

DRAFT

# Completeness and Cut-elimination in the Intuitionistic Theory of Types

Mary DeMarco  
Wesleyan University

James Lipton  
Wesleyan University  
and  
Technical University of Madrid  
jlipton@wesleyan.edu

April 27, 2004

## Abstract

In this paper we give a semantic proof of cut-elimination for ICTT. ICTT is an intuitionistic formulation of Church's theory of types defined by Miller, Scedrov, Nadathur and Pfenning in the late 1980s. It is the basis for the lambda-prolog programming language.

Our approach, extending techniques of Takahashi, Andrews and tableaux machinery of Fitting, Smullyan, Nerode and Shore, is to prove a completeness theorem for the cut-free fragment, and show, semantically, that cut is a derived rule.

We strengthen Andrews' notion of Hintikka sets to intuitionistic logic in a way that also defines tableau-provability for intuitionistic type theory. We develop a corresponding model theory for ICTT and, after giving a completeness theorem without using cut we then show, using cut, how to establish completeness of more conventional term models.

Our work supplies a declarative semantics for the logic underlying the lambda-Prolog programming language.

## 1 Introduction

The theory considered in this paper is an intuitionistic reformulation of Church's Theory of types we call ICTT, first studied by Miller, Nadathur, Pfenning and Scedrov in the early 1990's (see e.g. [16]) who used it as a basis for the  $\lambda$ -prolog programming language, the leading higher-order declarative language today. Church's original 1940 paper introduced the typed lambda calculus, which has had a major impact on the theory and design of programming languages.

In this paper he also defined classical higher order logic by adding suitable logical constants at higher types, and rules of inference. This system has played an important role in higher-order automated deduction [1].

The research in this paper originated in efforts to find a semantics for full  $\lambda$ -prolog (see [6]). A model theory was developed by Blair and Bai [4], and independently by Wolfram [26] for the higher-order Horn fragment, whose classical and intuitionistic theories coincide. When more connectives are allowed in programs and goals, as in the higher-order Hereditarily Harrop (HOHH) fragment underlying  $\lambda$ -prolog, which uses universal quantification and implication, the intuitionistic and classical theories do *not* coincide. As argued in e.g. [15], the use of intuitionistic implication is essential in capturing module structure and hypothetical queries logically, as in  $\lambda$ -prolog.

The development of a semantics and completeness theorem for the full higher-order Hereditarily Harrop intuitionistic theory has remained an open problem.

No cut-elimination proof has been developed explicitly for ICTT, although Andrews supplied a proof for the classical theory in 1970. Needless to say, Girard, Coquand and others have published normalization proofs that imply cut elimination for related impredicative intuitionistic theories, but the question for ICTT has remained in bit of a fog that this paper aims to dispel. Furthermore, as Andrews has pointed out in his recent Herbrand Award speech [2], his approach to cut-elimination via Hintikka sets permits an extension of Smullyan's Unifying Principle [22] that has played a critical role in automated deduction applications of the classical theory. It is hoped that our extension of his proof to intuitionistic Hintikka sets will lead to similar applications of ICTT.

It should be pointed out that logic programming depends in a vital way on cut-elimination, since proof search is restricted to rules which are, at the very least, cut-free. In fact, as described in [16], more is required of a proof system for a reasonably operational notion of proof search, namely the *uniformity* property described therein.

## 2 Cut elimination and completeness

In 1954 Takeuti conjectured that cut-elimination holds for the simple theory of types. For over a decade it was a leading open problem in proof theory. A semantic proof was given for (different formulations of) classical second and higher-order logic by Takahashi [20], Tait [23], Martin L of and Prawitz in the late 1960's and early 70's, and a constructive proof of strong normalization for an intuitionistic system for analysis (extending G odel's **T**) and for the theory of finite types (as defined in e.g. Takeuti or Sch utte) was given by Girard [10] in 1970, (see [24, 11] for a discussion).

In the same year the Andrews [1] gave a semantic proof of cut elimination (and completeness) for Church's *classical* Type Theory (CTT). A constructive proof of cut elimination for the intuitionistic theory of species (set quantification) was given by Dragalin in the mid-1980's.

It is a matter of debate to what extent a syntactic proof of cut-elimination has ever been found for higher-order logic. Obviously such a proof could not be formalized in the logic itself, by G odel's second incompleteness theorem, and, in particular, any algorithm extracted from such an argument would not be provably total in the theory in question.

In this paper we give a semantic proof of cut-elimination, by giving a completeness proof for the cut-free fragment. Most completeness proofs construct prime extensions of theories or build Lindenbaum algebras, and rely heavily on cut in the object logic. One can avoid this by use of tableau-style techniques of Hintikka, Smullyan and Beth, to build partial models, called semivaluations by Takahashi, and then carefully show how these may be extended to full valuations (see Girard's discussion of this problem in terms of three-valued logics in [11]). As a by-product of our approach to the construction of Hintikka sets, we obtain a tableau proof-technique for ICTT.

Once we have the cut-elimination theorem in hand, of course, we are free to use cut to show that a much more natural notion of model, namely Lindenbaum-style term models [25], will yield a completeness theorem. This is discussed in the last section of this paper.

It should be pointed out that a completeness theorem for any impredicative logic that *assumes* cut elimination would be, by itself, a very weak result, since most semantic proofs of cut elimination come very close<sup>1</sup> to giving a completeness theorem without cut. One is very nearly assuming something stronger than what one is trying to prove.

---

<sup>1</sup>A semantic proof of cut-elimination does not appear *inevitably* to amount to a completeness proof. Systems have been found in which semantic cut-elimination does not imply completeness [3].

Our approach draws heavily from the work of Andrews [1] and the tableau techniques described in Nerode and Shore [19].

### 3 The Calculus

Church's calculus consists of a collection of higher-order functional types together with a simply typed lambda-calculus of terms. Connectives and quantifiers are given explicitly as typed constants in the language. The set of base types must include a type of logical formulas, as well as a type of individuals. Proofs are defined by a sequent calculus for logical formulas. In more detail:

**Type expressions** are formed from a set of *base types* (which must include at least the type  $\iota$  of individuals and  $o$  of truth values, or *formulae*) via the functional (or arrow) type constructor: if  $\alpha$  and  $\beta$  are types then so is  $\alpha \rightarrow \beta$ . **Church's notation** for functional type expressions is sometimes more convenient because of its brevity:  $(\beta\alpha)$  denotes the type  $\alpha \rightarrow \beta$ . Association, in Church's notation, is to the *left* whereas arrow terms associate to the right:  $\alpha\beta\gamma$  denotes  $(\alpha\beta)\gamma$ , meaning  $\gamma \rightarrow \beta \rightarrow \alpha$ , which is the same as  $\gamma \rightarrow (\beta \rightarrow \alpha)$ . We will use his notation interchangeably with arrow notation, especially with complex type expressions.

**Terms** are always associated with a type expression, and are built up from typed constants and typed variables (countably many at each type  $\alpha$ ):  $x_\alpha^1, x_\alpha^2, \dots, y_\alpha^1, y_\alpha^2, \dots$  according to the following definition:

1. A variable  $x_\alpha$  is a term of type  $\alpha$ .
2. A constant of type  $\alpha$  is a term of type  $\alpha$ .
3. If  $t_1$  is a term of type  $\alpha \rightarrow \beta$  and  $t_2$  is a term of type  $\alpha$  then  $t_1 t_2$  is a term of type  $\beta$ , called an application.
4. If  $t$  is a term of type  $\beta$  then  $\lambda x_\alpha. t$  is a term of type  $\alpha \rightarrow \beta$ , called an abstraction.

Terms may be displayed subscripted with their associated type, when convenient. We omit such subscripts from variables when clear from context, or when the information is of no interest.

**Logical Constants** The following logical constants are available to define formulas:

$$\top_o, \perp_o, \wedge_{ooo}, \vee_{ooo}, \supset_{ooo},$$

as well as

$$\Sigma_{o(o\alpha)} \quad \text{and} \quad \Pi_{o(o\alpha)},$$

the latter two existing for all types  $\alpha$ .

Note that because of currying in Church's type theory, the constants  $\wedge, \vee$  corresponding to the usual binary logical connectives, have higher types *ooo*, i.e.  $o \rightarrow (o \rightarrow o)$ .

**Reduction** We recall that a  $\beta$ -*redex* is a term  $\lambda x_\alpha. t u$ , formed by application of an abstraction to any compatibly typed term. The *contractum* (or  $\beta$ -contraction) of such a term is defined to be any term  $t[u/x]$  resulting from the substitution (after renaming of bound variables if necessary to avoid capture) of  $u$  for all free occurrences of  $x$  in  $t$ .  $\beta$ -contraction also refers to the binary relation between redices and contracta. The transitive closure of this binary relation is called

$$\begin{array}{c}
\overline{\Gamma \longrightarrow \top} \quad \overline{\Gamma, U \longrightarrow U} \quad \overline{\Gamma, \perp \longrightarrow \perp} \\
\\
\frac{\Gamma, B, C \longrightarrow A}{\Gamma, B \wedge C \longrightarrow A} \wedge_L \quad \frac{\Gamma \longrightarrow B \quad \Gamma \longrightarrow C}{\Gamma \longrightarrow B \wedge C} \wedge_R \\
\\
\frac{\Gamma, B \longrightarrow A \quad \Gamma, C \longrightarrow A}{\Gamma, B \vee C \longrightarrow A} \vee_L \quad \frac{\Gamma \longrightarrow B_i}{\Gamma \longrightarrow B_1 \vee B_2} \vee_R \\
\\
\frac{\Gamma \longrightarrow B \quad \Gamma, C \longrightarrow A}{\Gamma, B \supset C \longrightarrow A} \supset_L \quad \frac{\Gamma, B \longrightarrow C}{\Gamma \longrightarrow B \supset C} \supset_R \\
\\
\frac{\Gamma, P[t/x] \longrightarrow A}{\Gamma, \forall x.P \longrightarrow A} \forall_L \quad \frac{\Gamma \longrightarrow P}{\Gamma \longrightarrow \forall x.P} \forall_R^* \\
\\
\frac{\Gamma, P \longrightarrow A}{\Gamma, \exists x.P \longrightarrow A} \exists_L^* \quad \frac{\Gamma \longrightarrow P[t/x]}{\Gamma \longrightarrow \exists x.P} \exists_R \\
\\
\frac{\Gamma' \longrightarrow A'}{\Gamma \longrightarrow A} \lambda \quad \frac{\Gamma \longrightarrow \perp}{\Gamma \longrightarrow B} \perp_R
\end{array}$$

Figure 1: Higher-order Sequent Rules

$\beta$ -reduction. Contraction of  $\eta$ -redices  $\lambda x.tx$  to  $t$  when  $x$  does not occur freely in  $t$  induces eta-reduction. In this paper we will refer to the combination of both reductions,  $\beta\eta$ -reduction as  $\lambda$ -reduction, and the reflexive, symmetric, transitive closure of this relation as  **$\lambda$ -equivalence**.

In Church's Theory, as in Andrews, Miller and Nadathur's work, a canonical listing of variables is used to ensure the existence of unique canonical normal forms (sometimes denoted  $\rho(t)$ , and denoted  $\eta(t)$  in the section on cut-elimination) for each term  $t$ .

**Definition 3.1** *A formula is any term of type  $o$ . An atomic formula is a term of type  $o$  which is not  $\lambda$ -equivalent to any term whose head is a logical constant.*

The meaning of the logical constants  $\wedge$ ,  $\vee$  and  $\supset$  is evident, while  $\Sigma$  and  $\Pi$  are used to define the quantifiers  $\exists$  and  $\forall$ :  $\exists x_\alpha P_o$  is an abbreviation for  $\Sigma_{o(o\alpha)} \lambda x_\alpha.P_o$  and  $\forall x_\alpha P_o$  is an abbreviation for  $\Pi_{o(o\alpha)} \lambda x_\alpha.P_o$ . Also  $f(x_\alpha)$  is an abbreviation of  $(f_{\beta\alpha} x_\alpha)$ , for some  $\beta$ .

### 3.1 Deduction

We assume familiarity with Gentzen's sequent calculus, and in particular the intuitionistic LJ (see e.g. [12]). We present here, in the figure above, a sequent system for the intuitionistic fragment of Church's Theory of Types, which we call ICTT, similar to that of e.g. [16] of Miller *et al.*

In the sequent calculus displayed above all *sequents*  $\Gamma \longrightarrow A$  consist of a *set* of formulas (antecedents)  $\Gamma$  and a single formula  $A$  (consequent). The symbol  $U$  in the identity rule stands for any atomic formula. The rules of inference shown in the figure consist of *premisses*: the zero, one or two sequents above the line and *conclusions*: the lone sequent below the line. Rules marked with an asterisk must not contain free occurrences of the index variable below the line. In rule  $\lambda$ , the primed and unprimed antecedents (resp. consequents) are  $\lambda$ -equivalent.

The following lemmas will prove useful. Their proofs are routine. The first says that a derived identity rule holds for nonatomic formulas. The second is a derived rule of weakening.

**Lemma 3.2** *There is a sequent proof of  $\Gamma, A \longrightarrow A$  for any set of formulas  $\Gamma$  and any formula  $A$ .*

**Lemma 3.3** *If  $\Gamma' \supseteq \Gamma$  and  $\Gamma \longrightarrow A$  then  $\Gamma' \longrightarrow A$ .*

## 4 Models for ICTT

We define a semantics with respect to which ICTT will be shown sound and, in subsequent sections, complete.

Mitchell and Moggi [18] incorporate applicative structures into Kripke models to model (only) equational reasoning in the typed lambda calculus. We use a related, but somewhat different approach: we combine applicative structures with Heyting algebra semantics [25, Chap. 13, §4] for intuitionistic logic, and we do not consider equality here.

In the interest of making this paper more self-contained we briefly recapitulate some elementary definitions.

**Definition 4.1** *A lattice is a partially ordered (nonempty) set with two operations  $\wedge$  and  $\vee$ , the meet and join, satisfying the following axioms for all members  $a, b$  and  $c$  of the set:*

$$a \leq a \vee b \quad b \leq a \vee b \tag{1}$$

$$a \wedge b \leq a \quad a \wedge b \leq b \tag{2}$$

$$a \leq c \text{ and } b \leq c \text{ implies } a \vee b \leq c \tag{3}$$

$$c \leq a \text{ and } c \leq b \text{ implies } c \leq a \wedge b \tag{4}$$

*A lattice is distributive if for all members  $a, b$  and  $c$  of the partially ordered set,*

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \tag{5}$$

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \tag{6}$$

*The join  $\bigvee S$  of an arbitrary set  $S$ , if it exists, is the least upper bound of the set. Likewise, the meet  $\bigwedge S$  would be its greatest lower bound. A lattice is called complete if arbitrary joins and meets exist.*

The meet and join serve to model conjunction and disjunction, but we need an additional operator corresponding to implication. The following definition of a Heyting (or pseudo-Boolean) algebra is taken from [25].

**Definition 4.2** *A Heyting algebra  $H$  is a lattice with a least element  $\perp$  and an operation  $\rightarrow$  defined on all pairs of elements of  $H$  such that*

$$a \wedge b \leq c \text{ if and only if } a \leq b \rightarrow c \tag{7}$$

*In a complete Heyting algebra, arbitrary meets and joins exist.*

In later sections, we will consider complete Heyting algebras, as well as those with only certain parametrized infinite meets and joins corresponding to universal and existential quantification.

It can be shown [25] that a Heyting algebra is distributive and that if  $\bigvee B$  exists then

$$a \wedge \bigvee B = \bigvee \{a \wedge b : b \in B\}. \tag{8}$$

an identity called  $\wedge \vee$  distributivity. It is clear from (2) and (7) that a Heyting algebra has a top element ( $a \rightarrow a$  for any  $a$ ), which we denote  $\top$ . The following characterization of the arrow connective will be useful below. Any complete lattice satisfying  $\wedge - \vee$  distributivity (8) can be turned into a Heyting algebra by defining

$$a \rightarrow b := \bigvee \{x : x \wedge a \leq b\}. \quad (9)$$

Furthermore, in any (not necessarily complete) Heyting algebra, the supremum on the right hand side of (9) exists, and the identity holds.

**Example 4.3** *The open subsets of  $\mathbb{R}$  form a Heyting algebra with  $\vee$  denoting union,  $\wedge$  the interior of the intersection,  $\leq$  the subset relation,  $\perp$  the empty set and  $a \rightarrow b := \bigcup \{w : w \cap a \subseteq b\}$ . Then  $\top$  is evidently  $\mathbb{R}$  and (8) holds, but the dual statement*

$$a \vee \bigwedge B = \bigwedge \{a \vee b : b \in B\}$$

*need not: let  $a$  be  $\mathbb{R} \setminus \{0\}$  and  $B$  be the set of intervals  $\{(-\frac{1}{n}, \frac{1}{n}) : n \in \mathbb{N}\}$ . Here  $a \vee \bigwedge B = a \vee \emptyset = a$  but  $\bigwedge \{a \vee b : b \in B\} = \bigwedge \{\mathbb{R} : b \in B\} = \mathbb{R}$ .*

The failed dual to  $\wedge \vee$ -distributivity corresponds, in standard semantics for first-order logic, to the truth of the formula  $A \vee \forall x.B(x)$  iff  $\forall x.(A \vee B(x))$ , which is not an intuitionistically valid principle, and therefore something we do not want to satisfy in our models. This formula corresponds in turn to Axiom  $6^\alpha$  of Church's classical Theory of Types [5].

## 4.1 Applicative Structures

We will make use of the notion of applicative structures, a well-known semantical framework for the simply-typed lambda calculus, first introduced systematically by H. Friedman in [9], although obviously implicit in [13, 14, 21]. (See also [17] for a detailed discussion.)

**Definition 4.4** *A typed applicative structure  $\langle D, \text{App}, \text{Const} \rangle$  consists of an indexed family  $D = \{D_\alpha\}$  of sets  $D_\alpha$  for each type  $\alpha$ , an indexed family **App** of functions  $\text{App}_{\alpha,\beta} : D_{\beta\alpha} \times D_\alpha \rightarrow D_\beta$  for each pair  $(\alpha, \beta)$  of types, and an (indexed) interpretation function  $\text{Const} = \{\text{Const}_\alpha\}$  taking constants of each type  $\alpha$  to elements of  $D_\alpha$ .*

A typed applicative structure is (functionally) *extensional* if for every  $f, g$  in  $D_{\beta\alpha}$ ,  $\text{App}(f, d) = \text{App}(g, d)$  for all  $d \in D_\alpha$  implies  $f = g$ . The proof of our first completeness theorem will require the construction of a model which is not functionally extensional, while still satisfying eta-conversion.

In the final section, we consider global models which are not necessarily extensional, however we show how to build a functionally extensional model which satisfies the logical intensionality requirements described in the appendix. Because of the way truth is defined and distinguished from denotation in our structures, there is simply no interference between the two ideas. The details are discussed in the appendix.

It should be noted that we do not work with equality, so there is no way to explicitly specify (or prove) extensionality in the theory. Soundness of the  $\lambda$ -rule, therefore, only requires e.g. for  $M$  a term without free occurrences of  $x$ , that  $\lambda x.Mx$  and  $M$  have the same truth value in the structures we are about to define. Extensionality will guarantee more: that they have the same denotation. The distinction between denotation and truth value is crucial for the use of ICTT for logic programming as discussed (in the paragraph on logical intensionality) in the appendix.

An assignment  $\varphi$  is a function from the free variables of the language into  $D$  which respects types. Given a typed applicative structure  $D$ , an environmental model consists of a total function

$\llbracket \_ \rrbracket_\varphi$  from the open terms of the language into  $D$  for each assignment  $\varphi$  respecting types, for which the following equalities hold:

$$\begin{aligned} \llbracket c \rrbracket_\varphi &= \text{Const}(c) && \text{for constants } c \\ \llbracket x \rrbracket_\varphi &= \varphi(x) && \text{for variables } x \\ \llbracket (MN) \rrbracket_\varphi &= \text{App}(\llbracket M \rrbracket_\varphi, \llbracket N \rrbracket_\varphi) \\ \text{App}(\llbracket \lambda x_\alpha. M_\beta \rrbracket_\varphi, d) &= \llbracket M \rrbracket_{\varphi[x \mapsto d]} \end{aligned}$$

Now we can define models for our intuitionistic theory of types. We add certain restrictions to typed applicative structures for ICTT to handle the logical constants and predicates.

**Definition 4.5** *A Heyting applicative structure  $\langle D, \text{App}, \text{Const}, \omega, \Omega \rangle$  for ICTT is a typed applicative structure with an associated Heyting algebra  $\Omega$  and function  $\omega$  from  $D_o$  to  $\Omega$  such that for each  $f$  in  $D_{o\alpha}$ ,  $\Omega$  contains the parametrized meets and joins*

$$\bigwedge \{ \omega(\text{App}(f, d)) : d \in D_\alpha \} \text{ and } \bigvee \{ \omega(\text{App}(f, d)) : d \in D_\alpha \},$$

and the following conditions are satisfied:

$$\begin{aligned} \omega(\text{Const}(\top_o)) &= \top_\Omega \\ \omega(\text{Const}(\perp_o)) &= \perp_\Omega \\ \omega(\text{App}(\text{App}(\text{Const}(\wedge_{ooo}), d_1), d_2)) &= \omega(d_1) \wedge \omega(d_2) \\ \omega(\text{App}(\text{App}(\text{Const}(\vee_{ooo}), d_1), d_2)) &= \omega(d_1) \vee \omega(d_2) \\ \omega(\text{App}(\text{App}(\text{Const}(\supset_{ooo}), d_1), d_2)) &= \omega(d_1) \rightarrow \omega(d_2) \\ \omega(\text{App}(\text{Const}(\Sigma_{o(o\alpha)}), f)) &= \bigvee \{ \omega(\text{App}(f, d)) : d \in D_\alpha \} \\ \omega(\text{App}(\text{Const}(\Pi_{o(o\alpha)}), f)) &= \bigwedge \{ \omega(\text{App}(f, d)) : d \in D_\alpha \} \end{aligned}$$

A definition, with suitable further restrictions on  $D_o$  identifying  $D_o$  with  $\Omega$  (i.e., restricting  $\omega$  to the identity function) might seem more natural but would make  $A \wedge B$  indiscernable from  $B \wedge A$  in the structure and thereby identify the truth values of  $P_{oo}(A_o \wedge B_o)$  and  $P_{oo}(B_o \wedge A_o)$ . This identity holds neither in ICTT as presented here nor in the HOHH sub-system used in the  $\lambda$ -Prolog programming language.

An assignment is a function from the variables in the language to the elements of the model which allows us to give meaning to open terms. An assignment  $\varphi$  must satisfy  $\varphi(x_\alpha) \in D_\alpha$ .

**Definition 4.6** *A (global) model for ICTT is a total assignment-indexed interpretation function  $\mathfrak{H} =$*

$\{\mathfrak{H}(\_ )_\varphi : \varphi \text{ an assignment}\}$  into a Heyting applicative structure  $\langle D, \text{App}, \text{Const}, \Omega, \Omega \rangle$  which takes (possibly open) terms of type  $\alpha$  into  $D_\alpha$  and satisfies the environmental model conditions cited above, following definition (4.4), as well as eta-conversion, that is to say:

$$\begin{aligned} \mathfrak{H}(c)_\varphi &= \text{Const}(c) && \text{for constants } c \\ \mathfrak{H}(x)_\varphi &= \varphi(x) && \text{for variables } x \\ \mathfrak{H}(M)_\varphi &= \mathfrak{H}(N)_\varphi && M \text{ eta-equivalent to } N \\ \mathfrak{H}((MN))_\varphi &= \text{App}(\mathfrak{H}(M)_\varphi, \mathfrak{H}(N)_\varphi) \\ \text{App}(\mathfrak{H}(\lambda x_\alpha. M_\beta)_\varphi, d) &= \mathfrak{H}(M)_\varphi[x \mapsto d] \end{aligned}$$

Given a model  $\mathfrak{H}$  and an assignment  $\varphi$ , we say that  $\varphi$  satisfies  $B$  in  $\mathfrak{H}$  if  $\omega(\mathfrak{H}(B_o)_\varphi) = \top_\Omega$ ; we abbreviate this assertion to  $\mathfrak{H} \models_\varphi B_o$ . We say  $B_o$  is valid in  $\mathfrak{H}$  (equivalently,  $\mathfrak{H} \models B_o$ )

if  $\mathfrak{H} \models_{\varphi} B_o$  for every assignment  $\varphi$ . We abbreviate  $\omega(\mathfrak{H}(B_o)_{\varphi})$  to  $\llbracket B_o \rrbracket_{\varphi}$ . We also omit the subscript  $\varphi$  when our intentions are clear. We often use the term *model* to refer to the mapping  $\llbracket - \rrbracket$  from logical formulas to  $\Omega$ .

We will need a more liberal notion of model in our work, especially for the cut-elimination result, below.

## 4.2 The Extent Operator and Local Models

Heyting applicative structures correspond to so-called global  $\Omega$ -structures for first-order intuitionistic logic in [25, 8], while the extension introduced below corresponds to a special (semilocal) class of the more generic (local)  $\Omega$ -structures introduced by Fourman and Scott for logic with equality and existence predicate. Having no need here to model equality, we use this more general notion of model to translate Kripke models into interpretations of ICTT. These ideas are based on the work of Michael Fourman and Dana Scott [8]. We first introduce the more general notion of local structures, and then the semilocal ones we will be using.

**Definition 4.7** *A local Heyting applicative structure  $\mathcal{D} = \langle \mathbf{D}, \text{App}, \text{Const}, \omega, \Omega, \mathbf{E} \rangle$  is a Heyting applicative structure with an indexed function  $\mathbf{E}_{\alpha} : \mathbf{D}_{\alpha} \rightarrow \Omega$  for each type  $\alpha$ , satisfying the following conditions:*

$$\begin{aligned} \text{Range}(\mathbf{E} \circ \text{Const}) &= \{\top_{\Omega}\} \\ \omega(\text{App}(\text{Const}(\Sigma_{o(o\alpha)}), f_{o\alpha})) &= \bigvee \{\mathbf{E}_{\alpha}(d) \wedge \omega(\text{App}(f_{o\alpha}, d)) : d \in \mathbf{D}_{\alpha}\} \\ \omega(\text{App}(\text{Const}(\Pi_{o(o\alpha)}), f_{o\alpha})) &= \bigwedge \{\mathbf{E}_{\alpha}(d) \rightarrow \omega(\text{App}(f_{o\alpha}, d)) : d \in \mathbf{D}_{\alpha}\} \end{aligned}$$

The Heyting Algebra  $\Omega$  is now required to be closed under the following relativized meets and joins:

$$\bigwedge \{\mathbf{E}_{\alpha}(d) \rightarrow \omega(\text{App}(f, d)) : d \in \mathbf{D}_{\alpha}\} \text{ and } \bigvee \{\mathbf{E}_{\alpha}(d) \wedge \omega(\text{App}(f, d)) : d \in \mathbf{D}_{\alpha}\}$$

for each  $f$  in  $\mathbf{D}_{o\alpha}$ .

**Definition 4.8** *Given a local Heyting applicative structure  $\mathcal{D}$ , a local  $\mathcal{D}$ -environment is a type-preserving mapping  $\phi$  from variables to the carriers  $\mathbf{D}$  of  $\mathcal{D}$ . The environment is called **global** if it satisfies the following condition for every term  $t$  in ICTT:*

$$\mathbf{E} \llbracket t \rrbracket_{\phi} = \top_{\Omega}.$$

We will refer to a local structure satisfying eta-conversion together with a local environment as a local model. If its environment is global, we will call it a **semilocal** model. Note that the condition on the environment does not make a semilocal model global. Elements  $d$  of the carrier  $\mathbf{D}$  which are not the image of a term in the language may have non-maximal extent.

## 4.3 Soundness

In the following we extend interpretations to sequents in a natural way.

**Definition 4.9** *We define the meaning of a sequent in a model to be the truth-value in  $\Omega$  given by:*

$$\llbracket \Gamma \longrightarrow \Delta \rrbracket := \llbracket \bigwedge \Gamma \supset \Delta \rrbracket$$

where  $\bigwedge \Gamma$  signifies the conjunction of the elements of  $\Gamma$  and where we recall that, in an intuitionistic calculus, the consequent  $\Delta$  is restricted to a single formula.

Note that  $\llbracket \bigwedge \Gamma \supset \Delta \rrbracket = \top$  if and only if  $\top \leq \llbracket \bigwedge \Gamma \supset \Delta \rrbracket$ , which is to say  $\top \leq \llbracket \bigwedge \Gamma \rrbracket \rightarrow \llbracket \Delta \rrbracket$ , which by the condition on  $\rightarrow$  is equivalent to  $\top \wedge \llbracket \bigwedge \Gamma \rrbracket \leq \llbracket \Delta \rrbracket$ , which in turn is equivalent to  $\llbracket \bigwedge \Gamma \rrbracket \leq \llbracket \Delta \rrbracket$  by (2) and (4). We will abbreviate  $\llbracket \bigwedge \Gamma \rrbracket$  to  $\llbracket \Gamma \rrbracket$  and express the validity of the indicated sequent by  $\llbracket \Gamma \rrbracket \leq \llbracket \Delta \rrbracket$  or, when referring to the environment, by  $\llbracket \Gamma \rrbracket_\varphi \leq \llbracket \Delta \rrbracket_\varphi$  henceforth.

The following technical lemmas will be needed to prove soundness of ICTT for semilocal models.

**Lemma 4.10** *Suppose  $\Omega$  is a (complete) Heyting algebra and  $u \in \Omega$ . Then the  $u$ -relativized algebra  $\Omega^u$ , defined by*

$$\begin{aligned} \|\Omega^u\| &:= \{v \wedge_\Omega u : v \in \|\Omega\|\}, \\ \top_u &= u \\ v \rightarrow_u w &:= (v \rightarrow_\Omega w) \wedge_\Omega u, \\ \bigvee_u, \bigwedge_u &:= \bigvee_\Omega, \bigwedge_\Omega, \\ v \leq_u w &:= v \leq w. \end{aligned}$$

and with  $\perp_u := \perp_\Omega$  and  $\vee_u, \vee_u, \wedge_u, \wedge_u$  just the restrictions of the corresponding  $\Omega$  operations to  $\Omega_u$ , is a (complete) Heyting algebra and the mapping  $i : \Omega \rightarrow \Omega^u$  given by

$$i(v) = v \wedge_\Omega u$$

is a surjective cHa homomorphism, i.e. it preserves all cHa operations. Thus  $\Omega_u$  is isomorphic to the quotient  $\Omega/\equiv_i$ .

**Proof.** First we observe that the underlying set  $\|\Omega^u\|$  can be described equivalently as  $\{v \in \|\Omega\| : v \leq u\}$ , so the mapping  $i$  is clearly surjective. Checking that  $\Omega^u$  is a complete Heyting algebra is easy. The only interesting case is implication. We leave it to the reader to verify that  $r \rightarrow_u s = \bigvee\{z \in \Omega_u : z \wedge_u r \leq s\}$ .

Since  $x \leq y \Rightarrow x \wedge u \leq y \wedge u$ ,  $i$  is clearly monotone, and it follows immediately from the distributivity axioms for  $\Omega$  that  $i$  preserves  $\vee, \wedge$ . By  $\wedge - \vee$  distributivity  $i(x \rightarrow y) = \bigvee\{z \wedge u : z \wedge x \leq y\}$ . Now

$$\begin{aligned} i(x) \rightarrow_u i(y) &= (i(x) \rightarrow i(y)) \wedge u \\ &= (x \wedge u \rightarrow y \wedge u) \wedge u \\ &= u \wedge \bigvee\{z : z \wedge x \wedge u \leq y \wedge u\} \\ &= \bigvee\{z \wedge u : (z \wedge u) \wedge x \leq y \wedge u\}. \end{aligned}$$

We must show the suprema  $\bigvee\{z \wedge u : z \wedge x \leq y\}$  and  $\bigvee\{z \wedge u : (z \wedge u) \wedge x \leq y \wedge u\}$  coincide. In fact, they are taken over sets that are equal. Since  $z \wedge x \leq y$  implies  $(z \wedge u) \wedge x \leq y \wedge u$  the former set is contained in the latter. If  $v$  is in  $\{z \wedge u : (z \wedge u) \wedge x \leq y \wedge u\}$ , i.e. is equal to some  $z \wedge u$  in the set, then, letting  $z_u = z \wedge u$ , we have that  $v = z_u \wedge u$  and so,  $z_u \wedge x \leq y \wedge u \leq y$  so  $v$  is also in  $\{z \wedge u : z \wedge x \leq y\}$ .  $\square$

**Lemma 4.11** *Suppose  $\mathcal{D} = \langle \mathsf{D}, \mathsf{App}, \mathsf{Const}, \Omega, \omega, \mathsf{E} \rangle$  is a local structure,  $\varphi$  a global  $\mathcal{D}$ -environment, and  $u \in \Omega$ . Define  $\mathcal{D}^u = \langle \mathsf{D}^u, \mathsf{App}^u, \mathsf{Const}^u, \Omega^u, \omega^u, \mathsf{E}^u \rangle$ , the  $u$ -relativized structure as follows:  $\Omega^u$  is the Heyting algebra defined above,  $\mathsf{D}^u, \mathsf{App}^u, \mathsf{Const}^u$  are the same as for  $\mathcal{D}$ ,  $\omega^u(d) = \omega(d) \wedge_\Omega u$ , and  $\mathsf{E}^u d = \mathsf{E}d \wedge_\Omega u$ .*

*Then  $\mathcal{D}^u$  is a local Heyting Applicative Structure. If  $\varphi$  is a global environment for  $\mathcal{D}$  and if  $d \in D_\alpha$  has extent  $u \in \Omega$ , then  $\varphi[x := d]$  is a global environment for  $\mathcal{D}^u$ , that is to say, the resulting  $u$ -relativized model  $\llbracket - \rrbracket_{\varphi[x:=d]}^{\mathsf{E}d}$  is a semilocal model.*

We use the notation  $\llbracket - \rrbracket_\varphi^u$  to refer to the model, i.e. mapping from formulas to truth values induced by  $\mathcal{D}^u$  and the appropriate environment  $\varphi$ . Note that for any formula  $B$ ,  $\llbracket B \rrbracket^u = \omega^u(\mathcal{D}(B)) = \omega(\mathcal{D}(B)) \wedge_\Omega u = \llbracket B \rrbracket \wedge_\Omega u$ .

**Proof.** Since only the definitions of  $\Omega^u, \omega^u, \mathbf{E}^u$  change, we just have to check the axioms for  $\omega$  and  $\mathbf{E}$  in definition (4.5). The axioms for  $\top, \perp, \wedge$  and  $\vee$  are immediate. The existence of the relativized meet and joins

$$\bigwedge \{\omega(\mathbf{App}(f, d)) : d \in D_\alpha\} \text{ and } \bigvee \{\omega(\mathbf{App}(f, d)) : d \in D_\alpha\},$$

follow from associativity of  $\wedge$  and  $\wedge \vee$  distributivity of the cHa  $\Omega$ . For implication, we have, using the results of the preceding lemma

$$\begin{aligned} \omega^u(\mathbf{App}(\mathbf{App}(\mathbf{Const}(\supset_{ooo}), d_1), d_2)) &= (\omega(d_1) \rightarrow \omega(d_2)) \wedge u \\ &= i(\omega(d) \rightarrow \omega(d_2)) \\ &= i(\omega(d)) \rightarrow_u i(\omega(d_2)) \\ &= \omega^u(d_1) \rightarrow_u \omega^u(d_2). \end{aligned}$$

We leave the  $\Sigma$  case to the reader and establish the required property for  $\Pi$ .

$$\begin{aligned} \omega^u(\mathbf{App}(\mathbf{Const}(\Pi_{o(o\alpha)}), f)) &= \omega(\mathbf{App}(\mathbf{Const}(\Pi_{o(o\alpha)}), f)) \wedge u \\ &= u \wedge \bigwedge_{D_\alpha} \mathbf{E}d \rightarrow \omega(\mathbf{App}(f_{o\alpha}, d)) \\ &= \bigwedge_{D_\alpha} u \wedge (\mathbf{E}d \rightarrow \omega(\mathbf{App}(f_{o\alpha}, d))) \\ &= \bigwedge_{D_\alpha} i(\mathbf{E}d \rightarrow \omega(\mathbf{App}(f_{o\alpha}, d))) \\ &= \bigwedge_{D_\alpha} i(\mathbf{E}d) \rightarrow_u i(\omega(\mathbf{App}(f_{o\alpha}, d))) \\ &= \bigwedge_{D_\alpha} \mathbf{E}^u d \rightarrow \omega^u(\mathbf{App}(f_{o\alpha}, d)). \end{aligned}$$

□

**Theorem 4.12 (Soundness)** *If  $\Gamma \longrightarrow A$  is provable in ICTT then  $\llbracket \Gamma \rrbracket \leq \llbracket A \rrbracket$  in every semi-local model  $\mathfrak{E}$  of ICTT.*

**Proof.** Let  $\mathcal{D} = \langle D, \mathbf{App}, \mathbf{Const}, \Omega, \mathbf{E} \rangle$  be a local applicative structure, and  $\varphi$  a global assignment.

If  $\Gamma \longrightarrow A$  is provable in ICTT then there is a sequent proof with  $\Gamma \longrightarrow A$  at the root. We prove  $\llbracket \Gamma \rrbracket_\varphi \leq \llbracket A \rrbracket_\varphi$  by induction on the height of the sequent tree.

The base cases are immediate.

Now, we assume that we have a proof of height greater than 0, and that for every deduction of smaller height, and every semilocal model, the conclusion holds. We must consider all possible rules used in the last step of the deduction.

The propositional cases are routine. We only treat  $\supset$ -left.

$$\frac{\Delta \longrightarrow B \quad \Delta, C \longrightarrow D}{\Delta, B \supset C \longrightarrow D} \supset_L$$

By induction hypothesis  $\llbracket \Delta \rrbracket_\varphi \leq_\Omega \llbracket B \rrbracket_\varphi$  and  $\llbracket \Delta \rrbracket_\varphi \wedge_\Omega \llbracket C \rrbracket_\varphi \leq_\Omega \llbracket D \rrbracket_\varphi$ . By elementary facts about Heyting algebras

$$\llbracket \Delta \rrbracket_\varphi \wedge_\Omega \llbracket B \supset C \rrbracket_\varphi \leq_\Omega \llbracket \Delta \rrbracket_\varphi \wedge_\Omega \llbracket B \rrbracket_\varphi \wedge_\Omega \llbracket B \supset C \rrbracket_\varphi$$

$$\begin{aligned} &\leq_{\Omega} \llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \llbracket C \rrbracket_{\varphi} \\ &\leq_{\Omega} \llbracket D \rrbracket_{\varphi} \end{aligned}$$

$\exists_L$ :

Suppose the last step in the deduction is

$$\frac{\Delta, B \longrightarrow C}{\Delta, \exists x_{\alpha}. B \longrightarrow C} \exists_L *$$

with  $x_{\alpha}$  not free in the conclusion. Then  $\llbracket \Delta, \exists x_{\alpha}. B \rrbracket_{\varphi} = \llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \bigvee_{d \in D_{\alpha}} \mathbf{E}d \wedge_{\Omega} \llbracket B \rrbracket_{\varphi[x:=d]}$ . Now,  $\llbracket - \rrbracket_{\varphi[x:=d]}^{\mathbf{E}d}$  is a semilocal model by lemma 4.11. So by the induction hypothesis

$$\llbracket \Delta \rrbracket_{\varphi[x:=d]}^{\mathbf{E}d} \wedge_{\Omega} \llbracket B \rrbracket_{\varphi[x:=d]}^{\mathbf{E}d} \leq \llbracket C \rrbracket_{\varphi[x:=d]}^{\mathbf{E}d}$$

Since  $x$  does not occur in  $\Delta$  or  $C$ , this is equivalent to

$$\llbracket \Delta \rrbracket_{\varphi}^{\mathbf{E}d} \wedge_{\Omega} \llbracket B \rrbracket_{\varphi[x:=d]}^{\mathbf{E}d} \leq \llbracket C \rrbracket_{\varphi}^{\mathbf{E}d}$$

Since  $\llbracket \Delta \rrbracket_{\varphi}^{\mathbf{E}d} = \llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \mathbf{E}d$  this implies that

$$\llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \mathbf{E}d \wedge_{\Omega} \llbracket B \rrbracket_{\varphi[x:=d]} \leq \llbracket C \rrbracket_{\varphi} \wedge_{\Omega} \mathbf{E}d \leq \llbracket C \rrbracket_{\varphi}.$$

Taking suprema over  $d \in D_{\alpha}$  on the left gives the desired result:  $\llbracket \Delta, \exists x_{\alpha}. B \rrbracket_{\varphi} \leq \llbracket C \rrbracket_{\varphi}$ .

$\exists_R$ :

Suppose the last step of the deduction was

$$\frac{\Delta \longrightarrow B[t/x]}{\Delta \longrightarrow \exists x_{\alpha}. B} \exists_R$$

By the induction hypothesis  $\llbracket \Delta \rrbracket_{\varphi} \leq \llbracket B[t/x] \rrbracket_{\varphi} = \llbracket B \rrbracket_{\varphi[x:=d_0]}$  where  $d_0 = \varphi(t)$ . Since  $\varphi$  is a global environment and  $t$  a term (of type  $\alpha$ ) in ICTT,  $\mathbf{E}d_0 = \top_{\Omega}$ . Thus the preceding inequality is equivalent to  $\llbracket \Delta \rrbracket_{\varphi} \leq \mathbf{E}d_0 \wedge_{\Omega} \llbracket B \rrbracket_{\varphi[x:=d_0]}$ , which is, in turn, bounded by  $\bigvee_{d \in D_{\alpha}} \mathbf{E}d \wedge_{\Omega} \llbracket B \rrbracket_{\varphi[x:=d]}$  and hence by  $\llbracket \exists x_{\alpha}. B \rrbracket_{\varphi}$ .

$\forall_L$ :

Suppose the last step of the deduction was

$$\frac{\Delta, B[t/x] \longrightarrow C}{\Delta, \forall x. B \longrightarrow C} \forall_L$$

and, by induction hypothesis  $\llbracket \Delta, B[t/x] \rrbracket_{\varphi} \leq \llbracket C \rrbracket_{\varphi}$ . Then  $\llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \llbracket B \rrbracket_{\varphi[x:=d_0]} \leq \llbracket C \rrbracket_{\varphi}$  where  $d_0 = \varphi(t)$ . Since  $\varphi$  is a global environment,  $\mathbf{E}d_0 = \top_{\Omega}$  and the preceding inequality is equivalent to  $\llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \mathbf{E}d_0 \wedge_{\Omega} \llbracket B \rrbracket_{\varphi[x:=d_0]} \leq \llbracket C \rrbracket_{\varphi}$ , from which the desired conclusion

$$\llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \llbracket \forall x. B \rrbracket_{\varphi} = \llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \bigwedge_{d \in D_{\alpha}} \llbracket B \rrbracket_{\varphi[x:=d]} \leq \llbracket C \rrbracket_{\varphi}$$

is immediate.

$\forall_R$ :

Suppose the last step of the deduction was

$$\frac{\Delta \longrightarrow B}{\Delta \longrightarrow \forall x_{\alpha}. B} \forall_R *$$

with  $x_\alpha$  not free in the conclusion. Pick any  $d \in D_\alpha$ . Then, by the preceding lemmas,  $\llbracket - \rrbracket_{\varphi[x:=d]}^{\mathbf{E}d}$  is a semilocal model, so, by induction hypothesis,  $\llbracket \Delta \rrbracket_{\varphi[x:=d]}^{\mathbf{E}d} \leq \llbracket B \rrbracket_{\varphi[x:=d]}^{\mathbf{E}d}$ . Since  $x$  is not free in  $\Delta$ , and for any formula  $D$ ,  $\llbracket D \rrbracket_{\varphi}^{\mathbf{E}d} = \llbracket D \rrbracket_{\varphi} \wedge_{\Omega} \mathbf{E}d$ , we have  $\llbracket \Delta \rrbracket_{\varphi} \wedge_{\Omega} \mathbf{E}d \leq \llbracket B \rrbracket_{\varphi[x:=d]} \wedge_{\Omega} \mathbf{E}d$ . Since  $u \wedge v \leq w \wedge v$  implies  $u \leq v \rightarrow w$  in any Heyting algebra,  $\llbracket \Delta \rrbracket_{\varphi} \leq \mathbf{E}d \rightarrow \llbracket B \rrbracket_{\varphi[x:=d]}$ . This holds for any  $d$ , so, taking the infimum of the right hand side we have  $\llbracket \Delta \rrbracket_{\varphi} \leq \llbracket \forall x_\alpha. B \rrbracket_{\varphi}$  as we wanted to show.

To show soundness of the  $\lambda$ -rule, one must show soundness of  $\beta$  reduction and  $\alpha$ -conversion in a typed applicative structure. The result is well-known, and we omit the details here. Soundness of eta-reduction is true by definition in local models.  $\square$

## 4.4 Cut-elimination for ICTT

We now sketch a semantic proof of cut-elimination in ICTT, which extends to the intuitionistic case similar arguments due to Andrews [1], [20], using techniques inspired by Smullyan [22], Fitting [7], and Nerode and Shore [19].

**Cut-free methods in the semantics** One should underscore why a Hintikka-set approach is essential for proving completeness without cut in the object logic, in other words, why we are going to such trouble. Most conventional completeness arguments use cut with a vengeance, and are therefore of no use to us here. Henkin's completeness theorem, builds for a theory  $\Gamma$  such that  $\Gamma \not\vdash A$  a structure in which  $\Gamma$  holds and  $A$  fails, by first extending  $\Gamma$  to a prime theory (satisfying disjunction and existence properties) that avoids proving  $A$ . The proof that this can be done, usually known as Lindenbaum's lemma, builds the prime extension incrementally, proving that consistency (or avoidance of  $A$ ) is maintained at each stage. One begins by enumerating all disjunctions and existential formulas over a language enriched with so-called Henkin constants. At the appropriate stage  $2n$ , one considers whether the extension  $\Gamma_{2n}$  built so far, with the inductive hypothesis that  $\Gamma_{2n} \not\vdash A$ , proves the  $n$ -th disjunctive formula  $B \vee C$ . If so, one of the disjuncts  $B, C$  must be added to  $\Gamma_{2n}$  *without spoiling* the requirement  $\Gamma_{2n} \not\vdash A$ . One can only do this using cut.

If  $\Gamma_{2n} \vdash B \vee C$  and  $\Gamma_{2n}, B \not\vdash A$  let  $\Gamma_{2n+1} := \Gamma_{2n} \cup \{B\}$  else  $\Gamma_{2n+1} := \Gamma_{2n} \cup \{C\}$  How do we know one of the two choices avoids  $A$ ? Suppose not. Then

$$\frac{\frac{\Gamma_{2n}, B \longrightarrow A \quad \Gamma_{2n}, C \longrightarrow A}{\Gamma_{2n}, B \vee C \longrightarrow A} \vee - L \quad \Gamma_{2n} \longrightarrow B \vee C}{\Gamma_{2n} \longrightarrow A} \text{CUT}$$

which contradicts the hypothesis, but *uses cut!*

Another approach to completeness, using Boolean or Heyting-valued models (as we do), is to build the so-called Lindenbaum algebra of equivalence classes of formulas

$$[A] := \{B : \Gamma, B \vdash A \text{ and } \Gamma, A \vdash B\}$$

with  $[A] \leq_{\Omega} [B]$  if  $\Gamma, A \vdash B$ .

One needs to show  $\Omega$  is a Boolean (Heyting) algebra with enough limits for the quantifiers, and then interpret formulas  $A$  by their own equivalence classes. But first we have to show  $\Omega$  is a poset! And that means showing  $\leq_{\Omega}$  is transitive, and that is equivalent to CUT! Hence the need for the cut-free methods of Hintikka, Smullyan and others. The crux of the method is this: instead of completing the given theory  $\Gamma$ , one must extend it "downwards", by putting in the appropriate subformulas and their substitution instances. The cost of this is that, initially, one only produces a partial model (or semi-valuation), which must then be extended, using Takahashi's ideas, with some care.

Also, instead of a Lindenbaum algebra made up of equivalence classes of formulas, we use a topological Heyting algebra built from sequences of natural numbers. We now proceed to the details.

## 4.5 Hintikka Sets

We start with a partially ordered set  $K$  with least element  $p_0$  and a set  $S$ .  $K$  will supply nodes of a rooted Kripke model, which will then be used to construct a semilocal model. In the completeness theorem  $K$  will consist of sequences of natural numbers with prefix order.  $S$  will supply us with a set of constants to function as witnesses for e.g. true existential statements. One should point out that the entire construction can be easily adapted to the use of fresh *variables* to play this role, as in Andrews' proof.

A  $K$ -Hintikka set (or, strictly speaking, an  $(S, K)$ -Hintikka set)  $\mathcal{H}$  for ICTT is a set of *signed forcing formulas* of the form

$$\mathbf{B} p \Vdash A_o$$

where  $\mathbf{B} \in \{\mathbf{T}, \mathbf{F}\}$ ,  $p \in K$ , and  $A_o$  is a formula in ICTT, satisfying certain conditions listed below. In order to define these conditions we need some terminology. An element  $p$  of  $K$  is said to occur in  $\mathcal{H}$  if it occurs in some signed forcing formula in  $\mathcal{H}$ .

**Definition 4.13** *An element  $c_\alpha$  of  $S$  is a  $p$ -constant, or a constant of level  $p$  on  $\mathcal{H}$  if it appears in some signed forcing statement  $\mathbf{B} q \Vdash A_o$  in  $\mathcal{H}$ , with  $q \leq p$ . A formula or term is a  $p$ -formula if it is an ICTT formula over the language consisting of (all the logical and nonlogical constants already in ICTT and) the  $p$ -constants in  $\mathcal{H}$ . In particular, any term over the original language, i.e. over ICTT and the logical and nonlogical constants already in ICTT is a  $p$ -term for every  $p$ .*

**Definition 4.14** *A set of signed forcing formulas is a  $K$ -Hintikka set if it satisfies the following conditions.*

1. if  $\mathbf{B}p \Vdash A \in \mathcal{H}$  and  $A$  is not in  $\eta$ -nf, then  $\mathbf{B}p \Vdash \eta A \in \mathcal{H}$
2. if  $\mathbf{T}p \Vdash \wedge BC \in \mathcal{H}$  then  $\mathbf{T}p \Vdash B \in \mathcal{H}$  and  $\mathbf{T}p \Vdash C \in \mathcal{H}$ .
3. if  $\mathbf{F}p \Vdash \wedge BC \in \mathcal{H}$  then either  $\mathbf{F}p \Vdash B \in \mathcal{H}$  or  $\mathbf{F}p \Vdash C \in \mathcal{H}$ .
4. if  $\mathbf{T}p \Vdash \vee BC \in \mathcal{H}$  then either  $\mathbf{T}p \Vdash B \in \mathcal{H}$  or  $\mathbf{T}p \Vdash C \in \mathcal{H}$ .
5. if  $\mathbf{F}p \Vdash \vee BC \in \mathcal{H}$  then  $\mathbf{F}p \Vdash B \in \mathcal{H}$  and  $\mathbf{F}p \Vdash C \in \mathcal{H}$ .
6. if  $\mathbf{T}p \Vdash \supset BC \in \mathcal{H}$  then for every  $p' \geq p$  occurring in  $\mathcal{H}$ , either  $\mathbf{F}p' \Vdash B \in \mathcal{H}$  or  $\mathbf{T}p' \Vdash C \in \mathcal{H}$ .
7. if  $\mathbf{F}p \Vdash \supset BC \in \mathcal{H}$  then for some  $p' \geq p$ ,  $\mathbf{T}p' \Vdash B \in \mathcal{H}$  and  $\mathbf{F}p' \Vdash C \in \mathcal{H}$ .
8. if  $\mathbf{T}p \Vdash \Sigma_{o(o\alpha)} B$  then for some constant  $c_\alpha$  of  $S$ ,  $\mathbf{T}p \Vdash Bc_\alpha$ .
9. if  $\mathbf{F}p \Vdash \Sigma_{o(o\alpha)} B$  then for each  $p$ -formula  $C_\alpha$  in  $\mathcal{H}$  of type  $\alpha$ ,  $\mathbf{F}p \Vdash BC$  occurs in  $\mathcal{H}$ .
10. if  $\mathbf{T}p \Vdash \Pi_{o(o\alpha)} B$  then for each  $q \geq p$  in  $\mathcal{H}$ , and each  $q$ -formula  $C_\alpha$  in  $\mathcal{H}$  of type  $\alpha$ ,  $\mathbf{T}q \Vdash BC$  occurs in  $\mathcal{H}$ .
11. if  $\mathbf{F}p \Vdash \Pi_{o(o\alpha)} B$  then for some  $p' \geq p$  in  $\mathcal{H}$ , and some constant  $c_\alpha$  of  $S$ ,  $\mathbf{F}p' \Vdash Bc_\alpha$  occurs in  $\mathcal{H}$ .

The conditions for negation  $\sim B$  are identical to  $B \supset \perp$ , and hence a special case of the implication clauses above.

A Hintikka set  $\mathcal{H}$  is contradictory if one of the following occurs in  $\mathcal{H}$ :

- for some formula  $A$  and  $p \in K$  both  $\top p \Vdash A$  and  $\text{F}p \Vdash A$  occurs in  $\mathcal{H}$ .
- For some  $p$ ,  $\top p \Vdash \perp$  occurs in  $\mathcal{H}$ .
- For some  $p$ ,  $\text{F}p \Vdash \top$  occurs in  $\mathcal{H}$ .

**Definition 4.15** For any ICTT sequent  $\Gamma \longrightarrow A$ , a Hintikka set **for**  $(\Gamma, A)$  is a  $K$ -Hintikka set such that

$$\{\top p_0 \Vdash \gamma : \gamma \in \Gamma\} \cup \{\text{F}p_0 \Vdash A\} \subset \mathcal{H}.$$

**Lemma 4.16** For every sequent  $\Gamma \longrightarrow A$  there is a  $K$ -Hintikka set for  $(\Gamma, A)$  (for some partially ordered set  $K$  with a least element and set of constants  $S$ ).

**Proof.** The proof proceeds by explicit construction of all Hintikka sets generated in a certain way, essentially using the algorithm generating a complete systematic tableau proof. Since our aim is to produce a model rather than a tableau, we make some changes to the standard tableau procedure, such as enumerating all possible witnesses to univocal true statements. More efficiency is obtained in automated deduction, by attempting to enumerate the least number of witnesses possible, but this makes the semantics harder to define.

Let  $K$  be the set of finite sequences of natural numbers under the prefix ordering, with root node  $p_0 = \langle \rangle$  (the empty sequence) and  $S$  a sequence  $\{S_\alpha : \alpha \in \text{TY}\}$  of countable sets. Each member  $S_\alpha$  is called the set of constants of type  $\alpha$ , whose members will be written  $c_\alpha$  to recall their intended type. Let  $\kappa$  be the set of constants already in the language of ICTT.

We will define a tree of Hintikka sets, the *complete systematic  $K$ -Hintikka tree*, by stages. Under the assumption that  $\Gamma$  is finite, we may handle the requirement that each path in the tree satisfies  $\Gamma$  on the opening stage. With infinite  $\Gamma$  we need only change the construction to enumerate  $\top \langle \rangle \Vdash \gamma$  for each  $\gamma$  in  $\Gamma$  at the odd stages.

We simultaneously define the following in stages.

1. A sequence  $\{\tau_n : n \in \omega\}$  of labelled trees, each path of which is an approximation to a Hintikka set, i.e. consists of signed forcing formulas.  $[\tau_n]$  will denote the set of paths in  $\tau_n$ .
2. A sequence  $\{S_{\alpha,n}^\pi(p) : \alpha \in \text{TY}, p \in K, n \in \omega, \pi \in [\tau_n]\}$  of subsets of  $S \cup \kappa$ . Each  $S_{\alpha,n}^\pi(p)$  contains those constants of type  $\alpha$  occurring in some signed formula  $\text{B}q \Vdash A$  in path  $\pi$  of  $\tau_n$ , for some  $q \leq p$ . We will always enforce monotonicity of the  $S^\pi$  in  $n$ . Note that at each stage we extend some path  $\pi$  of  $\tau_n$  to either a new path  $\pi'$ , or, by adding a forking tree to the leaf node of  $\pi$ , to a pair of paths  $\pi_0, \pi_1$  of  $\tau_{n+1}$ . Thus, strictly speaking, the set of constants occurring on  $\pi$  at stage  $n$  is never updated. Rather, one or two new sets of constants,  $S_{n+1}^{\pi'}$ , or  $S_{n+1}^{\pi_0}$  and  $S_{n+1}^{\pi_1}$ , are created from  $S_n^\pi$ . To simplify notation, we do not mention the names of the new path(s) created. Another convention used below is that  $S^\pi$  denotes the (indexed) family  $\{S_{\alpha,n}^\pi\}$  where each  $S_{\alpha,n}^\pi$  is viewed as a function mapping each  $p \in K$  to a (possibly empty) set of constants. Thus when we wish to add a constant  $c_\alpha$  to  $S_{\alpha,n}^\pi(p)$ , we add the pair  $(p, c_\alpha)$  to  $S_{\alpha,n}^\pi$ . We are thus able to speak, for example, of all the constants of any type occurring at or below  $p \in K$  on path  $\pi$  as all those  $\{(q, c) \in S_n^\pi : q \leq p\}$
3. A partition of the formula occurrences at each node  $\nu$  of  $\tau_n$  into *used* and *unused*.

We assume some standard ordering exists on the nodes of each tree, e.g. induced by depth and left-to-right ordering.

Stage 0:

$\tau_0$  has the unique path  $\pi = \{\text{F} \langle \rangle \Vdash A, \top \langle \rangle \Vdash \gamma_1, \dots, \top \langle \rangle \Vdash \gamma_n\}$ , where  $\Gamma = \{\gamma_1, \dots, \gamma_n\}$ .

$$S_0^\pi(\langle \rangle) = \kappa$$

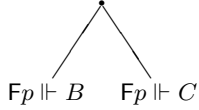
Stage  $n + 1$ :

Assume  $\tau_n$  and the  $S_n$  have been constructed. and node  $\nu$  on path  $\pi$  is the least unused node on the tree, labelled with  $\mathbf{B}p \Vdash A$ . We now build  $\tau_n$  by attaching to *each* path  $\pi$  through  $\nu$  the tree described below, and updating each  $S_n^\pi$  according to the structure of the signed formula  $\mathbf{B}p \Vdash A$ . After each such step declare the entry at node  $\nu$  a *used* occurrence.

**A not in  $\eta$ -nf:** Attach to the leaf of path  $\pi$  through  $\nu$  the formula  $\mathbf{B}p \Vdash \eta A$  unless it is already on  $\pi$ . Let  $S_{n+1} = S_n$ .

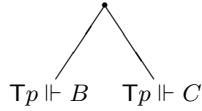
$\mathbf{T}p \Vdash \wedge BC$ : Extend each  $\pi$  through  $\nu$  with  $\{\mathbf{T}p \Vdash B, \mathbf{T}p \Vdash C\}$ , thus increasing its length by two nodes. Let  $S_{n+1} = S_n$ .

$\mathbf{F}p \Vdash \wedge BC$ : Extend each  $\pi$  through  $\nu$  with



Thus  $\pi$  now splits into two paths, each one node longer than  $\pi$ . Let  $S_{n+1} = S_n$ .

$\mathbf{T}p \Vdash \vee BC$ : Extend each  $\pi$  through  $\nu$  with

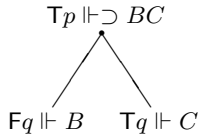


Let  $S_{n+1} = S_n$ .

$\mathbf{F}p \Vdash \vee BC$ : Extend each  $\pi$  through  $\nu$  with  $\{\mathbf{F}p \Vdash B, \mathbf{F}p \Vdash C\}$ . Let  $S_{n+1} = S_n$ .

$\mathbf{T}p \Vdash \supset BC$ : For each path  $\pi$  through  $\nu$  do the following: let  $p'$  be the least  $q$  in  $K$  which occurs on  $\pi$  with  $q \geq p$  such that neither  $\mathbf{F}q \Vdash B$  nor  $\mathbf{T}q \Vdash C$  occurs on  $\pi$ .

Extend each  $\pi$  through  $\nu$  with



Thus  $\pi$  splits into two paths, each *two* nodes longer than  $\pi$ . The entry that was used at this step is written anew at the apex of the attached subtree so that a fresh copy of it will be unused and receive attention at a later stage, spawning a similar tree for the *next*  $q$  on  $K$  which occurs on  $\pi$  with  $q \geq p$  such that no entry of the appropriate form occurs on  $\pi$ . This is a way of making true implications permanently *active*. Let  $S_{n+1} = S_n$ .

$\mathbf{F}p \Vdash \supset BC$ : Extend each  $\pi$  through  $\nu$  as follows: let  $k$  be the least natural number such that  $p * k$  (denoting the concatenation of  $p$  with the one-element sequence  $\langle k \rangle$ ) has not occurred on  $\pi$ , put  $p' = p * k$  (note that it is incomparable with any  $q$  on  $\pi$  not below or equal to  $p$ ). Attach the two formulas  $\{\mathbf{T}p' \Vdash B, \mathbf{F}p' \Vdash C\}$ . Let  $S_{n+1}^\pi(p') = \bigcup_{q \leq p} S_n^\pi(q)$ .

$\top p \Vdash \Sigma_{o(o\alpha)} B$ : Extend each path  $\pi$  through  $\nu$  as follows. Let  $c$  be the first constant in  $c$  of type  $\alpha$  that has not occurred in  $\pi$  (i.e.  $c \notin S_{\alpha,n}^\pi$ ). Attach  $\top p \Vdash Bc_\alpha$  to  $\pi$  and put  $S_{\alpha,n+1}^\pi = S_{\alpha,n}^\pi \cup \{(p, c_\alpha)\}$

$\text{F}p \Vdash \Sigma_{o(o\alpha)} B$ : Let  $C$  be the least ICTT formula over the language of  $S_n^\pi(p)$  (i.e., using constants occurring in entries of the form  $\text{B}q \Vdash E$  for  $q \leq p$ ). such that  $\text{F}p \Vdash BC$  does not already occur on  $\pi$ . Attach the *two* nodes  $\{\text{F}p \Vdash \Sigma_{o(o\alpha)} B, \text{F}p \Vdash BC\}$  to the end of  $\pi$ . As discussed in the  $\top p \Vdash \supset BC$  case, this keeps a copy of  $\text{F}p \Vdash \Sigma_{o(o\alpha)} B$  active.

Let  $S_{n+1} = S_n$ .

$\top p \Vdash \Pi_{o(o\alpha)} B$ : Let  $(p', C)$  be the least pair  $(r, D)$  with  $r \in K$  occurring on  $\pi$ ,  $r \geq p$  and with  $D$  an ICTT term over the language of  $S_{\alpha,n}^\pi(p)$ , and with  $\top r \Vdash BD$  not already occurring in  $\pi$ . Then extend  $\pi$  with the two entries  $\{\top p \Vdash \Pi_{o(o\alpha)} B, \top p' \Vdash BC\}$ . That is to say, we attach a fresh unused copy of the entry that has just received attention, followed by an instance. Let  $S_{n+1} = S_n$ .

$\text{F}p \Vdash \Pi_{o(o\alpha)} B$ : Let  $k \in \mathbb{N}$  be the least natural number such that  $p * k$  does not occur in  $\pi$  and let  $p' = p * k$ . Let  $c_\alpha$  be the first constant in  $S_{.alpha}$  not occurring in  $\pi$ . Attach  $\text{F}p' \Vdash Bc_\alpha$  to  $\pi$  and update  $S_n$  as follows:

$$S_{\alpha,(n+1)}^\pi(p') = \bigcup_{q \leq p', q \text{ on } \pi} S_{\alpha,n}^\pi(q) \cup \{(p', c_\alpha)\}$$

The cases for negation  $\top p \Vdash \sim B$ , and  $\text{F}p \Vdash \sim B$  are treated like  $B \supset \perp$ .

Now let  $\tau = \bigcup_n \tau_n$ . It is obvious that every node of  $\tau$  is used, i.e. for some  $m_0$  it occurs used on all  $\tau_m$  for  $m \geq m_0$ . For each path  $\pi$  in  $\tau$ , define  $K^\pi$  to be the restriction of the poset  $K$  to those worlds  $p$  occurring in  $\pi$ , and  $S^\pi, S_\alpha^\pi, S_\alpha^\pi(p), S^\pi(p)$  to be the unions of the appropriate sets corresponding to paths that are initial segments of  $\pi$  built at some stage of the construction. Let  $\hat{\pi}$  be the set of entries occurring on  $\pi$ . Then it is straightforward to show that each  $\hat{\pi}$  is a  $K^\pi$ -Hintikka set over the constants in  $S^\pi$ .  $\square$

**Definition 4.17** A  $K$ -Hintikka set is **closed** if all nodes force  $\top_o$  and it is *montone*: whenever  $\top p \Vdash A \in \mathcal{H}$  and  $q \geq p$  occurs in  $\mathcal{H}$ , then  $\top q \Vdash A \in \mathcal{H}$ . Note that any Hintikka set is contained in a closed one, obtained by adding all the necessary signed forcing statements.

**Definition 4.18** If  $\mathcal{H}$  is a  $K$ -Hintikka path, or any initial segment thereof, and  $p$  is in  $K$ , then  $\top_p(\mathcal{H})$  is the set

$$\{B : \top q \Vdash B \in \mathcal{H}, \text{ some } q \leq p\}$$

of all positively  $p$ -forced formulae on  $\mathcal{H}$  for the world  $p \in K$ . Let  $\text{F}_p(\mathcal{H})$  be the set of all negatively forced formulae exactly at world  $p$ :

$$\{B : \text{F}p \Vdash B \in \mathcal{H}\}$$

A  $K$ -Hintikka set (resp. Hintikka path) is **ICTT-consistent** if for any  $p$ , and any  $B$  in  $\text{F}_p(\mathcal{H})$ ,

$$\top_p(\mathcal{H}) \not\vdash B,$$

where provability means provability in the ICTT sequent calculus.

**Theorem 4.19** Suppose  $\Gamma \not\vdash A$ . Then there is a consistent  $K$ -Hintikka path  $\pi$  for  $(\Gamma, A)$ .

**Proof.** We will show that if  $\pi$  is a consistent finite path in some partially developed tableau  $\tau_n$  for  $(\Gamma, A)$  then at least one of the ways it is extended at some stage, as one of its entries is used, preserves consistency. As remarked at the beginning of this section, the whole point of using a tableau construction is that the proof that consistency is maintained does not require cut in the object logic.

Suppose  $\nu$  is a node on  $\pi$  with entry  $\mathbf{B}p \Vdash B$ , the least unused occurrence in  $\tau_n$ . We consider all cases:

$B$  not in  $\eta$ -nf: The path  $\pi$  is extended with a single entry  $\mathbf{B}p \Vdash \eta A$ . Let us assume  $\pi$  consistent and new path inconsistent, and that  $\mathbf{B} = \mathbf{T}$ . Then, for some  $D$  in  $\mathbf{F}_p(\pi)$ , we must have  $\mathbf{T}_p(\pi), \eta A \vdash D$ . By the  $\lambda$ -rule, we must have  $\mathbf{T}_p(\pi), A \vdash D$ , contradicting the consistency of  $\pi$ . The dual case  $\mathbf{B} = \mathbf{F}$  is similar.

$\mathbf{T}p \Vdash \wedge BC$ : If the original path is consistent and the new one is not, then for some formula  $D$  in  $N(\pi)$

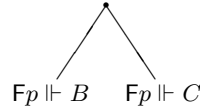
$$\mathbf{T}_p(\pi), B, C \vdash D$$

but then

$$\mathbf{T}_p(\pi), B \wedge C \vdash D$$

so  $\mathbf{T}_p(\pi) \vdash B$  contradicting the fact that  $\pi$  is consistent.

$\mathbf{F}p \Vdash \wedge BC$ :  $\pi$  is extended to two new paths  $\pi_0, \pi_1$  as follows:

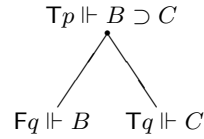


Observe that  $\mathbf{T}_p(\pi_0) = \mathbf{T}_p(\pi_1) = \mathbf{T}_p(\pi)$ . If both extensions are inconsistent, and the original path  $\pi$  consistent, the only possible violations of consistency are

$$\mathbf{T}_p(\pi) \vdash B \text{ and } \mathbf{T}_p(\pi) \vdash C$$

But then  $\mathbf{T}_p(\pi) \vdash B \wedge C$ , and  $\pi$  is inconsistent.

$\mathbf{T}p \Vdash \vee BC$  and  $\mathbf{F}p \Vdash \vee BC$ : Similar to the above two cases and left to the reader.



$\mathbf{T}p \Vdash B \supset C$ : the path is extended as follows:

If neither extended path is consistent (and the original one is assumed consistent) then  $\mathbf{T}_q(\pi) \vdash B$  and  $\mathbf{T}_q(\pi), C \vdash D$  for some  $D$  in  $\mathbf{F}_q(\pi)$ . But then, by the  $\supset_L$  rule,  $\mathbf{T}_q(\pi), B \supset C \vdash D$  whence, since  $\mathbf{T}p \Vdash \supset BC$  is in  $\pi$ ,  $\pi$  is inconsistent.

$\mathbf{F}p \Vdash \supset BC$ : Then  $\pi$  is extended to the path  $\pi'$  with the two signed formulas  $\{\mathbf{T}p' \Vdash B, \mathbf{F}p' \Vdash C\}$  for an appropriate world  $p'$ . If this path is inconsistent, and the original one is not, the inconsistency must take place at  $p'$ .  $\mathbf{T}_{p'}(\pi)$  inherits all true forced statements for worlds  $q$  below  $p'$ , but  $\mathbf{F}_{p'}(\pi)$  does *not*. So the only possibility is  $\mathbf{T}_{p'}(\pi) \vdash C$ , i.e.  $\mathbf{T}_p(\pi), B \vdash C$  which implies  $\mathbf{T}_p(\pi) \vdash B \supset C$  and hence the inconsistency of  $\pi$ .

$\mathbf{T}p \Vdash \Sigma_{o(o\alpha)}B$ : The path is extended with the single formula  $\mathbf{T}p \Vdash Bc_\alpha$ , for a fresh constant  $c_\alpha$  not occurring in  $\pi$ . If inconsistent,  $\mathbf{T}_p(\pi), Bc_\alpha \vdash D$  for some  $D \in \mathbf{F}_p(\pi)$  then  $\mathbf{T}_p(\pi), \Sigma B \vdash D$  (by generalization on fresh constants, left to the reader) and we have the inconsistency of  $\pi$ .

$Fp \Vdash \Sigma_{o(o\alpha)}B$ : The path  $\pi$  is extended with the two nodes  $\{Fp \Vdash \Sigma_{o(o\alpha)}B, Fp \Vdash BC\}$  for the appropriate  $C$ . If the new path is inconsistent, so is the old by  $\exists_R$ .

$\top p \Vdash \Pi_{o(o\alpha)}B$ : Similar to the last case, using  $\forall_L$ .

$Fp \Vdash \Pi_{o(o\alpha)}B$ : The path  $\pi$  is extended to a new path  $\pi'$  with the single entry  $Fp' \Vdash Bc_\alpha$ , for the appropriate world  $p' \geq p$  and *fresh* constant  $c_\alpha$ . If  $\pi$  is consistent, and the new path fails to be, and  $\top_{p'}(\pi') = \top_p(\pi)$ , we must have

$$\top_p(\pi) \vdash Bc_\alpha.$$

Generalizing on the fresh constant, we obtain, by  $\forall_R$ ,

$$\top_p(\pi) \vdash \Pi_{o(o\alpha)}B$$

yielding the inconsistency of  $\pi$  for a contradiction, which completes the proof.  $\square$

**Corollary 4.20** *If a Hintikka path (resp. set) is consistent, it is non-contradictory.*

**Proof.** If  $\top p \Vdash B$  and  $Fp \Vdash B$  are members of a Hintikka set  $\mathcal{H}$ , then, since  $B \in \top_p(\mathcal{H})$  and hence  $\top_p(\mathcal{H}) \vdash B$ , consistency is violated.  $\square$

**Theorem 4.21** *Let  $\mathcal{H}$  be a non-contradictory closed  $K$ -Hintikka set for the pair  $(\Gamma, A)$ . Then there is a local HAS model  $\mathfrak{D}$  for ICTT agreeing with  $\mathcal{H}$ , i.e. for every closed  $\eta$ -nf-formula  $B$  occurring in some signed formula in  $\mathcal{H}$*

$$\mathfrak{D} \models B \quad \text{iff} \quad \top_{p_0} \Vdash B \in \mathcal{H}$$

**Proof.** We define a local HAS  $\mathfrak{D}_H = \langle D, \cdot, \text{Const}, \Omega, \omega, E \rangle$  as follows. The carriers  $D_\alpha$  consist of pairs  $\langle M, r \rangle$  called  $V$ -complexes by Takahashi and Andrews, with  $M$  an open term in canonical  $\eta$ -nf-normal form of type  $\alpha$ , and  $r$  a realizer, essentially a member of the full-type hierarchy over the carriers  $D_i$  and  $D_o$ .

The base carrier  $D_o$  contains all possible truth valuations of logical propositions (terms of type  $o$ ) consistent with  $\mathcal{H}$ . This is where the most significant changes to Andrews' original proof take place, to deal with intuitionistic logic. Classical logic can be handled with two truth values, but ICTT will require a greatly expanded set of truth values meeting certain conditions.

To make these conditions precise, we must define two ancillary concepts, the minimal and maximal  $\mathcal{H}$ -consistent truth values for a formula  $A$ .

**Definition 4.22** *The weak and strong support on  $\mathcal{H}$  of a formula  $A$  are given by*

$$\mathcal{H}_A^\top = \{p : \top p \Vdash A \text{ occurs on } \mathcal{H}\} \tag{10}$$

$$\mathcal{H}_A^{\neg F} = \{p : \forall q \geq p \text{ } Fq \Vdash A \text{ does not occur on } \mathcal{H}\} \tag{11}$$

**The model  $\mathfrak{D}_H$**

- $\Omega = \mathcal{O}(K)$ , the (topological) cHa of all upwards closed subsets of  $K$ . Thus  $\top_\Omega = K$ ,  $\perp_\Omega = \emptyset$  and  $\vee_\Omega$  is union,  $\wedge_\Omega$  intersection,  $\Rightarrow_\Omega$ , the standard Heyting implication operator:  $u \Rightarrow v = \text{Int}(K - u) \cup v$ , and the pseudo-complement  $\bar{u}$  is the interior of the set-theoretic complement of  $u$ .
- $D_o = \{\langle A_o, a \rangle : A_o \text{ is in } \eta\text{-nf, occurs in } \mathcal{H} \text{ and } \mathcal{H}_A^\top \subseteq a \subseteq \mathcal{H}_A^{\neg F}\} \cup \{\langle A_o, u \rangle : A_o \text{ is in } \eta\text{-nf, does not occur in } \mathcal{H} \text{ and } u \in \Omega\}$ .  
Note that if  $A_o$  is a formula not occurring in any signed forcing formula in  $\mathcal{H}$ , it appears paired with *every* possible truth value in  $\Omega$ .

- $D_\iota = \{\langle A_\iota, \iota \rangle : A_\iota \text{ is in } \eta\text{-nf}\}$ .
- $D_{\beta\alpha} = \{\langle A_{\beta\alpha}, f \rangle : A_{\beta\alpha} \text{ is in } \eta\text{-nf} : D_\alpha \rightarrow D_\beta \text{ and for every } \langle B_\alpha, b \rangle \in D_\alpha$   
 $f\langle B_\alpha, b \rangle$  is of the form  
 $\langle \eta[AB], r \rangle$  for some  $r\}$ .

In order to show the model (and in particular the **Const** operator below) is well-defined, we need to establish that for every  $\eta\text{-nf}$  term  $A_\alpha$  over the language of  $\mathcal{H}$ , there is an  $a$  such that  $\langle A, a \rangle \in D_\alpha$ . To this end we define a function  $\rho$  from  $\eta\text{-nf}$  terms  $A_\alpha$  to the second component of a particular V-complex in  $D_\alpha$ . We will establish several useful properties of this function in lemma (4.23) below.

- Define  $\rho : Tm_\alpha(\mathcal{H}) \rightarrow D_\alpha^2$  by:
  - if  $\alpha$  is  $o$ :  $\rho(A_o) = \{p : \top p \Vdash A_o \in \mathcal{H}\}$  if  $A_o$  occurs in  $\mathcal{H}$ , else choose any  $u \in \Omega$  e.g.  $\emptyset$ .
  - if  $\alpha$  is  $\iota$ :  $\rho(A_\iota) = \iota$ .
  - if  $\alpha$  is  $(\gamma\beta)$ :  $\rho(A_{(\gamma\beta)})\langle B_\beta, b \rangle = \langle \eta[AB], \rho(\eta[AB]) \rangle$ .
- $\langle A_{\beta\alpha}, a \rangle \cdot \langle B_\alpha, b \rangle = \langle \eta[AB], (a\langle B_\alpha, b \rangle)^2 \rangle$  where the superscript 2 means the second component of the indicated ordered pair.
- For each  $\alpha$ ,  $E : D_\alpha \rightarrow \Omega$  is given by  $E(\langle N, n \rangle) = \{p \in K : N \text{ is } ap\text{-term}\}$  (see definition 4.13). Note that this set may be empty. If  $c_{(\alpha\beta)}, d_\beta$  are constants at levels  $p$  and  $q$  of  $\mathcal{H}$ , and  $p$  and  $q$  are incomparable worlds, then the term  $c \cdot d$  exists in  $D_\alpha$ , but has extent  $\emptyset$ . By definition 4.13 of  $p$ -constants if  $N$  is a term not occurring in  $\mathcal{H}$ , or *any* term over the language of ICTT,  $E\langle N, n \rangle = \top_\Omega$ . The extent of an element of  $D_o$  has nothing to do with its truth value. In particular,  $E\langle \perp_o, \perp_\Omega \rangle = \top_\Omega$ , since  $\perp_o$  is a term of ICTT not containing any special constants in  $\mathcal{H}$ , and hence defined in every world  $p \in K$ .
- $\omega : D_o \rightarrow \Omega$  is given by  $\omega(\langle B, b \rangle) = b$
- **Const** is defined as follows:
  - $\text{Const}(\top_o) = \langle \top_o, \top_\Omega \rangle$ .
  - $\text{Const}(\perp_o) = \langle \perp_o, \perp_\Omega \rangle$ .
  - $\text{Const}(c_\alpha) = \langle c_\alpha, \rho c_\alpha \rangle$ , for non-logical constants  $c_\alpha$ .
  - $\text{Const}(\sim) = \langle \sim, \lambda\langle B_o, b \rangle. \langle \sim B_o, \bar{b} \rangle \rangle$ , where  $\bar{b}$  denotes the pseudo-complement of the element  $b$  in the Heyting algebra  $\Omega$ .
  - $\text{Const}(\wedge) = \langle \wedge, \lambda\langle B_o, b \rangle. \langle \wedge B_o, \lambda\langle D_o, d \rangle. \langle \wedge B_o D_o, b \cap d \rangle \rangle \rangle$ , where  $b \cap d$  is  $\wedge_\Omega$ , the meet of  $b$  and  $d$  in  $\Omega$  (which happens to be intersection).
  - $\text{Const}(\vee) = \langle \vee, \lambda\langle B_o, b \rangle. \langle \vee B_o, \lambda\langle D_o, d \rangle. \langle \vee B_o D_o, b \cup d \rangle \rangle \rangle$ , where  $b \cup d$  is  $\vee_\Omega$ , the join of  $b$  and  $d$  in  $\Omega$  (which happens to be union).
  - $\text{Const}(\supset) = \langle \supset, \lambda\langle B_o, b \rangle. \langle \supset B_o, \lambda\langle D_o, d \rangle. \langle \supset B_o D_o, b \Rightarrow_\Omega d \rangle \rangle \rangle$ .
  - $\text{Const}(\Sigma_{o(o\alpha)}) = \langle \Sigma, \lambda\langle M_{o\alpha}, m \rangle. \langle \Sigma M, \bigvee_{d \in D_\alpha} E d \wedge_\Omega (md)^2 \rangle \rangle$ .
  - $\text{Const}(\Pi_{o(o\alpha)}) = \langle \Pi, \lambda\langle M_{o\alpha}, m \rangle. \langle \Pi M, \bigwedge_{d \in D_\alpha} E d \Rightarrow_\Omega (md)^2 \rangle \rangle$ .

Where the meta-lambda notation means set-theoretic function abstraction:  $\lambda\langle A_\alpha, a \rangle. \langle M_\beta, m \rangle$  denotes the function  $f : D_\alpha \rightarrow D_\beta$  given by  $f(\langle A_\alpha, a \rangle) = \langle M_\beta, m \rangle$ .

**Lemma 4.23 ( $\rho$ -lemma)** *The function  $\rho : Tm_\alpha(\mathcal{H}) \rightarrow D_\alpha^2$  given above is well-defined. In fact, for every  $A_\alpha$  in  $\eta\text{-nf}$ ,*

$$\langle A_\alpha, \rho A_\alpha \rangle \in D_\alpha.$$

*In particular, for every  $\eta\text{-nf}$  formula  $A_\alpha$  there is an  $a$  such that  $\langle A, a \rangle \in D_\alpha$ .*

**Proof.** The proof is by induction on type structure. If  $\alpha = \iota$  the result is obvious, and it is also immediate for  $\alpha = o$ , since  $\mathcal{H}_{A_o}^T = \rho A_o \subseteq \mathcal{H}_{A_o}^{\neg F}$ . If  $\alpha$  is  $(\gamma\beta)$  and  $\langle B_\beta, b \rangle \in D_\beta$  then  $\rho(A_{(\gamma\beta)})\langle B_\beta, b \rangle = \langle \eta[AB], \rho(\eta[AB]) \rangle$  is in  $D_\gamma$  by inductive hypothesis.  $\square$

**Lemma 4.24** *The function Const is well-defined: for every logical constant  $k_\alpha$ ,  $\text{Const}(k) \in D_\alpha$ .*

**Proof.** What we must show is that e.g. for constants of type  $o$ , if  $\text{Const}(k) = \langle k, c \rangle$ , the condition

$$\mathcal{H}_k^T \subseteq c \subseteq \mathcal{H}_k^{\neg F}$$

required of members of  $D_o$  holds, and that for constants of higher type (e.g.  $\wedge$ ), it is respected on application:  $\text{Const}(\wedge)\langle B_o, b \rangle\langle C_o, c \rangle$  is in  $D_o$ .

Const( $\top_o$ ):

We have  $\mathcal{H}_{\top_o}^T = \top_\Omega$  by definition in a closed Hintikka set. Since  $\mathcal{H}$  is non-contradictory,  $\top_\Omega \subseteq \mathcal{H}_{\top_o}^{\neg F}$ .

Const( $\perp_o$ ):

Immediate, since  $\mathcal{H}_{\perp_o}^T$  is empty.

Const( $\sim$ ):

is  $\langle \sim, \lambda\langle B_o, b \rangle.\langle \sim B_o, \bar{b} \rangle \rangle$ . We must show  $\lambda\langle B_o, b \rangle.\langle \sim B_o, \bar{b} \rangle$  really maps each  $\langle B_o, b \rangle \in D_o$  to  $D_o$ , that is to say, if

$$\mathcal{H}_{B_o}^T \subseteq b \subseteq \mathcal{H}_{B_o}^{\neg F}$$

then

$$\mathcal{H}_{\sim B_o}^T \subseteq \bar{b} \subseteq \mathcal{H}_{\sim B_o}^{\neg F}.$$

Suppose  $p \in \mathcal{H}_{\sim B_o}^T$ . Then  $\top p \Vdash \sim B$  occurs on  $\mathcal{H}$ . By the properties of (4.14) for each  $q \geq p$  on  $\mathcal{H}$ ,  $\text{F}q \Vdash B_o$  occurs on  $\mathcal{H}$ , so  $q \notin \mathcal{H}_{B_o}^{\neg F}$  hence no  $q \geq p$  is in  $b$ , so  $p \in \bar{b}$ , establishing the first inequality we wish to show.

Now suppose  $p \in \bar{b}$ . If  $q \geq p$ , then  $q$  is not in  $b$  hence not in  $\mathcal{H}_{B_o}^T$  so  $\top q \Vdash B_o$  does not occur on  $\mathcal{H}$ . If  $\text{F}q \Vdash \sim B_o$  then for some  $q' \geq q \geq p$  we would have  $\top q' \Vdash B$ , so  $p \in \mathcal{H}_{\sim B_o}^{\neg F}$  as we wanted to show.

Const( $\wedge$ ):

Suppose  $\langle B_o, b \rangle$  and  $\langle C_o, c \rangle$  are in  $D_o$ . Then  $\mathcal{H}_{B_o}^T \subseteq b \subseteq \mathcal{H}_{B_o}^{\neg F}$  and  $\mathcal{H}_{C_o}^T \subseteq c \subseteq \mathcal{H}_{C_o}^{\neg F}$ . If  $p \in \mathcal{H}_{B \wedge C}^T$  then  $\top p \Vdash B \wedge C \in \mathcal{H}$ . Consequently  $\top p \Vdash B$  and  $\top p \Vdash C$  are both on  $\mathcal{H}$ , so  $p \in \mathcal{H}_B^T \cap \mathcal{H}_C^T \subseteq b \cap c$ .

Suppose  $p \in b \cap c$ . Then  $p \in \mathcal{H}_B^{\neg F} \cap \mathcal{H}_C^{\neg F}$ , so for every  $q \geq p$ ,  $\text{F}q \Vdash B$ ,  $\text{F}q \Vdash C$ , and hence  $\text{F}q \Vdash B \wedge C$  do not occur in  $\mathcal{H}$ , so  $p \in \mathcal{H}_{B \wedge C}^{\neg F}$  as we needed to show.

Const( $\vee$ ) and Const( $\supset$ ):

left to the reader.

Const( $\Sigma_{o(o\alpha)}$ ):

$$= \langle \Sigma, \lambda\langle M_{o\alpha}, m \rangle.\langle \Sigma M, \bigvee_{d \in D_\alpha} \text{E}d \wedge_\Omega (md)^2 \rangle \rangle.$$

Suppose  $\langle M_{o\alpha}, m \rangle \in D_{o\alpha}$  and  $d = \langle c, k \rangle \in D_\alpha$ . Then  $\langle Mc, (md)^2 \rangle \in D_o$ , so

$$\mathcal{H}_{Mc}^T \subseteq (md)^2 \subseteq \mathcal{H}_{Mc}^{\neg F}. \quad (12)$$

We must show

$$\mathcal{H}_{\Sigma M}^T \subseteq \bigvee_{d \in D_\alpha} \text{E}d \wedge_\Omega (md)^2 \subseteq \mathcal{H}_{\Sigma M}^{\neg F}.$$

If  $p \in \mathcal{H}_{\Sigma M}^T$  then  $\top p \Vdash \Sigma M \in \mathcal{H}$ , and hence, for some  $c_\alpha$ ,  $\top p \Vdash Mc \in \mathcal{H}$ , and so,  $p \in \mathcal{H}_{Mc}^T$ .

By lemma (4.23) there is a  $k$  such that  $\langle c, k \rangle \in D_\alpha$ . Let  $d = \langle c, k \rangle$ . Then, by hypothesis (12)  $p \in (md)^2$ . Also,  $p \in \text{E}d$  (since  $c$  is in the language at world  $p$ ), so  $p \in \text{E}d \wedge_\Omega (md)^2 \subseteq \bigvee \text{E}d \wedge_\Omega (md)^2$ .

To show the remaining inequality, suppose  $p \in \bigvee \text{Ed} \wedge_{\Omega} (md)^2$ . We want to show that for any  $q \geq p$  the signed formula  $\text{F}q \Vdash \Sigma M$  is not on  $\mathcal{H}$ . Suppose otherwise for some  $q \geq p$ . Then for any  $c_{\alpha}$  occurring at the appropriate worlds on  $\mathcal{H}$ ,  $\text{F}q \Vdash Mc$  also occurs on  $\mathcal{H}$ . Let  $k$  and  $d$  be such that  $d = \langle c, k \rangle \in \text{D}_{\alpha}$  for such a  $c$ . Then  $q \in \text{Ed}$  and  $\langle Mc, (md)^2 \rangle \in \text{D}_{\alpha}$ . Furthermore  $q$  is not in  $\mathcal{H}_{Mc}^{\neg \text{F}}$  hence not in  $(md)^2$ . We have shown that for any  $d \in \text{D}_{\alpha}$  if  $q \in \text{Ed}$  then  $q \notin (md)^2$ . Thus for any such  $d$ ,  $q \notin \text{Ed} \wedge (md)^2$  hence  $q$  is not in the supremum of these sets (their union) contradicting the assumption that  $p$  was, and that the set is upwards closed.

Const( $\Pi_{o(o\alpha)}$ ):

Similar, and left to the reader. □

We must now show that the conditions on the **Const** function in definitions (4.5) subject to the modifications of definition (4.7) are satisfied.

- $\omega(\text{Const}(\top_o)) = \top_{\Omega}$  and  $\omega(\text{Const}(\perp_o)) = \perp_{\Omega}$  by definition.
- We verify the condition for  $\supset$  and leave those for  $\wedge, \vee$  to the reader.  $\omega(\supset \cdot \langle B_o, b \rangle \cdot \langle D_o, d \rangle) = \omega(\langle \supset B_o D_o, b \Rightarrow_{\Omega} d \rangle) = \omega(\langle B_o, b \rangle) \Rightarrow_{\Omega} \omega(\langle D_o, d \rangle)$  as required.
- $\omega(\Sigma_{o(o\alpha)} \cdot \langle M_{o\alpha}, m \rangle) = \omega(\langle \Sigma M, \bigvee_{d \in \text{D}_{\alpha}} \text{Ed} \wedge_{\Omega} (m \cdot d)^2 \rangle) = \bigvee_{d \in \text{D}_{\alpha}} \text{Ed} \wedge_{\Omega} (m(d))^2 = \bigvee_{d \in \text{D}_{\alpha}} \text{Ed} \wedge_{\Omega}$   
 $\omega(\langle M_{o\alpha}, m \rangle \cdot d)$  as required.

Finally, let  $\varphi$  be an assignment of typed variables to  $\text{D}$ , and let  $\varphi^1$  denote the projection onto the first component of the pairs produced by  $\varphi$ . We now define the extension of  $\varphi$  to a map  $\widehat{\varphi}$  from open terms (to open terms) by:

$$\widehat{\varphi}B = \eta(\lambda x_1 \lambda x_2 \cdots \lambda x_n. B) \varphi^1(x_1) \varphi^1(x_2) \cdots \varphi^1(x_n)$$

where  $x, \dots, x_n$  are the free variables in  $B$ . If there are no free variables in  $B$ ,  $\widehat{\varphi}$  just returns its canonical normal form. Let  $\mathfrak{I}_{\varphi}$  be an interpretation over  $\varphi$  of open terms of type  $\alpha$  into  $\text{D}_{\alpha}$  given as follows, by induction on term structure. Let  $\mathfrak{I}_{\varphi}^1$  and  $\mathfrak{I}_{\varphi}^2$  be the first and second components of  $\mathfrak{I}_{\varphi}$ .

1.  $\mathfrak{I}_{\varphi}(c) = \text{Const}(c)$  for all constants  $c$ , logical or otherwise.
2.  $\mathfrak{I}_{\varphi}(x) = \varphi(x)$  for variables  $x$ .
3.  $\mathfrak{I}_{\varphi}(MN) = \langle \widehat{\varphi}[MN], (\mathfrak{I}_{\varphi}^2(M) \cdot \mathfrak{I}_{\varphi}^2(N))^2 \rangle$ .
4.  $\mathfrak{I}_{\varphi}(\lambda x_{\alpha}. M_{\beta}) = \langle \widehat{\varphi}[\lambda x_{\alpha}. M_{\beta}], \lambda d. \mathfrak{I}_{\varphi[x:=d]}(M) \rangle$

One must now check that  $\mathfrak{I}_{\varphi}$  is well-defined, i.e. takes values in  $\text{D}$ , and that  $\mathfrak{I}_{\varphi}$  defines a local model, that is to say, satisfies the conditions in definitions (4.5) and (4.7), in particular

$$\begin{aligned} \mathfrak{H}(c)_{\varphi} &= \text{Const}(c) && \text{for constants } c \\ \mathfrak{H}(x)_{\varphi} &= \varphi(x) && \text{for variables } x \\ \mathfrak{H}(M)_{\varphi} &= \mathfrak{H}(N)_{\varphi} && M \lambda\text{-equivalent to } N \\ \mathfrak{H}((MN))_{\varphi} &= \text{App}(\mathfrak{H}(M)_{\varphi}, \mathfrak{H}(N)_{\varphi}) \\ \text{App}(\mathfrak{H}(\lambda x_{\alpha}. M_{\beta})_{\varphi}, d) &= \mathfrak{H}(M)_{\varphi[d/x]} \end{aligned}$$

To show the soundness of  $\lambda$ -equivalence one must show that  $\beta$  and eta-conversion are sound. Since the model we have built is not functionally extensional, eta-soundness is not immediate,

but straightforward. If  $x$  is not free in  $M$  then,  $\mathfrak{I}_\varphi(\lambda x.Mx) = \langle \widehat{\varphi}M, \lambda d.\langle \widehat{\varphi}[Md^1], (\mathfrak{I}_\varphi^2(M) \cdot d)^2 \rangle \rangle$  whereas

$\mathfrak{I}_\varphi(M) = \langle \widehat{\varphi}M, \mathfrak{I}_\varphi^2(M) \rangle$ . But, trivially,

$$\begin{aligned} \mathfrak{I}_\varphi^2(M).d &= \langle (\mathfrak{I}_\varphi^2(M).d)^1, (\mathfrak{I}_\varphi^2(M) \cdot d)^2 \rangle \\ &= \langle \widehat{\varphi}[Md^1], (\mathfrak{I}_\varphi^2(M) \cdot d)^2 \rangle \end{aligned}$$

so  $\mathfrak{I}_\varphi(M) = \langle \widehat{\varphi}M, \lambda d.\langle \widehat{\varphi}[Md^1], (\mathfrak{I}_\varphi^2(M) \cdot d)^2 \rangle \rangle = \mathfrak{I}_\varphi(\lambda x.Mx)$ .

The remaining details are routine (see e.g. [1]) and are left to the reader.

For any environment  $\varphi$  our model agrees with the Hintikka set  $\mathcal{H}$  on closed formulas. Since we need to prove cut elimination for open formulas as well, we must choose an environment that will extend to all formulas. The obvious choice is  $\varphi_0(x) = \langle x, \rho(x) \rangle$ , which is a global assignment, since the extension of  $\varphi_0$  to any term of ICTT will only return ordered pairs whose first component is a term of ICTT in canonical normal form. These terms are global, i.e. have extent  $\top_\Omega$ . Thus, with this environment, our model is semilocal.  $\square$

Let  $\Gamma$  be a consistent set of logical formulae of ICTT, i.e. for some  $A$ ,  $\Gamma \not\vdash A$ . Then there is a consistent Hintikka set for  $(\Gamma, A)$ , i.e. making  $\Gamma$  true and  $A$  false at the root world of the poset associated with  $\mathcal{H}$ .

By the preceding theorem, there is an induced semilocal model  $\mathfrak{I}_{\varphi_0}^{\Gamma, A}$  agreeing with  $\mathcal{H}$ . Thus we have established the completeness of ICTT with respect to semilocal models.

**Theorem 4.25 (Completeness of ICTT)**  $\Gamma \vdash A$  in ICTT iff  $\omega(\mathfrak{I}_{\varphi_0}^{\Gamma, A}(A)) = \top_\Omega$ .

One direction follows by soundness of local models, the other by the arguments given before the statement of the theorem.

Finally, we can derive the cut-rule for ICTT. Equivalently, have cut-elimination for ICTT with cut.

**Corollary 4.26 (Cut-elimination)** *ICTT admits cut. That is to say, if  $\Gamma \vdash A$  and  $\Gamma, A \vdash B$  in ICTT, then  $\Gamma \vdash B$ .*

**Proof.** Suppose  $\Gamma \not\vdash B$ . Let  $\llbracket G \rrbracket$  denote the value  $\omega(\mathfrak{I}_{\varphi_0}^{\Gamma, B}(G))$  of  $\Omega$ , in the local model induced by a non-contradictory Hintikka set for  $(\Gamma, B)$ . By soundness,  $\llbracket A \rrbracket = \top$ . Thus  $\mathfrak{I}_{\varphi_0}^{\Gamma, B}$  is a model of  $\Gamma \cup \{A\}$ , hence, by soundness again,  $\llbracket B \rrbracket = \top$ , a contradiction.  $\square$

The model just produced owes its peculiar form to the need to establish completeness using only cut-free methods. But now that we have established the admissibility of cut for ICTT, we may show that more conventional, global HAS-models form a complete class, using the familiar Lindenbaum model construction.

## 5 Completeness of Global Models

**Theorem 5.1 (Completeness)** *If  $\llbracket \Gamma \rrbracket \leq \llbracket B \rrbracket$  in all global models  $\mathfrak{H}$ , then there is a sequent proof of  $\Gamma \longrightarrow B$ .*

**Proof.** We construct the *Lindenbaum model*, i.e. a Heyting applicative structure in which each domain  $D_\alpha$  consists of terms of type  $\alpha$  in canonical normal form,  $\Omega$  is a Lindenbaum algebra, and the interpretation function  $\mathfrak{L}$  maps terms their normal forms.

Let  $\mathbf{App}(t_1, t_2)$  be  $\rho(t_1 t_2)$  (the canonical normal form of the term  $(t_1 t_2)$ ), and, for  $\theta$  an environment, i.e. a substitution from variables to terms, put  $\mathfrak{L}(B)_\theta = \rho(\widehat{\theta}B)$ , where  $\rho$  denotes

the canonical normal form, and where  $\widehat{\theta}$  is the extension of the substitution  $\theta$  to terms that avoids capture by suitable renaming of bound variables in a canonical way, e.g.

$$\widehat{\theta}B = (\lambda x_1 \lambda x_2 \cdots \lambda x_n. B)\theta(x_1)\theta(x_2) \cdots \theta(x_n)$$

where  $x, \dots, x_n$  are the free variables in  $B$ , as in the cut-free completeness theorem of the preceding section.

Let  $\text{Const}(c) = c$  for any constant in the language of ICTT.

Let  $\Omega$  be the set of equivalence classes

$$[B] := \{C : \Gamma, B \longrightarrow C \text{ and } \Gamma, C \longrightarrow B \text{ are theorems in ICTT} \}.$$

We turn  $\Omega$  into a partially ordered set (now that we have the cut-rule!) by defining the order relation

$$[B] \leq [C] := \Gamma, B \longrightarrow C \text{ is derivable in ICTT}$$

The cut gives us transitivity of this relation. The set  $\Omega$  is well-defined: suppose  $[B] \leq [C]$ ,  $B' \in [B]$ , and  $C' \in [C]$ . We show  $[B'] \leq [C']$  by giving a sequent tree proof:

$$\frac{\frac{\Gamma, B' \longrightarrow B \quad \frac{\Gamma, B \longrightarrow C \quad \Gamma, C \longrightarrow C'}{\Gamma, B \longrightarrow C'} \text{ cut}}{\Gamma, B' \longrightarrow C'} \text{ cut}}{\Gamma, B' \longrightarrow C'} \text{ cut}$$

The leaves follow from  $B' \in [B]$ ,  $[B] \leq [C]$ , and  $C' \in [C]$ , respectively.

In fact,  $\Omega$  is a Heyting algebra under the following interpretation of the algebra operations:

$$\begin{aligned} \perp &:= [\perp_o] \\ [B] \wedge [C] &:= [B \wedge C] \\ [B] \vee [C] &:= [B \vee C] \\ [B] \rightarrow [C] &:= [B \supset C] \end{aligned}$$

We begin by showing  $\Omega$  is a lattice with  $\wedge$  and  $\vee$  defined as above. The first lattice axiom in (1) is  $[B] \leq [B] \vee [C]$ , which means  $[B] \leq [B \vee C]$ . The sequent proof of this is

$$\frac{\overline{\Gamma, B \longrightarrow B}}{\Gamma, B \longrightarrow B \vee C} \vee_R$$

The proof of  $[C] \leq [B] \vee [C]$  is analogous.

Of the two axioms (2), we prove  $[B] \wedge [C] \leq [B]$ , *i.e.*,  $[B \wedge C] \leq [B]$ :

$$\frac{\overline{\Gamma, B, C \longrightarrow B}}{\overline{\Gamma, B \wedge C \longrightarrow B}} \wedge_L$$

Axiom (3) states that  $[B] \leq [E]$  and  $[C] \leq [E]$  imply  $[B] \vee [C] \leq [E]$ . This is immediate once translated into sequents:

$$\frac{\Gamma, B \longrightarrow E \quad \Gamma, C \longrightarrow E}{\Gamma, B \vee C \longrightarrow E} \vee_L$$

Similarly for (4), we have  $[E] \leq [B]$  and  $[E] \leq [C]$  imply  $[E] \leq [B] \wedge [C]$  since

$$\frac{\Gamma, E \longrightarrow B \quad \Gamma, E \longrightarrow C}{\Gamma, E \longrightarrow B \wedge C} \wedge_R$$

Next, we show that Lindenbaum algebra is not only a lattice but a Heyting algebra. We mentioned above that the least element  $\perp$  should be  $[\perp]$ ; now we prove  $[\perp] \leq [B]$  for all  $[B]$  in the lattice:

$$\frac{\overline{\Gamma, \perp \longrightarrow \perp}}{\overline{\Gamma, \perp \longrightarrow B}} \perp_R$$

We prove both directions of (7). To show  $[B] \wedge [C] \leq [E]$  implies  $[B] \leq [C] \rightarrow [E]$ , we construct a sequent tree in which  $\Gamma, B \wedge C \longrightarrow E$  is a premise:

$$\frac{\frac{\overline{\Gamma, B, C \longrightarrow B} \quad \overline{\Gamma, B, C \longrightarrow C}}{\overline{\Gamma, B, C \longrightarrow B \wedge C}} \wedge_R \quad \Gamma, B \wedge C \longrightarrow E}{\frac{\overline{\Gamma, B, C \longrightarrow E}}{\overline{\Gamma, B \longrightarrow C \supset E}} \supset_R} \text{cut}$$

To show the other direction, we assume  $[B] \leq [C] \rightarrow [E]$ , *i.e.*,  $\Gamma, B \longrightarrow C \supset E$ . Using cut we easily have a sequent proof of  $\Gamma, B, C \longrightarrow E$ , with which we construct the desired proof

$$\frac{\Gamma, B, C \longrightarrow E}{\Gamma, B \wedge C \longrightarrow E} \wedge_L$$

We have established that  $\Omega$  is a Heyting algebra. To turn  $\mathfrak{L}$  into a model for ICTT, we must specify a map  $\omega$  and show  $\Omega$  is definitionally complete, that is, that the meet and join  $\bigwedge \{\omega(\mathbf{App}(\lambda x.B, d)) : d \in D_\alpha\}$  and  $\bigvee \{\omega(\mathbf{App}(\lambda x.B, d)) : d \in D_\alpha\}$  exist for all  $B_{o\alpha}$  and equal  $[\forall x_\alpha B]$  and  $[\exists x_\alpha B]$ , respectively.

Define  $\omega : D_o \longrightarrow \Omega$  by  $\omega(B) = [B]$ .

As mentioned in the definition of a Heyting algebra,  $\bigwedge S$  is the greatest lower bound of an arbitrary set  $S$ , *i.e.*,  $\bigwedge S$  satisfies

- $(\bigwedge S) \leq s$  for all  $s \in S$  and
- for any  $r$ ,  $r \leq s$  for all  $s \in S$  implies  $r \leq \bigwedge S$

Since we want to prove that the meet  $\bigwedge \{\omega(\mathbf{App}(\lambda x.B, d)) : d \in D_\alpha\}$  exists and is in fact  $[\forall x_\alpha B]$ , we must show that  $[\forall x_\alpha B] \leq \omega(\mathbf{App}(\lambda x.B, d))$  for all  $d \in D_\alpha$  and also that if  $[C] \leq \omega(\mathbf{App}(\lambda x.B, d))$  for all  $d \in D_\alpha$  then  $[C] \leq [\forall x_\alpha B]$ . A proof of the former is given by

$$\frac{\frac{\rho(\Gamma), \rho(B[d/x]) \longrightarrow \rho(B[d/x])}{\rho(\Gamma), (B[d/x])' \longrightarrow \rho(B[d/x])} \lambda}{\rho(\Gamma), \rho(\forall x_\alpha B) \longrightarrow \rho(B[d/x])} \forall_L$$

where  $(B[d/x])'$  is  $\lambda$ -equivalent to  $B[d/x]$ . The uppermost sequent above follows from Lemma 3.2. Note that  $d$  is actually a term in the language, so there was no need to add constants to the language. To show the latter condition, we assume  $[C] \leq \omega(\mathbf{App}(\lambda x_\alpha.B, d))$  for all  $d \in D_\alpha$ . Hence, under any assignment  $\varphi$ ,  $[C] \leq [B]$  where  $x$  of type  $\alpha$  does not occur in  $\Gamma$  or  $C$ ,  $\Gamma, C \longrightarrow B$  holds. Thus we have a sequent tree

$$\frac{\frac{\rho(\Gamma), \rho(C) \longrightarrow \rho(B)}{\rho(\Gamma), \rho(C) \longrightarrow (B)'} \lambda}{\rho(\Gamma), \rho(C) \longrightarrow \rho(\forall x_\alpha B)} \forall_R$$

and we may conclude that  $[C] \leq [\forall x_\alpha B]$ , as desired.

To show  $\bigvee \{ \omega(\mathbf{App}(\lambda x.B, d)) : d \in D_\alpha \}$  exists and is  $[\exists x_\alpha B(x)]$ , we first show that for all  $d \in D_\alpha$ ,  $\omega(\mathbf{App}(\lambda x_\alpha.B, d)) \leq [\exists x_\alpha B]$ :

$$\frac{\frac{\rho(\Gamma), \rho(B[d/x]) \longrightarrow \rho(B[d/x])}{\rho(\Gamma), \rho(B[d/x]) \longrightarrow (B[d/x])'} \lambda}{\rho(\Gamma), \rho(B[d/x]) \longrightarrow \rho(\exists x.B)} \exists_R$$

where the upper sequent again follows from Lemma 3.2. Next we assume that  $\omega(\mathbf{App}(\lambda x.B, d)) \leq [C]$  for all  $d \in D_\alpha$ . As above, for any interpretation  $\varphi$  and appropriate variable  $x$ , we have  $[B] \leq [C]$ , *i.e.*,  $\Gamma, B \longrightarrow C$ . Thus we have a sequent proof

$$\frac{\frac{\rho(\Gamma), \rho(B) \longrightarrow \rho(C)}{\rho(\Gamma), (B)' \longrightarrow \rho(C)} \lambda}{\rho(\Gamma), \rho(\exists x.B) \longrightarrow \rho(C)} \exists_L$$

and therefore  $[\exists x.B] \leq [C]$ , as desired.

We have shown  $\langle \mathbf{D}, \mathbf{App}, \mathbf{Const}, \omega, \Omega \rangle$  is a Heyting applicative structure. We only need to show  $\mathfrak{L}$  satisfies the ICTT model axioms of definition 4.6. They are straightforward and left to the reader.

Finally, let  $\varphi : \mathbf{Vars} \longrightarrow \mathbf{D}$  be the environment given by  $\varphi(x) = x$ . Suppose  $[\Gamma] \leq [B]$  in  $\mathfrak{L}_\varphi$ . Then  $[\Gamma, \top_o] \leq [B]$ , and  $\Gamma, \top_o \longrightarrow B$  is derivable. It follows easily that there is a sequent proof of  $\Gamma \longrightarrow B$ .  $\square$

## 5.1 Remarks

The model used to prove Theorem 5.1 above is extensional for those functions which are not propositional: any two elements  $f, f'$  of  $D_{\beta\alpha}$  for which  $\mathbf{App}(f, d) = \mathbf{App}(f', d)$  for all  $d$  in  $D_\alpha$  must be identical. This is a consequence of our use of normal forms of open terms as the elements of the model. However, since there is no notion of equality in the language, extensionality for this model does not indicate that all models would be extensional.

Strictly speaking, it is the typed applicative structure in the model above which is extensional. The model itself is non-extensional in the sense we desire for  $\lambda$ Prolog: two predicate functions  $P_{o\alpha}$  and  $Q_{o\alpha}$  may have the same truth value for all inputs (that is,  $\omega(\mathbf{App}(\llbracket P \rrbracket, d)) = \omega(\mathbf{App}(\llbracket Q \rrbracket, d))$  for all  $d \in D_\alpha$ ) and yet other functions need not agree on  $P$  and  $Q$ ; *e.g.*, we need not have  $\llbracket R_{oo}(P(t)) \rrbracket = \llbracket R_{oo}(Q(t)) \rrbracket$ .

## 6 Acknowledgements

The authors wish to express a special debt to Chad Brown of Carnegie-Mellon University for pointing out certain pitfalls in the proof of cut-elimination, and correcting significant errors in an earlier manuscript. The formulations of  $\mathcal{H}_A^T$  and  $\mathcal{H}_A^{\neg F}$  in definition 4.22 are his. The authors also benefited from discussions with Steve Awodey, Peter Andrews, Gopalan Nadathur and Dale Miller.

## References

- [1] Peter Andrews. Resolution in type theory. *Journal of Symbolic Logic*, 36(3), 1971.
- [2] Peter Andrews. Herbrand award acceptance speech. Technical report, Dept. of Mathematical Sciences, Carnegie Mellon University, 2003.
- [3] Matthias Baaz. Personal communication.

- [4] Mino Bai and Harold Blair. General model theoretic semantics for higher-order horn logic programming. In *Logic Programming and Automated Reasoning, International Conference LPAR'92*, volume 624 of *LNCS*. Springer, 1992.
- [5] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.
- [6] M. DeMarco and J. Lipton. Uniform algebras. To appear.
- [7] Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel, 1983.
- [8] M. P. Fourman and D. S. Scott. In M. P. Fourman, C. J. Mulvey, and D. S. Scott, editors, *Applications of Sheaves: Proceedings of the Research Symposium on Applications of Sheaf Theory to Logic, Algebra, and Analysis*, volume 753 of *LNCS*, chapter Sheaves and Logic. Springer Verlag, July 9–21, 1977.
- [9] Harvey Friedman. Equality between functionals. In R. Parikh, editor, *Logic Colloquium*, volume 453 of *Lecture Notes in Mathematics*, pages 22–37. Springer, 1975.
- [10] J. Y. Girard. Une extension de l'interprétation de Gödel à l'analyse et son application à l'élimination de coupures dans l'analyse et la théorie des types. In J. E. Fenstad, editor, *Proceedings of the second Scandinavian proof theory symposium*. North-Holland, 1971.
- [11] J. Y. Girard. *Proof Theory and Logical Complexity*. Bibliopolis, 1987.
- [12] J.Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge University Press, 1998.
- [13] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, June 1950.
- [14] H. Läuchli. An abstract notion of realizability for which intuitionistic predicate calculus is complete. In A. Kino, J. Myhill, and R. E. Vesley, editors, *Intuitionism and Proof Theory*, studies in logic, pages 277–234. North-holland, Amsterdam, London, 1970. Proceedings of the Conference on Intuitionism and Proof Theory, Buffalo, New York, August 1968.
- [15] Dale Miller. A logical analysis of modules in logic programming. *Journal of Logic Programming*, pages 79–108, 1989.
- [16] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51(1-2):125–157, 1991.
- [17] John Mitchell. *Foundations for Programming Languages*. MIT Press, Cambridge, Massachusetts, 1996.
- [18] John Mitchell and Eugenio Moggi. Kripke-style models for typed lambda calculus. *Annals of Pure and Applied Logic*, 51:99–124, 1991.
- [19] A. Nerode and R. Shore. *Logic for Applications*. Springer, 1993.
- [20] Moto o Takahashi. A proof of cut-elimination in simple type theory. *J. Math. Soc. Japan*, 19(4), 1967.
- [21] Gordon Plotkin. Lambda definability in the full type hierarchy. In J.P. Seldin and J. R. Hindley, editors, *To H.B. Curry: Essays in Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, New York, 1980.
- [22] Raymond Smullyan. *First-Order Logic*. Dover, 1968,1995.
- [23] W. Tait. A non-constructive proof of Gentzen's Hauptsatz for second-order predicate logic. *Bulletin of the American Mathematical Society*, 72:980–983, 1966.
- [24] G. Takeuti. *Proof Theory*. North-Holland, 1987.
- [25] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction*, volume 2. Elsevier Science Publishers, 1988.
- [26] D. A. Wolfram. A semantics for  $\lambda$ prolog. *Theoretical Computer Science*, 136:277–289, 1994.

# Appendix

## .1 Higher-order Logic Programming

Although we are not concerned with logic programming in this paper, it provided a powerful motive for studying the semantics of ICTT, and justifies some of the choices made in the presentation of the theory [16], and therefore, some of the choices we must make in the semantics. In particular, we must work with a higher-order logic in which a certain *logical extensionality* must be avoided. This is discussed in the following paragraphs. Readers not familiar with  $\lambda$ -prolog may skip this section.

**Breaking truth-functionality of higher-order predicates:** In lambda-prolog we observe the following behavior.

Program:

```

module ntf.
  type p o.           p :- q.
  type q o.           q :- p.
  type f o -> o.     f(p).

```

Queries:

```

?- f(p).
solved

?- f(q).
no

?- f(p & p).
no

```

The logical equivalence of  $p$  and  $q$ , or of  $p$  and  $p \wedge p$  plays no role in the search for a proof of  $f(q)$  or of  $f(p \wedge p)$ . In other words, the following inference fails in the underlying logic of  $\lambda$ -Prolog, and in ICTT, when  $A, B$  are propositions and the type of  $p$  is  $o \rightarrow o$ :

$$\frac{A \iff B}{p(A) \rightarrow p(B)}$$

Thus, If HOHH-resolution is to be sound and complete with respect to our model theory, the meaning of the formula  $f(p \wedge p)$  must not be identified with  $f(p)$ . This is achieved by separating the cHa object of truth values  $\Omega$  from the carrier  $D_o$  of propositions. Although  $p$  and  $p \wedge p$  will have the same truth value in  $\Omega$ , they will not be identified in  $D_o$ , making it possible to consistently assign different truth values to  $f(p \wedge p)$  and  $f(p)$ .

**The logical intensionality problem:** Models may be extensional on lambda terms: if  $f$  and  $g$  are terms, say, of type  $\alpha \rightarrow \beta$ , with  $\alpha$  not the type  $o$  of truth-values, and if for all individuals  $u$  in the model's carrier of type  $\alpha$

$$\llbracket f \rrbracket u = \llbracket g \rrbracket u$$

then

$$\llbracket f \rrbracket = \llbracket g \rrbracket$$

However, we need to enforce a certain kind of meta-level failure of *logical extensionality*. Wolfram [26] gives an example of the non-extensional behavior of  $\lambda$ Prolog, which we translate into Church's type-theoretic notation:

p X :- q X.	$\forall x_o.(q_{oo}(x_o) \supset p_{oo}(x_o))$
q X :- p X.	$\forall x.(p(x) \supset q(x))$
r p.	$r_{o(oo)}(p_{oo})$
?- r q.	The goal $r_{o(oo)}(q_{oo})$
no.	is not derivable from the program clauses above.

$p$  and  $q$  are interpreted as two truth-valued functions which agree on all arguments, which must not be allowed, in general, to have the same meaning in a model.

This is also solved by distinguishing between the carriers of the types of logical formulae and the object  $\Omega$  of truth-values. The logical equivalence of  $p$  and  $q$  on predicates takes place in  $\Omega$ , not in the carrier of their type, so they are not required to have the same denotation in the carrier, even in a functionally extensional model.