Kripke Semantics for Dependent Type Theory and Realizability Interpretations *

EXTENDED ABSTRACT

by

James Lipton
Department of Mathematics
University of Pennsylvania
Philadelphia, PA 19104

Abstract

Constructive reasoning has played an increasingly important role in the development of provably correct software. Both typed and type-free frameworks stemming from ideas of Heyting, Kleene, and Curry have been developed for extracting computations from constructive specifications. These include Realizability, and Theories based on the Curry-Howard isomorphism. Realizability – in its various typed and type-free formulations – brings out the algorithmic content of theories and proofs and supplies models of the "recursive universe". Formal systems based on the propositions-as-types paradigm, such as Martin-Löf's dependent type theories, incorporate term extraction into the logic itself.

Another, major tradition in constructive semantics originated in the model theory developed by Gödel, Herbrand and Tarski, resulting in the interpretations developed by Kripke and Beth, and in subsequent categorical generalizations. They provide a *complete* semantics for constructive logic. These models are a powerful tool for building counterexamples and establishing independence and conservativity results, but they are often less constructive and less computationally oriented.

It is highly desirable to combine the power of these approaches to constructive semantics, and to elucidate some connections between them. We define modified Kripke and Beth models for syntactic Realizability and Dependent Type theory, in particular for the one-universe Intensional Martin-Löf Theory $\mathbf{ML}_0^i$. These models provide a new framework for reasoning about computational evidence and the process of term-extraction. They are defined over a *constructive* type-free metatheory based on the Feferman-Beeson theories of abstract applicative structure.

Our models have a feature which is shared by all published constructive completeness theorems for intuitionistic logic, known in the literature as "fallibility": there may be worlds in which some sentences are both false and true, a phenomenon which corresponds to the presence of empty types in various type disciplines. We also identify a natural lattice of truth values associated with type theory and realizability: the *degrees of inhabitation*.

1 Introduction

Kripke models were developed in 1963 by Saul Kripke. Similar interpretations were implicit in earlier, topological models of Tarski from the 1940's and in Beth's work in the 1950's. They were subsequently generalized by a number of researchers who strengthened some of the algebraic and topological features of the semantics. These models have proven to be a powerful tool for studying the metamathematics of

*from *Constructivity in Computer Science*, LNCS 613

intuitionistic formal systems. Much of this power lies precisely in the fact that they bring to bear on the study of formal systems an exceptionally powerful and versatile arsenal of algebraic, topological and categorical tools, yielding new consistency, conservativity and independence results. Perhaps the greatest strength of the semantics as developed in Saul Kripke's original paper, is that it supplies the means for effective systematic construction of counterexamples forced by specific requirements, in the spirit of Cohen's independence results for set theory. Often a simple diagram suffices to give intuitionistic counterexamples. These models have also been used in computer science to provide interpretations of computations in terms of state transitions, or properties invariant under certain transformations ([27, 16, 28])

Our objective here is to develop a similar tool for type-theory, and other formal systems for carrying out term extraction from constructive reasoning ([5]), extending the line of research initiated by Mitchell and Moggi for simply typed λ-Calculus in [23], and for realizability-style interpretations ([3, 30, 13]). The models developed in this paper present a number of characteristics of interest: they require the use of covers, or *delayed satisfaction of disjunctive and existential formulas*, as well as the inhabitation of void in the semantics: *we allow inconsistent, or "fallible" nodes*. This has proven to be a critical component in all known intuitionistically valid completeness theorems (see [17, 18]). As with the realizability-Kripke models to be discussed below, and in the Effective Topos (see Hyland's [12]), the truth-value structure imposed by type theory is that of the *degrees of inhabitation* of definable sets [1]. That is to say, the Heyting Algebra formed by taking, for all definable (almost-negative) sets in HA or some applicative theory like APP , equivalence classes of their existential closures under provable equivalence, constitutes a natural domain of truth values of type theory, in a sense that will be made precise below. [2]

A Thumbnail Sketch of Kripke and Realizability semantics

The kind of Kripke model $\mathcal{K}$ we will be concerned with here is, in fact, a variant of the standard one in the literature. Strictly speaking, it is a *fallible Beth model*, because

- (fallibility) nodes are allowed to be inconsistent, i.e. some p may force *every* φ, provided not every node does so. We require $p \Vdash \bot \rightarrow p \Vdash \varphi$, a node forcing an inconsistency must force everything.
- We relax the definition of forcing of disjunction and existential quantification. Our variant, an instance of *forcing with covers over a site* (see Troelstra and van Dalen's [30], or Grayson's [10]), can be thought of as an analogue of Beth's forcing with *bars*. [3]

Definition 1.1 *A fallible* **Kripke/Beth structure** $\mathcal{B} = \langle B, \leq_B, \mathbf{D}, \Vdash_B, Cov\rangle$ *over a language* $\mathcal{L}$ *(containing e.g., relation symbols,* R_i*, and constant symbols* c_i [4] *) as follows:*
$\langle B, \leq_B\rangle$ *is a pre-order. Members of* B *are called* nodes*.* $\mathbf{D}$ *assigns a* domain *of individuals to nodes in a monotone way:* $p \leq q \rightarrow \mathbf{D}(p) \subseteq \mathbf{D}(q)$*.* $\Vdash$ *is a* monotone *binary relation between* nodes *and atomic*

[1] or, in some cases, just the almost-negative ones

[2] Our work is, in some respects, a syntactic analogue of Hyland's *Effective Topos*. But the syntactic nature of the realizability makes a cruicial difference here, as can be seen from the main arguments. Our model is not a Topos, but rather, a locally cartesian category, with first order semantics. Our "object of truth values" is the set of formulas in one free variable in **APP**, or, in one case a subquotient we have called the inhabitation degrees.

[3] As a first approximation, we can think of forcing with covers as relaxing the condition that a node p of a Kripke model force a series of formulas by allowing instead that some set of nodes associated with p do the forcing. Such an associated set could be interpreted as a set of future states after state p: in which case we are tolerating some delay in the confirmation that p really does force something. See the references cited for other, more topological insights into this idea.

[4] since **APP** can be formulated relationally by taking *App* as a ternary predicate, this is all we need. Adding function symbols to the formalism is easy: the n-ary function symbol $\bar{f}$ is interpreted by the function f at node p if for every q above p $f : \mathbf{D}(q) \rightarrow \mathbf{D}(q)^n$ and the graph of f restricted to $\mathbf{D}(p)$ is contained in the graph of f at all higher nodes.

sentences over $\mathcal{L}$:

$$p \le q \quad \& \quad p \Vdash R(a) \quad \Rightarrow q \Vdash R(a).$$

satisfying the covering property *(if every member of a cover of p forces a sentence, then so does p):*

$$Cov(p, \mathbf{S}) \,\&\, (\forall q \in \mathbf{S})(q \Vdash \varphi) \Rightarrow p \Vdash \varphi$$

where: Cov is a binary relation between nodes p and *sets* of nodes $\mathbf{S} \subseteq B$, satisfying a series of *cover axioms*. Rather than give a general formulation of cover axioms here, we will limit ourselves to giving the one required for our argument, at the beginning of section 2 below. [5]

The forcing relation is extended to all sentences φ over the language $\mathcal{L}$ as follows:

1. $p \Vdash \varphi \,\&\, \psi$ iff $p \Vdash \varphi$ and $p \Vdash \psi$
2. $p \Vdash \varphi \vee \psi$ iff $(\exists \mathbf{S}) Cov(p, \mathbf{S})$ and $(\forall q \in \mathbf{S}) q \Vdash \varphi$ or $q \Vdash \psi$
3. p $\Vdash \varphi \to \psi$ iff $(\forall \ge p) q \Vdash \varphi \Rightarrow q \Vdash \psi$
4. p $\Vdash \exists x \varphi(x)$ iff $(\exists \mathbf{S}) Cov(p, \mathbf{S})$ and $(\forall q \in \mathbf{S})(\exists a \in \mathbf{D}(q)) q \Vdash \varphi(a)$
5. p $\Vdash \forall x \varphi(x)$ iff $(\forall q \ge p)(\forall a \in \mathbf{D}(q)) q \Vdash \varphi(a)$.

We give a formulation of realizability due originally to Feferman, which has been extensively used in the literature, and modified by e.g. Troelstra, Van Dalen, Diller, Beeson and others (see [30]). It is defined over an abstract applicative theory, APP, of *partial application* using the logic of existence or *partial terms*. We refer the reader to [3] or [30] for details.

Abstract realizability and interpreting theories into APP In the presentation below, we may take the approach described in detail in [18] to incorporate other theories into our formal system: interpret an arbitrary first order theory $\mathbf{S}$ into APP by adding a universe or domain predicate U whose extension is the target of the interpretation. Then we leave "up to the reader" how to define the atomic realizability $|A|(x)$ of formulas A from the object language $\mathcal{L}(\mathbf{S})$ by terms x from the *realizing metalanguage* **APP** . We will only require that the atomic realizability be faithfully copied by the atomic forcing assignments. The essential point is that we wish to consider almost any syntactic realizability over any theory. The role of **APP** is only to supply abstract realizers. We will not explicitly develop this approach here, however, since the notation is cumbersome and the details not especially enlightening.

Definition 1.2 *Let A, B be sentences over the language of APP (APP$\underline{C}$). Then we define inductively the realizability formulas $|A|$ in one free variable as follows:*

If A is prime $|A|(x)$ is $A \,\&\, x\downarrow$

$$|A \,\&\, B|(x) \equiv (A \times B)(x) \stackrel{\text{def}}{\equiv} A(\pi_0 x) \,\&\, B(\pi_1 x) \tag{1}$$

$$|A \vee B|(x) \equiv (A + B)(x) \stackrel{\text{def}}{\equiv} N(\pi_0 x) \,\&\, (\pi_0 x = 0 \to |A|(\pi_1 x)) \,\&\, (\pi_0 x \ne 0 \to |B|(\pi_1 x)) \tag{2}$$

$$|A \to B|(x) \equiv (A \Rightarrow B)(x) \stackrel{\text{def}}{\equiv} \forall y[A(y) \to xy\downarrow \,\&\, |B|(xy)] \tag{3}$$

$$|\exists y A(y)|(x) \equiv (\textstyle\sum_x A)(z) \stackrel{\text{def}}{\equiv} |A(\pi_0 z)|(\pi_1 z) \tag{4}$$

$$|\forall y A(y)|(x) \equiv (\textstyle\prod_x A)(z) \stackrel{\text{def}}{\equiv} \forall y[(zy)\downarrow \,\&\, A(y, zy)] \tag{5}$$

[5] See Grayson's [10] for a quite general formulation, or Troelstra and van Dalen, *op.cit.* for the original definition of forcing over a site, due to Joyal, and based on earlier ideas of Grothendieck.

$|A|(x)$ is usually written $x \underset{\sim}{\mathbf{r}} A$. Note that if A is a formula in n variables over APP (or APP$\underline{C}$) then the above clauses defined an associated realizability formula in $n+1$ variables. Note that if A is *prime*, A is logically equivalent to $|A|(x)$ for any variable x.

2 A Kripke Model for abstract realizability

Let $\underline{C} = \{c_i | i \in \omega\}$ be a denumerable set of fresh constants. APP$\underline{C}$ is the theory APP, together with the constants in $\underline{C}$, the axioms $c\downarrow$ for each $c \in \underline{C}$ and all schemas extended to the new language in the way specified in e.g. [3]. Let $\mathcal{L}$ be the language of APP$\underline{C}$, and Δ the set of all formal closed (variable-free) terms of $\mathcal{L}$.

We are now in a position to define the Kripke model $\mathcal{K}$ we want. The **nodes** of $\mathcal{K}$ are formulas of **APP**$\underline{C}$ in one free variable, with ordering given by

$$A \geq B \text{ iff } (\exists t \in \Delta) \text{ APP}\underline{C} \vdash \forall x[A(x) \rightarrow \quad tx\downarrow \;\&\; B(tx)]$$

(We will sometimes denote this state of affairs by A $\overset{t}{\geq} B$, or $t : A \rightarrow B$). The **domain** of the Kripke Model is the constant domain Δ. Our notion of **covers** (see section 1) will be as follows: For any node A and set of nodes $\mathbf{S}$ we have $Cov(A, \mathbf{S})$ iff

1. $\forall B \in \mathbf{S}(\exists t \in \Delta)\ (t : B \rightarrow A)$
2. if whenever, for some object D $\forall B \in \mathbf{S} \exists t_B \in \Delta\ t_B : B \rightarrow \quad D$ then $(\exists t \in \Delta)\ t : A \rightarrow D$,

i.e., $A = \inf(\mathbf{S})$ in $\langle | \mathcal{K} |, \leq \rangle$.

Finally, our atomic forcing **assignment** is given by:

$$A \Vdash \theta \quad \text{iff} \quad (\exists t \in \Delta) t : A \rightarrow |\theta| \tag{6}$$

for *atomic* [6] θ. In particular for $\theta \in \mathcal{L}(\,\mathbf{APP}\,)$ this means true ground θ are forced by every node, and false ground θ only by provably uninhabited ones. Our main result is that the equivalence (6) holds for all formulas, not just the atomic ones.

Theorem 2.1 *For every node A and sentence φ*

$$A \Vdash \varphi \quad \textit{iff} \quad (\exists t \in \Delta) t : A \rightarrow |\varphi|$$

(**proof:** omitted in this abstract)

Now, we immediately have the desired result, to wit, that the model just constructed is elementarily equivalent to abstract realizability over APP.

Corollary 2.2 *Let φ be a sentence in the language of APP, $\mathcal{K}_{APP}$ the Kripke model described above. Then*

$$\mathcal{K}_{APP} \models \varphi \quad \textit{iff} \quad \mathbf{APP} \vdash \exists x(x \underset{\sim}{\mathbf{r}} \varphi)$$

(where $x \underset{\sim}{\mathbf{r}} \varphi$ is the traditional notation for $|\varphi|(x)$).

[6] As remarked above, this atomic assignment can be replaced with little modification of the main arguments, by any formal interpreted realizability $|(\theta)^*|(t)$, where $|(\theta)^*|$ is the interpretation of a formula θ from any *object language* $\mathcal{L}'$ into the applicative metalanguage **APP**.

On the "Degrees of Inhabitation"

In the preceding section, we constructed a Kripke model elementarily equivalent to abstract realizability over **APP**. In fact, via the modifications outlined in the remarks preceding definition (1.2) we have defined a uniform class of Kripke models for a quite general notion of realizability, in which the coding of the object theory and its atomic realizability are free to vary.

In particular, the model constructed above satisfies Church's thesis and the strong computational properties found in realizability semantics. But what is the structure of such models, and what light do they shed on the truth-value structure implicit in realizability? In this section we will briefly address this issue. To begin with, we note that the models have unusual properties. They are *closed under finite suprema and infima:* The cartesian product is the supremum, and + the infimum, in fact they are weakly cartesian closed (we omit the proof in this abstract).

Also, in these models, every set of nodes has an upper bound, since we have "fallible" or inconsistent nodes at the top. Such fallible nodes were first discussed in the papers of Läuchli [16], Veldman [31] and de Swart [29], and seem to play a fundamental role in Kripke models associated with realizability, as well as in the intuitionistic completeness theorem of Friedman, Veldman, de Swart and Troelstra (see the discussion in [30]). This suggests that one should think of Kripke models of the type studied here as being developed in an intuitionistic metatheory, where consistency of nodes is not necessarily decidable. Our proofs are fully constructive and can be seen as a kind of syntactic counterpart to a constructive completeness theorem for Kripke semantics (this is brought out in more detail in [18]). These models are perhaps best conceived as *internal Kripke models*, that is to say, as models developed within a Topos or Kripke model. Another way of looking at the existence of inconsistent nodes is in terms of tableaux proofs in the style of Nerode, Fitting or Odifreddi (see e.g., [24, 25, 8]) in which we are not always constructively able to recognize when branches are infinite and consistent or finite and closed off.

One may regard the lattice-theoretic structure of these models as a syntactic analogue of reducibility orderings in recursion theory. Nodes ordered by

$$A \geq B \iff \exists t \ \mathbf{APP} \vdash \forall x(A(x) \to tx \downarrow \ \& \ B(tx))$$

constitute a structure we might call syntactic *degrees of inhabitation*, not unlike provable M-degrees (which are a stronger ordering: functions must preserve complements). A slight modification of our Kripke structures gives us a sharper picture, however. We will need a few definitions to make this precise.

Definition 2.3 *A formula over a language L extending HA or* **APP** *is called* **negative** *if it contains no disjunctions or existential quantifiers, and* **almost negative** *if it contains no disjunctions and existential quantifiers are present only next to atomic subformulas.*

Negative and almost negative formulas play a role in realizability interpretations similar to that of absolute formulas in set theory: they are equivalent to their own realizability in a uniform way.

Definition 2.4 *A formula A is called* **self-realizing** *if there is a term $\mathcal{J}_A$ of* **APP** *such that, provably in* **APP**,

(i) $A(x) \to \mathcal{J}_A(x) \ \underset{\sim}{\mathbf{r}} \ A$

(ii) $(q \ \underset{\sim}{\mathbf{r}} \ A) \to A.$

The following properties of negative and almost-negative formulas are well-known. See Beeson *op.cit.* and [30] for proofs.

Lemma 2.5 *Every negative formula is self-realizing. If A is almost negative then A is equivalent to some negative formula B, provably in* **APP** *. Furthermore, every realizability formula $|A|(x)$, or $x \,\underset{\sim}{\mathfrak{r}}\, A$ is almost negative.*

Theorem 2.6 *If B is negative and C is any formula in two free variables, then*

$$\mathbf{APP} \vdash \forall x(B(x) \rightarrow \exists z C(x,z))$$

implies that for some closed term f

$$\mathbf{APP} \vdash \forall x(B(x) \rightarrow fx \downarrow \ \& \ C(x, fx)).$$

Inspection of the proofs of theorem 3.6, above, and, e.g., theorems 2.5 and 3.9 in [17] shows that the Kripke model $\mathcal{K}$, $\mathcal{K}_{APP}$, ($\mathcal{K}_{HA}$ in [17]) , are elementarily equivalent to their *almost-negative* **reducts** $\mathcal{K}^{an}$, $\mathcal{K}^{an}_{APP}$, $\mathcal{K}^{an}_{HA}$ given by restricting the corresponding partial orders $\langle K, \leq \rangle$ to

$$\{A : A \simeq \text{ an almost-negative formula } \}.$$

What is the significance of cutting the models down to the **a-n** reducts? Theorem 2.6 provides the key. For almost negative formulas, the partial order in $\mathcal{K}$ can be easily characterized: $A \geq B$ is equivalent to the existence of a proof

$$APP \vdash (\exists x A(x)) \rightarrow (\exists y B(y)).$$

In short, our model is simply the *Lindenbaum algebra of existential closures of almost negative formulas* with the order reversed, a structure we have dubbed the *provable degrees of inhabitation*. In fact, the result is not entirely surprising. Realizability semantics means *inhabiting* statements with computational *evidence*. Therefore a natural algebraic interpretation of realizability is obtained by taking as truth-values the different degrees of consistency of provable inhabitation. To be realized is to be forced by the provably inhabited formulas. Other models are obtained by relativizing to any degree that is consistent and independent over HA or APP. This also points the way towards the sort of converse studied in Läuchli's [16] and the author's [18]. If we are to construe arbitrary Kripke models as abstract realizability interpretations we must arrange to imbed the Heyting algebra of truth-values generated by the Kripke model into an algebra of degrees of inhabitation.

3 Interpreting Dependent Types

We now develop Kripke models for type theory in the same vein. We will take as our basic theory the formulation $\mathbf{ML}_0^i$ of "one-universe" Martin-Löf type theory presented in [30]. A similar presentation can be found in Beeson *op.cit.*. In order to harness the framework developed above to model type theories, we need to add one twist to the definitions just given. Using a syntax close to that of conventional first-order logic, we will be interpreting *dependent type expressions* of the form, e.g., $\prod x : D \cdot A(x)$. At first sight, this might seem like a natural "correlate" to the first-order predicate $(\forall x \in D)(A(x))$. However, in the type-free first-order language of arithmetic (or of **APP**) the latter formula is usually understood to be an abbreviation for $(\forall x)(D(x) \rightarrow A(x))$. This clearly violates the spirit of the Martin-Löf theory, in identifying type membership $x : D$, with what –for want of a better word – we call *parametricity*, $D(x)$. In our type-free *semantics* we will remedy this by using *realizability* to distinguish between the two notions: $x : D$ will become $|(D)^r|(x)$ where the bars denote realizability, and the r- superscript denotes a translation to be defined below. However, in the *syntax* we will require an extension of the definition of well-formed formula to include *formalized bounded quantification* in order to model dependent Π and Σ types. We call such new formulas *special* or *extended formulas* in this section.

Definition 3.1 *If D is a formula and $\theta(x)$ is a formula with x free, then $(\forall_{x\in D})\theta(x)$ and $(\exists_{x\in D})\theta(x)$ are formulas, with free variables $FV(\theta)\cup FV(D)\setminus\{x\}$. If D is a formula then τ_D is an* atomic *formula with one more variable free than D. $\tau_D(a)$ is simply a formalization of $a : D$ into our "extended first-order syntax." We also write $a : D$ for $\tau_D(a)$.*[7]

All schemas of first-order logic and the axioms of **APP** are to be extended to the new formulae. Their *meaning* and role in our work should be made clear by the way they are treated in the semantics.

Definition 3.2 *An* **extended** *Kripke model for* $\mathbf{APP}\underline{C}$ *is a Kripke model which satisfies the usual definition of covers and forcing for standard formulas of* $\mathbf{APP}\underline{C}$ *, as well as the following conditions for each node p:*

$$p \Vdash (\exists_{x\in D})\theta(x) \quad \stackrel{\text{def}}{\equiv} \quad (\exists\mathbf{S})(Cov(p,\mathbf{S}))(\forall q\in\mathbf{S})(\exists a\in\mathbf{D}(q))p\Vdash a:D \ \ \& \ \ \theta(a)$$

$$p \Vdash (\forall_{x\in D})\theta(x) \quad \stackrel{\text{def}}{\equiv} \quad (\forall q\geq p)(\forall a\in\mathbf{D}(q)) \quad q\Vdash a:D\rightarrow\theta(a)$$

We must also define forcing of proofs *as well as formulas. For a a constant in $D(p)$,*

$$p\Vdash a:D \qquad (\ \textit{or} \qquad p\Vdash \tau_D(a))$$

is defined as follows:

$$p\Vdash a:C\vee D \quad \stackrel{\text{def}}{\equiv} \quad (\exists\mathbf{S})(Cov(p,\mathbf{S}))(\forall q\in\mathbf{S})$$
$$(q\Vdash \mathbf{p}_0a=0 \ \textit{and} \ q\Vdash \mathbf{p}_1a:C) \ \textit{or} \ (q\Vdash \mathbf{p}_0a\neq 0 \ \textit{and} \ q\Vdash \mathbf{p}_1a:D)$$
$$p\Vdash a:(\exists_{x\in D})\theta(x) \quad \stackrel{\text{def}}{\equiv} \quad p\Vdash \mathbf{p}_0a:D \ \textit{and} \ p\Vdash \mathbf{p}_1a:\theta(\mathbf{p}_0a)$$
$$p\Vdash a:(\forall_{x\in D})\theta(x) \quad \stackrel{\text{def}}{\equiv} \quad (\forall q\geq p)(\forall a\in\mathbf{D}(q))(q\Vdash u:D\Rightarrow q\Vdash au:\theta(u))$$

For *atomic* D, the $p\Vdash a:D$ must also be specified by the atomic forcing assignment for the model in question. Every occurrence of an application au above is *strict*: it is understood that $au\downarrow$. Also, every occurrence of the disjunctive flag condition $\mathbf{p}_0a=0$ or $\mathbf{p}_0a\neq 0$ is preceded by a tacit $N(\mathbf{p}_0a)$: the condition is decidable for natural numbers. We now need only modify our definition of realizability slightly, to take the new special formulas defined in (3.1) into account.

Now we define (extended) realizability for special formulas as follows:

Definition 3.3

$$|a:D|(x) \quad \stackrel{\text{def}}{\equiv} \quad |D|(a)$$
$$|\exists_{y\in D}A(y)| \quad \stackrel{\text{def}}{\equiv} \quad |D|(\mathbf{p}_0z)\ \&\ |A(\mathbf{p}_0z)|(\mathbf{p}_1z)$$
$$|\forall_{y\in D}A(y)| \quad \stackrel{\text{def}}{\equiv} \quad \forall y[|D|(y)\rightarrow zy\downarrow\ \&\ |A(y)|(zy)].$$

We now make use of a well-known result in the semantics of Martin-Löf type theory: there is a natural translation of the theory into **APP** via abstract realizability, stemming from Martin-Löf's own informal semantics, and developed (in roughly similar ways) in the work of Troelstra and van Dalen, Allen, Beeson, and Diller, amongst others (see [30, 2]). The details of the translation of contexts Γ and judgements θ into formulas $[\![\Gamma]\!]$ and $[\![\theta]\!]$ of **APP** are and omitted in this abstract. Troelstra and van Dalen obtain the following soundness result for their translation. A similar result is to be found in Beeson, *op.cit.*

[7] The context will make it clear whether we are referring to the syntax of the type theory or the extended first-order formulae.

Theorem 3.4 *If* $\mathbf{ML}_0^i \vdash \Gamma \gg \theta$, *then* $\mathbf{APP} \vdash [\![\Gamma]\!] \rightarrow [\![\theta]\!]$. *In particular, suppose the type A is provably inhabited in* $\mathbf{ML}_0^i$, *that is to say, there is a proof in* $\mathbf{ML}_0^i$ *ending with the sequent*

$$\gg t : A$$

for some term t. Then $\mathbf{APP} \vdash [\![A]\!](t^*)$

In order to mediate between type theory and the first-order language of partial application, we define a "reverse" r-translation $(A)^r$ of $\mathbf{ML}_0^i$ types into extended **APP** formulae. The key property of the translation is:

Lemma 3.5 (Reverse-Translation lemma) *Let A be a type in* $\mathbf{ML}_0^i$. *Then the following equivalence is provable in* **APP** *:*

$$[\![A]\!](e) \equiv |(A)^r|(e)$$

Now it is straightforward to define a Kripke model, $\mathcal{K}$, for $\mathbf{ML}_0^i$ along the lines of the preceding section, with adjustments for the special formulas defined above, which is sound in the sense that for any *extended* formula A:

$$A \text{ is provably inhabited in } \mathbf{ML}_0^i \Rightarrow \mathcal{K} \models (A)^r$$

A straightforward modification of the definitions and arguments in section 2 (with some attention paid to bounded quantification) provides is with a Kripke model $\mathcal{K}$, for *extended* formulas, as defined in 3.2 which models extended realizability.

Theorem 3.6 *Let* φ *be any (possibly nonatomic) extended (i.e. special) sentence over the language of* $\mathbf{APP}\underline{C}$ *. Then*

$$A \Vdash \varphi \iff A \geq |\varphi|.$$

From this we can immediately conclude:

Corollary 3.7 *Special formulas* φ *are true in the Kripke/Beth model* $\mathcal{K} \iff$ *they are provably realizable in* **APP**.

Finally, we have

Corollary 3.8 (Soundness of the interpretation) *Let A be a type in* $\mathbf{ML}_0^i$, *provably inhabited in that theory. In other words, for some term u, the sequent* $\gg u : A$ *is provable in* $\mathbf{ML}_0^i$. *Then* $\mathcal{K} \models (A)^r$

3.1 Extensions, Constructive Completeness, Conclusion

Our construction provides a countable collection of non-equivalent models for $\mathbf{ML}_0^i$ since the model $\mathcal{K}_A$ taken by restricting attention to all nodes above a given node A is itself a Kripke model. It is not hard to see that $\mathcal{K}$ has countably many nonequivalent nodes. Pick any node A, and let B be a *sentence* over the language of $\mathbf{APP}\underline{C}$ independent of $\mathbf{APP}\underline{C} \cup \{\exists x A(x)\}$ We cannot have $A \geq B$, since this means that for some term f

$$\mathbf{APP} \vdash \forall x(A(x) \rightarrow fx \downarrow \ \& \ B).$$

But then, by existential elimination and arrow introduction, we have

$$\mathbf{APP} \vdash (\exists x A(x)) \rightarrow B,$$

contradicting independence. (Similarly we cannot have $A \geq \neg B$). We can, of course iterate this (tacitly using Gödel's incompleteness theorem to supply new sentences), obtaining a B_1 not below A or B, and then a B_2, etc. In effect our Kripke model *lifts independence results in* **APP** *to independence results in* $\mathbf{ML}_0^i$. If the realizability of $(\theta)^r$ is independent of **APP** then θ cannot be forced by the root node of $\mathcal{K}$, nor can its negation, hence neither is provably inhabited in $\mathbf{ML}_0^i$.

Fallible Kripke/Beth models first appeared (to the author's knowledge) in a 1970 paper by Läuchli ([16]) of which this paper (as well as many others in the bibliography) is a descendant. These models continued to appear in a seminal series of papers on *constructive proofs of completeness* due to Veldman, Friedman and others (see [30] for history and references). Along lines almost identical to [18] we can adapt the Veldman-Friedman-Troelstra-Van Dalen proof to establish, constructively, completeness of *extended fallible Kripke/Beth models* for extended **APP** formulas and for "one-universe" dependent type theory. The arguments are straightforward and are omitted in this abstract.

What are the main directions for continuing this work, what are the open problems? The results just shown provide a framework for developing a tableau-based refutation method for type theory (similar to Nuprl [4], but with Kripke counter-models) as well as for new conservativity and independence results. A natural question here is: How do we extend these results to second and higher order theories. Two directions suggest themselves: developing the second-order model theory of the subject (as initiated in [30], and extended in [6]), or formalizing these arguments in Constructive Set Theory. We discuss both approaches briefly. Many questions remain open here, along with the matter of subrecursive realizability models, to be taken up in [19].

References

[1] Aczel, P. [1977] The Strength of Martin-Löf's Type Theory with One Universe, in: Mietissen, S. and Väänänen, J., (eds), *The proceedings of the symposiums on Mathematical Logic Helsinki 1975*, University of Helsinki, 1977.

[2] Allen,S. [1988], Ph. D. Dissertation, Cornell University, Ithaca, N.Y.

[3] Beeson, M. J. [1985a], *Foundations of Constructive Mathematics*, Springer-Verlag, Berlin.

[4] Constable, R. L., et al [1986], *Implementing Mathematics with the NUPRL Development System*, Prentice-Hall, N.J.

[5] Coquand, T. [1990], "On the analogy between propositions and types", in: *Logic Foundations of Functional Programming*, Huet, G. ed., Addison-Wesley, Reading, MA.

[6] Dragalin, A. G., [1988], "A Completeness Theorem for Higher Order Intuitionistic Logic: An Intuitionistic Proof", in *Mathematical Logic and its Applications*, D. Skordev, ed., Plenum.

[7] Feferman, S. [1975], "A language and axioms for explicit mathematics", in: *Algebra and Logic*, Lecture Notes in Mathematics No. 450, pp. 87-139, Springer, Berlin.

[8] Fitting, M. [1983], *Proof Methods for Modal and Intuitionistic Logics*, D. Reidel, Dordrecht, The Netherlands.

[9] Fourman, M. P. and D. S. Scott [1979], "Sheaves and logic", in: Fourman, Mulvey and Scott, (eds.), *Applications of Sheaves*, Mathematical Lecture Notes 753, pp.302-401, Springer-Verlag, Berlin.

[10] Grayson, R. J. [1983], "Forcing in intuitionistic systems without power set", Journal of Symbolic Logic 48, 670-682.

[11] Hyland, J. M. E., P. T. Johnstone and A. M. Pitts [1980], "Tripos Theory", Math. Proceedings of the Cambridge Phil. Society 88, 205-252.

[12] Hyland, J. M. E. [1982], "The effective topos". in: Troelstra, A. S. and D. S. van Dalen (eds.), *L.E.J. Brouwer Centenary Symposium*, North-Holland, Amsterdam.

[13] Kleene, S. C. [1952], *Introduction to Metamathematics*, North-Holland (1971 edition), Amsterdam.

[14] Kripke, S. [1965], "Semantical analysis of intuitionistic logic I", in: Crossley, J. N. and M. Dummett (eds.), *Formal Systems and Recursive Functions*, Proceedings of the Eighth Logic Colloquium, Oxford, 1963, North-Holland, Amsterdam, 92-130.

[15] Lambek, J. and P. J. Scott [1986], *Introduction to higher order categorical logic*, Cambridge Studies in Advanced Mathematics 7, Cambridge.

[16] Läuchli, H. [1970], "An abstract notion of realizability for which predicate calculus is complete", in: Myhill, J., A. Kino, and R. E. Vesley (eds.), *Intuitionism and Proof Theory*, North-Holland, Amsterdam, 227-234.

[17] Lipton, J. [1990], "Realizability and Kripke Forcing", in the *Annals of Mathematics and Artificial Intelligence*, Vol.4, North-Holland, Amsterdam. An expanded version appeared as technical report 90-1163, Dept. of Computer Science, Cornell University, Ithaca, NY.

[18] Lipton, J. [1990], "Constructive Kripke Semantics and Realizability", to appear in the proceedings of the *Logic for Computer Science* conference held at the Math. Sci. Research Institute, Berkeley, Nov. 1989.

[19] Lipton, J], "Kripke Models for Subrecursive computation", to appear.

[20] Martin-Löf, P. [1982], "Constructive Mathematics and Computer Programming", in Logic, Methodology and Philosophy of Science IV, North Holland, Amsterdam.

[21] Martin-Löf, P. [1984], *Intuitionistic Type Theory*, Studies in Proof Theory Lecture Notes, BIBLIOPOLIS, Napoli, Italy.

[22] McCarty, D. C. [1986], "Realizability and recursive set theory", Annals of Pure and Applied Logic 32, 11-194.

[23] Mitchell, J. and E. Moggi [1987], "Kripke-style models for typed lambda calculus", *Proceedings from Symposium on Logic in Computer Science*, Cornell University, June 1987, IEEE, Washington, D.C..

[24] Nerode, A. [1989b], "Some lecures on Intuitionistic Logic I", Proceedings on Summer School on Logical Foundations of Computer Science, CIME, Montecatini, 1988, Lecture Notes in Mathematics, Springer-Verlag, Berlin.

[25] Nerode, A., and P. Odifreddi, [1990], *Lambda Calculi and Constructive Logics*, MSI Tech. Report '90-55

[26] Odifreddi, P. [1989], *Classical Recursion Theory*, North-Holland, Amsterdam.

[27] Plotkin, G., [1980], "Lambda definability in the full type hierarchy", in : Seldin, J.P. and J. R. Hindley (eds.), *To H.B. Curry: Essays in Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, New York.

[28] Statman, R., [1982],"Logical relations and the typed Lambda Calculus", Information and Control.

[29] Swart, H. C. M. de [1976], "Another intuitionistic completeness proof", Journal of Symbolic Logic 41, 644-662.

[30] Troelstra, A. S. and D. van Dalen [1988], *Constructivism in Mathematics: An Introduction*, Vol. II, Studies in Logic and the Foundations of Mathematics, Vol. 123, North-Holland, Amsterdam.

[31] Veldman, W. [1976], "An intuitionistic completeness theorem for intuitionistic predicate logic", Journal of Symbolic Logic 41, 159-166.