant}(\tau), y : \text{ant}(\chi \vee \tau)$, $\Delta_3 = \Delta, y : \text{ant}(\chi \vee \tau)$, $\Theta_1 = \Theta, z_1 : \text{succ}(\phi), z_2 : \text{succ}(\phi \wedge \psi)$, $\Theta_2 = \Theta, z_2 : \text{succ}(\psi), z : \text{succ}(\phi \wedge \psi)$, $\Theta_3 = \Theta, z : \text{succ}(\phi \wedge \psi)$ and δ'_3 and δ''_3 are δ_3 followed by weakenings.

We encode the first proof-tree in $\text{NR}_{\Sigma_{G_3c}}$ as the following derivation:

$$\frac{\frac{\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_1} \vdash M_{\delta_1} : \#}{\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_3} \vdash \lambda z_1 : \text{succ}(\phi) . M_{\delta_1} : \text{succ}(\phi) \rightarrow \#}}{\frac{\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_2} \vdash M_{\delta_2} : \#}{\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_3} \vdash \lambda z_2 : \text{succ}(\psi) . M_{\delta_2} : \text{succ}(\psi) \rightarrow \#}}$$

$$\begin{array}{c}
\frac{\Gamma_1 \vdash \lambda z_1 : \text{succ}(\phi) . M_{\delta_1} : \text{succ}(\phi) \rightarrow \# \quad \Gamma_1 \vdash \lambda z_2 : \text{succ}(\psi) . M_{\delta_2} : \text{succ}(\psi) \rightarrow \#}{\Gamma_1 \vdash \text{AND-R}(\lambda z_1 : \text{succ}(\phi) . M_{\delta_1})(\lambda z_2 : \text{succ}(\psi) . M_{\delta_2})z : \#} \\
\frac{\Gamma_1 \vdash \text{AND-R}(\lambda z_1 : \text{succ}(\phi) . M_{\delta_1})(\lambda z_2 : \text{succ}(\psi) . M_{\delta_2})z : \#}{\Gamma_2 \vdash \lambda y_1 : \text{ant}(\chi) . \text{AND-R}(\lambda z_1 : \text{succ}(\phi) . M_{\delta_1})(\lambda z_2 : \text{succ}(\psi) . M_{\delta_2}) : \text{ant}(\chi) \rightarrow \#} \\
\frac{\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_3} \vdash M_{\delta_3} : \#}{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_3} \vdash \lambda y_2 : \text{ant}(\tau) . M_{\delta_3} : \text{ant}(\tau) \rightarrow \#} \\
\frac{\Gamma_2 \vdash P : \text{ant}(\chi) \rightarrow \# \quad \Gamma_2 \vdash \lambda y_2 : \text{ant}(\tau) . M_{\delta_3} : \text{ant}(\tau) \rightarrow \#}{\Gamma_2 \vdash \text{OR-L}(P)(\lambda y_2 : \text{ant}(\tau) . M_{\delta_3})y : \#}
\end{array}$$

where $\Gamma_1 = \Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_3}$, $\Gamma_2 = \Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_3}$, $P = \lambda y_1 : \text{ant}(\chi) . \text{AND-R}(\lambda z_1 : \text{succ}(\phi) . M_{\delta_1})(\lambda z_2 : \text{succ}(\psi) . M_{\delta_2})z$ and the second proof-tree is represented by the derivation done in stages due to the size:

$$\begin{array}{c}
\frac{\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_1} \vdash M_{\delta_1} : \#}{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_1} \vdash \lambda y_1 : \text{ant}(\chi) . M_{\delta_1} : \text{ant}(\chi) \rightarrow \#} \\
\frac{\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_1} \vdash M_{\delta'_3} : \#}{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_1} \vdash \lambda y_2 : \text{ant}(\tau) . M_{\delta'_3} : \text{ant}(\tau) \rightarrow \#} \\
\frac{\Gamma_3 \vdash \lambda y_1 : \text{ant}(\chi) . M_{\delta_1} : \text{ant}(\chi) \rightarrow \# \quad \Gamma_3 \vdash \lambda y_2 : \text{ant}(\tau) . M_{\delta'_3} : \text{ant}(\tau) \rightarrow \#}{\Gamma_3 \vdash \text{OR-L}(\lambda y_1 : \text{ant}(\chi) . M_{\delta_1})(\lambda y_2 : \text{ant}(\tau) . M_{\delta'_3})y : \#} \\
\frac{\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_2} \vdash M_{\delta_2} : \#}{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_2} \vdash \lambda y_1 : \text{ant}(\chi) . M_{\delta_2} : \text{ant}(\chi) \rightarrow \#} \\
\frac{\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_2} \vdash M_{\delta''_3} : \#}{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_2} \vdash \lambda y_2 : \text{ant}(\tau) . M_{\delta''_3} : \text{ant}(\tau) \rightarrow \#} \\
\frac{\Gamma_4 \vdash \lambda y_1 : \text{ant}(\chi) . M_{\delta_2} : \text{ant}(\chi) \rightarrow \# \quad \Gamma_4 \vdash \lambda y_2 : \text{ant}(\tau) . M_{\delta''_3} : \text{ant}(\tau) \rightarrow \#}{\Gamma_4 \vdash \text{OR-L}(\lambda y_1 : \text{ant}(\chi) . M_{\delta_2})(\lambda y_2 : \text{ant}(\tau) . M_{\delta''_3})z : \#} \\
\frac{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_1} \vdash M_{\delta'} : \#}{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_3} \vdash \lambda z_1 : M_{\delta'} . \text{succ}(\phi) \rightarrow \#} \\
\frac{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_2} \vdash M_{\delta''} : \#}{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_3} \vdash \lambda y_2 : \text{succ}(\psi) . M_{\delta''} : \text{succ}(\psi) \rightarrow \#} \\
\frac{\Gamma_5 \vdash \lambda z_1 : \text{succ}(\phi) . M_{\delta'} : \text{succ}(\phi) \rightarrow \# \quad \Gamma_5 \vdash \lambda z_2 : \text{succ}(\psi) . M_{\delta''} : \text{succ}(\psi) \rightarrow \#}{\Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_3} \vdash \text{AND-R}(\lambda z_1 : \text{succ}(\phi) . M_{\delta'})(\lambda z_2 : \text{succ}(\psi) . M_{\delta''}) : \#}
\end{array}$$

where $\Gamma_3 = \Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_1}$, $\Gamma_4 = \Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_2}$, $\Gamma_5 = \Gamma_X, \Gamma_{\Delta_3}, \Gamma_{\Theta_3}$, $M_{\delta'} = \text{OR-L} \epsilon_s(\chi) \epsilon_s(\tau)(\lambda y_1 : \text{ant}(\chi) . M_{\delta_1})(\lambda y_2 : \text{ant}(\tau) . M_{\delta'_3})y$ and $M_{\delta''} = \text{OR-L} \epsilon_s(\chi) \epsilon_s(\tau)(\lambda y_1 :$

$\text{ant}(\chi) . M_{\delta_2})(\lambda y_2 : \text{ant}(\tau) . M_{\delta_3'})y$.

We observe that the two derivations in $\text{NR}_{\Sigma_{G_3^c}}$ only differ by a rule permutation.

We do one other case. Here we have $v\text{-IMP-L}$ being permuted past $v\text{-AND-R}$. We initially have

$$\frac{\frac{\frac{\vdash (X) \Delta_1 \xrightarrow{\delta_2} \Theta_1 \quad \vdash (X) \Delta_1 \xrightarrow{\delta_3} \Theta_2}{\vdash (X) \Delta_1 \xrightarrow{\text{AND-R}(z_1 : \delta_2, z_2 : \delta_3, z)} z : \text{suc}(\chi \wedge \tau), \Theta}}{(X) \Delta, y : \text{ant}(\phi \supset \psi) \xrightarrow{\delta_1} \Theta_3 \quad \vdash X \Delta_1 \xrightarrow{\text{AND-R}(z_1 : \delta_1, z_2 : \delta_2, y)} \Theta_3}}{\vdash (X) \Delta, y : \text{ant}(\phi \supset \psi) \xrightarrow{\text{IMP-L}(z_3 : \delta_1, y_1 : \text{AND-R}(z_1 : \delta_2, z_2 : \delta_3, z), y)} \Theta_3}}$$

where $\Delta_1 = \Delta, y : \text{ant}(\phi \supset \psi), u_1 : \text{ant}(\psi), \Theta_1 = z_1 : \text{suc}(\chi), z : \text{suc}(\chi \wedge \tau), \Theta, \Theta_2 = z_2 : \text{suc}(\tau), z : \text{suc}(\chi \wedge \tau), \Theta, \Theta_3 = z : \text{suc}(\chi \wedge \tau), \Theta$, which is permuted to

$$\frac{\frac{\frac{\frac{\vdash (X) \Delta_1 \xrightarrow{\delta_1} \Theta_1 \quad \vdash (X) \Delta_2 \xrightarrow{\delta_2} \Theta_2}{\vdash (X) \Delta_1 \xrightarrow{\text{IMP-L}(z_3 : \delta_1, y_1 : \delta_2, y)} \Theta_2}}{\vdash (X) \Delta_1 \xrightarrow{\delta_1} \Theta_1 \quad \vdash (X) \Delta_2 \xrightarrow{\delta_3} \Theta_3}}{\vdash (X) \Delta_1 \xrightarrow{\text{IMP-L}(z_3 : \delta_1, y_1 : \delta_3, y)} \Theta_3}}{\frac{\frac{\vdash (X) \Delta_1 \xrightarrow{\text{IMP-L}(z_3 : \delta_1, y_1 : \delta_2, y)} \Theta_2 \quad \vdash (X) \Delta_1 \xrightarrow{\text{IMP-L}(z_3 : \delta_1, y_1 : \delta_3, y)} \Theta_3}{\vdash (X) \Delta_1 \xrightarrow{\text{AND-R}(z_1 : \text{IMP-L}(z_3 : \delta_1, y_1 : \delta_2, y), z_2 : \text{IMP-L}(z_3 : \delta_1, y_1 : \delta_3, y), z)} \Theta_4}}{\vdash (X) \Delta_1 \xrightarrow{\text{AND-R}(z_1 : \text{IMP-L}(z_3 : \delta_1, y_1 : \delta_2, y), z_2 : \text{IMP-L}(z_3 : \delta_1, y_1 : \delta_3, y), z)} \Theta_4}}$$

where $\Delta_1 = \Delta, y : \text{ant}(\phi \supset \psi), \Delta_2 = \Delta, y : \text{ant}(\phi \supset \psi), y_1 : \text{ant}(\psi), \Theta_1 = z_3 : \text{suc}(\phi), \Theta, \Theta_2 = z_1 : \text{suc}(\chi), z : \text{suc}(\chi \wedge \tau), \Theta, \Theta_3 = z_2 : \text{suc}(\tau), z : \text{suc}(\chi \wedge \tau), \Theta$ and $\Theta_4 = z : \text{suc}(\chi \wedge \tau), \Theta$.

We encode the first proof-tree in $\text{NR}_{\Sigma_{G_3^c}}$ and obtain the following derivation:

$$\frac{\frac{\frac{\frac{\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_1} \vdash M_{\delta_1} : \#}{\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_4} \vdash \lambda z_3 : \text{suc}(\phi) . M_{\delta_1} : \text{suc}(\phi) \rightarrow \#}}{\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_2} \vdash M_{\delta_2} : \#}}{\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_4} \vdash \lambda z_1 : \text{suc}(\chi) . M_{\delta_2} : \text{suc}(\chi) \rightarrow \#}}{\frac{\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_3} \vdash M_{\delta_3} : \#}{\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_4} \vdash \lambda z_2 : \text{suc}(\tau) . M_{\delta_3} : \text{suc}(\tau) \rightarrow \#}}{\frac{\Gamma_1 \vdash P : \text{suc}(\chi) \rightarrow \# \quad \Gamma_1 \vdash Q : \text{suc}(\tau) \rightarrow \#}{\Gamma_1 \vdash z : \text{suc}(\chi \wedge \tau)}}{\Gamma_1 \vdash \text{AND-R}(P)(Q)z : \#}}$$

$$\begin{array}{c}
\Gamma_1 \vdash \text{AND-R}(P)(Q)z : \# \\
\hline
\Gamma_2 \vdash \lambda y_1 : \text{ant}(\psi) . \text{AND-R}(P)(Q)z : \text{ant}(\psi) \rightarrow \# \\
\Gamma_2 \vdash \lambda z_3 : \text{suc}(\phi) . M_{\delta_3} : \# \quad \Gamma_2 \vdash R : \text{ant}(\psi) \rightarrow \# \quad \Gamma_2 \vdash y : \text{ant}(\phi \supset \psi) \\
\hline
\Gamma_2 \vdash \text{IMP-L}(\lambda z_3 : \text{suc}(\phi) . M_{\delta_3})(R)y : \#
\end{array}$$

where $\Gamma_1 = \Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_4}$, $\Gamma_2 = \Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_4}$, $P = \lambda z_1 : \text{suc}(\chi) . M_{\delta_2}$, $Q = \lambda z_2 : \text{suc}(\tau) . M_{\delta_3}$ and $R = \lambda y_1 : \text{ant}(\psi) . \text{AND-R}(P)(Q)z$. The second proof-tree is encoded in $\text{NR}_{\Sigma_{G_3^c}}$ as the following derivation. The derivation is in parts due to its size.

$$\begin{array}{c}
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_1} \vdash M_{\delta_1} : \# \\
\hline
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_2} \vdash \lambda z_3 : \text{suc}(\psi) . M_{\delta_1} : \text{suc}(\phi) \rightarrow \# \\
\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_2} \vdash M_{\delta_2} : \# \\
\hline
\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_2} \vdash \lambda y_1 : \text{ant}(\psi) . M_{\delta_2} : \text{ant}(\psi) \rightarrow \# \\
\Gamma_3 \vdash \lambda z_3 : \text{suc}(\psi) . M_{\delta_1} : \text{suc}(\phi) \rightarrow \# \quad \Gamma_3 \vdash \lambda y_1 : \text{ant}(\psi) . M_{\delta_2} : \text{ant}(\psi) \rightarrow \# \\
\hline
\Gamma_3 \vdash \text{IMP-L}(\lambda z_3 : \text{suc}(\psi) . M_{\delta_1})(\lambda y_1 : \text{ant}(\psi) . M_{\delta_2})y : \# \\
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_1} \vdash M_{\delta_1} : \# \\
\hline
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_3} \vdash \lambda z_3 : \text{suc}(\phi) . M_{\delta_1} : \text{suc}(\phi) \rightarrow \# \\
\Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_3} \vdash M_{\delta_3} : \# \\
\hline
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_3} \vdash \lambda y_1 : \text{ant}(\psi) . M_{\delta_3} : \text{ant}(\psi) \rightarrow \# \\
\Gamma_4 \vdash T : \text{suc}(\phi) \rightarrow \# \quad \Gamma_4 \vdash U : \text{ant}(\psi) \rightarrow \# \quad \Gamma_4 \vdash y : \text{ant}(\phi \supset \psi) \\
\hline
\Gamma_4 \vdash \text{IMP-L}(U)(T)y : \# \\
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_2} \vdash M_{\delta'} : \# \\
\hline
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_4} \vdash \lambda z_1 : \text{suc}(\chi) . M_{\delta'} : \text{suc}(\chi) \rightarrow \# \\
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_3} \vdash M_{\delta''} : \# \\
\hline
\Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_4} \vdash \lambda z_2 : \text{suc}(\tau) . M_{\delta''} : \text{suc}(\tau) \rightarrow \# \\
\Gamma_5 \vdash V : \text{suc}(\chi) \rightarrow \# \quad \Gamma_5 \vdash W : \text{suc}(\tau) \rightarrow \# \quad \Gamma_5 \vdash z : \text{suc}(\chi \wedge \tau) \\
\hline
\Gamma_5 \vdash \text{AND-R}(V)(W)z : \#
\end{array}$$

where $\Gamma_3 = \Gamma_X, \Gamma_{\Delta_1}, \Gamma_{\Theta_2}$, $\Gamma_4 = \Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_3}$, $\Gamma_5 = \Gamma_X, \Gamma_{\Delta_2}, \Gamma_{\Theta_4}$, $T = \lambda z_3 : \text{suc}(\phi) . M_{\delta_1}$, $U = \lambda y_1 : \text{ant}(\psi) . M_{\delta_3}$, $V = \lambda z_1 : \text{suc}(\chi) . M_{\delta'}$, $W = \lambda z_2 : \text{suc}(\tau) . M_{\delta''}$, $M_{\delta'} = \text{IMP-L}\epsilon_s(\phi)\epsilon_s(\psi)(\lambda z_3 : \text{suc}(\psi) . M_{\delta_1})(\lambda y_1 : \text{ant}(\psi) . M_{\delta_2})y$ and $M_{\delta''} = \text{IMP-L}\epsilon_s(\phi)\epsilon_s(\psi)(\lambda z_3 : \text{suc}(\psi) . M_{\delta_1})(\lambda y_1 : \text{ant}(\psi) . M_{\delta_3})y$.

Again we observe that the two derivations in $\text{NR}_{\Sigma_{G_3^c}}$ differ by a permutation of the *Res* and \rightarrow *R* rules. ■

13.5 Summary

The analysis in the previous section can be extended to higher-order classical logic. The following definition and lemmas cover both classical and higher-order classical logic.

Definition 13.37 (Uniform Proof-Term for Multi-Concluioned Sequent Systems)

Let \mathcal{L} be a judged proof system with multi-conclusioned sequent rules, which can be adequately represented in LF. Let M be a term in the $\lambda\Pi_{\Sigma_{\mathcal{L}}}$ calculus in long $\beta\eta$ -normal form. We say that M is a *uniform proof-term for a multi-conclusioned sequent system* if all subterms $Ny : \#$ typed in a context $\Gamma_{\mathcal{A}}, \Gamma_{\Delta}, \Gamma_{\Theta}$, satisfy the following conditions:

- if y has type $\text{ant}(@(\phi_1, \dots, \phi_n))$, where $@ \in \{\top, \wedge, \vee, \supset, \forall_{\sigma}, \exists_{\sigma}\}$ and Γ_{Θ} contains $z : \text{suc}(\psi)$, where ψ is atomic;
- if $y : \text{suc}(\phi)$, then $\phi = @(\phi_1, \dots, \phi_n)$, where $@ \in \{\wedge, \vee, \supset, \forall_{\sigma}, \exists_{\sigma}\}$ ■

Lemma 13.38

Let \mathcal{L} be a judged proof system with multi-conclusioned sequents, which can be adequately represented in LF. If every proof-term M in $\text{NR}_{\Sigma_{\mathcal{L}}}$ is a uniform proof-term or can be permuted into one, then \mathcal{L} is an abstract logic programming language.

Proof (Sketch) Follows from Lemma 13.32 and Miller *et al.*'s definition of an abstract logic programming language. We leave higher-order classical logic to the reader. ■

It would be nice to have a result about the representation of an abstract logic programming language in LF which works for judged proof systems with single- and multi-conclusioned sequents. Our problem arises from the method used to encode multi-conclusioned sequent systems in LF. It is fundamentally different from the way that single-conclusioned systems are encoded. Ideally, we would prefer to be able to restrict multi-conclusioned systems to their single-conclusioned fragment. This would mirror Miller *et al.*'s method for defining uniform proof in classical logic. We cannot do this because we need to use the G_3 -type systems to be able to obtain an adequate encoding in LF. Since this does not work, the next simplest solution is to consider the single-conclusioned systems to be a special case of the multi-conclusioned systems and then encode them as multi-conclusioned systems. This also does not work since we are unable to prove fullness for the right rules. We illustrate the problem which arises.

Suppose we have the rule *v-AND-R*. We apply the induction hypothesis to obtain $\Gamma_X, \Gamma_{\Delta}, z_1 : \text{suc}(\phi) \vdash_{\Sigma_{G'_3i}} M_{\delta_1} : \#$ and $\Gamma_X, \Gamma_{\Delta}, z_2 : \text{suc}(\psi) \vdash_{\Sigma_{G'_3i}} M_{\delta_2} : \#$, where $\Sigma_{G'_3i}$ is the $\lambda\Pi$ -context with constants encoding all the valid proof rules of G_3i using $\#$. We now abstract these two rules to obtain $\Gamma_X, \Gamma_{\Delta} \vdash_{\Sigma_{G'_3i}} \lambda z_1 : \text{suc}(\phi) . M_{\delta_1} : \text{suc}(\phi) \rightarrow \#$ and $\Gamma_X, \Gamma_{\Delta} \vdash_{\Sigma_{G'_3i}} \lambda y_2 : \text{suc}(\psi) . M_{\delta_2} : \text{suc}(\psi) \rightarrow \#$. We

also have $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G'_3 i}} \text{AND-R} : (\text{suc}(\phi) \rightarrow \#) \rightarrow (\text{suc}(\psi) \rightarrow \#) \rightarrow (\text{suc}(\phi \wedge \psi) \rightarrow \#)$. We now have a problem since we are unable to obtain the assertion $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_{G'_3 i}} z : \text{suc}(\phi \wedge \psi)$. This does not occur in classical logic since $z : \text{suc}(\phi \wedge \psi)$ is present in the succedent of each premiss and hence $z : \text{suc}(\phi \wedge \psi)$ does occur in Γ_Θ .

The author claims that because LF is not a natural metatheory for sequent calculi and the method of encoding single- and multi-conclusioned sequent calculi are fundamentally different, it will not be possible to solve this problem. It is, however, left as an open problem.

Chapter 14

Conclusion

The main results of this thesis are summarized as follows:

- Kripke $\lambda\Pi$ -models. These models provide a semantics for the $\lambda\Pi$ -calculus and are a generalization of the Kripke lambda models of (Mitchell & Moggi 1991).
- Propositions-as-types isomorphism. We have shown that the propositions-as-types correspondence between the $\lambda\Pi$ -calculus and its internal logic induces an isomorphism between a Kripke $\lambda\Pi$ -model and a suitable Kripke model of the internal logic.
- Judged proof systems. We have provided a general proof- and model-theoretic account of these, which includes a class of Kripke models.
- Judgements-as-types epimorphism. We have shown that the judgements-as-types correspondence induces an epimorphism between a Kripke model of a judged proof system and a Kripke $\lambda\Pi$ -model. Furthermore, this epimorphism was used to provide a semantic proof of fullness.
- Worlds-as-parameters. We have shown that labelled natural deduction systems presented as judged proof systems provide a natural account of the worlds-as-parameters representation mechanism and allow us to observe it is a special case of the judgements-as-types correspondence. Furthermore, we show that we can interpret labels as worlds in the appropriate Kripke models, which allows us to obtain a worlds-as-parameters epimorphism.
- Least fixed-point construction. We have shown that a class of Kripke $\lambda\Pi$ -models can be seen to be Herbrand models and a least fixed-point can be constructed.
- Abstract logic programming languages. We have provided a characterization of abstract logic programming languages in terms of proof-terms in LF.

We conclude with a discussion of various possible avenues of future research. This thesis is, in a certain sense, a tidying up of various loose ends and setting the scene for a foundation for proof-search and logic programming in terms of logical frameworks. The results of Chapter 11 are not used in this thesis and further work on proof-search needs to take these results as its starting point. The next stage would be to consider the search space generated by resolution in \mathbf{C} and examine its relationship to the search space generated by resolution in the object-logic. An important text to consider would be (Kowalski 1979) and the hope would be to provide a formal account of the ideas contained therein.

On a similar vein, an account of backtracking in terms of Kripke worlds (*cf.* (Pym & Ritter 2004)) could be provided for Kripke $\lambda\Pi$ -models. This could then be compared to backtracking in Kripke models of object-logics.

There is a strong possibility that semantic characterizations of fullness and faithfulness can be obtained. These would be similar to those found in (Simpson 1993).

In Chapter 7 we encountered problems with a general semantic characterization of connectives. It is not clear whether a general satisfaction condition, analogous to Prawitz's schemata for natural deduction inference rules, is possible. It seems that it is impossible to decide *a priori* whether a given connective is local or non-local, for example. The most likely approach to this problem is through labelled natural deduction systems. The author hopes that Chapter 9 goes some way to helping formalize the problem. It would also be interesting to analyse in detail the relationship between a general satisfaction condition, if one exists, and the categorical structure this induces in the Kripke model.

It is possible that the analysis in this thesis can be carried out for other logical frameworks. The author hopes that this will be done in the future and that model-theoretic analysis of logical frameworks is as common as proof-theoretic analysis.

Finally, we suggest a line of research suggested to the author by Matthew Collinson. He has suggested that the proof of functional completeness found in (Wansing 1993) is a metalogical argument and that it could be used, after a suitable generalization, to determine functional completeness for object-logics which can be adequately represented in LF.

Bibliography

LaTeX 2 ϵ (1994).

URL: <http://www.latex-project.org/>

- Aczel, P. (1980), Frege structures and the notions of propositions, truth and set, in J. Barwise, H. J. Keisler & K. Kunen, eds, 'The Kleene Symposium', North-Holland, pp. 31–59.
- Alechina, N., de Paiva, V. & Ritter, E. (1998), 'Relating categorical and kripke semantics for intuitionistic modal logics'.
- Arnon Avron, F. H. & Mason, I. (1996), An overview of the edinburgh logical framework, Technical report, University of Edinburgh.
- Avron, A. (1991), 'Simple consequence relations', *Information and Computation* **92**, 105–139.
- Avron, A., Honsell, F., Mason, I. & Pollack, R. (1992), 'Using typed lambda calculus to implement formal systems on a machine', *Journal of Automated Reasoning* **9**, 309–354.
- Avron, A., Honsell, F., Miculan, M. & Paravano, C. (1997), 'Encoding modal logics in logical frameworks', *Studia Logical* **60**, 161–208.
- Barendregt, H. (1992), Lambda calculi with types, in S. Abramsky, D. M. Gabbay & T. S. E. Maibaum, eds, 'Handbook of Logic in Computer Science: Volume 2', Oxford University Press.
- Barendregt, H. P. (1991), 'Introduction to generalized type systems', *Journal of Functional Programming* **1**(2), 125–154.
- Basin, D. & Matthews, S. (2002), Logical frameworks, in D. Gabbay & F. Guenther, eds, 'Handbook of Philosophical Logic: Volume 9', Kluwer Academic Publishers.
- Basin, D., Matthews, S. & Viganò, L. (1997), 'Labelled propositional modal logics: Theory and practice', *Journal of Logic, Language and Computation* **7**(1), 685–717.

- Basin, D., Matthews, S. & Viganò, L. (1998), ‘Natural deduction for non-classical logics’, *Studia Logica* **60**(1), 685–717.
- Brouwer, L. E. J. (1924a), ‘Intuitionistische zerlegung mathematischer grundbegriffe’, *Jahresbericht der Deutsche Mathematiker-Vereinigung* **33**, 241–256.
- Brouwer, L. E. J. (1924b), ‘Over de onbetrouwbaarheid’, *Tijdschrift voor Wijsbegeerte* **2**, 21–256.
- Brown, J. & Wallen, L. (1995), Representing unification in a logical framework, in ‘CSL’, pp. 106–129.
- Cartmell, J. (1986), ‘Generalized algebraic theories and contextual categories’, *Annals of pure and applied logic* **32**(33), 209–243.
- Cartmell, J. (1990), Generalized algebraic theories and contextual categories, PhD thesis, University of Oxford.
- Chellas, B. (1980), *Modal Logic: An Introduction*, Cambridge University Press.
- Church, A. (1940), ‘A formulation of the simple theory of types’, *The journal of symbolic logic* **5**(2), 56–68.
- Coquand, T. (1991), An algorithm for testing conversion in type theory, in G. Huet & G. Plotkin, eds, ‘Logical Frameworks’, Cambridge University Press, pp. 255–279.
- Curry, H. (1934), Functionality in combinatory logic, in ‘Proceedings of the National Academy of Sciences’, Vol. 20, pp. 584–590.
- Curry, H. & Feys, R. (1958), *Combinatory Logic Volume I*, Amsterdam: North-Holland.
- Dowek, G., Hardin, T., Kirchner, C. & Pfenning, F. (1996), Unification via explicit substitutions: The case of higher-order patterns, in M. Maher, ed., ‘Proceedings of the Joint International Conference and Symposium on Logic Programming’, MIT Press, pp. 259–273.
- Dybjer, P. (1995), Internal type theory, in ‘TYPES’, pp. 120–134.
- Elliott, C. M. (1990), Extensions and Applications of Higher-order Unification, PhD thesis, Carnegie Mellon University.
- Felty, A. (1991), Encoding dependent types in a intuitionistic logic, in G. Huet & G. Plotkin, eds, ‘Logical Frameworks’, Cambridge University Press, pp. 241–251.

- Friedman, H. (1975), Equality between functionals, *in* R. Parikh, ed., ‘Logic Colloquium: Symposium on Logic held at Boston, 1972-73’, Lecture Notes in Mathematics, 453, Springer.
- Gallier, J. (1997), ‘Kripke models and the (in)equational logic of the second-order lambda-calculus’, *Annals of Pure and Applied Logic* **84**, 257–316.
- Galmiche, D. & Pym, D. (2000), ‘Proof-search in type-theoretic languages: An introduction’, *Theoretical Computer Science* **232**, 5–53.
- Gardner, P. (1992a), Equivalences between logics and their representing type theories, Technical report, University of Edinburgh. LFCS report ECS-LFCS-92-251.
- Gardner, P. (1992b), Representing Logics in Type Theory, PhD thesis, University of Edinburgh.
- Gentzen, G. (1934), ‘Untersuchungen über das logische schliessen’, *Mathematische Zeitschrift* **34**, 176–210 and 405–431.
- Girard, J. (1987), ‘Linear logic’, *Theoretical Computer Science* **50**(1), 1–102.
- Harper, R. (1988), An equational formulation of LF, Technical report, University of Edinburgh. LFCS report ECS-LFCS-88-67.
- Harper, R., Honsell, F. & Plotkin, G. (1987), A framework for defining logics, *in* ‘Proceedings of the 2nd Annual IEEE Symposium on Logic in Computer Science, LICS’87, Ithaca, NY, USA, 22–25 June 1987’, IEEE Computer Society Press, New York, pp. 194–204.
- Harper, R., Honsell, F. & Plotkin, G. (1993), ‘A framework for defining logics’, *Journal of the Association for Computing Machinery* **40**(1), 143–184.
- Harper, R., Sannella, D. & Tarlecki, A. (1994), ‘Structured theory presentations and logic representations’, *Annals of Pure and Applied Logic* **67**, 113–160.
- Hayashi, S. (1985), ‘Adjunction of semifunctors: categorical structures in non-extensional lambda calculus’, *Theoretical computer science* **41**, 95–104.
- Herbrand, J. (1967), Investigations in proof theory: the properties of true propositions, *in* J. van Heijenoort, ed., ‘From Frege to Gödel: A Source Book in Mathematical Logic’, Harvard University Press, pp. 525–581.
- Heyting, A. (1934), *Mathematische Grundlagenforschung. Intuitionismus. Beweistheorie*, Springer-Verlag.

- Hindley, J. R. & Seldin, J. P. (1986), *Introduction to Combinators and λ -calculus*, Cambridge University Press.
- Hodges, W. (2001), Elementary predicate logic, *in* D. Gabbay & F. Guenther, eds, 'Handbook of Philosophical Logic: Volume 1', Kluwer Academic Publishers.
- Hofmann, M. (1996), Syntax and semantics of dependent types, *in* A. Pitts & P. Dybjer, eds, 'Semantics and Logics of Computation', Cambridge University Press, pp. 79–130.
- Howard, W. A. (1980), The formulæ-as-types notion of construction, *in* J. R. Hindley & J. P. Seldin, eds, 'To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism', Academic Press, pp. 479–490.
- Ishtiaq, S. & Pym, D. J. (2002), 'Dependently-typed, bunched λ -calculus', *Journal of logic and computation* **12**(6), 1061–1104.
- Ishtiaq, S. S. & Pym, D. J. (1998), 'A relevant analysis of natural deduction', *Journal of Logic and Computation* **8**(6), 809–838.
- Jacobs, B. (1991), *Categorical Type Theory*, PhD thesis, University of Nijmegen.
- Jacobs, B. (1993), 'Comprehension categories and the semantics of type dependency', *Theoretical Computer Science* **107**, 169–207.
- Jacobs, B. (1999), *Categorical Logic and Type Theory*, Elsevier.
- Kant, I. (1800), *Immanuel Kant's Logik: Ein Handbuch zu Vorlesungen*, Gottlob Benjamin Jäsche, Königsberg.
- Kleene, S. (1952), *Introduction to Metamathematics*, North-Holland.
- Kock, A. & Reyes, G. E. (1977), Doctrines in categorical logic, *in* J. Barwise, ed., 'Handbook of Mathematical Logic', North-Holland.
- Kolmogorov, A. N. (1932), 'Zur deutung der intuitionistischen logik', *Mathematische Zeitschrift* **35**, 58–65.
- Kowalski, R. (1979), *Logic for Problem Solving*, North Holland.
- Kripke, S. (1963), 'Semantical analysis of modal logic I', *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **16**, 83–94.
- Kripke, S. (1965), Semantical analysis of intuitionistic logic, *in* 'Formal Systems and Recursive Functions', North-Holland Publishing Company, pp. 92–130.

- Lambek, J. & Scott, P. J. (1986), *Introduction to Higher-Order Categorical Logic*, Cambridge University Press.
- Lawvere, W. (1969), ‘Adjointness in foundations’, *Dialectica* **23**. Reprinted in Reprints in Theory and Applications of Categories, No. 16 (2006), pp 1-16.
- Lawvere, W. (1970), Equality in hyperdoctrines and comprehension schema as an adjoint functor, *in* ‘Proceedings of the American Mathematical Society Symposium on Pure Mathematics XVII’, pp. 1–14.
- Lemmon, E. & Scott, D. (1977), *An Introduction to Modal Logic*, Blackwell.
- Lloyd, J. W. (1984), *Foundations of Logic Programming*, Springer Verlag.
- Lorenzen, P. (1955), *Einführung die operative Logik und Mathematik*, Berlin.
- Martin-Löf, P. (1971), Hauptatz for the intuitionistic theory of iterated inductive definitions, *in* J. E. Fenstad, ed., ‘Second Scandinavian Logic Symposium’, North-Holland.
- Martin-Löf, P. (1975), An intuitionistic theory of types: Predicative part, *in* H. E. Rose & J. C. Shepherdson, eds, ‘Logic Colloquium 1973’, North-Holland, pp. 73–188.
- Martin-Löf, P. (1982), ‘On the meaning of logical constants and the justifications of the logical laws’.
- Mason, I. A. (1903), Hoare’s logic in the LF, Technical Report ECS-LFCS-87-32, University of Edinburgh.
- Meyer, A. R. (1982), ‘What is a model of the lambda calculus?’, *Information and Control* **52**(1), 87–122.
- Miller, D. (1989), ‘A logical analysis of modules in logic programming’, *journal of logic programming* **6**, 79–108.
- Miller, D., Nadathur, G., Pfenning, F. & Šcedrov, A. (1991), ‘Uniform proofs as a foundation for logic programming’, *Annals of pure and applied logic* **51**, 125–157.
- Mitchell, J. C. & Moggi, E. (1991), ‘Kripke-style models for typed λ -calculus’, *Annals of Pure and Applied Logic* **51**, 99–124.
- Paré, R. & Schumacher, D. (1978), Abstract families and the adjoint functor theorem, *in* P. T. Johnstone & R. Paré, eds, ‘Indexed categories and their applications’, lecture notes in mathematics 661, Springer-Verlag.

- Pfenning, F. & Schürmann, C. (1998), Algorithms for equality and unification in the presence of notational definitions, *in* T. Altenkirch, W. Naraschewski & B. Rues, eds, ‘Types for Proofs and Programs’, Lecture Notes in Computer Science, Springer Verlag, pp. 179–193.
- Pfenning, F. (1991), Logic programming in the LF logical framework, *in* ‘Logical Frameworks’, Cambridge University Press, pp. 149–181.
- Pfenning, F. (2000), ‘Structural cut elimination I. classical and intuitionistic logic’, *Information and Control* **157**(1-2), 84–141.
- Pfenning, F. (2002), ‘A linear logical framework’, *Information and Computation* **179**, 19–75.
- Pfenning, F. & Elliott, C. (1988), Higher-order abstract syntax, *in* ‘Proceedings of the ACM SIGPLAN PLDI ’88’, pp. 199–208.
- Pfenning, F. & Schürmann, C. (1999), System description: Twelf - a meta-logical framework for deductive systems, *in* H. Ganzinger, ed., ‘Proceedings of the 16th International Conference of Automated Deduction (CADE-16)’, Springer Verlag, pp. 202–206.
- Pientka, B. & Pfenning, F. (2003), Optimizing higher-order pattern unification, *in* F. Baader, ed., ‘Proceedings of the 19th Conference on Automated Deduction (CADE-19)’, Springer Verlag, pp. 473–487.
- Pitts, A. M. (2000), Categorical logic, *in* S. Abramsky, D. M. Gabbay & T. S. E. Maibaum, eds, ‘Handbook of Logic in Computer Science, Volume 5. Algebraic and Logical Structures’, Oxford University Press, pp. 39–128.
- Plotkin, G. (1980), Lambda-definability in the full-type hierarchy, *in* ‘To H. B. Curry: Essays on Combinatory logic, lambda calculus and formalism’, Academic press, pp. 363–374.
- Plotkin, G. (n.d.), Lambda-definability in the full-type hierarchy. Internal memorandum, school of artificial intelligence, university of Edinburgh.
- Prawitz, D. (1965), *Natural Deduction: A Proof-theoretic Study*, Almqvist and Wiksell.
- Prawitz, D. (1978), Proofs and the meanings and completeness of the logical constants, *in* J. Hintikka, I. Niiniluoto & E. Saarinen, eds, ‘Essays in mathematical and philosophical logic’, Reidel, pp. 25–39.
- Prawitz, D. (1982), ‘Beweise und die bedeutung und vollständigkeit der logischen konstanten’, *Conceptus* **16**(38), 3–44.

- Prawitz, D. (2008). Personal Communication.
- Prior, A. (1960), ‘The runabout inference ticket’, *Analysis* **21**, 38–39.
- Pym, D. (1992), ‘A unification for the $\lambda\Pi$ -calculus’, *International Journal of Foundations of Computer Science* **2**, 333–378.
- Pym, D. (2004a), Functorial Kripke-Beth-Joyal models of the $\lambda\Pi$ -calculus I: type theory and internal logic. Personal correspondence.
- Pym, D. (2004b), Functorial Kripke-Beth-Joyal models of the $\lambda\Pi$ -calculus II: the lf logical framework. Personal correspondence.
- Pym, D. (2004c), Functorial Kripke-Beth-Joyal models of the $\lambda\Pi$ -calculus III: logic programming and its semantics. Personal correspondence.
- Pym, D. & Harland, J. (1994), ‘A uniform proof-theoretic investigation of linear logic programming’, *Journal of Logic and Computation* **2**, 175–207.
- Pym, D. J. (1990), Proofs, Search and Computation in General Logic, PhD thesis, University of Edinburgh. Available as report CST-69-90, Department of Computer Science, University of Edinburgh, 1990. (Also published as LFCS report ECS-LFCS-90-125.).
- Pym, D. J. (1995), ‘A note on the proof theory [of] the $\lambda\Pi$ -calculus’, *Studia Logica* **54**(2), 199–230.
- Pym, D. & Ritter, E. (2004), *Reductive Logic and Proof-search: Proof Theory, Semantics and Control*, Oxford University Press.
- Pym, D. & Wallen, L. (1991), Proof-search in the $\lambda\Pi$ -calculus, in ‘Logical Frameworks’, Cambridge University Press, pp. 309–340.
- Pym, D. & Wallen, L. (1992), Logic programming via proof-valued computations, in K. Broda, ed., ‘ALPUK92, Proceedings of the 4th U.K. Conference on logic Programming’, Springer Verlag, pp. 253–262.
- Read, S. (1988), *Relevance Logics*, Blackwell.
- Restall, G. (2008), Proof theory and philosophy. Draft available at <http://consequently.org/papers/ptp.pdf>.
- Ritter, E. (1992), Categorical Abstract Machines for Higher-Order Typed Lambda Calculi, PhD thesis, University of Cambridge.

- Ritter, E., Pym, D. & Wallen, L. (1996*a*), On the intuitionistic force of classical search, *in* ‘Proceedings of the Fifth Workshop on Theorem Proving and Analytic Tableaux and Related Methods’, Lecture Notes in Artificial Intelligence, Springer Verlag, pp. 295–311.
- Ritter, E., Pym, D. & Wallen, L. (1996*b*), Proof-terms for classical and intuitionistic logic (extended abstract), *in* ‘Proceedings of the 13th International Conference on Automate Deduction’, Lecture Notes in Artificial Intelligence, pp. 17–31.
- Robinson, J. A. (1965), ‘A machine-oriented logic based on the resolution principle’, *Journal of the Association for Computing Machinery* **12**, 23–49.
- Salvesen, A. (1990), A proof of the Church-Rosser property for the Edinburgh LF with η -conversion. Lecture given at the First Workshop on Logical Frameworks, Sophia-Antipolis, France.
- Schröder-Heister, P. & Dösen, K., eds (1993), *Substructural Logics*, Clarendon Press.
- Schroeder-Heister, P. (1982), ‘Logische konstanten und regeln. zur deutung von aussagenoperatoren’, *Conceptus* **16**(38), 45–59.
- Seely, R. A. (1983), ‘Hyperdoctrines, naturality and the the Beck condition’, *Zeitschrift für Mathematik Logic und Grundlagen* **52**, 505–542.
- Simpson, A. (1993), Kripke semantics for a logical framework. Presented at Workshop on Types for Proofs and Programs, Baastad, Sweden.
- Simpson, A. (1994), The Proof Theory and Semantics of Intuitionistic Modal Logic, PhD thesis, University of Edinburgh.
- Statman, R. (1985), ‘Logical relations and typed lambda-calculus’, *Information and control* **65**, 85–97.
- Streicher, T. (1989), Completeness and Correctness of a Categorical Semantics of the Calculus of Constructions, PhD thesis, University of Passau.
- Sundholm, G. (2001), Systems of deduction, *in* D. Gabbay & F. Guentner, eds, ‘Handbook of Philosophical Logic: Volume 2’, Kluwer Academic Publishers.
- Tarski, A. (1955), ‘A lattice-theoretic fixpoint theorem and its applications’, *Pacific journal of mathematics* **5**, 285–309.
- Taylor, P. (1986), ‘Diagrams’.
URL: <http://tug.ctan.org/tex-archive/macros/generic/diagrams/taylor/>

- Troelstra, A. & Schwichtenberg, H. (1996), *Basic Proof Theory*, Cambridge University Press.
- van Benthem, J. (1984), Correspondence theory, *in* 'Handbook of Philosophical Logic, Volume II', Reidel, Dordrecht.
- Van Daalen, D. T. (1980), The Language Theory of Automath, PhD thesis, Technical University of Eindhoven.
- Van Dalen, D. (1994), *Logic and Structure*, Springer.
- van Emden, M. H. & Kowalski, R. A. (1976), 'The semantics of predicate logic as a programming language', *Journal of the association for computing machinery* **23**, 733–742.
- Viganò, L. (2000), *Labelled Non-Classical Logics*, Kluwer Academic Publishers.
- Wansing, H. (1993), 'Functional completeness for subsystems of intuitionistic propositional logic', *Journal of Philosophical Logic* **22**, 303–321.

Appendix A

Signatures of Object-logics

This appendix contains a various signatures of object-logics.

Minimal Logic

Definition A.1 (Σ_{ML})

Σ_{ML} is the $\lambda\Pi$ -signature containing the following constants:

o : Type

ι : Type

proof: $o \rightarrow \text{Type}$

\wedge : $o \rightarrow o \rightarrow o \rightarrow o$

\vee : $o \rightarrow o \rightarrow o \rightarrow o$

\supset : $o \rightarrow o \rightarrow o \rightarrow o$

\forall : $(\iota \rightarrow o) \rightarrow o$

\exists : $(\iota \rightarrow o) \rightarrow o$

$\wedge I$: $\Pi p: o. \Pi q: o. \text{true}(p) \rightarrow \text{true}(q) \rightarrow \text{true}(p \wedge q)$

$\vee I1$: $\Pi p: o. \Pi q: o. \text{true}(p) \rightarrow \text{true}(p \vee q)$

$\vee I2$: $\Pi p: o. \Pi q: o. \text{true}(q) \rightarrow \text{true}(p \vee q)$

$\supset I$: $\Pi p: o. \Pi q: o. (\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(p \supset q)$

$\forall I$: $\Pi F: \iota \rightarrow o. (\Pi x: \iota. \text{true}(Fx)) \rightarrow \text{true}(\forall(\lambda x: \iota. Fx))$

$\exists I$: $\Pi F: \iota \rightarrow o. (\Pi x: o. \text{true}(Fx)) \rightarrow \text{true}(\exists(\lambda x: \iota. Fx))$

$\wedge E$: $\Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \vee q) \rightarrow ((\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(r)) \rightarrow \text{true}(r)$

$\vee E$: $\Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \vee q) \rightarrow (\text{true}(p) \rightarrow \text{true}(r)) \rightarrow (\text{true}(q) \rightarrow \text{true}(r))$

$$\begin{aligned}
& \rightarrow \text{true}(r) \\
\supset E: \Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \supset q) \rightarrow \text{true}(p) \rightarrow (\text{true}(q) \rightarrow \text{true}(r)) \rightarrow \text{true}(r) \\
\forall E: \Pi F: \iota \rightarrow o. \Pi x: \iota. \Pi r: o. \text{true}(\forall(\lambda x: \iota. Fx)) \rightarrow (\text{true}(Fx) \rightarrow \text{true}(r)) \\
& \rightarrow \text{true}(r) \\
\exists E: \Pi F: \iota \rightarrow o. \Pi r: o. \text{true}(\exists(\lambda x: \iota. Fx)) \rightarrow (\Pi x: \iota. \text{true}(Fx) \rightarrow \text{true}(r)) \\
& \rightarrow \text{true}(r) \quad \blacksquare
\end{aligned}$$

Intuitionistic Logic

Definition A.2 (Σ_{IL})

Σ_{IL} is the $\lambda\Pi$ -signature containing the following constants:

$$\begin{aligned}
& o: \text{Type} \\
& \iota: \text{Type} \\
& \text{proof}: o \rightarrow \text{Type} \\
& \wedge: o \rightarrow o \rightarrow o \rightarrow o \\
& \vee: o \rightarrow o \rightarrow o \rightarrow o \\
& \supset: o \rightarrow o \rightarrow o \rightarrow o \\
& \perp: o \\
& \forall: (\iota \rightarrow o) \rightarrow o \\
& \exists: (\iota \rightarrow o) \rightarrow o \\
& \wedge I: \Pi p: o. \Pi q: o. \text{true}(p) \rightarrow \text{true}(q) \rightarrow \text{true}(p \wedge q) \\
& \vee I1: \Pi p: o. \Pi q: o. \text{true}(p) \rightarrow \text{true}(p \vee q) \\
& \vee I2: \Pi p: o. \Pi q: o. \text{true}(q) \rightarrow \text{true}(p \vee q) \\
& \supset I: \Pi p: o. \Pi q: o. (\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(p \supset q) \\
& \forall I: \Pi F: \iota \rightarrow o. (\Pi x: \iota. \text{true}(Fx)) \rightarrow \text{true}(\forall(\lambda x: \iota. Fx)) \\
& \exists I: \Pi F: \iota \rightarrow o. (\Pi x: o. \text{true}(Fx)) \rightarrow \text{true}(\exists(\lambda x: \iota. Fx)) \\
& \wedge E: \Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \wedge q) \rightarrow ((\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(r)) \rightarrow \text{true}(r) \\
& \vee E: \Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \vee q) \rightarrow (\text{true}(p) \rightarrow \text{true}(r)) \rightarrow (\text{true}(q) \rightarrow \text{true}(r)) \\
& \rightarrow \text{true}(r) \\
& \supset E: \Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \supset q) \rightarrow \text{true}(p) \rightarrow (\text{true}(q) \rightarrow \text{true}(r)) \rightarrow \text{true}(r) \\
& \forall E: \Pi F: \iota \rightarrow o. \Pi x: \iota. \Pi r: o. \text{true}(\forall(\lambda x: \iota. Fx)) \rightarrow (\text{true}(Fx) \rightarrow \text{true}(r))
\end{aligned}$$

$$\begin{aligned} & \rightarrow \text{true}(r) \\ \exists E: \Pi F: \iota \rightarrow o. \Pi r: o. \text{true}(\exists(\lambda x: \iota. Fx)) & \rightarrow (\Pi x: \iota. \text{true}(Fx) \rightarrow \text{true}(r)) \\ & \rightarrow \text{true}(r) \end{aligned} \quad \blacksquare$$

$$BOT: \Pi p: o. \text{proof}(\perp) \rightarrow \text{proof}(p)$$

Modal Logics

Definition A.3 (Σ_G)

Σ_G is the $\lambda\Pi$ -signature containing the following constants:

$$\begin{aligned} o: & \text{Type} \\ \iota: & \text{Type} \\ \text{true}: & o \rightarrow \text{Type} \\ \text{valid}: & o \rightarrow \text{Type} \\ \wedge: & o \rightarrow o \rightarrow o \rightarrow o \\ \vee: & o \rightarrow o \rightarrow o \rightarrow o \\ \supset: & o \rightarrow o \rightarrow o \rightarrow o \\ \Box: & o \rightarrow o \\ \Diamond: & o \rightarrow o \\ G: & \text{valid}(\Diamond^j \Box^m \phi \supset \Box^i \Diamond^n \phi) \\ \wedge I: & \Pi p: o. \Pi q: o. \text{true}(p) \rightarrow \text{true}(q) \rightarrow \text{true}(p \wedge q) \\ \vee I1: & \Pi p: o. \Pi q: o. \text{true}(p) \rightarrow \text{true}(p \vee q) \\ \vee I2: & \Pi p: o. \Pi q: o. \text{true}(q) \rightarrow \text{true}(p \vee q) \\ \supset I: & \Pi p: o. \Pi q: o. (\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(p \supset q) \\ \text{Nec } I: & \Pi p: o. \text{valid}(p) \rightarrow \text{valid}(\Box p) \\ \text{Pos } I: & \Pi p: o. \text{valid}(p) \rightarrow \text{valid}(\Diamond p) \\ \wedge E: & \Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \vee q) \rightarrow ((\text{true}(p) \rightarrow \text{true}(q)) \rightarrow \text{true}(r)) \rightarrow \text{true}(r) \\ \vee E: & \Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \vee q) \rightarrow (\text{true}(p) \rightarrow \text{true}(r)) \rightarrow (\text{true}(q) \rightarrow \text{true}(r)) \\ & \rightarrow \text{true}(r) \\ \supset E: & \Pi p: o. \Pi q: o. \Pi r: o. \text{true}(p \supset q) \rightarrow \text{true}(p) \rightarrow (\text{true}(q) \rightarrow \text{true}(r)) \rightarrow \text{true}(r) \\ \text{Nec } E: & \Pi p: o. \Pi q: o. \text{valid}(\Box p) \rightarrow (\text{valid}(p) \rightarrow \text{valid}(q)) \rightarrow \text{valid}(q) \\ \text{Pos } E: & \Pi p: o. \Pi q: o. \text{valid}(\Box p) \rightarrow (\text{valid}(p) \rightarrow \text{valid}(q)) \rightarrow \text{valid}(q) \end{aligned}$$

$$C : \Pi p : o . \text{valid}(p) \rightarrow \text{true}(p) \quad \blacksquare$$

Higher-order Intuitionistic Logic

Definition A.4 (Σ_{HOIL})

The signature Σ_{HOIL} contains the following constants:

$$\begin{aligned}
& \text{holtype} : \text{Type} \\
& \iota : \text{holtype} \\
& o : \text{holtype} \\
& \Rightarrow : \text{holtype} \rightarrow \text{holtype} \rightarrow \text{holtype} \\
& \text{obj} : \text{holtype} \rightarrow \text{Type} \\
& \text{true} : \text{obj}(o) \rightarrow \text{Type} \\
& \top : \text{obj}(o) \\
& \wedge : \text{obj}(o \Rightarrow o \Rightarrow o) \\
& \vee : \text{obj}(o \Rightarrow o \Rightarrow o) \\
& \supset : \text{obj}(o \Rightarrow o \Rightarrow o) \\
& \forall_\sigma : \Pi s : \text{holtype} . \text{obj}((s \Rightarrow o) \Rightarrow o) \\
& \exists_\sigma : \Pi s : \text{holtype} . \text{obj}((s \Rightarrow o) \Rightarrow o) \\
& \Lambda : \Pi s : \text{holtype} . \Pi t : \text{holtype} . (\text{obj}(s) \rightarrow \text{obj}(t)) \rightarrow \text{obj}(s \Rightarrow t) \\
& \text{ap} : \Pi s : \text{holtype} . \Pi t : \text{holtype} . \text{obj}(s \Rightarrow t) \rightarrow \text{obj}(s) \rightarrow \text{obj}(t) \\
& = : \Pi s : \text{holtype} . \text{obj}(s \Rightarrow s \Rightarrow o) \\
& \text{TOP} : \text{true}(\top) \\
& \text{AND-R} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \text{true}(p) \rightarrow \text{true}(q) \rightarrow \text{true}(\text{ap}(\text{ap} \wedge p)q) \\
& \text{AND-L} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \Pi r : \text{obj}(o) . (\text{true}(p) \rightarrow \text{true}(q) \rightarrow \text{true}(r)) \\
& \quad \rightarrow (\text{true}(\text{ap}(\text{ap} \wedge p)q) \rightarrow \text{true}(r)) \\
& \text{OR-R1} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \text{true}(p) \rightarrow \text{true}(\text{ap}(\text{ap} \vee p)q) \\
& \text{OR-R2} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \text{true}(q) \rightarrow \text{true}(\text{ap}(\text{ap} \vee p)q) \\
& \text{OR-L} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \Pi r : \text{obj}(o) . (\text{true}(p) \rightarrow \text{true}(r)) \\
& \quad \rightarrow (\text{true}(p) \rightarrow \text{true}(r)) \rightarrow (\text{true}(\text{ap}(\text{ap} \vee p)q) \rightarrow \text{true}(r)) \\
& \text{IMP-R} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . (\text{true}(p) \rightarrow \text{true}(q)) \\
& \rightarrow \text{true}(\text{ap}(\text{ap} \wedge p)q) \text{IMP-L} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \Pi r : \text{obj}(o) . \text{true}(p)
\end{aligned}$$

$$\begin{aligned}
& \rightarrow (\text{true}(q) \rightarrow \text{true}(r)) \rightarrow (\text{true}(\text{ap}(\text{ap} \supset p)q) \rightarrow \text{true}(r)) \\
\text{FORALL-R: } & \Pi s : \text{holtype} . \Pi F : \text{obj}(s \Rightarrow o) . (\Pi x : \text{obj}(s) . \\
& \quad \text{true}(\text{ap}Fx)) \rightarrow \text{true}(\text{ap}\forall_s F) \\
\text{FORALL-L: } & \Pi s : \text{holtype} . \Pi F : \text{obj}(s \Rightarrow o) . \Pi r : \text{obj}(o) . \Pi x : \text{obj}(s) . \\
& \quad (\text{true}(\text{ap}Fx) \rightarrow \text{true}(r)) \rightarrow (\text{true}(\text{ap}\forall_s F) \rightarrow \text{true}(r)) \\
\text{EXISTS-R: } & \Pi s : \text{holtype} . \Pi F : \text{obj}(s \Rightarrow o) . \Pi x : \text{obj}(s) . \\
& \quad \text{true}(\text{ap}Fx) \rightarrow \text{true}(\text{ap}\exists_s F) \\
\text{EXISTS-L: } & \Pi s : \text{holtype} . \Pi F : \text{obj}(s \Rightarrow o) . \Pi r : \text{obj}(o) . (\Pi x : \text{obj}(s) . \\
& \quad \text{true}(\text{ap}Fx) \\
& \quad \rightarrow \text{true}(r)) \rightarrow (\text{true}(\text{ap}\exists_s F) \rightarrow \text{true}(r)) \\
\approx \equiv & \lambda s : \text{holtype} . \lambda x : . \text{obj}(s) . \text{ap}_{s,o}(\text{ap}_{s,s \Rightarrow o} =_s x)y : \Pi s : \text{holtype} . \\
& \quad \text{obj}(s) \rightarrow \text{obj}(s) \rightarrow \text{obj}(o) \\
\text{EQ: } & \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \text{true}(p) \rightarrow \text{true}(p \approx_o q) \rightarrow \text{true}(q) \\
\text{LAM: } & \Pi s, t : \text{holtype} . \Pi f, g : \text{obj}(s) \rightarrow \text{obj}(t) . (\Pi x : \text{obj}(s) . \text{true}(fx \approx_t gx)) \\
& \quad \rightarrow \text{true}(\Lambda_{s,t} \lambda x : \text{obj}(s) . fx \approx_{s \Rightarrow t} \Lambda_{s,t} \lambda x : \text{obj}(s) . gx) \\
\beta : & \Pi s, t : \text{holtype} . \Pi f : \text{obj}(s) \rightarrow \text{obj}(t) . \Pi x : \text{obj}(s) . \text{true}(\text{ap}_{s,t}(\Lambda_{s,t}(\lambda x : \text{obj}(s) \\
& \quad . fx))x \approx_t fx) \\
\eta : & \Pi s, t : \text{holtype} . \Pi f : \text{obj}(s \Rightarrow t) . \text{true}(\Lambda_{s,t}(\lambda x : \text{obj}(s) . \text{ap}_{s,t}fx) \approx_{s \Rightarrow t} f) \quad \blacksquare
\end{aligned}$$

Labelled KT

Definition A.5 (Σ_{KT})

The signature Σ_{KT} contains the following constants:

$$\begin{aligned}
U & : \text{Type} \\
o & : \text{Type} \\
\text{true} & : U \rightarrow o \rightarrow \text{Type} \\
\text{related} & : U \rightarrow U \rightarrow \text{Type} \\
\perp & : o \\
\supset & : o \rightarrow o \rightarrow o \\
\Box & : o \rightarrow o \\
\text{REFL} & : \Pi a : U . \text{related}(a)(a)
\end{aligned}$$

$BOT-E: \Pi p: o . \Pi a: U . \Pi b: U . (\text{true}(a)(p \supset \perp) \rightarrow \text{true}(b)\perp) \rightarrow \text{true}(a)p$

$IMP-I: \Pi p: o . \Pi q: o . \Pi a: U . (\text{true}(a)p \rightarrow \text{true}(a)q) \rightarrow \text{true}(a)(p \supset q)$

$IMP-E: \Pi p: o . \Pi q: o . \Pi r: o . \Pi a: U . \Pi b: U . \text{true}(a)(p \supset q) \rightarrow \text{true}(a)p \rightarrow$
 $(\text{true}(a)q \rightarrow \text{true}(b)r) \rightarrow \text{true}(b)(r)$

$BOX-I: \Pi p: o . \Pi a: U . (\Pi b: U . \text{related}(a)b \rightarrow \text{true}(b)p) \rightarrow \text{true}(a)(\Box p)$

$BOX-E: \Pi p: o . \Pi a: U . \Pi b: U . \text{true}(a)(\Box p) \rightarrow \text{related}(a)b \rightarrow \text{true}(b)p$ ■

Labelled KB

Definition A.6 (Σ_{KB})

The signature Σ_{KB} contains the following constants:

$U: \text{Type}$

$o: \text{Type}$

$\text{true}: U \rightarrow o \rightarrow \text{Type}$

$\text{related}: U \rightarrow U \rightarrow \text{Type}$

$\perp: o$

$\supset: o \rightarrow o \rightarrow o$

$\Box: o \rightarrow o$

$SYMM: \Pi a: U . \Pi b: o . \text{related}(a)(b) \rightarrow \text{related}(b)(a)$

$BOT-E: \Pi p: o . \Pi a: U . \Pi b: U . (\text{true}(a)(p \supset \perp) \rightarrow \text{true}(b)\perp) \rightarrow \text{true}(a)p$

$IMP-I: \Pi p: o . \Pi q: o . \Pi a: U . (\text{true}(a)p \rightarrow \text{true}(a)q) \rightarrow \text{true}(a)(p \supset q)$

$IMP-E: \Pi p: o . \Pi q: o . \Pi r: o . \Pi a: U . \Pi b: U . \text{true}(a)(p \supset q) \rightarrow \text{true}(a)p \rightarrow$
 $(\text{true}(a)q \rightarrow \text{true}(b)r) \rightarrow \text{true}(b)(r)$

$BOX-I: \Pi p: o . \Pi a: U . (\Pi b: U . \text{related}(a)b \rightarrow \text{true}(b)p) \rightarrow \text{true}(a)(\Box p)$

$BOX-E: \Pi p: o . \Pi a: U . \Pi b: U . \text{true}(a)(\Box p) \rightarrow \text{related}(a)b \rightarrow \text{true}(b)p$ ■

Labelled $K4$

Definition A.7 (Σ_{K4})

The signature Σ_{K4} contains the following constants:

$U: \text{Type}$

$o: \text{Type}$

$\text{true}: U \rightarrow o \rightarrow \text{Type}$

$\text{related}: U \rightarrow U \rightarrow \text{Type}$

$\perp: o$

$\supset: o \rightarrow o \rightarrow o$

$\Box: o \rightarrow o$

$\text{TRANS}: \Pi a:U . \Pi b:U . \Pi c:U . \text{related}(a)(b) \rightarrow \text{related}(b)(c) \rightarrow \text{related}(a)(c)$

$\text{BOT-E}: \Pi p:o . \Pi a:U . \Pi b:U . (\text{true}(a)(p \supset \perp) \rightarrow \text{true}(b)\perp) \rightarrow \text{true}(a)p$

$\text{IMP-I}: \Pi p:o . \Pi q:o . \Pi a:U . (\text{true}(a)p \rightarrow \text{true}(a)q) \rightarrow \text{true}(a)(p \supset q)$

$\text{IMP-E}: \Pi p:o . \Pi q:o . \Pi r:o . \Pi a:U . \Pi b:U . \text{true}(a)(p \supset q) \rightarrow \text{true}(a)p \rightarrow$
 $(\text{true}(a)q \rightarrow \text{true}(b)r) \rightarrow \text{true}(b)(r)$

$\text{BOX-I}: \Pi p:o . \Pi a:U . (\Pi b:U . \text{related}(a)b \rightarrow \text{true}(b)p) \rightarrow \text{true}(a)(\Box p)$

$\text{BOX-E}: \Pi p:o . \Pi a:U . \Pi b:U . \text{true}(a)(\Box p) \rightarrow \text{related}(a)b \rightarrow \text{true}(b)p$ ■

Labelled $K5$

Definition A.8 (Σ_{K5})

The signature Σ_{K5} contains the following constants:

$U: \text{Type}$

$o: \text{Type}$

$\text{true}: U \rightarrow o \rightarrow \text{Type}$

$\text{related}: U \rightarrow U \rightarrow \text{Type}$

$\perp: o$

$\supset: o \rightarrow o \rightarrow o$

$\Box: o \rightarrow o$

$\text{EUCL}: \Pi a:U . \Pi b:U . \Pi c:U . \text{related}(a)(b) \rightarrow \text{related}(b)(c) \rightarrow \text{related}(b)(a)$

$\text{BOT-E}: \Pi p:o . \Pi a:U . \Pi b:U . (\text{true}(a)(p \supset \perp) \rightarrow \text{true}(b)\perp) \rightarrow \text{true}(a)p$

$\text{IMP-I}: \Pi p:o . \Pi q:o . \Pi a:U . (\text{true}(a)p \rightarrow \text{true}(a)q) \rightarrow \text{true}(a)(p \supset q)$

$\text{IMP-E}: \Pi p:o . \Pi q:o . \Pi r:o . \Pi a:U . \Pi b:U . \text{true}(a)(p \supset q) \rightarrow \text{true}(a)p \rightarrow$
 $(\text{true}(a)q \rightarrow \text{true}(b)r) \rightarrow \text{true}(b)(r)$

$\text{BOX-I}: \Pi p:o . \Pi a:U . (\Pi b:U . \text{related}(a)b \rightarrow \text{true}(b)p) \rightarrow \text{true}(a)(\Box p)$

$\text{BOX-E}: \Pi p:o . \Pi a:U . \Pi b:U . \text{true}(a)(\Box p) \rightarrow \text{related}(a)b \rightarrow \text{true}(b)p$ ■

Labelled Classical Propositional Logic

Definition A.9 (Σ_{LCL})

The signature Σ_{LCL} contains the following constants:

$$U : \text{Type}$$

$$o : \text{Type}$$

$$\text{true} : U \rightarrow o \rightarrow \text{Type}$$

$$AND-I : \Pi a : U . \Pi p : o . \Pi q : o . \text{true}(a)p \rightarrow \text{true}(a)q \rightarrow \text{true}(a)p \wedge q$$

$$AND-E : \Pi a : U . \Pi b : U . \Pi p : o . \Pi q : o . \Pi r : o . \text{true}(a)p \wedge q \rightarrow ((\text{true}(a)p \rightarrow \text{true}(a)q) \rightarrow \text{true}(b)r) \rightarrow \text{true}(b)r$$

$$OR1-I : \Pi a : U . \Pi p : o . \Pi q : o . \text{true}(a)p \rightarrow \text{true}(a)p \vee q$$

$$OR2-I : \Pi a : U . \Pi p : o . \Pi q : o . \text{true}(a)q \rightarrow \text{true}(a)p \vee q$$

$$OR-E : \Pi a : U . \Pi b : U . \Pi p : o . \Pi q : o . \Pi r : o . \text{true}(a)p \vee q \rightarrow (\text{true}(a)p \rightarrow \text{true}(b)r) \rightarrow (\text{true}(a)q \rightarrow \text{true}(b)r) \rightarrow \text{true}(b)r$$

$$IMP-I : \Pi a : U . \Pi p : o . \Pi q : o . (\text{true}(a)p \rightarrow \text{true}(a)q) \rightarrow \text{true}(a)p \supset q$$

$$IMP-E : \Pi a : U . \Pi b : U . \Pi p : o . \Pi q : o . \Pi r : o . \text{true}(a)p \supset q \rightarrow \text{true}(a)p \rightarrow (\text{true}(a)q \rightarrow \text{true}(b)r) \rightarrow \text{true}(b)r$$

$$NEG-I : \Pi a : U . \Pi b : U . \Pi p : o . (\text{true}(a)p \rightarrow \text{true}(b)\perp) \rightarrow \text{true}(a)\neg p$$

$$BOT-E : \Pi a : U . \Pi b : U . \Pi p : o . \text{true}(b)\perp \rightarrow \text{true}(a)p$$

■

Labelled Intuitionistic Propositional Logic

Definition A.10 (Σ_{LIL})

The signature Σ_{LIL} contains the following constants:

$$U : \text{Type}$$

$$o : \text{Type}$$

$$\text{true} : U \rightarrow o \rightarrow \text{Type}$$

$$\text{related} : U \rightarrow U \rightarrow \text{Type}$$

$$AND-I : \Pi a : U . \Pi p : o . \Pi q : o . \text{proof}(a)p \rightarrow \text{proof}(a)q \rightarrow \text{proof}(a)p \wedge q$$

$$AND-E : \Pi a : U . \Pi b : U . \Pi p : o . \Pi q : o . \Pi r : o . \text{proof}(a)p \wedge q \rightarrow ((\text{proof}(a)p \rightarrow \text{proof}(a)q) \rightarrow \text{proof}(b)r) \rightarrow \text{proof}(b)r$$

$$OR1-I : \Pi a : U . \Pi p : o . \Pi q : o . \text{proof}(a)p \rightarrow \text{proof}(a)p \vee q$$

$$OR2-I : \Pi a : U . \Pi p : o . \Pi q : o . \text{proof}(a)q \rightarrow \text{proof}(a)p \vee q$$

$$OR-E: \Pi a:U . \Pi b:U . \Pi p:o . \Pi q:o . \Pi r:o . \text{proof}(a)p \vee q \rightarrow (\text{proof}(a)p \rightarrow \text{proof}(b)r) \rightarrow (\text{proof}(a)q \rightarrow \text{proof}(b)r) \rightarrow \text{proof}(b)r$$

$$IMP-I: \Pi a:U . \Pi b:U . \Pi p:o . \Pi q:o . ((\text{proof}(b)p \rightarrow \text{related}(a)b) \rightarrow \text{proof}(b)q) \rightarrow \text{proof}(a)p \supset q$$

$$IMP-E: \Pi a:U . \Pi b:U . \Pi p:o . \Pi q:o . \text{proof}(a)p \supset q \rightarrow \text{proof}(b)p \rightarrow \text{related}(a)b \rightarrow \text{proof}(b)q$$

$$BOT-E: \Pi a:U . \Pi b:U . \Pi p:o . \text{proof}(a)\perp \rightarrow \text{proof}(b)p \quad \blacksquare$$

Labelled Minimal Propositional Logic

Definition A.11 (Σ_{LML})

The signature Σ_{LML} contains the following constants:

$$U: \text{Type}$$

$$o: \text{Type}$$

$$\text{true}: U \rightarrow o \rightarrow \text{Type}$$

$$AND-I: \Pi a:U . \Pi p:o . \Pi q:o . \text{proof}(a)p \rightarrow \text{proof}(a)q \rightarrow \text{proof}(a)p \wedge q$$

$$AND-E: \Pi a:U . \Pi b:U . \Pi p:o . \Pi q:o . \Pi r:o . \text{proof}(a)p \wedge q \rightarrow ((\text{proof}(a)p \rightarrow \text{proof}(a)q) \rightarrow \text{proof}(b)r) \rightarrow \text{proof}(b)r$$

$$OR1-I: \Pi a:U . \Pi p:o . \Pi q:o . \text{proof}(a)p \rightarrow \text{proof}(a)p \vee q$$

$$OR2-I: \Pi a:U . \Pi p:o . \Pi q:o . \text{proof}(a)q \rightarrow \text{proof}(a)p \vee q$$

$$OR-E: \Pi a:U . \Pi b:U . \Pi p:o . \Pi q:o . \Pi r:o . \text{proof}(a)p \vee q \rightarrow (\text{proof}(a)p \rightarrow \text{proof}(b)r) \rightarrow (\text{proof}(a)q \rightarrow \text{proof}(b)r) \rightarrow \text{proof}(b)r$$

$$IMP-I: \Pi a:U . \Pi b:U . \Pi p:o . \Pi q:o . ((\text{proof}(b)p \rightarrow \text{related}(a)b) \rightarrow \text{proof}(b)q) \rightarrow \text{proof}(a)p \supset q$$

$$IMP-E: \Pi a:U . \Pi b:U . \Pi p:o . \Pi q:o . \text{proof}(a)p \supset q \rightarrow \text{proof}(b)p \rightarrow \text{related}(a)b \rightarrow \text{proof}(b)q \quad \blacksquare$$

Sequent Intuitionistic Logic

Definition A.12 ($\Sigma_{G_{3i}}$)

$\Sigma_{G_{3i}}$ is the $\lambda\Pi$ -signature containing the following constants:

$$o: \text{Type}$$

$$\iota: \text{Type}$$

$$\begin{aligned}
& \text{proof} : o \rightarrow \text{Type} \\
& \top : o \\
& \wedge : o \rightarrow o \rightarrow o \\
& \vee : o \rightarrow o \rightarrow o \\
& \supset : o \rightarrow o \rightarrow o \\
& \neg : o \rightarrow o \\
& \forall : (\iota \rightarrow o) \rightarrow o \\
& \exists : (\iota \rightarrow o) \rightarrow o \\
& \text{AND-L} : \Pi p, q, r : o . (\text{proof}(p) \rightarrow \text{proof}(q) \rightarrow \text{proof}(r)) \\
& \quad \rightarrow (\text{proof}(p \wedge q) \rightarrow \text{proof}(r)) \\
& \text{AND-R} : \Pi p, q : o . \text{proof}(p) \rightarrow \text{proof}(q) \rightarrow \text{proof}(p \wedge q) \\
& \text{OR-L} : \Pi p, q, r : o . (\text{proof}(p) \rightarrow \text{proof}(r) \rightarrow \text{proof}(q) \rightarrow \text{proof}(r)) \\
& \quad \rightarrow (\text{proof}(p \vee q) \rightarrow \text{proof}(r)) \\
& \text{OR-R}_1 : \Pi p, q : o . \text{proof}(p) \rightarrow \text{proof}(p \vee q) \\
& \text{OR-R}_2 : \Pi p, q : o . \text{proof}(q) \rightarrow \text{proof}(p \vee q) \\
& \text{IMP-L} : \Pi p, q, r : o . (\text{proof}(p) \rightarrow \text{proof}(q) \rightarrow \text{proof}(r)) \\
& \quad \rightarrow (\text{proof}(p \supset q) \rightarrow \text{proof}(r)) \\
& \text{IMP-R} : \Pi p, q : o . (\text{proof}(p) \rightarrow \text{proof}(q)) \rightarrow \text{proof}(p \supset q) \\
& \text{TOP} : \text{proof}(\top) \\
& \text{FORALL-L} : \Pi F : \iota \rightarrow o . \Pi r : o . \Pi x : \iota . (\text{proof}(Fx) \rightarrow \text{proof}(r)) \\
& \quad \rightarrow (\text{proof}(\forall(\lambda x : \iota . Fx)) \rightarrow \text{proof}(r)) \\
& \text{FORALL-R} : \Pi F : \iota \rightarrow o . (\Pi x : \iota . \text{proof}(Fx)) \rightarrow \text{proof}(\forall(\lambda x : \iota . Fx)) \\
& \text{EXISTS-L} : \Pi F : \iota \rightarrow o . \Pi r : o . (\Pi x : \iota . \text{proof}(Fx) \rightarrow \text{proof}(r)) \\
& \quad \rightarrow (\text{proof}(\exists(\lambda x : \iota . Fx)) \rightarrow \text{proof}(r)) \\
& \text{EXISTS-R} : \Pi F : \iota \rightarrow o . \Pi x : \iota . \text{proof}(Fx) \rightarrow \text{proof}(\exists(\lambda x : \iota . Fx)) \quad \blacksquare
\end{aligned}$$

Sequent Classical Logic

Definition A.13 ($\Sigma_{G_{3c}}$)

$\Sigma_{G_{3c}}$ is the $\lambda\Pi$ -signature which contains the following constants:

$\iota : \text{Type}$

$$\begin{aligned}
& o: \text{Type} \\
& \#: \text{Type} \\
& \text{ant}: o \rightarrow \text{Type} \\
& \text{suc}: o \rightarrow \text{Type} \\
& \top: o \\
& \wedge: o \rightarrow o \rightarrow o \\
& \vee: o \rightarrow o \rightarrow o \\
& \supset: o \rightarrow o \rightarrow o \\
& \neg: o \rightarrow o \\
& \forall: (\iota \rightarrow o) \rightarrow o \\
& \exists: (\iota \rightarrow o) \rightarrow o \\
& \text{HYP}: \Pi p: o. \text{ant}(p) \rightarrow \text{suc}(p) \rightarrow \# \\
& \text{TOP}: \text{ant}(\top) \rightarrow \# \\
& \text{AND-L}: \Pi p: o. \Pi q: o. (\text{ant}(p) \rightarrow \text{ant}(q) \rightarrow \#) \rightarrow (\text{ant}(p \wedge q) \rightarrow \#) \\
& \text{AND-R}: \Pi p: o. \Pi q: o. (\text{suc}(p) \rightarrow \#) \rightarrow (\text{suc}(q) \rightarrow \#) \rightarrow (\text{suc}(p \wedge q) \rightarrow \#) \\
& \text{OR-L}: \Pi p: o. \Pi q: o. (\text{ant}(p) \rightarrow \#) \rightarrow (\text{ant}(q) \rightarrow \#) \rightarrow (\text{ant}(p \vee q) \rightarrow \#) \\
& \text{OR-R}: \Pi p: o. \Pi q: o. (\text{suc}(p) \rightarrow \text{suc}(q) \rightarrow \#) \rightarrow (\text{suc}(p \vee q) \rightarrow \#) \\
& \text{IMP-L}: \Pi p: o. \Pi q: o. (\text{suc}(p) \rightarrow \#) \rightarrow (\text{ant}(q) \rightarrow \#) \rightarrow (\text{ant}(p \supset q) \rightarrow \#) \\
& \text{IMP-R}: \Pi p: o. \Pi q: o. (\text{ant}(p) \rightarrow \text{suc}(q) \rightarrow \#) \rightarrow (\text{suc}(p \supset q) \rightarrow \#) \\
& \text{NEG-L}: \Pi p: o. (\text{suc}(p) \rightarrow \#) \rightarrow (\text{ant}(\neg p) \rightarrow \#) \\
& \text{NEG-R}: \Pi p: o. (\text{ant}(p) \rightarrow \#) \rightarrow (\text{suc}(\neg p) \rightarrow \#) \\
& \text{FORALL-L}: \Pi F: \iota \rightarrow o. \Pi x: \iota. (\text{ant}(Fx) \rightarrow \#) \rightarrow (\text{ant}(\forall(\lambda x: \iota. Fx)) \rightarrow \#) \\
& \text{FORALL-R}: \Pi F: \iota \rightarrow o. (\Pi x: \iota. \text{suc}(Fx) \rightarrow \#) \rightarrow (\text{suc}(\forall(\lambda x: \iota. Fx)) \rightarrow \#) \\
& \text{EXISTS-L}: \Pi F: \iota \rightarrow o. (\Pi x: \iota. \text{ant}(Fx) \rightarrow \#) \rightarrow (\text{ant}(\exists(\lambda x: \iota. Fx)) \rightarrow \#) \\
& \text{EXISTS-R}: \Pi F: \iota \rightarrow o. \Pi x: \iota. (\text{suc}(Fx) \rightarrow \#) \rightarrow (\text{suc}(\exists(\lambda x: \iota. Fx)) \rightarrow \#) \blacksquare
\end{aligned}$$

Sequent Higher-order Intuitionistic Logic

Definition A.14 (Σ_{G_3HOIL})

The $\lambda\Pi$ -signature Σ_{G_3HOIL} contains the following constants:

$$holtype: \text{Type}$$

$$\begin{aligned}
& \iota : \text{holtype} \\
& o : \text{holtype} \\
& \Rightarrow : \text{holtype} \rightarrow \text{holtype} \rightarrow \text{holtype} \\
& \text{obj} : \text{holtype} \rightarrow \text{Type} \\
& \top : \text{obj}(o) \\
& \wedge : \text{obj}(o \Rightarrow o \Rightarrow o) \\
& \vee : \text{obj}(o \Rightarrow o \Rightarrow o) \\
& \supset : \text{obj}(o \Rightarrow o \Rightarrow o) \\
& \forall_\sigma : \Pi s : \text{holtype} . \text{obj}((s \Rightarrow o) \Rightarrow o) \\
& \exists_\sigma : \Pi s : \text{holtype} . \text{obj}((s \Rightarrow o) \Rightarrow o) \\
& \Lambda : \Pi s : \text{holtype} . \Pi t : \text{holtype} . (\text{obj}(s) \rightarrow \text{obj}(t)) \rightarrow \text{obj}(s \Rightarrow t) \\
& \text{ap} : \Pi s : \text{holtype} . \Pi t : \text{holtype} . \text{obj}(s \Rightarrow t) \rightarrow \text{obj}(s) \rightarrow \text{obj}(t) \\
& = : \Pi s : \text{holtype} . \text{obj}(s \Rightarrow s \Rightarrow o) \\
& \text{proof} : \text{obj}(o) \rightarrow \text{Type} \\
& \text{TOP} : \text{proof}(\top) \\
& \text{AND-R} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \text{proof}(p) \rightarrow \text{proof}(q) \rightarrow \text{proof}(\text{ap}(\text{ap} \wedge p)q) \\
& \text{AND-L} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \Pi r : \text{obj}(o) . (\text{proof}(p) \rightarrow \text{proof}(q) \rightarrow \text{proof}(r)) \\
& \quad \rightarrow (\text{proof}(\text{ap}(\text{ap} \wedge p)q) \rightarrow \text{proof}(r)) \\
& \text{OR-R1} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \text{proof}(p) \rightarrow \text{proof}(\text{ap}(\text{ap} \vee p)q) \\
& \text{OR-R2} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \text{proof}(q) \rightarrow \text{proof}(\text{ap}(\text{ap} \vee p)q) \\
& \text{OR-L} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \Pi r : \text{obj}(o) . (\text{proof}(p) \rightarrow \text{proof}(r)) \\
& \quad \rightarrow (\text{proof}(p) \rightarrow \text{proof}(r)) \rightarrow (\text{proof}(\text{ap}(\text{ap} \vee p)q) \rightarrow \text{proof}(r)) \\
& \text{IMP-R} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . (\text{proof}(p) \rightarrow \text{proof}(q)) \\
& \rightarrow \text{proof}(\text{ap}(\text{ap} \wedge p)q) \text{IMP-L} : \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \Pi r : \text{obj}(o) . \text{proof}(p) \\
& \quad \rightarrow (\text{proof}(q) \rightarrow \text{proof}(r)) \rightarrow (\text{proof}(\text{ap}(\text{ap} \supset p)q) \rightarrow \text{proof}(r)) \\
& \text{FORALL-R} : \Pi s : \text{holtype} . \Pi F : \text{obj}(s \Rightarrow o) . (\Pi x : \text{obj}(s) . \\
& \quad \text{proof}(\text{ap}F x)) \rightarrow \text{proof}(\text{ap}\forall_s F) \\
& \text{FORALL-L} : \Pi s : \text{holtype} . \Pi F : \text{obj}(s \Rightarrow o) . \Pi r : \text{obj}(o) . \Pi x : \text{obj}(s) . \\
& \quad (\text{proof}(\text{ap}F x) \rightarrow \text{proof}(r)) \rightarrow (\text{proof}(\text{ap}\forall_s F) \rightarrow \text{proof}(r))
\end{aligned}$$

$$\begin{aligned}
& \text{EXISTS-R: } \Pi s : \text{holtype} . \Pi F : \text{obj}(s \Rightarrow o) . \Pi x : \text{obj}(s) . \\
& \quad \text{proof}(\text{ap}F x) \rightarrow \text{proof}(\text{ap}\exists_s F) \\
& \text{EXISTS-L: } \Pi s : \text{holtype} . \Pi F : \text{obj}(s \Rightarrow o) . \Pi r : \text{obj}(o) . (\Pi x : \text{obj}(s) . \\
& \quad \text{proof}(\text{ap}F x) \\
& \quad \rightarrow \text{proof}(r)) \rightarrow (\text{proof}(\text{ap}\exists_s F) \rightarrow \text{proof}(r)) \\
& \approx \equiv \lambda s : \text{holtype} . \lambda x : . \text{obj}(s) . \text{ap}_{s,o}(\text{ap}_{s,s \Rightarrow o} =_s x)y : \Pi s : \text{holtype} . \\
& \quad \text{obj}(s) \rightarrow \text{obj}(s) \rightarrow \text{obj}(o) \\
& \text{EQ: } \Pi p : \text{obj}(o) . \Pi q : \text{obj}(o) . \text{proof}(p) \rightarrow \text{proof}(p \approx_o q) \rightarrow \text{proof}(q) \\
& \text{LAM: } \Pi s, t : \text{holtype} . \Pi f, g : \text{obj}(s) \rightarrow \text{obj}(t) . (\Pi x : \text{obj}(s) . \text{proof}(fx \approx_t gx)) \\
& \quad \rightarrow \text{proof}(\Lambda_{s,t} \lambda x : \text{obj}(s) . fx \approx_{s \Rightarrow t} \Lambda_{s,t} \lambda x : \text{obj}(s) . gx) \\
& \beta : \Pi s, t : \text{holtype} . \Pi f : \text{obj}(s) \rightarrow \text{obj}(t) . \Pi x : \text{obj}(s) . \text{proof}(\text{ap}_{s,t}(\Lambda_{s,t}(\lambda x : \text{obj}(s) \\
& \quad . fx))x \approx_t fx) \\
& \eta : \Pi s, t : \text{holtype} . \Pi f : \text{obj}(s \Rightarrow t) . \text{proof}(\Lambda_{s,t}(\lambda x : \text{obj}(s) . \text{ap}_{s,t}fx) \approx_{s \Rightarrow t} f) \quad \blacksquare
\end{aligned}$$

Appendix B

Examples of Labelled Natural Deductive Systems

In this appendix, we provide labelled natural deduction presentations of minimal, intuitionistic and classical logic.

Definition B.1 (Minimal Logic)

The labelled natural deduction system for minimal logic has a base system consisting of the following inference rules:

$$\begin{array}{c}
 \frac{\text{proof}(w, \phi) \quad \text{proof}(w, \psi)}{\text{proof}(w, \phi \supset \psi)} \wedge I \quad \frac{\begin{array}{c} [\text{proof}(w, \phi)] [\text{proof}(w, \psi)] \\ \vdots \\ \text{proof}(z, \tau) \end{array}}{\text{proof}(z, \tau)} \wedge E \\
 \\
 \frac{\text{proof}(w, \phi)}{\text{proof}(w, \phi \vee \psi)} \vee I1 \quad \frac{\text{proof}(w, \psi)}{\text{proof}(w, \phi \vee \psi)} \vee I2 \\
 \frac{\begin{array}{c} [\text{proof}(w, \phi)] \quad [\text{proof}(w, \psi)] \\ \vdots \quad \vdots \\ \text{proof}(w, \phi \vee \psi) \quad \text{proof}(z, \tau) \quad \text{proof}(z, \tau) \end{array}}{\text{proof}(z, \tau)} \vee E \\
 \\
 \frac{\begin{array}{c} [\text{proof}(z, \phi)] [\text{related}_{\leq}(w, z)] \\ \vdots \\ \text{proof}(z, \psi) \end{array}}{\text{proof}(w, \phi \supset \psi)} \supset I \quad \frac{\text{proof}(w, \phi \supset \psi) \quad \text{proof}(z, \phi) \quad \text{related}_{\leq}(w, z)}{\text{proof}(z, \psi)} \supset E
 \end{array}$$

and no Horn relational theory. ■

Definition B.2 (Intuitionistic Logic)

The labelled natural deduction system for intuitionistic logic has a base system consisting of the following inference rules:

$$\begin{array}{c}
 \frac{\text{proof}(w, \phi) \quad \text{proof}(w, \psi)}{\text{proof}(w, \phi \supset \psi)} \wedge I \quad \frac{\begin{array}{c} [\text{proof}(w, \phi)] [\text{proof}(w, \psi)] \\ \vdots \\ \text{proof}(z, \tau) \end{array}}{\text{proof}(z, \tau)} \wedge E \\
 \\
 \frac{\text{proof}(w, \phi)}{\text{proof}(w, \phi \vee \psi)} \vee I1 \quad \frac{\text{proof}(w, \psi)}{\text{proof}(w, \phi \vee \psi)} \vee I2 \\
 \frac{\begin{array}{c} [\text{proof}(w, \phi)] \quad [\text{proof}(w, \psi)] \\ \vdots \\ \text{proof}(z, \tau) \quad \text{proof}(z, \tau) \end{array}}{\text{proof}(z, \tau)} \vee E \\
 \\
 \frac{\begin{array}{c} [\text{proof}(z, \phi)] [\text{related}_{\leq}(w, z)] \\ \vdots \\ \text{proof}(z, \psi) \end{array}}{\text{proof}(w, \phi \supset \psi)} \supset I \quad \frac{\text{proof}(w, \phi \supset \psi) \quad \text{proof}(z, \phi) \quad \text{related}_{\leq}(w, z)}{\text{proof}(z, \psi)} \supset E \\
 \\
 \frac{\text{proof}(w, \perp)}{\text{proof}(z, \phi)} \perp E
 \end{array}$$

and no Horn relational theory. ■

Definition B.3 (Classical Logic)

The labelled natural deduction system for classical logic has a base system consisting of the following inference rules:

$$\begin{array}{c}
 \frac{\text{true}(w, \phi) \quad \text{true}(w, \psi)}{\text{true}(w, \phi \supset \psi)} \wedge I \quad \frac{\begin{array}{c} [\text{true}(w, \phi)] [\text{true}(w, \psi)] \\ \vdots \\ \text{true}(z, \tau) \end{array}}{\text{true}(z, \tau)} \wedge E \\
 \\
 \frac{\text{true}(w, \phi)}{\text{true}(w, \phi \vee \psi)} \vee I1 \quad \frac{\text{true}(w, \psi)}{\text{true}(w, \phi \vee \psi)} \vee I2
 \end{array}$$

$$\begin{array}{c}
\frac{\frac{\frac{[\text{true}(w, \phi)]}{\vdots} \quad \frac{[\text{true}(w, \psi)]}{\vdots}}{\text{true}(w, \phi \vee \psi) \quad \text{true}(z, \tau) \quad \text{true}(z, \tau)} \vee E}{\text{true}(z, \tau)} \\
\frac{\frac{[\text{true}(w, \phi)]}{\vdots} \quad \text{true}(w, \psi)}{\text{true}(w, \phi \supset \psi)} \supset I \quad \frac{\text{true}(w, \phi \supset \psi) \quad \text{true}(w, \phi) \quad \frac{[\text{true}(w, \phi)]}{\vdots} \quad \text{true}(z, \tau)}{\text{true}(z, \tau)} \supset E}{\text{true}(z, \perp)} \neg I \\
\frac{\text{true}(z, \perp)}{\text{true}(w, \phi)} \perp E
\end{array}$$

and no Horn relational theory. ■

Appendix C

Various Sequent Calculi

In this appendix we present sequent systems for LJ , LK

Definition C.1 (LJ)

The judged proof system LJ is given by the alphabet $A = (S, V, E, C, J)$, where:

- $S = \{o, \iota\}$;
- $V = \{\iota\}$;
- $E = \{\top, \wedge, \vee, \supset, \forall, \exists\}$;
- $C = \{\top, \wedge, \vee, \supset, \forall, \exists\}$;
- $J = \{\text{proof}\}$.

Each connective is assigned an arity and a level. \top has arity o and level 1, \wedge, \vee and \supset all have arity $(o, o) \rightarrow o$ and level 1. \forall and \exists have arity $(\iota \rightarrow o) \rightarrow o$ and level 2. The judgement **proof** has arity 0. Together with the following rules:

$$\begin{array}{c} \frac{}{\Delta, \text{proof}(\phi) \vdash_{LJ} \text{proof}(\phi)} Ax \\ \frac{\Delta, \text{proof}(\phi), \text{proof}(\psi) \vdash_{LJ} \text{proof}(\chi)}{\Delta, \text{proof}(\phi \wedge \psi) \vdash_{LJ} \text{proof}(\chi)} \wedge l \\ \frac{\Delta \vdash_{LJ} \text{proof}(\phi) \quad \Delta \vdash_{LJ} \text{proof}(\psi)}{\Delta \vdash_{LJ} \text{proof}(\phi \wedge \psi)} \wedge r \\ \frac{\Delta, \text{proof}(\phi) \vdash_{LJ} \text{proof}(\chi) \quad \Delta, \text{proof}(\phi \vee \psi), \text{proof}(\psi) \vdash_{LJ} \text{proof}(\chi)}{\Delta, \text{proof}(\phi \vee \psi) \vdash_{LJ} \text{proof}(\chi)} \vee l \\ \frac{\Delta \vdash_{LJ} \text{proof}(\phi_i)}{\Delta \vdash_{LJ} \text{proof}(\phi_1 \vee \phi_2)} \vee r_i \end{array}$$

$$\begin{array}{c}
\frac{\Delta \vdash_{LJ} \text{proof}(\phi) \quad \Delta, \text{proof}(\phi \supset \psi), \text{proof}(\psi) \vdash_{LJ} \text{proof}(\chi)}{\Delta, \text{proof}(\phi \supset \psi) \vdash_{LJ} \text{proof}(\chi)} \supset l \\
\frac{\Delta, \text{proof}(\phi) \vdash_{LJ} \text{proof}(\psi)}{\delta \vdash_{LJ} \text{proof}(\phi \supset \psi)} \supset r \\
\frac{}{\Delta \vdash_{LJ} \top} \top \\
\frac{\Delta, \text{proof}(\phi[x]) \vdash_{LJ} \text{proof}(\chi)}{\Delta, \text{proof}(\forall x\phi) \vdash_{LJ} \text{proof}(\chi)} \forall l \\
\frac{\Delta \vdash_{LJ} \text{proof}(\phi[x])}{\Delta \vdash_{LJ} \text{proof}(\forall x\phi)} \forall r \\
\frac{\Delta, \text{proof}(\phi[x]) \vdash_{LJ} \text{proof}(\chi)}{\Delta, \text{proof}(\exists x\phi) \vdash_{LJ} \text{proof}(\chi)} \exists l \\
\frac{\Delta \vdash_{LJ} \text{proof}(\phi[x])}{\Delta \vdash_{LJ} \text{proof}(\exists x\phi)} \exists r \\
\frac{\Delta \vdash_{LJ} \text{proof}(\phi)}{\Delta, \text{proof}(\psi) \vdash_{LJ} \text{proof}(\chi)} \text{weakening} \quad \blacksquare
\end{array}$$

Definition C.2 (LK)

The judged proof system LK is given by the alphabet $A = (S, V, E, C, J)$ where

- $S = \{o, \iota\}$,
- $V = \{\iota\}$,
- $E = \{\top, \wedge, \vee, \supset, \neg, \forall, \exists\}$,
- $C = \{\wedge, \vee, \supset, \neg, \forall, \exists\}$,
- $J = \{\text{ant}, \text{suc}\}$.

Each connective is assigned an arity and a level. \top has arity o and level 0, \neg has arity $o \rightarrow o$ and level 1, \wedge, \vee, \supset have arity $(o, o) \rightarrow o$ and level 1 and \forall and \exists have arity $(\iota \rightarrow o) \rightarrow o$ and level 2. The judgements both have arity 0. Together with the rules:

$$\begin{array}{c}
\frac{}{\Delta, \text{ant}(\phi) \vdash_{LK} \text{suc}(\phi)\Theta} Ax \\
\frac{}{\Delta \vdash_{LK} \text{suc}(\top), \Theta} \top \\
\frac{\Delta, \text{ant}(\phi), \text{ant}(\psi) \vdash_{LK} \Theta}{\Delta, \text{ant}(\phi \wedge \psi) \vdash_{LK} \Theta} \wedge l
\end{array}$$

$$\begin{array}{c}
\frac{\Delta \vdash_{LK} \text{suc}(\phi), \Theta \quad \Delta \vdash_{LK} \text{suc}(\psi), \Theta}{\Delta \vdash_{LK} \text{suc}(\phi \wedge \psi), \Theta} \wedge r \\
\frac{\Delta, \text{ant}(\phi) \vdash_{LK} \Theta \quad \Delta, \text{ant}(\psi) \vdash_{LK} \Theta}{\Delta, \text{ant}(\phi \vee \psi) \vdash_{LK} \Theta} \vee l \\
\frac{\Delta \vdash_{LK} \text{suc}(\phi_i), \Theta}{\Delta \vdash_{LK} \text{suc}(\phi_1 \vee \phi_2), \Theta} \vee r_i \\
\frac{\Delta \vdash_{LK} \text{suc}(\phi), \Theta \quad \Delta, \text{ant}(\psi) \vdash_{LK} \Theta}{\Delta, \text{ant}(\phi \supset \psi) \vdash_{LK} \Theta} \supset l \\
\frac{\Delta, \text{ant}(\phi) \vdash_{LK} \text{suc}(\psi), \Theta}{\Delta \vdash_{LK} \text{suc}(\phi \supset \psi), \Theta} \supset r \\
\frac{\Delta \vdash_{LK} \text{suc}(\phi), \Theta}{\Delta, \text{ant}(\neg\phi) \vdash_{LK} \Theta} \neg l \\
\frac{\Delta, \text{ant}(\phi) \vdash_{LK} \Theta}{\Delta \vdash_{LK} \text{suc}(\neg\phi), \Theta} \neg r \\
\frac{\Delta, \text{ant}(\phi[x]) \vdash_{LK} \Theta}{\Delta, \text{ant}(\forall x\phi) \vdash_{LK} \Theta} \forall l \\
\frac{\Delta \vdash_{LK} \text{suc}(\phi[x]), \Theta}{\Delta \vdash_{LK} \text{suc}(\forall x\phi), \Theta} \forall r \\
\frac{\Delta, \text{ant}(\phi[x]) \vdash_{LK} \Theta}{\Delta, \text{ant}(\exists x\phi) \vdash_{LK} \Theta} \exists l \\
\frac{\Delta \vdash_{LK} \text{suc}(\phi[x]), \Theta}{\Delta \vdash_{LK} \text{suc}(\exists x\phi), \Theta} \exists r \\
\frac{\Delta \vdash_{LK} \Theta}{\Delta, \text{ant}(\phi) \vdash_{LK} \text{suc}(\phi), \Theta} \textit{weakening}
\end{array}$$

■

Definition C.3 (Miller's *HOIL* (Miller et al. (1991)))

HOIL consists of well-formed formulæ, inference and structural rules for sequents. We begin by defining the well-formed formulæ. To do this we need a language to generate them. This language consists of types, defined by the grammar:

$$\textit{Types} := \iota \mid o \mid \sigma \rightarrow \tau$$

and expression symbols $\wedge, \vee, \supset, \top, \forall_\sigma, \exists_\sigma, \lambda_\sigma$ each with a type. \top has type o ; \wedge, \vee and \supset have type $o \rightarrow o \rightarrow o$, \forall_σ and \exists_σ have type $(\sigma \rightarrow o) \rightarrow o$ and λ_σ has

type $\sigma \rightarrow \tau \rightarrow (\sigma \rightarrow \tau)$. Every expression symbol is a proper symbol apart from λ_σ . For a given type there are countably many variables of that type. Every variable is a proper formula. The well-formed formulæ are the smallest class of formulæ which satisfy the following rules:

1. Every proper formula is a well-formed formula with type as defined above;
2. If x is a variable with type β and M is a well-formed formula with type α then $\lambda_\beta x . M$ is a well-formed formula with type $\beta \rightarrow \alpha$;
3. If F and A are well-formed formulæ with type $\alpha \rightarrow \beta$ and type α respectively then FA is a well-formed formula with type β .

We omit the subscript on the λ_σ and abbreviate $\forall_\sigma \lambda_\sigma x . \phi$ to $\forall x . \phi$ and $\exists_\sigma \lambda_\sigma x . \phi$ to $\exists x . \phi$ when it is clear from the context what the intended meaning is. We say that a well-formed formula has been (λ -)converted if one of the following conversions has been applied:

- (α) A well-formed formula $\lambda x . M$, where x has type β and M has type α can be converted to a well formed-formula $\lambda y . M'$, where y has type β and M' has type α and M' is the result of substituting y for x in M , provided y does not occur in M and x is not bound in M ;
- (β) A well-formed formula $(\lambda x . M)N$, where x and N have type β and M has type α , can be converted to M' of type α , where M' is the result of substituting N for x in M , provided that the bound variables of M are distinct from x and N ;
- (η) A well-formed formula $\lambda x . (Mx)$, where x has type α and M has type $\alpha \rightarrow \beta$, can be converted to M provided y does not occur free in M .

We now define the inference rules, where for each rule Δ is a set of well-formed formulæ. ϕ, ψ and χ are well-formed formulæ:

$$\begin{array}{c}
\frac{}{\Gamma \vdash_{HOIL} \top} \top \\
\\
\frac{}{\Gamma, \phi \vdash_{HOIL} \phi} Ax \\
\\
\frac{\Delta, \phi, \psi \vdash_{HOIL} \chi}{\Delta, \phi \wedge \psi \vdash_{HOIL} \chi} \wedge L \\
\\
\frac{\Delta \vdash_{HOIL} \phi \quad \Delta \vdash_{HOIL} \psi}{\Delta \vdash_{HOIL} \phi \wedge \psi} \wedge R \\
\\
\frac{\Delta, \phi \vdash_{HOIL} \chi \quad \Delta, \psi \vdash_{HOIL} \chi}{\Delta, \phi \vee \psi \vdash_{HOIL} \chi} \vee L
\end{array}$$

$$\begin{array}{c}
\frac{\Delta \vdash_{HOIL} \phi}{\Delta \vdash_{HOIL} \phi \vee \psi} \vee R1 \\
\frac{\Delta \vdash_{HOIL} \psi}{\Delta \vdash_{HOIL} \phi \vee \psi} \vee R2 \\
\frac{\Delta \vdash_{HOIL} \phi \quad \Delta, \psi \vdash_{HOIL} \chi}{\Delta, \phi \supset \psi \vdash_{HOIL} \chi} \supset L \\
\frac{\Delta, \phi \vdash_{HOIL} \psi}{\Delta \vdash_{HOIL} \phi \supset \psi} \supset R \\
\frac{\Delta, \phi[x] \vdash_{HOIL} \chi}{\Delta, \forall x \phi \vdash_{HOIL} \chi} \forall L \\
\frac{\Delta \vdash_{HOIL} \phi[x]}{\Delta \vdash_{HOIL} \forall x \phi} \forall R \\
\frac{\Delta, \phi[x] \vdash_{HOIL} \chi}{\Delta, \exists x \phi \vdash_{HOIL} \chi} \exists L \\
\frac{\Delta \vdash_{HOIL} \phi[x]}{\Delta \vdash_{HOIL} \exists x \phi} \exists R \\
\frac{\Delta \vdash_{HOIL} \phi}{\Delta' \vdash_{HOIL} \phi'} \lambda
\end{array}$$

where in λ , Δ' and ϕ' are obtained by applying λ -conversions to ϕ and formulæ in Δ . We also have the following structural rule:

$$\frac{\Delta \vdash_{HOIL} \phi}{\Delta, \psi \vdash_{HOIL} \phi} \textit{weakening}$$

We do not need to explicitly state contraction and permutation since Δ is taken to be a set. ■