CONSTRUCTIVE KRIPKE SEMANTICS AND REALIZABILITY

JAMES LIPTON

Abstract. What is the truth-value structure of realizability? How can realizability style models be integrated with forcing techniques from Kripke and Beth semantics, and conversely? These questions have received answers in Hyland’s [33], Läuchli’s [43] and in other, related or more syntactic developments cited below. Here we re-open the investigation with the aim of providing more constructive answers to both questions. A special, constructive class of so-called fallible Beth models has been shown intuitionistically to be complete for intuitionistic logic by Friedman, Veldman and others. Here we build such intuitionistic Beth models elementarily equivalent to a natural and broad class of realizabilities, thus showing that realizability interpretations correspond to the particularly effective kind of models yielded by the Veldman–Friedman–de Swart completeness theorem. In the other direction, an abstract realizability is shown constructively to be complete for intuitionistic logic. This extends earlier results along these lines due to Läuchli and others.

1. Introduction

Kripke models are a powerful metamathematical tool in constructive and computational mathematics. Often a simple diagram suffices to exhibit an intuitionistic counterexample. Such models, and their generalizations as toposes, Heyting-Valued sets, Beth models, or permutation models, also provide useful interpretations of constructive theories in terms of sets evolving through time, state-transitions or properties fixed by certain transformations.

These models were introduced and shown sound and complete by Kripke in 1963 [41], although similar ideas can be found in the work of Beth in the mid-fifties [3], in the topological models of Tarski and McKinsey in 1948 [53], and even in the early work of Jaskowsky (1936, [35]).

Kripke models are usually constructed in a classical metatheory. Most completeness proofs depend upon non-constructive principles such as the use of the Fan theorem, which is incompatible with ‘strong’ constructivity principles such as Church’s thesis. A series of papers by Friedman [24, 23], Veldman [81], de Swart [72], Lopez-Escobar [50], Troelstra [77] and Van Dalen [76] showed that if one considers Beth—rather than Kripke—models, and relaxes the requirement that the set of formulas forced at a node of such a model be consistent, a completeness theorem can be established in a fully constructive metatheory. The arguments developed by these authors constitute a major untapped resource in theoretical computer science, despite their constructivity. They have played a major role in shaping the results in this paper.

Realizability, introduced by Kleene in 1945 [39], provides an inherently more constructive interpretation of intuitionistic reasoning, close in spirit to the propositions-as-types paradigm [13]. Propositions are witnessed by computations, which constitute constructive ‘evidence’ for assertions. Their use in constructive foundations and computer science ranges from extraction of computations from constructive specifications (McCarty [51], Beeson [1], Feferman [19], Hayashi [29]) to modelling the recursive ‘universe’ (e.g., McCarty [52], Hyland [33], Scedrov [67]) to furnishing models for computation and for consistency, conservativity and independence results for theories with a strong computational component (Goodman [27], Beeson [1, 2], McCarty [52], Mitchell-Moggi [55], Scedrov [66], Kreisel and Troelstra [40], Cook-Urquardt [12], Buss [6], Nerode-Remmel-Scedrov [59], Stein [70], Dragalin [15] and many others, see e.g. Lipton [46] for references).

The question of how to relate these two paradigms: truth-value semantics and evidence semantics has been addressed by a number of researchers. In 1982 Martin Hyland constructed a topos with the same semantics as Kreisel-Troelstra realizability ([76, 40]), extended to intuitionistic
set theory [33]. This work was based on the Tripos-theory foundation worked out in Hyland, Johnstone and Pitts’ 1980 paper [34]. A syntactic, first-order analogue of this is developed in the author’s [46, 47], and a very similar Heyting-valued structure in Troelstra and van Dalen’s [76]. Various authors (especially Rosolini [65]) have shown deep connections between Hyland’s topos and “effective categories” studied by Ershov and Mulry (see Rosolini op.cit for references).

In the other direction, in 1970 [43], H. Läuchli showed how to build, out of a given Kripke model, a permutation model of (a variant of) the lambda calculus in which formulas true in the original Kripke structure are realized by certain fixpoints. In a classical metatheory he showed that one could restrict the class of fixpoints to those elements which were lambda-definable over a certain type theory. The arguments used, however, were nonconstructive.

A good understanding of the links between algebraic and realizability semantics for constructive formal systems is of great interest in designing semantics for analyzing the metamathematics of type theory, and theories based on the Curry-Howard isomorphism. Kripke models inspired by related work of H. Läuchli have been used in the study of logical relations and the semantics of programming languages by Mitchell and Moggi op.cit. and G. Plotkin [63]. Constructions similar to those below and in [47] have been used to model dependent types ([48]). The correspondences studied here are also of interest in automated deduction, in particular for obtaining the most effective possible information out of tableaux proofs.

In this paper we investigate both directions, endeavoring to remain as constructive as possible. In the first part, we show, in a constructive metatheory, that to each of a broad family of realizability interpretations, there corresponds an elementarily equivalent ‘fallible’ Beth model of the same computational complexity as the realizing theory, in the style of Troelstra-Van Dalen [76]. This result is achieved via a meta-theoretic translation of their proof of the intuitionistic completeness theorem for such models.

In the second part we give a constructive proof of a Läuchli-style converse. For each countable Kripke model we construct an elementarily equivalent realizability interpretation. The realizers are indices of functions partial recursive in the satisfaction predicate of the original Kripke model.

2. CONSTRUCTIVE BETH MODELS FOR GENERAL REALIZABILITIES

In this section we apply a translation to the (constructive) completeness theorem for Beth semantics, obtaining a functional version of this theorem. We are thus able to proceed from

\[ \varphi \text{ is realized} \]

in a theory to

\[ \varphi \text{ is everywhere forced in a constructive Beth model} \]

in a uniform constructive way.

We consider a broad notion of realizability. We start with any theory \( T \) extending a set of axioms formalizing the theory of partial application such as Beeson’s \( \text{PCA}^+ \), or EON [1], and we leave unspecified how realizability of atomic sentences is defined. We only require that the language over which the realized sentences are defined (the object language) be formalized in \( T \).

Given this situation we define a constructive, fallible Beth Model which is elementarily equivalent to the given realizability interpretation. By constructive we mean:

(1) No appeal to the Fan theorem is made

(2) The construction uses intuitionistic reasoning for recursive theories \( T \), and otherwise uses only decidability of \( T \), i.e. is recursive in \( T \).

Fallibility means \( \bot \) (falsehood) may be forced at some nodes.
Our construction is a straightforward adaptation of the Friedman–Beth–Veldman–Lopez-Escobar–Troelstra–van Dalen intuitionistic completeness theorem as formulated in [76]. It is a straightforward corollary of that theorem that Kripke models for syntactic realizability exist, given the axiomatizability of such models in suitable extensions of the theories APP discussed below. But the aim here is to explicitly exhibit such a class of models by translation of the arguments of the completeness theorem. By interpreting ‘and’ as a suitable product ×, ‘or’ as a ‘realizability’ coproduct, implication as a function space, and so on, we obtain a metatheoretic translation of the proof of the completeness theorem itself in the spirit of the Curry-Howard isomorphism. While we do not wish to reproduce all the details of the translated argument here, it is important to spell out enough to show the kind of correspondence that mediates between Kripke semantics and realizability. Furthermore, our translation requires a more careful approach to some of the problems arising in the original construction. See [76], Veldman op.cit., de Swart op.cit. and Troelstra [77] for details of the original completeness proof.

**Definition 2.1.** ([76]) A fallible, uniform (or strong) Beth Model

\[ B = \langle K, \preceq, \models, D \rangle \] is given by the following data

1. A fan (see below) \( \langle K, \preceq \rangle \) where \( \preceq \) is an ordering of the nodes by the initial segment relation.

2. \( D \), a domain function, which assigns inhabited sets to each \( k \in K \) such that, if \( k \preceq k' \), then \( D(k) \subseteq D(k') \). Each constant symbol \( c \) in \( L \) has an interpretation \( c \) in every \( D(k) \).

3. A forcing relation \( \models : \) a binary relation between nodes \( k \in K \) and prime sentences \( P(d_1, \ldots, d_n) \) over \( L \cup D(k) \) such that

   \((B1a)\) \( \exists z \in N \forall k' \preceq z \kappa(k' \models \neg P) \Rightarrow k \models \neg P \) (covering)

   \((B1b)\) \( k \models \neg P \) and \( k' \preceq k \Rightarrow k' \models \neg P \) (monotonicity).

The notation \( k' \preceq z k \) means \( k' \) is a string extending \( k \) by length \( z \), i.e.

\[ k' \preceq z k \equiv k' \preceq z k \text{ and } lth(k') - lth(k) = z. \]

**Definition 2.2.** A fan \( T \) is a finitely branching tree, that is to say an inhabited, decidable set of finite sequences of natural numbers closed under initial segments in which each node has at least one successor and which is finitely branching. More formally:

1. \( \langle T, \sigma \rangle \) \( \forall \sigma (\sigma \in T \lor \sigma \notin T) \) and \( \forall \sigma \tau (\sigma \in T \land \tau < \sigma \rightarrow \tau \in T) \)

2. \( \forall \sigma \in T \exists x \in N \sigma * < x \geq \sigma \)

3. \( \forall \sigma \in T \exists z \in N \forall x \in N (\sigma * < x \geq \sigma \rightarrow x \leq z) \)

where * denotes concatenation of strings.

We now define truth in a uniform Beth model.

**Definition 2.3.** For a node \( k \in K \) we say a sentence is true at \( k \), or is forced at \( k \) by the following inductive definition.

The prime case has already been specified.

\[ k \models A \lor B \equiv \exists z \in N \forall k' \preceq z k(k' \models A \lor k' \models \neg B) \]

\[ k \models \exists x A(x) \equiv \exists z \in N \forall k' \preceq z k \exists d \in D(k')(k' \models A(d)) \]

\[ k \models \neg A \land B \equiv k \models A \land k \models \neg B \]

\[ k \models \neg A \rightarrow B \equiv \forall k' \preceq k(k' \models \neg A \rightarrow k' \models B) \]

\[ k \models \forall x A(x) \equiv \forall k' \preceq k \forall d \in D(k')(k' \models A(d)) \]

\[ k \models \neg A \equiv k \models A \rightarrow \bot \]
In a fallible Beth model, falsity $\bot$ may be forced at a node (but not at every node) and the following condition must be met

$$\text{if } k \models \bot \text{ then for every sentence } A, k \models A.$$ 

A few remarks are in order here about the definition just given. What gives a uniform, or strong Beth model its name is the fact that the usual definition of a bar over a node $k$ in a Beth model, namely a set of nodes which intersects every path through $k$, is here made stronger: a uniform bar for $k$ is a set of nodes which intersects every path through $k$ and is bounded. Equivalently, as has been done here, it is simply defined as the set of all nodes of a given height $n$ above $k$. The point is to build compactness into the definition so that no appeal to the Fan theorem is necessary. Such a definition of bars is just a special case of forcing with covers (e.g. Grayson’s [28]). Grayson’s cover axioms, essentially a reformulation of the definition—due to Joyal—of forcing over a Grothendieck topology (see the discussion of forcing over sites in [76]), guarantee soundness with respect to intuitionistic logic. These axioms, in our case, boil down to verifying the monotonicity and covering properties defined below. The proof that these properties guarantee soundness is routine (see, e.g., Grayson’s paper, or Troelstra and Van Dalen op.cit.) and left to the reader.

**Lemma 2.4.** In a uniform Beth model, for all sentences $A$

1. (monotonicity) $k \models A$ and $k' \succeq k \Rightarrow k' \models A$
2. (covering) $(\exists z \in N)(\forall k' \succeq z, k)(k' \models A) \rightarrow k \models A$

**Proof:** By simultaneous induction on the structure of $A$: the details are straightforward.

**Realizability in Abstract Applicative Structures.** We briefly sketch how realizability interpretations are defined in an abstract applicative theory. There are various versions of such a theory, APP introduced by Feferman in the mid 1970’s (see [76]), the system of the same name formalized in set theory in McCarty’s [51] and Beeson’s EON [1]. In these theories the notions of convergence and partial application are usually taken as primitive. We adopt Beeson’s EON here. Readers familiar with this theory may skip to the end of definition (2.9).

To avoid introducing two sets of variables, formal variables and metavariables (which range over terms built up using the former) EON adopts the convention that every variable converges, but does not allow substitution of terms for variables in a universally quantified statement unless said terms converge. Thus $(\forall x)x \downarrow$ is a theorem, but $t \downarrow$ is not, for an arbitrary term $t$ since the axiom $\forall x A & t \downarrow \rightarrow A[t/x]$ cannot be applied unless $t \downarrow$ has already been shown.

We now briefly describe the theory EON, as does Beeson, as an extension of a more basic theory PCA. We refer the reader to Beeson’s book for the details.

The logic of partial terms (LPT): This logic includes the usual rules for propositional logic plus the following rules of inference:

$$R\forall \frac{B \rightarrow A}{B \rightarrow \forall x A} \quad R\exists \frac{A \rightarrow B}{\exists x A \rightarrow B} \quad (x \text{ not free in } B)$$

and the following axioms (note that A1, A2, A4, A5, A6 are axiom schemas using metavariables $t$, $s$, $t_i$ for arbitrary terms and A7, A8 schemas for special terms.)

(A1) $\forall x A & t \downarrow \rightarrow A[t/x]$

(A2) $A[t/x] & t \downarrow \rightarrow \exists x A$

$\downarrow$ is a unary (post-fix) relation symbol in the language

$\simeq$ is a defined binary relation symbol

$$t \simeq s \equiv t \downarrow \forall s \downarrow \rightarrow t = s.$$ 

We have the following axioms governing $\downarrow$, $\simeq$ and $=$
CONSTRUCTIVE KRIPEK SEMANTICS AND REALIZABILITY

(A3) \( x = x \land (x = y \rightarrow y = x) \)
(A4) \( t \approx s \land \varphi(t) \rightarrow \varphi(s) \)
(A5) \( t = s \rightarrow t \downarrow \land s \downarrow \)
(A6) \( R(t_1, ..., t_n) \rightarrow t_1 \downarrow \land ... \land t_n \downarrow \) for any atomic formula \( R(t_1, ..., t_n) \) and any terms \( t_1, ..., t_n \).
(A7) (i) For each constant symbol \( c : c \downarrow \)
(ii) For each variable \( x : x \downarrow \)

We note that (A5) is a special case of (A6). Another important special case of (A6) is
(A6') \( f(t_1, t_2, ..., t_n) \downarrow \rightarrow t_1 \downarrow \land t_2 \downarrow \land ... \land t_n \downarrow \)

N.B. (A6) does NOT imply that for any formula \( \varphi, \varphi(t_1, ..., t_n) \rightarrow t_1 \downarrow \land ... \land t_n \downarrow \). Consider, e.g. \( \neg t \downarrow \rightarrow t \downarrow \).

PCA.: We now introduce the theory PCA over the logic of partial terms.

Language: Two constants, \( k \) and \( s \).
A binary function symbol \( Ap \)
We will never explicitly write \( Ap \). We use juxtaposition, \( (st) \), or just \( st \) , to denote \( Ap(s,t) \).

Axioms of PCA:

Those of LPT together with

(PCA1) \( kxy = x \)
(PCA2) \( sxyz \simeq xz(yz) \land sxy \downarrow \)
(PCA3) \( k \neq s \)

We now define EON, Beeson’s Elementary Theory of Operations and Numbers. It is PCA together with

constants \( \pi_0, \pi_1, p, d, S_N, P_N, 0, \) and a predicate letter \( N, \) with axioms

EON1: \( pxy \downarrow \land \pi_0(pxy) = x \land \pi_1(pxy) = y \)
EON2: \( N(0) \land \forall x(N(x) \rightarrow [N(S_N(x)) \land P_N(S_N x) = x \land S_N x \neq 0]) \)
EON3: \( \forall x(N(x) \land x \neq 0 \rightarrow N(P_N x) \land S_N(P_N x) = x) \)
EON4: Definition by integer cases

\[ N(a) \land N(b) \land a = b \rightarrow d(a, b, x, y) = x \]
\[ N(a) \land N(b) \land a \neq b \rightarrow d(a, b, x, y) = y \]

EON5: Induction schema: for each formula \( \varphi \)

\( \varphi(0) \land \forall x[N(x) \land \varphi(x) \rightarrow \varphi(S_N x)] \rightarrow \forall x(N(x) \rightarrow \varphi(x)). \)

We will often write

if \( a = b \) then \( x \) else \( y \) for \( d(a, b, x, y) \)
\( \langle x, y \rangle \) for \( pxy. \)

Proofs of the following results about EON (and related systems) can be found in Beeson’s book or in [76].

Theorem 2.5 (The Recursion Theorem). There is a term \( R \) such that PCA proves

\[ Rf \downarrow \land [g = Rf \rightarrow \forall x(gx \simeq fgx)] \]

Theorem 2.6. Let \( M \) be a model of EON. Then every partial recursive function is numerically representable in \( M \).

Theorem 2.7 (Numerical and Term Existence properties).
If \( EON \vdash \exists x A \) then there is a term \( t \) such that

\[ EON \vdash t \downarrow \land A(t) \].
If \( EON \vdash \exists n(N(n) \land A(n)) \) there is a numeral \( \bar{m} = s(s(\ldots s(0)\ldots) \text{ such that } EON \vdash A(\bar{m}). \)

The following definitions hold for \( EON \) as well as the enrichment by new constants \( EONC \) we will be considering below.

**Definition 2.8.** Let \( A, B \) be formulas in one free variable over the language of \( EON \) (with possibly a denumerable set of new constants added). Then

\[
(A \times B)\langle x \rangle \equiv A(\pi_0 x) \land B(\pi_1 x)
\]

\[
(A + B)\langle x \rangle \equiv N(\pi_0 x) \land (\pi_0 x \neq 0 \rightarrow A(\pi_1 x))
\]

\[
(A \Rightarrow B)\langle x \rangle \equiv \forall y[A(y) \rightarrow xy \downarrow \land B(xy)]
\]

Let \( A(x, y) \) be a formula in two free variables. Then \( (\Sigma_x A) \) and \( (\Pi_x A) \) are formulas in one free variable given by

\[
(\Sigma A)(z) \equiv A(\pi_0 z, \pi_1 z)
\]

\[
(\Pi A)(z) \equiv \forall y[(zy) \downarrow \land A(y, zy)]
\]

**Definition 2.9.** Let \( A, B \) be sentences over the language of \( EON \) (\( EONC \)). Then we define inductively the realizability formulas \( |A| \) in one free variable as follows:

If \( A \) is prime \(|A|\langle x \rangle \) is \( A \land x \downarrow \)

\[|A \land B| \equiv |A| \times |B|
\]

\[|A \lor B| \equiv |A| + |B|
\]

\[|A \rightarrow B| \equiv |A| \Rightarrow |B|
\]

\[|\exists y A(y)| \equiv (\Sigma y|A(y)|), \text{ i.e. } |\exists y A(y)|(z) \equiv |A(\pi_0 z)|(\pi_1 z)
\]

\[|\forall y A(y)| \equiv (\Pi y|A(y)|), \text{ i.e. } |\forall y A(y)|(z) \equiv \forall y[(zy) \downarrow \land |A(y)|(zy)]
\]

\[|\neg A| \equiv |A \rightarrow \bot| \equiv \forall y|\neg A|(y)
\]

\(|A|\langle x \rangle \) is usually written \( x \not\in A \). Note that if \( A \) is a formula in \( n \) variables over \( EON \) (or \( EONC \)) then the above clauses define an associated realizability formula in \( n + 1 \) variables. Note that if \( A \) is prime, \( A \) is logically equivalent to \(|A|\langle x \rangle \) for any variable \( x \), since by the LPT conventions, \( A \rightarrow A \land (x \downarrow) \). However \(|A|\langle t/x \rangle \) is not logically equivalent to \( A \) since \(|A|\langle t/x \rangle \) is \( A \land (t \downarrow) \), which requires additional proof. The point of this definition is to guarantee the base case of the following lemma, which is easily proved by induction.

**Lemma 2.10.** \( EON \vdash |A|\langle t \rangle \rightarrow t \downarrow \) for every sentence \( A \), term \( t \).

Remarks, Conventions & Definitions: We now make precise the assumptions on the theory \( T \). Let \( \mathcal{L} \) be a language. Let \( T \) be a theory extending \( EON \) with (names for) the constants \( a \) of \( \mathcal{L} \) and the members of a denumerable set of fresh constants \( C = \{c_i | i \in \omega \} \) together with axioms \( a \downarrow \), \( c_i \downarrow \), as well as the necessary extensions of the axiom schemas of \( EON \) to include the new terms so generated. We are distinguishing between the language of the sentences to be realized, \( \mathcal{L} \cup C \), which will be called the object language, and the language \( \mathcal{L}_T \) of the theory \( T \), over which the realizability interpretation is taking place. \( \mathcal{L}_T \) will have names e.g., for combinators and the usual vocabulary of the theory of applicative structures (in this paper Beeson’s \( PCA+ \)). The object language is formalized in \( T \), i.e., we include in \( T \) a predicate \( U(x) \) such that \( U(c) \) is an axiom for each \( c \) of \( \mathcal{L} \) or \( C \).

We define realizability in \( T \) for \( \mathcal{L} \cup C \) — sentences in the usual way, but with quantifiers over individuals relativized to the object language, \( U \). We use the notation \( |\varphi|\langle x \rangle \) for the usual \( x \not\in \varphi \).
Thus, for every sentence $\varphi$ in the object language, $|\varphi|$ is a formula in one free variable in the language of the realizing theory.

For prime $P$, $|P|(x)$ will remain unspecified.

$|\varphi \& \psi|(x) \overset{\text{def}}{=} (|\varphi| \times |\psi|)(x) \equiv |\varphi|(\pi_0 x) \& |\psi|(\pi_1 x)$

$|\varphi \lor \psi|(x) \overset{\text{def}}{=} (|\varphi| + |\psi|)(x) \equiv \exists (\pi_0 x) \& (\pi_0 x \neq 0 \rightarrow |\varphi|(\pi_1 x))$

If $|\varphi| \rightarrow |\psi|(x) = (|\varphi| \rightarrow |\psi|(x) \forall z)[|\varphi|(z) \rightarrow \pi_0 x \& |\psi|(\pi_1 x)]$

$|\exists y \varphi(y)|(x) \overset{\text{def}}{=} (\Sigma_{y \in U} |\varphi(y)|)(x) \equiv D(\pi_0 x) \& |\varphi(\pi_0 x)|(\pi_1 x)$

$|\forall y \varphi(y)|(x) \overset{\text{def}}{=} (\Pi_{y \in U} |\varphi(y)|)(x) \equiv (\forall y)[D(y) \rightarrow \pi_0 x \& |\varphi(y)|(\pi_1 x)]$

We will sometimes omit all reference to relativization of quantifiers to the domain $U$. In such cases, we will recall the dependency on $U$ by writing $|\varphi|_U$. Notice that with this shorthand, e.g., $|\exists x \theta(x)| \equiv |\theta(\pi_0 x)|_{U(\pi_1 x)}$.

Now, let $\Delta$ be the set of all variable-free terms over the language $L_T$ of $T$. Let $\{\varphi_n\}$ be an enumeration of all sentences of $L \cup C$ (not $L_T$, but of the object language) with infinitely many repetitions. We now define a family of formulas of $L_T$ in one free variable indexed by all finite binary strings $\{G_k | k \in 2^{<\omega}\}$.

**Construction of $G_k$. $G_{<\omega}$ is the formula $x = 0$.** Suppose the $G_k$ have been defined for all $k \in 2^{<\omega}$ of length $u$. We now show how to define $G_{k+0}$ and $G_{k+1}$. There will be four cases. In all cases, if $k' \geq k$ then $G_{k'} = G_k$ or $G_{k'} \equiv G_k \times B$ for some formula $B$ (modulo associativity of the Cartesian product $\times$). Before proceeding with the construction we need one more bit of notation.

We write $T \vdash_{u} j : G_k \rightarrow A$ if, upon enumerating all proofs in the theory $T$ of code $\leq u$, we find for some $i \in \Delta$ and some $k' \leq k$, a proof of $T \vdash i : G_{k'} \rightarrow A$. If $k' = k$ then $j$ is $i$ itself. If $k' < k$, then, by the remarks just made, $G_k$ is either $G_{k'}$ or $G_{k'} \times B$ for some $B$. In the former case $j$ is $i$, in the latter $j$ is $i \circ \pi$ where $\pi$ is the projection function such that $T \vdash \pi : G_k \rightarrow G_{k'}$. In this case, we have not actually witnessed a proof of $j : G_k \rightarrow A$ directly, but we already have evidence that such a proof will turn up (when the code is large enough to permit the proof of $\pi : G_k \rightarrow G_{k'}$ and the relevant composition and associativity facts to be combined with the proof of $i : G_{k'} \rightarrow A$). The point of the device just defined is to permit looking back at proofs earlier formulas $G_{k'}$, so that once $u$ is sufficiently large so that $T \vdash_{u} j : G_{k'} \rightarrow A$, then for all $k'' \geq k'$, $T \vdash_{u} j : G_{k''} \rightarrow A$ for some $j$. This property will only be needed in the last clause of the proof of Theorem 2.12.

Define $L(G_k)$ to be the language $L$ together with those constants $c \in C$ occurring in $G_k$.

Now, back to the construction. Recall, we have $G_k$ defined for all $k$ up to a certain length $u$, and we are about to define $G_{k+0}$ for $n = 0, 1$.

Case 1.: $|\varphi_u| \not\in L(G_k)$: If $|\varphi_u| \not\in L(G_k)$ just means that all fresh constants from $C$ appearing in $\varphi_u$ (or $|\varphi_u|$) already appear in $G_k$. Then $G_{k+0} \equiv G_k$, for $j = 0, 1$.

Case 2.: $|\varphi_u| \in L(G_k)$, $\varphi_u = B \lor C$ (hence $|\varphi_u| = |B| + |C|$) and $\forall i \in \Delta)T \vdash_{u} \forall x[G_k(x) \rightarrow ix \downarrow \& \langle |B| + |C| \rangle(ix)]$. Then

$G_{k+0} \equiv G_k \times |B|$

$G_{k+1} \equiv G_k \times |C|$

Case 3.: $|\varphi_u| \in L(G_k), \varphi_u = \exists x \theta(x)$ and $\exists i \in \Delta)T \vdash_{u} \forall z[G_k(z) \rightarrow iz \downarrow \& \exists x \theta(x)|(iz)]$. Then

$G_{k+0} \equiv G_k \times |\theta(a)|_U$, $j = 0, 1$

where

$a = \mu_v [c \in C \& c \not\in L(G_k \cup \{|\exists x \theta(x)|\})]$. 

and, we recall, $|\theta(a)|_U$ is shorthand notation for $(U(a)\&|\theta(a))$.

Case 4: None of the above cases apply. Then

$$G_{k^*0} \equiv G_k, \quad G_{k^*1} \equiv G_k \times |\varphi_u|$$

This completes the construction of the $\{G_k|k \in 2^{<\omega}\}$.

**Lemma 2.11 (The Bar – condition lemma).** Let $k \in 2^{<\omega}$, $z \in N$, $|A| \in \mathcal{L}(G_k)$. (Recall that for $k$, $k' \in 2^{<\omega}$, $k' \geq z$ means $k'$ is a binary string extending $k$ by precisely $z$ bits.) Then:

1. $$(\exists i \in \Delta) \vdash \forall x[G_k(x) \rightarrow ix \downarrow \& |A|(ix)]$$

   iff

2. $$\forall k' \geq z \exists i \in \Delta \vdash \forall x[G_k(x) \rightarrow ix \downarrow \& |A|(ix)]$$

**Notation:** We will write $T \vdash i : G_k \rightarrow |A|$ for $T \vdash \forall x[G_k(x) \rightarrow ix \downarrow \& |A|(ix)]$.

**Proof:** By induction on $z$. The base case is

$$(\exists i \in \Delta) \vdash i : G_k \rightarrow |A|$$

iff

$$(\exists i \in \Delta) \vdash i : G_{k^*0} \rightarrow |A| \text{ and } (\exists j \in \Delta) \vdash j : G_{k^*1} \rightarrow |A|$$

Notice that the $\Rightarrow$ direction is trivial, since in all cases $G_{k^*j} \equiv G_k$ or $G_k \times |B|$ for some $B$.

Thus $\pi_0 : G_k \times |B| \rightarrow G_k$, so if $i : G_k \rightarrow A$ then $i \circ \pi_0 : G_k \times |B| \rightarrow A$.

We prove the $\Leftarrow$ direction of the base case. We investigate the four cases in the construction of the $G_k$.

Case 1:

$G_{k^*j} \equiv G_k$, trivial.

Case 2: Let $u = lh(k)$, and suppose $\varphi_u \equiv B_0 \vee B_1$, $|\varphi_u| \in \mathcal{L}(G_k)$ and

$$(\exists i \in \Delta) \vdash i : G_k \rightarrow |B_0| + |B_1|.$$  

Then $G_{k^*<j>} \equiv G_k \times |B_j| \quad (j = 0, 1)$. By hypothesis, for some $f_0$, $f_1$ in $\Delta$

$$T \vdash f_0 : G_{k^*<0>} \rightarrow |A| \text{ and } T \vdash f_1 : G_{k^*<1>} \rightarrow |A|.$$  

Then we have

$$T \vdash \{f_0, f_1\} : (G_k \times |B_0|) + (G_k \times |B_1|) \rightarrow |A|$$

and

$$T \vdash \ll id, i \gg : G_k \rightarrow G_k \times (|B_0| + |B_1|)$$

where $id \equiv \lambda x \cdot x$, and where, for suitable $f$, $g$, $\ll f, g \gg$ is the canonical map out of the coproduct, in this case $\lambda x \cdot \text{ if } x_0 = 0 \text{ then } f x_1 \text{ else } g x_1$, and $\ll f, g \gg$ is the canonical product map $\lambda x < f x, g x >$.

Then, the (self-inverse) map $h \equiv \lambda x \cdot < \pi_0(\pi_1(x)), < \pi_0 x, \pi_1 x > >$ gives

$$T \vdash (G_k \times |B_0|) + (G_k \times |B_1|) \overset{h}{=} G_k \times (|B_0| + |B_1|)$$

hence

$$T \vdash f_0, f_1 \circ h \ll id, i \gg : G_k \rightarrow |A|$$

i.e., $(\exists j \in \Delta), T \vdash j : G_k \rightarrow |A|$.
Case 3.: Suppose \( \varphi_u = \exists x \theta(x), |\varphi_u| \in \mathcal{L}(G_k) \) and \((\exists i \in \Delta)T \vdash_i : G_k \rightarrow |\exists x \theta(x)|\), meaning
\[
T \vdash \forall y[G_k(y) \rightarrow iy \downarrow \& |\theta(\pi_0(iy))|_{U}(|\pi_1(iy)|)].
\]

(3) \[
T \vdash \forall y[G_k(y) \rightarrow \ll id, i \gg y \downarrow \& G_k(y) & |\theta(\pi_0(iy))|_{U}(|\pi_1(iy)|)].
\]

By construction \( G_{k^{*,<j>} } \equiv G_k \times |\theta(c)|_U \) for some new \( c, (j = 0,1) \), and

(4) \[
(\exists f \in \Delta)T \vdash f : G_k \times |\theta(c)|_U \rightarrow |A|.
\]

Now, by the definition of case 3 in the construction of the \( G_k, c \notin \mathcal{L}(T \cup \{G_k\}) \). By lambda abstraction it is easy to see that there is a \( c \)-free term \( f' \) such that \( T \vdash f'c \equiv f \) in (4), hence, generalizing on \( c \) in (4),

(5) \[
T \vdash (\forall z \in U)\forall y[(G_k \times |\theta(z)|)(y) \rightarrow (f'z)y & |A|((f'z)y)].
\]

Combining (3) and (5)

\[
T \vdash \forall y[G_k(y) \rightarrow \]
\[
(f'(\pi_0(iy)))\ll y, \pi_1(iy) \gg \downarrow \& |A|((f'(\pi_0(iy)))\ll y, \pi_1(iy) \gg)]
\]

hence

(\exists g \in \Delta)T \vdash g : G_k \rightarrow A.

Case 4.: \( G_{k^{*,<0>} } = G_k \). Then the result is immediate.

Now for the inductive step: Assume the result is true for \( x \leq n \). Then \( k' \geq n+1 \) \( k \Rightarrow \exists k''(k' \geq_1 k'' \geq_1 n) \).

Apply the inductive hypothesis to get

(\exists i \in \Delta)T \vdash_i : G_k \rightarrow |A| \iff

(\forall k'' \geq_1 n)(\exists i \in \Delta), T \vdash_i : G_{k''} \rightarrow |A|

and use the same argument used in the base case to extend the result from \( k'' \) of length \( n \) greater than \( k \) to \( k'' \) of length \( n+1 \) over \( k \). \( \square \)

We are now in a position to define the uniform Beth model \( B \). The partially ordered set \( < P, \leq > \) is the full binary tree \( 2^{\omega} \) with the initial segment order. The domain function is \( D(k) \equiv constants \) of \( \mathcal{L}(G_k) \), that is to say, all constants of \( \mathcal{L} \) plus those fresh constants \( c \in C \) which have shown up in \( G_k \). The atomic forcing assignment is

\[
k \models P \text{ iff } P \in \mathcal{L}(G_k) \text{ and (}\exists i \in \Delta)T \vdash_i : G_k(y) \rightarrow (iy) \downarrow \& |P|(iy)].
\]

We have to show this satisfies monotonicity and covering (see the definition of uniform Beth model, conditions (B1a) and (B1b)).

Monotonicity: if \( k' \geq k \) and (\( \exists i \in \Delta \)\( T \vdash_i : G_k \rightarrow |P| \)) then, by construction \( G_{k'} \equiv G_k \) or \( G_{k'} = G_k \times A \) for some \( T \)-formula \( A \). (We are, of course, identifying \( ((G_k \times A_1) \times A_2) \times \ldots \times A_n) \) with \( G_k \times (A_1 \times \ldots \times A_n) \). So, for some projection function \( \pi \) (modulo associativity) \( T \vdash \pi : G_{k'} \rightarrow G_k \), hence \( T \vdash \pi \circ \pi : G_{k'} \rightarrow |P| \). Thus \( k' \geq k \) and \( k \models \neg P \Rightarrow k' \models \neg P \). Covering follows immediately from Lemma 2.11, just proved:

(\exists z)(\forall k' \geq z)k' \models \neg P \Rightarrow k \models \neg P.
Theorem 2.12. For every sentence $A \in \mathcal{L}(G_k)$,

$$k \vdash A \iff (\exists i \in \Delta) T \vdash i : G_k \rightarrow |A|$$

Proof: Atomic case: by definition of $B$.

and: Let $A \equiv \varphi \& \theta, A \in \mathcal{L}(G_k)$. Then $\varphi, \theta \in \mathcal{L}(G_k)$, and

$$k \vdash \varphi \& \theta \Rightarrow k \vdash \varphi \text{ and } k \vdash \theta$$

which implies, by inductive hypothesis

$$(\exists i, j \in \Delta), T \vdash i : G_k \rightarrow |\varphi| \text{ and } T \vdash j : G_k \rightarrow |\theta|$$

hence

$$T \vdash \lambda x. \langle i, j \rangle : G_k \rightarrow |\varphi| \times |\theta|.$$ Conversely: $T \vdash f : G_k \rightarrow |\varphi \& \theta| \Rightarrow T \vdash \pi_0 \circ f : G_k \rightarrow |\varphi| \text{ and } T \vdash \pi_1 \circ f : G_k \rightarrow |\theta|$. Hence, by inductive hypothesis

$$k \vdash \varphi \text{ and } k \vdash \theta, \text{ so } k \vdash \varphi \& \theta.$$ or: Say $A \equiv \varphi \lor \theta, A \in \mathcal{L}(G_k)$. Then $k \vdash \varphi \lor \theta \Rightarrow (\exists z) \forall k' \geq z, k' \vdash \theta \text{ or } k' \vdash \varphi$. If $\theta, \varphi$ are both in $\mathcal{L}(G_k)$ then $\theta$ and $\varphi$ are in $\mathcal{L}(G_{k'})$ so by inductive hypothesis

$$\forall k' \geq z \exists i \in \Delta) T \vdash i : G_{k'} \rightarrow |\varphi| \text{ or } (\exists i \in \Delta) T \vdash i : G_{k'} \rightarrow |\theta|.$$ For any $i \in \Delta$ witnessing the former case

$$T \vdash \lambda x. < 0, i \rangle : G_{k'} \rightarrow |\varphi| + |\theta|$$

and in the latter

$$T \vdash \lambda x. < 1, i \rangle : G_{k'} \rightarrow |\varphi| + |\theta|$$

hence, in all cases $\forall k' \geq z \exists h \in \Delta) T \vdash h : G_{k'} \rightarrow |\varphi| + |\theta|$. By Lemma 2.11

$$(\exists h \in \Delta) T \vdash h : G_{k'} \rightarrow |\varphi| + |\theta|.$$ Conversely, suppose $T \vdash j : G_k \rightarrow |\varphi| + |\theta|$. Then, for some $z' \in N,$

$$T \vdash \lambda x. < z', i \rangle : G_{k'} \rightarrow |\varphi| + |\theta|$$

(i.e. some number bounds the code of the proof). Let $z'' = \mu_y [y \geq z', lth(k) \text{ and } \varphi_y = \varphi \lor \theta]$ where, we recall that $\varphi_y$ is the $y$th formula in our original enumeration of sentences over $\mathcal{L} \cup C$ with infinitely many repetitions. Then, letting $z = z'' - lth(k)$

$$\forall k' \geq z, \exists j' \in \Delta) T \vdash j' : G_{k'} \rightarrow |\varphi \lor \theta|,$$

where $\varphi \lor \theta$ is $\varphi_{z''}$, hence, by (case 2 of) the construction of the $G_k$

$$G_{k' \leq < 0)} = G_{k'} \times |\varphi| \text{ and } G_{k' \leq < 1)} = G_k \times |\theta|$$

so for every $k''$ with $k'' \geq z + 1 k$ and for $i = \text{projection} \text{ onto second factor}$

$$T \vdash i : G_{k''} \rightarrow |\varphi|, \text{ or } T \vdash i : G_{k''} \rightarrow |\theta|$$

so, by the inductive hypothesis

$$(\forall k'' \geq z + 1 k) k'' \vdash |\varphi| \text{ or } k'' \vdash |\theta|$$

hence $k \vdash |\varphi \lor \theta$ (definition of forcing $\lor$).

implies: Suppose $k \vdash \psi \rightarrow \theta$ and $\psi \rightarrow \theta \in \mathcal{L}(G_k)$. Then $(\forall k') \geq k' \vdash \psi \Rightarrow k' \vdash \theta$. Let $z' = \mu_y [y \geq lth(k) \text{ and } \varphi_y = \psi \lor \psi]$. (in other words, the disjunction of $\psi$ and itself is the $y$th formula in our master enumeration $\{\varphi_y\}$). Now let $z = z' - lth(k)$. Then, for any $k' \geq z k$
Now, as observed earlier, for some projection function \( \rho \)

\[ \rho \] gives the desired conclusion \( F \) or each \( g \).

But, since \( \lambda x \) to show that a weak form of the cartesian closure axiom holds here:

\[ (\exists h \in \Delta), \ T \vdash h : G_{k'^*} \rightarrow |\psi|, \]

and by the induction hypothesis \( k'^* < 1 \rightarrow \neg \psi \). Now, by the assumption \( k \rightarrow \neg \psi \rightarrow \theta \), we must have \( k'^* < 1 \rightarrow \neg \theta \), so by the inductive hypothesis again

\[ (\exists g \in \Delta)T\vdash g : G_{k'^*} \rightarrow |\theta|. \]

Depending on which case obtains above in (6,7), we have

\[ (\exists g \in \Delta)T\vdash g : G_k \times |\psi| \rightarrow |\theta| \]

or

\[ (\exists g \in \Delta)T\vdash g : G_k \times (|\psi| + |\psi|) \rightarrow |\theta|. \]

But, since \( \lambda x \cdot 0, x : |\psi| \rightarrow |\psi| + |\psi| \), (10) implies (9), so (9) always obtains. Now, it is easy to show that a weak form of the cartesian closure axiom holds here:

\[ (\exists g \in \Delta)T\vdash g : G_k \times |\psi| \rightarrow |\theta| \]

\[ \Rightarrow \]

\[ (\exists h \in \Delta)T\vdash h : G_k \rightarrow (|\psi| \Rightarrow |\theta|). \]

For each \( g \) in the first case take \( h = \lambda x \lambda y : g((< x, y >) \). Thus, we have established

\[ \forall k' \geq z \ k(\exists h \in \Delta)T\vdash h : G_{k'} \rightarrow |\psi\rightarrow \theta| \]

which, by Lemma 2.11, gives the desired conclusion

\[ (\exists h \in \Delta)T\vdash h : G_k \rightarrow |\psi\rightarrow \theta|. \]

Conversely, suppose \( T\vdash h : G_k \rightarrow |\psi\rightarrow \theta| \) and \( k' \geq k \) with \( k' \neg \neg \psi \). Then, by the inductive hypothesis

\[ (\exists f \in \Delta)T\vdash f : G_{k'} \rightarrow |\psi|. \]

Now, as observed earlier, for some projection function \( \rho \), \( T\vdash \rho : G_{k'} \rightarrow G_k \), so we have

\[ (\exists g \in \Delta)T\vdash g : G_{k'} \rightarrow (|\psi| \Rightarrow |\theta|). \]

So if \( w = \lambda x \cdot (fx)(fx) \) i.e., \( w = sgf \), then \( T\vdash w : G_{k'} \rightarrow |\psi| \), hence, by the induction hypothesis \( k' \neg \neg \theta \), which shows \( k \neg \neg \psi \rightarrow \theta \).

**exists:** Suppose \( k \vdash \neg \exists x \theta(x) \) and \( \exists x \theta(x) \in \mathcal{L}(G_k) \) Then \( \exists y \in N \forall k' \geq y \ k \exists c \in D(k')k' \vdash \neg \theta(c) \). Thus, for some \( z \geq y \)

\[ \forall k' \geq z \ k \exists c \in \mathcal{L}(G_{k'}) \ k' \vdash \neg \theta(c). \]

Therefore, by the inductive hypothesis

\[ \forall k' \geq z \ k \exists c \in \mathcal{L}(G_{k'}) \exists f \in \Delta \ T\vdash f : G_{k'} \rightarrow |\theta(c)|. \]
But $c \in \mathcal{L}(G_k)$ means $c \in \mathcal{L} \cup C$, i.e. $U(c)$ is an axiom of $T$ and hence, we have, above, $T \vdash f : G_{k'} \rightarrow U(c) \& |\theta(c)|$, or, using our shorthand:

$$\forall k' \geq k \exists c \in \mathcal{L}(G_{k'}) \exists f \in \Delta \ T \vdash f : G_{k'} \rightarrow |\theta(c)|_U.$$ 

Let $h = \lambda x \cdot < c, f x >$ and $T \vdash h : G_{k'} \rightarrow |\exists x \theta(x)|$. By Lemma 2.11

$$\exists g \in \Delta \ T \vdash g : G_k \rightarrow |\exists x \theta(x)|.$$ 

Conversely, suppose $T \vdash g : G_k \rightarrow |\exists x \theta(x)|$ with $|\exists x \theta(x)| \in \mathcal{L}(G_k)$. Then, for some $y \in N$, $T \vdash g : G_k \rightarrow |\exists x \theta(x)|$. Let $y' = \mu w[w > y, lth(k)$ and $\varphi_w = \exists x \theta(x)]$ and let $z = y' - lth(k)$. Then, for $k' \geq k$,

$$G_{k', <j>} = G_{k'} \times |\theta(c)|_U \quad j = 0, 1$$ 

for some $c \in \mathcal{L}(G_{k', <j>}) \setminus \mathcal{L}(G_{k'})$, by construction of the $G_k$ (case 3), hence

$$T \vdash \pi_1 : G_{k', <j>} \rightarrow |\theta(c)|$$ 

and, by inductive hypothesis, $k' \leq j > ||\theta(c)|$. So,

$$\forall k'' \geq z+1 \exists k \exists c \in D(k'') \ k'' ||\theta(c)|, \text{ and hence } k ||\exists x \theta(x).$$ 

for all: Let $\forall x \theta(x) \in \mathcal{L}(G_k)$. Suppose $k ||\forall x \theta(x)|$. To begin with, we must show that we can find a sufficiently large $n \in N$ such that for some $c = c_{i(k')}$

$$\forall k' \geq n \exists c \in \mathcal{L}(G_{k', <j>}) \setminus \mathcal{L}(G_{k'}) \quad j = 0, 1.$$ 

The reason is that for some tautological existential sentence, e.g., $\exists x(P(x) \rightarrow P(x))$ for $P$ a prime formula, some $c$ is eventually used to Henkinize this sentence in the construction of the $G_k$ (case 3), and this is done infinitely often. To be more precise, if $P$ is a prime formula in one free variable, $a$ a constant in the language $\mathcal{L}$, then

$$T \vdash \forall z[|P(a)|(z) \rightarrow (id)z \downarrow |P(a)|(z)]$$ 

hence $T \vdash |\exists x(P(x) \rightarrow P(x))(< 0, id >)$, whence

$$i = \lambda x \cdot if \ x = 0 \ then < 0, id > \ else \ x$$

we must have

$$T \vdash \forall y[y = 0 \rightarrow iy \downarrow \& |\exists x(P(x) \rightarrow P(x))|(iy)].$$ 

Thus $T \vdash i : G_{<k'} \rightarrow |\exists x(P(x) \rightarrow P(x))|$ so for every $G_k$,

$$T \vdash j : G_k \rightarrow |\exists x(P(x) \rightarrow P(x))|$$ 

for some $j$. Now let $n$ be a number bounding a code for a proof of

$$T \vdash i : G_{<k'} \rightarrow |\exists x(P(x) \rightarrow P(x))|.$$ 

Then, for each $k$, there is a $j \in \Delta$ such that $T \vdash u_j : G_k \rightarrow |\exists x(P(x) \rightarrow P(x))|$, $j$ being the composition of $i$ with the relevant projection function of $G_k \rightarrow G_{<k'}$. Now, increase $lth(k)$ until we find a $k'$ with $lth(k') \geq u$ and $lth(k')$ is some $u'$ such that $\varphi_u = \exists x(P(x) \rightarrow P(x))$ in our enumeration of all $\mathcal{L} \cup C$-sentences with infinitely many repetitions. Then for this $k'$ we have, for some $j' \in \Delta$

$$T \vdash u_j' : G_{k'} \rightarrow |\exists x(P(x) \rightarrow P(x))|$$ 

with $u' = lth(k')$ satisfying the requirements of case 3 of the construction of the $G_{k'}$. (This argument is the point of our somewhat bizarre definition of $\vdash u$).
Thus, the next stage of the construction of the $G_k$ required us to find a $c \notin L(G_{k'})$ and put $G_{k' < j>} = G_k \times \{P(c) \rightarrow P(c)\}$. Thus $c \in L(G_{k' < j>}) \setminus L(G_k)$ and, in particular, given our assumption $\forall x \theta(x) \in L(G_k)$, we have $\theta(c) \notin L(G_{k' < j>}) \setminus L(G_k)$.

Since $k' \geq k$ and $k \vdash \forall x \theta(x)$ and $c \in D(k' < j>)$ we may conclude $k' < j > \vdash \neg \theta(c)$. The induction hypothesis then gives

$$
(\exists f \in \Delta) \quad T \vdash f : G_{k' < j>} \rightarrow |\theta(c)| \quad (j = 0, 1)
$$

But, in fact,

$$
(\exists g \in \Delta) \quad T \vdash g : G_k \rightarrow |\theta(c)|,
$$
due to the special nature of $G_{k' < j>} = G_k \times \{P(c) \rightarrow P(c)\}$, as we now show. Since the identity realizes $\{P(c) \rightarrow P(c)\}$ we have

$$
T \vdash (\lambda x \cdot id) : G_k \rightarrow |P(c) \rightarrow P(c)|_U
$$
hence

$$
T \vdash \lambda x \cdot <id, \lambda x \cdot id> : G_k \rightarrow |P(c) \rightarrow P(c)|
$$

so

$$
T \vdash f \circ (\lambda x \cdot <id, \lambda x \cdot id>) : G_k \rightarrow |\theta(c)|.
$$

By $\lambda$-abstraction (easy induction proof on the structure of terms in $\Delta$) we can find an $h \in \Delta$ which is $c$-free (notation: $h \in \Delta/c$) such that

$$
T \vdash hc \simeq f \circ (\lambda x \cdot <id, \lambda x \cdot id>).
$$

Thus

$$
\exists h \in \Delta/c, T \vdash hc : G_k \rightarrow |\theta(c)|
$$

with $c \notin L(G_k) \cup \{\theta(x)\}$. Writing this out in full, we have

$$
T \vdash G_k(x) \rightarrow hc \downarrow \& |\theta(c)|(hcx)
$$

(11)

$$
T + G_k(x) \vdash hc \downarrow \& |\theta(c)|(hcx).
$$

(12)

Quantifying over $c$ (with a little care we move logical axioms involving $c$, such as $U(c)$ to the right of the $\vdash$ symbol, generalize on $c$, obtaining universal quantification relativized to $U$)

$$
T + G_k(x) \vdash (\forall z \in U)(hzx \downarrow \& |\theta(z)|(hzx))
$$

hence

$$
T \vdash \forall x[G_k(x) \rightarrow (\forall z \in U)(hzx \downarrow \& |\theta(z)|(hzx))].
$$

Now let $g = \lambda x \cdot \lambda z \cdot hzx$. By the combinatory completeness theorem for partial applicative structures (see, e.g., [1]) $gx \downarrow$ and

$$
T \vdash \forall x[G_k(x) \rightarrow gx \downarrow \& (\forall z \in U)(gzx \downarrow \& |\theta(z)|(gzx))]
$$

which is to say,

$$
T \vdash \forall x[G_k(x) \rightarrow gx \downarrow \& (\forall z \in U)(U(x) \rightarrow (gz)x \downarrow \& |\theta(z)|(gzx))]
$$

\[1\] The finitely many sentences $\Gamma(c)$ involving $c$ that might have occurred here are, as explained in the remarks after lemma (2.10), $U(c)$ and instances of the LPT-schemas (A1)-(A8), such as, e.g., $c \downarrow$, or $x \simeq c \& \varphi(x) \rightarrow \varphi(c)$. We form their conjunction $\Gamma_0(c)$, move them to the right of the turnstile, and generalize on $c$ (which is now no longer present to the left of the $\vdash$), replacing it with e.g. the variable $z$. With the exception of $U(z)$ which is handled above, every conjunct of $\Gamma_0(z)$ is now an axiom of the $c$-free theory, and can therefore be dropped.
i.e., \( T \vdash g : G_{k'} \rightarrow [\forall x \theta(x)] \). This holds for all \( k' \geq k \) with \( lth(k') = u' \), so by Lemma 2.11

\[
(\exists g' \in \Delta) \ T \vdash g' : G_k \rightarrow [\forall x \theta(x)].
\]

Conversely, suppose \( (\exists h \in \Delta) \ T \vdash h : G_k \rightarrow [\forall x \theta(x)] \), i.e.,

\[
(13) \quad T \vdash \forall y[G_k(y) \rightarrow hy \downarrow \& \forall z(U(z) \rightarrow hyz \downarrow \& |\theta(z)|(hyz))]
\]

Pick \( k' \geq k \) and \( c \in L(G_{k'}) \), i.e., \( c \in \theta(k') \). Then \( U(c) \) is an axiom of \( T \) and can be dropped.

Now, by (13) and by monotonicity, i.e. the existence of a projection \( \pi \) such that

\[
T \vdash \pi : G_{k'} \rightarrow G_k,
\]

there is an \( h' \in \Delta \) with

\[
T \vdash \forall y[G_k(y) \rightarrow h'y \downarrow \& h'yc \downarrow \& |\theta(c)|(hyc)].
\]

Let \( g = \lambda y \cdot h'yc \). Then \( T \vdash \forall y[G_k(y) \rightarrow gy \downarrow \& |\theta(c)|(gy)] \). So, by induction hypothesis, \( k' \models \theta(c) \).

Thus \( k \models [\forall x \theta(x)] \). \( \Box \)

**Corollary 2.13.** Let \( \varphi \) be a sentence over the language \( L \), and \( B \) the Beth model constructed above. Then

\[
B \models \varphi \iff (\exists e \in L(T)) T \vdash |\varphi|(e) \& e \downarrow
\]

**Proof:** \( B \models \varphi \Rightarrow [\varphi] \iff (\exists h \in \Delta) \ T \vdash \forall x[G_{<\varphi}(x) \rightarrow hx \downarrow \& |\varphi|(hx)] \)

\[
\Rightarrow T \vdash \forall x(x = 0 \rightarrow hx \downarrow \& |\varphi|(hx))
\]

\[
\Rightarrow T \vdash h0 \downarrow \& |\varphi|(h0).
\]

We have the problem that \( h \in \Delta \) but \( h \) is not necessarily a term of \( T \). As already discussed in the proof above, we repeatedly abstract out the finitely many parameters \( c_1, \ldots, c_m \) of \( h \) which are not in the language of \( T \), obtaining \( h' \in T \) with \( T \vdash h'c_1 \ldots c_m \simeq h \). Then universally quantify over these constants, and instantiate them to constants of \( T \), obtaining, for some \( g \) in the language of \( T \),

\[
T \vdash g \downarrow \& |\varphi|(g).
\]

Conversely, if \( T \vdash e \downarrow \& |\varphi|(e) \) then \( T \vdash ke0 \downarrow \& |\varphi|(ke0) \) hence

\[
T \vdash ke : G_{<\varphi} \rightarrow [\varphi].
\]

By the theorem, \( [\varphi] \models [\varphi] \) and \( B \models \varphi \). \( \Box \)

Thus every realizability notion formalized in some applicative theory \( T \) corresponds, uniformly to some elementarily equivalent constructive Beth model. The class of realizability interpretations so describable is quite broad: \( T \) may come equipped with a formalized diagram, which captures the theory of some model or of, say, fragments of analysis or set theory. We refer the reader to Beeson’s detailed formulation (op. cit.) of EONb, where \( b \) is a member of \( 2^\omega \), satisfying some axiom \( \varphi(b) \), and with reduction rules \( b(n) \rightarrow b(\bar{n}) \). We do not require that \( \varphi \) be self-realizing (if it is, the interpretation will be sound with respect to EONb – deductibility). We may take the theory \( T \) to be EONb together with defining axioms \( \varphi \), and use these special facts (i.e. the information supplied by \( \varphi \)) about \( b \) to define atomic realizability. Then the construction just given will produce a Beth model for this interpretation. A \( q \) – realizability variant of the above construction is discussed in [49].
3. A Realizability Interpretation Corresponding to Kripke Models

We now pursue the connections between Kripke and Kleene semantics in the other direction. We start with an arbitrary denumerable Kripke model $\mathcal{K}$ and construct a ‘stratified’ realizability interpretation which is elementarily equivalent to the original Kripke model. Informally, our aim is to build a nontrivial (all provably recursive functions are representable) category of realizers with the same semantics. We will construct a weakly cartesian closed category $\mathcal{C}$, with weak terminal object $T$ such that $\mathcal{K} \models \varphi$ iff there is a ‘realizing morphism’ $e : T \to \models \varphi$ in $\mathcal{C}$. By weakly cartesian closed we mean that exponentials exist, but do not have a unique witnessing morphism. An object $T$ of $\mathcal{C}$ is weakly terminal if for each object $A$ of $\mathcal{C}$ there is a not necessarily unique morphism $A \to T$. These notions will be made precise below.

Let $\mathcal{L}$ be a countable language and $\mathcal{K} = \langle P, \leq, D, \models \rangle$ a countable Kripke model over $\mathcal{L}$. We first need to make $\mathcal{K}$ into a fallible model, i.e., one with a top node $*$ which forces falsehood $\bot$ as well as every other $\mathcal{L}$-sentence. We simply add the relation $* \geq p$ for each node $p \in P$, and extend the $D$-function and the forcing relation as follows:

$$D(* \to \varphi) = \bigcup\{D(p) | p \in P\}. \text{ For } p \in P, p \models \varphi \equiv p \models \neg \varphi, \text{ and }$$

$* \models \varphi$ for every atomic sentence over the language of $D(*)$.

The new model $\mathcal{K}_* = \langle P^*, \leq^*, D^*, \models \rangle$ (which will hereafter be identified with $\mathcal{K}$) is elementarily equivalent to $\mathcal{K}$. We leave the straightforward proof to the reader. We now drop the $*$ notation, but the reader should remember that the element $*$ is now assumed to be in the underlying partial order of $\mathcal{K}$.

Now we need to formalize $\mathcal{K}$ within $\mathbb{N}$, the natural numbers. Let $\mathcal{L}_* = \mathcal{L}$, except for the individuals of $\mathcal{K}$ (i.e., $\mathcal{L}_* = \mathcal{L} \cup D(*)$). For each $p \in P$, $P$ now includes *) we have a Gödel number $\lceil p \rceil \in \mathbb{N}$. For each $c \in \mathcal{L}_*$, $\lceil c \rceil \in \mathbb{N}$. We also have the following functions and relations on $N$:

- $P(\lceil p \rceil) \iff p \in P$
- $D(\lceil c \rceil) \iff c \in D(*)$
- $D(\lceil p \rceil, \lceil c \rceil) \iff c \in D(p)$ (we distinguish the binary $D$ by writing $D^2$).

We Gödel-number all sentences over $\mathcal{L}_*$. Then define the following relations. For each prime $\mathcal{L}_*$-sentence $\varphi$ we have:

- $ax(\lceil p \rceil, \lceil \varphi \rceil) \iff p \models \varphi$ in $\mathcal{K}$ and also
- $O(\lceil p \rceil, \lceil q \rceil) \iff p \leq q$ in $\mathcal{K}$.

We also have the full collection of Gödel-numbering auxiliary predicates and functions available, e.g.,

- $Sent(n) \iff n = \lceil \varphi \rceil$ and $\varphi$ is an $\mathcal{L}_*$-sentence
- $Prime(n) \iff Sent(n) \& n = \lceil \varphi \rceil \& \varphi$ is prime
- $Sub(n, m, r) \iff r$ is the Gödel number of the formula resulting from the substitution of the term with Gödel number $n$ for the variable whose Gödel number is $m$.

Let $n$ be the Gödel number of an $\mathcal{L}_*$-sentence $\varphi$. Then:

- $Dis(n)$ is true iff $\varphi$ is a disjunction, in which case $l_\lor(n)$ and $r_\lor(n)$ give the Gödel numbers of its left and right disjuncts
- $Conj(n)$ is true iff $\varphi$ is a conjunction, with $l_\land(n)$, $r_\land(n)$ the Gödel numbers of the conjuncts.
- $Imp(n)$ is true iff $\varphi$ is an implication $\theta \to \psi$ with $\lceil \theta \rceil = \text{Ant}(n)$, $\lceil \psi \rceil = \text{Cons}(n)$
- $Ex(n)$ is true iff $\varphi$ is an existential sentence $\exists x \theta(x)$ with $B_{\text{var}}(n) = \lceil x \rceil$ (the outermost bound variable) and $\text{Pred}(n) = \lceil \theta(x) \rceil$.

Thus, if $\lceil c \rceil = m$, $Sub(m, B_{\text{var}}(n), \text{Pred}(n)) = \lceil \theta(c) \rceil$.

- $Univ(n)$ is true iff $\varphi$ is a universal sentence $\forall x \theta(x)$ with $B_{\text{var}}(n) = \lceil x \rceil$ and $\text{Pred}(n) = \lceil \theta(x) \rceil$.

We now need the following special Skolem functions $w : w(\lceil p \rceil, \lceil \varphi \lor \psi \rceil) = 0$ iff $p \models \varphi$.
some of the trivial conjuncts (e.g. \text{Prime} to be proven below."

the base case. Strictly for notational convenience we give a definition of \(p\).

Negation is handled by

\[ w([p], [\varphi \lor \psi]) = 1 \text{ iff not } (p \models \varphi) \text{ and } p \models \psi \]

\[ w([p], [\varphi \lor \psi]) \text{ is undefined otherwise} \]

\(w\) is also a witness function for existential sentences

\[ w([p], [\exists x \theta(x)]) = \mu m [m = [c] \& D(p, c) \& p \models \theta(c)] \]

if a witness exists.

We have in addition the following characteristic functions:

\[ \chi_p, \chi_{D^1}, \chi_{D^2}, \chi_{Ax}, \chi_{O}. \]

To summarize, we have added to \(N\) the following collection of functions

\[ F_0 = \{w, \text{ and the characteristic functions of } P, D(), D(, ), ax, O\} \]

as well as the corresponding predicates. Formally none of these predicates are present: all information is carried by the functions, although the predicates will be used in the discussion below.

All other Gödel numbering predicates and functions can be effectively defined in terms of the functions in \(F_0\).

Now let \(F\) be the set of functions partial recursive in \(F_0\), i.e., the least class containing \(F_0\), projections, constant functions, successor, and closed under primitive recursion and the \(\mu\)-operator. Via the enumeration theorem (relativized to arbitrary functions or oracles, see, e.g. Kleene's textbook [38]), we can associate indices in \(N\) to each relativized algorithm computing functions in \(F\), satisfying the (relativized) \(s\)-\(m\)-\(n\) and recursion theorems. We will informally write \(e \in F\) to mean \(\{e\}^{F_0}\), \(e \in N\). We assume standard pairing and un-pairing. We write \(e = \langle e_0, e_1 \rangle\).

For any term \(M\), \(\Lambda x : M\) will be the usual \(F_0\)-recursive code whose existence is guaranteed by the \(s\)-\(m\)-\(n\) theorem.

For codes \(e \in F\) we now define \(\xi\) - realizability over \(N\) (not over the language of the original Kripke model) enriched with the predicates and functions given above, in the usual way:

**Definition 3.1.** if \(R\) is an atomic relation, e.g. \(k = n + m\), \(D(n, m), O(m, n), P(n)\) etc., then

\[ e _{\xi} R \text{ iff } R \text{ is true in } N. \]

The inductive cases are as usual.

Now we introduce an (inductively) defined predicate \(Sat(n, m)\) which formalizes satisfaction in the original Kripke model (not in \(N\), i.e., \(Sat([p], [\varphi])\) formalizes \(p \models \varphi\) in \(K\).

\(Sat(m, n) \equiv P(m) \& sent(n) \&\)

\{[Prime(n) \& ax(m, n)]

\[\lor [Conj(n) \lor Sat(m, l_{\&}(n)) \& Sat(m, r_{\&}(n))]\]

\[\lor [Dis(n) \& [(w(m, n) = 0 \& Sat(m, l_{\lor}(n))) \lor (w(m, n) = 1 \& Sat(m, r_{\lor}(n)))]]]\]

\[\lor [Imp(n) \& \forall y (P(y) \& O(m, y) \& Sat(y, Ant(n)) \rightarrow Sat(y, Cons(n)))]\]

\[\lor [Ex(n) \& D(m, w(m, n)) \& Sat[m, \text{Sub}[w(m, n), B_0var(n), \text{Pred}(n)]]]\]

\[\lor [Univ(n) \& \forall y \forall z (P(y) \& O(m, y) \& D(y, z) \rightarrow Sat[y, Sub(z, B_0var(n), \text{Pred}(n))]]]\]

Negation is handled by \(p \models \neg \varphi \equiv p \models \varphi \rightarrow \bot\) where, we recall, at least one node, *, does force \(\bot\).

Observe that realizability of \(Sat(m, n)\) is already defined by induction, as it is already defined for the base case. Strictly for notational convenience we give a definition of \(e \ _{\xi} Sat(m, n)\) which ‘skips’ some of the trivial conjuncts (e.g. \(Prime(n)\)) which would otherwise lead to a proliferation of projections and subscripts. Such a definition will be legitimzed by soundness of the interpretation, to be proven below.
Theorem 3.3. Sat is self-realizing i.e.,
\( (1) \exists \hat{\varepsilon} \in F \text{ such that } \text{Sat}(m, n) \rightarrow \hat{\varepsilon} m n \not{\equiv} \text{Sat}(m, n). \)
\( (2) \forall q \in F \ (q \not{\equiv} \text{Sat}(m, n)). \)

\( \hat{\varepsilon} m n \) in (1) is Curried. It means \( \{\{\hat{\varepsilon}\}\}^{F_0}(m) \).

**Proof:** By simultaneous induction on the length of sentences (e.g., define \( lth(n) = lth(\varphi) \) if \( n = ('\varphi') \), and 0 otherwise). First we give the definition of \( \hat{\varepsilon} \) : By the (relativized) recursion theorem, pick \( \hat{\varepsilon} \) satisfying:
\( \hat{\varepsilon} = \Lambda m \cdot \Lambda \text{if } \chi_P(m) = 1 \& \chi_{\text{Sent}}(n) = 1 \text{ then } \)
\( \text{if } \text{Prime}(n) \& \text{ax}(m, n) \text{ then } \langle 0, 0 \rangle \text{ else } \)
\( \text{if } \text{Conj}(n) \text{ then } \langle 1, \langle \hat{\varepsilon} m l n, \hat{\varepsilon} m r n \rangle \rangle \text{ else } \)
\( \text{if } \text{Dis}(n) \text{ then } \langle 2, \langle w(m, n), \hat{\varepsilon} m l n \rangle \rangle \text{ if } w(m, n) = 0 \text{ then } \hat{\varepsilon} m l n \)
\( \text{else } \hat{\varepsilon} m r n \rangle \rangle \text{ else } \)
\( \text{if } \text{Imp}(n) \text{ then } \langle 3, \Lambda y \Lambda q \ 	ext{if } P(y) \& Q(m, y) \text{ then } \hat{\varepsilon} y \text{Cons}(n) \rangle \rangle \text{ else } \)
\( \text{if } \text{Ex}(n) \text{ then } \langle 4, \langle w(m, n), \hat{\varepsilon} m \text{Sub}(w(m, n), B_0 \text{var}(n), \text{Pred}(n)) \rangle \rangle \text{ else } \)
\( < 5, \Lambda z \Lambda c \ 	ext{if } P(z) \& Q(c, z) \& Q(m, z) \text{then } \hat{\varepsilon} z \text{Sub}(c, B_0 \text{var}(n), \text{Pred}(n)) \rangle \rangle \)

Now for the proof: (1) and (2) will refer to the two conclusions of the theorem and the corresponding inductive hypothesis, here labelled \( H(1) \) and \( H(2) \).

**Prime case:**
Suppose \( P(m) \) and \( \text{Sent}(n) \) and \( \text{Prime}(n) \& \text{ax}(m, n) \).
\( (1) \) Then \( \hat{\varepsilon} m n = \langle 0, 0 \rangle \), so \( \langle \hat{\varepsilon} m n \rangle_0 = 0 \& \text{Prime}(n) \& \text{ax}(m, n) \) so \( \hat{\varepsilon} m n \not{\equiv} \text{Sat}(m, n) \).
\( (2) \) Suppose \( q \not{\equiv} \text{Sat}(m, n) \), i.e. \( q_0 = 0 \& \text{Prime}(n) \& \text{ax}(m, n) \).
Then \( \text{Prime}(n) \& \text{ax}(m, n) \), i.e., \( \text{Sat}(m, n) \).

**and:**
\( (1) \) Suppose \( \text{Conj}(n) \& \text{Sat}(m, l_1(n)) \& \text{Sat}(m, r_2(n)) \).
Then \( \langle \hat{\varepsilon} m n \rangle_0 = 1 \) and by \( H(1) \ (\langle \hat{\varepsilon} m n \rangle_0 \not{\equiv} \text{Sat}(m, l_1(n)) \) and \( \langle \hat{\varepsilon} m n \rangle_11 \not{\equiv} \text{Sat}(m, r_2(n)) \) so \( \hat{\varepsilon} m n \not{\equiv} \text{Sat}(m, n) \).
\( (2) \) Suppose \( q_0 = 1 \) and \( \text{Conj}(n) \& q_{10} \not{\equiv} \text{Sat}(m, l_1(n)) \& q_{11} \not{\equiv} \text{Sat}(m, r_2(n)) \).
By \( H(2) \text{Conj}(n) \& \text{Sat}(m, l_1(n)) \& \text{Sat}(m, r_2(n)) \), so \( \text{Sat}(m, n) \).

**or:**
\( (1) \) Suppose \( \text{Dis}(n) \& [(w(m, n) = 0 \& \text{Sat}(m, l_1(n))) \lor (w(m, n) = 1 \& \text{Sat}(m, r_2(n)))]. \) Then \( \langle \hat{\varepsilon} m n \rangle_0 = 2 \) and
enriched with the constants $c$ associated with realizability, see \cite{47}, then $pq$.

Hence $\hat{emn} \not\in Sat(m,n)$.

(2) Suppose $q_0 = 2$ and

$\text{Dis}(n) \land [q_{10} = 0 \land q_{11} \not\in Sat(m,l_v(n))] \lor (q_{10} \neq 0 \land q_{11} \not\in Sat(m,r_v(n))]$.

By $H(2)$ and some elementary logic $q_{10} = 0 \lor q_{10} \neq 0$ and $\text{Dis}(n) \land [Sat(m,l_v(n)) \lor Sat(m,r_v(n))]$. Hence, by the definition of $w$,

$\text{Dis}(n) \land [(w(m,n) = 0 \land Sat(m,l_v(n)) \lor (w(m,n) = 1 \land Sat(m,r_v(n)))]$ so $Sat(m,n)$.

implies:

(1) Suppose $Sat(m,n)$ and $\text{Imp}(n)$.

Then $\forall y (P(y) \land \mathcal{O}(m,y) \land Sat(y,Ant(n)) \rightarrow Sat(y,Cons(n))]$, and $(\hat{emn})_0 = 3$ and

$(\hat{emn})_1 = \Lambda y.\lambda q : (P(y) \land \mathcal{O}(m,y) \land \text{eyCons}(n))$.

Now suppose $y$ and $q$ are given with $P(y) \land \mathcal{O}(m,y) \land q \not\in Sat(y,Ant(n))$.

By $H(2)$, $Sat(y,Ant(n))$, hence $Sat(y,Cons(n))$.

By $H(1)$, $\text{eyCons}(n) \not\in Sat(y,Cons(n))$ hence

$(\hat{emn})_1 y \downarrow$, $(\hat{emn})_1 y q \downarrow$ and $(\hat{emn})_1 y q \not\in Sat(y,Cons(n))$. Therefore $\hat{emn} \not\in Sat(m,n)$.

(2) Suppose $q_0 = 3$ and

$\forall y \forall u (P(y) \land \mathcal{O}(m,y) \land u \not\in Sat(y,Ant(n)) \rightarrow q_1 y \downarrow \land q_1 y u \downarrow \land q_1 y u \not\in Sat(y,Cons(n))]$.

Further suppose $y$ is given with $P(y) \land \mathcal{O}(m,y)$, and $Sat(y,Ant(n))$.

By $H(1)$

$$\text{eyAnt}(n) \not\in Sat(y,Ant(n)),$$

hence if $e' = \text{eyAnt}(n)$ then

$$q_1 ye' \downarrow \land q_1 ye' \not\in Sat(y,Cons(n)).$$

By $H(2)Sat(y,Cons(n))$. Hence $Sat(m,n)$.

The remaining cases are straightforward. See \cite{46} for details.

**Stratified Realizability.** Recall $\mathcal{L}_k$ is the original language for which $\mathcal{K}$ was a Kripke model, enriched with the constants $c$ in $D(*) = \cup \{D(p) : p \in P\}$, where * is the top node of $\mathcal{K}$, which forces all sentences over $\mathcal{L}_k$.

Let $\varphi$ be a sentence over $\mathcal{L}_k$. We now define an associated realizability formula $|\varphi|$ in one free variable $|\varphi(x)| \equiv x \not\in \varphi$.

Realizers are pairs $< e,p >$ with $e \in F$, $p \in P$. Formally, they are integers $< e,|p| >$ where $e$ is a code for an $F_0$-recursive function. Notationally, we will identify nodes $p$ with their Gödel numbers. We define a truncated notion of application $pq$ of node $p$ to node $q$ by

$$pq = \text{if } (p \leq q) \text{then } q \text{ else }*.$$

Note that this is a ‘degenerate supremum’. Our arguments work as well if we define $pq = \text{if } (p \leq q \text{ or } q \leq p) \text{ then max}(p,q) \text{ else }*$. In fact, if $\mathcal{K}$ were sup-closed (as are many Kripke Models naturally associated with realizability, see \cite{47}), then $pq = p \lor q$ would work.
We now define the realizability we will associate with the Kripke model \( K \). For the sake of legibility, we write out the definitions of, e.g., \(|\theta|\), as sets of realizers. Thus the following are all notational variants of the same notion:

\[
x \in |\varphi|, |\varphi|(x), x \xi \varphi, x \not\in \varphi
\]

**Definition 3.4.** For \( \varphi \) atomic \(|\varphi| = \{ < n, p > | ax(p, [\varphi]) \}\)

\[|\varphi| \times |\psi| = \{ < e, p > | < e_0, p > \in |\varphi| \land < e_1, p > \in |\psi| \}\]

\[|\varphi| + |\psi| = \{ < e, p > | (e_0 = 0 \land < e_1, p > \in |\varphi|) \lor (e_0 \neq 0 \land < e_1, p > \in |\psi|) \}\]

\[|\varphi| \Rightarrow |\psi| = \{ < e, p > | (\forall u, q \geq 0) e(u, q) \downarrow \land e(e(u, q) \land p q \in |\psi|) \}\]

\[\exists x \theta(x) = \{ < e, p > | < e, p > \in |\theta(e_0)| \land < e, p > \in |\varphi| \}\]

\[\forall x \theta(x) = \{ < e, p > | (\forall c, z \geq 0 D) e(e(c, z) \downarrow \land e(e(c, z) \land p z \in |\theta(c)|) \}\]

**Lemma 3.5.** For each \( L_k \)-sentence \( \varphi \) there are functions \( s_\varphi, t_\varphi \in F \) depending only on the parameter-free structure of \( \varphi \) such that

1. \( < e, p > \in |\varphi| \Rightarrow t_\varphi(e, p) \xi Sat(p, [\varphi]) \)
2. \( e \xi Sat(p, [\varphi]) \Rightarrow < s_\varphi(e, p), p > \in |\varphi| \) for any \( K \)

The functions can in fact be computed uniformly in a code for the parameter-free structure of \( \varphi \).

Note: By parameter-free structure, we mean that \( \varphi \) can be replaced by \( \hat{\varphi} \) which has variables in place of the parameters of \( \varphi \). Thus, e.g.,

\[s_{\varphi(c)} = s_{\varphi(a)} \text{ for distinct parameters } c \text{ and } a.\]

**Proof.** We sketch the routine simultaneous induction proof of (1), (2). The uniformity in \( \varphi \) will not be discussed, but is easy to see from the proof.

**prime case:**

1. \( < e, p > \in |\varphi| \Rightarrow ax(p, [\varphi]) \Rightarrow Sat(p, [\varphi]) \Rightarrow < 0, e > \xi Sat(p, [\varphi]) \)
   
   for any \( e \). Hence \( t_\varphi = \Lambda x \cdot < 0, x > \) will do.

2. \( e \xi Sat(p, [\varphi]) \Rightarrow e_0 = 0 \land ax(p, [\varphi]) \Rightarrow < x, p > \in |\varphi| \) for any \( K \)
   
   and:

1. Suppose \( < e, p > \in |\varphi| \times |\psi| \). Then \( < e_0, p > \in |\varphi| \) and \( < e_1, p > \in |\psi| \). By \( H(1) \) there are codes \( t_\varphi, t_\psi \) such that \( t_\varphi(p, e_0) \xi Sat(p, [\varphi]) \) and \( t_\psi(p, e_1) \xi Sat(p, [\psi]) \) hence
   
   \( < 1, t_\varphi(p, e_0), t_\psi(p, e_1) > > \xi Sat(p, \varphi \land \psi) \) and

\[t_\varphi \land \psi = \Lambda z \Lambda x \cdot < 1, < t_\varphi(z, x_0), t_\psi(z, x_1) >>.\]

2. A similar argument shows we should take

\[s_\varphi \land \psi = \Lambda z \Lambda x \cdot < s_\varphi(z, x_0), s_\psi(z, x_1) >>.\]

**or:**

Take \( t_\varphi \land \psi = \Lambda z \Lambda x \cdot < 2, < x_0, if x_0 = 0 \text{ then } t_\varphi(z, x_1) \else t_\psi(z, x_1) >>.\)

\[s_\varphi \land \psi = \Lambda z \Lambda x \cdot < x_0, if x_0 = 0 \text{ then } s_\varphi(z, x_1) \else s_\psi(z, x_1) >.

**implies:**

Suppose \( < e, p > \in |\varphi| \Rightarrow |\psi| \) and \( P(q), p \leq q, u \xi Sat(q, [\varphi]) \).

By \( H(2) \), \( < s_\varphi(u, q), q > \in |\varphi| \), hence \( < e(< s_\varphi(u, q), q >), q > \in |\varphi| \).

By \( H(1) \), \( t_\varphi(e(< s_\varphi(u, q), q >), q) \xi Sat(p, [\varphi]) \) so
A similar argument shows that

\[ \exists \psi \text{ Sat}(\varphi \rightarrow \psi) \]

hence

\[ t_{\varphi \rightarrow \psi} = \Lambda w \varphi p. < 3, \Lambda q \Lambda u. \text{ if } O(p, q) \text{ then } t_{\varphi}(e(< s_{\varphi}(u, q), q >), q) > . \]

A similar argument shows

\[ s_{\varphi \rightarrow \psi} = \Lambda w \Lambda p \Lambda z. \text{ if } O(p, z_1) \text{ then } s_{\varphi}(w(< t_{\varphi}(z_0, z_1), z_1 >), z_1) \text{ else } r_o. \]

where \( r_o \) denotes the (code of the) root node of \( K \).

**Corollary 3.6.**

\[ p \models \varphi \iff (\exists x) x.p > \in |\varphi|. \]

**Proof:** \( p \models \varphi \equiv \text{ Sat}([p], [\varphi]) \Rightarrow \) by Theorem 3.3 \( \exists e \models \text{ Sat}([p], [\varphi]) \). By Lemma 3.5,

\[ < s_{\varphi}(e, p), p > \in |\varphi|. \]

Conversely, suppose \( < e, p > \in |\varphi| \). Then \( t_{\varphi}(e, p) \models \text{ Sat}([p], [\varphi]) \). Hence, by Theorem 3.3, \( \text{ Sat}([p], [\varphi]) \), whence \( p \models \varphi \). \( \square \)

We now briefly sketch how a realizing category can be built from the notions just defined, which is elementarily equivalent to the original Kripke model. We construct a weakly cartesian closed category \( C \) which is nontrivial in the sense that all partial recursive functions are representable by (equivalence classes of) morphisms of \( C \) (in a way that will be made precise below), and define an abstract realizability in \( C \) as follows:

\( \varphi \) is realizable in \( C \) iff there is a morphism from the (weakly) terminal object \( T \) of \( C \) into the object \( |\varphi| \) of \( C \) representing \( \varphi \).

Extensional equivalence classes of local realizers of \( \varphi \) in \( C \) will be morphisms \( e : A \rightarrow |\varphi| \) for arbitrary objects \( A \) of \( C \).

**Definition 3.7.** \( C_K \) is the category whose objects are upward-closed sets of pairs \( < e, [p] > \) such that \( e \in N \) and \( [p] \in N \) is a code for a node \( p \) of the Kripke Model \( K \).

We will drop the brackets and identify \( p \) with its code. By \( A \) upward closed, we mean

\[ < e, p > \in A \& r \geq p \Rightarrow < e, r > \in A. \]

(A somewhat more constructive category is obtained if we restrict \( C_K \) to objects generated by \( \times, +, \Rightarrow, \Sigma, \Pi \) from the basic sets

\[ |\varphi| = \{ < n, p > : ax(p, \varphi), n \in N \} \]

and

\[ Q_p = \{ < n, r > | n \in N, r \geq p \} \]

where \( \varphi \) is atomic and \( r \geq p \) is shorthand for \( O([p], [r]). \)

Morphisms are triples \( (A, [\varphi], B) \) where \( e \in N \) and \( e \) denotes the function \( \{e\}^{F_0} \) whose action is given by

\[ e | < n, p > = < e(< n, p >), p >. \]
and \([e]\) is the equivalence class of \(e\) under extensional equality. To be precise, let “pre-morphisms” be triples \((A, e, B)\), and define equivalence \((A, e, B) \sim (A, f, B)\) by extensional equality, i.e.

\[
<n, p> \in A \Rightarrow e|<n, p> = f|<n, p>.
\]

Morphisms are classes under this equivalence. Our somewhat abusive applicative notation means the following: the left hand side is defining a new application operator ‘|’, which is the application of the category \(C_K\) just defined. The right hand side is application as defined earlier in this section, to wit, \(e(<n, p>)\) means \(\{e\}_F^0(<n, p>)\). Furthermore, \(e\) must satisfy the requirement

\[
<n, p> \in A \Rightarrow e(<n, p>) \downarrow \& e|<n, p> \in B.
\]

As in the preceding section, morphisms \((A, e, B)\) will just be referred to by the code \(e\) (i.e., domain and codomain will be tacit), and we will drop the brackets. The identity \(id\) defined by \(\pi_0 \equiv \Lambda x \cdot x_0\) satisfies \(id|<n, p> = <n, p>\), so \(1_A = <A, \pi_0, A>\). Composition of two morphisms, \(f\) and \(e\) is denoted \(f|e\), and is given by \((f|g)|x = f|(g|x)\). We will not bother to distinguish notationally between \(e\) as a morphism in \(C_K\) and \(e\) as a map from \(N\) to \(N\). The different application operators \(e|<n, p>\) and \(e(<n, p>)\) should make this distinction clear.

We recall that a category \(C\) has **weak exponents** if for every pair of objects \(A, B\) of \(C\) there is an object \((A \Rightarrow B)\) called the **weak exponent determined by** \(A\) and \(B\) and a morphism \(app_{A,B}\) in \(Hom_C((A \Rightarrow B) \times A, B)\) such that for any object \(C\) and any morphism \(f \in Hom_C(C \times A, B)\) there is a not necessarily unique map \(\Lambda f\) in \(Hom_C(C, A \Rightarrow B)\) making the diagram

\[
\begin{array}{ccc}
(A \Rightarrow B) \times A & \xrightarrow{app} & B \\
\Lambda^*f \times 1_A & \downarrow & \\
C \times A & \xrightarrow{f} & B
\end{array}
\]

commute.

If the morphism \(\Lambda f\) is unique for every such situation, we drop the word *weak* in the definition just given.

**Lemma 3.8.** \(C_K\) is weakly cartesian closed. That is to say, it is closed under finite products, coproducts, and weak exponentiation, and has a weak terminal object \(T\) (there is a not necessarily unique morphism from every object \(A\) into \(T\)).

**Proof:** We sketch the key definitions and leave the verifications to the reader. Products are given by

\[
A \times B = \{<e, f>, pq | <e, p> \in A, <f, q> \in B\}
\]

or, equivalently,

\[
\{<e, f>, p > | <e, p> \in A, <f, p> \in B\}.
\]

The projection maps are \(\rho_0\) and \(\rho_1\), where

\[
\rho_0 = \Lambda z \cdot z_{00}, \quad \rho_1 = \Lambda z \cdot z_{01}.
\]
Define

\[ \ll e, f \gg = \Lambda z \cdot \ll ez, fz \gg. \]

Then \( \ll e, f \gg : C \rightarrow A \times B \), and \( \rho_0 \| \ll e, f \gg = e \), and \( \rho_1 \| \ll e, f \gg = f \) and the map is easily shown unique (up to extensional equivalence).

Coproducts \( A + B \) are defined as in 3.4:

\[ A + B = \{ \ll e, p \| (e_0 = 0 & \ll e_1, p \gg \notin A) \vee (e_0 \neq 0 & \ll e_1, p \gg \in B) \} \]

with the associated canonical inclusions:

\[ h_0 : A \rightarrow A + B, \text{ given by } \Lambda z \cdot \ll 0, z_0 \gg \text{ and } \]

\[ h_1 : B \rightarrow A + B \text{ by } \Lambda z \cdot \ll 1, z_0 \gg. \]

If \( e : A \rightarrow C \) and \( f : B \rightarrow C \) in \( C_K \), then let \( \ll e, f \gg = \Lambda z \cdot \ll e(\ll y_0, z_0 \gg, y, z_1 \gg) \|. \) It is easily seen that \( \ll e, f \gg : A + B \rightarrow C \) is in \( C_K \) and

\[ \ll e, f \gg \| h_0 = e, \ll e, f \gg \| h_1 = f. \]

Uniqueness is straightforward, and left to the reader.

Exponents: Define, for objects \( B, C \) of \( C_K \)

\[ B \Rightarrow C = \{ \ll e, p \| [(\forall < u, q \gg \notin B) \ll e(< u, q \gg), pq \gg \in G] \}. \]

Suppose \( e : A \times B \rightarrow C \) is \( C_K \). Let \( \Lambda^* e = \Lambda y \Lambda z \cdot e(\ll y_0, z_0 \gg, y, z_1 \gg) \). Then, if \( < w, s \gg \in A. \)

\[ \Lambda^* e|< w, s \gg = \Lambda^* e(\ll w, s \gg), s \gg \]

We claim that \( \Lambda^* e|< w, s \gg \in B \Rightarrow C \). For suppose \( < w, q \gg \in B \). Then

\[ < [\Lambda z \cdot e(\ll w, z_0 \gg, sz_1 \gg)](\ll v, q \gg), sq \gg = < e(\ll w, v \gg, sq \gg), sq \gg \]

is in \( C \) since \( < w, s \gg \in A \), and \( < r, q \gg \in B \Rightarrow < < w, v \gg, sq \gg \in A \times B \),

hence, by the assumption that \( e : A \times B \rightarrow C \),

\[ e|< w, v \gg, sq \gg = < e(\ll w, v \gg, sq \gg), sq \gg \in C. \]

Thus, if \( e : A \times B \rightarrow C \) then \( \Lambda^* e : A \rightarrow (B \Rightarrow C) \).

Now suppose \( e : A \rightarrow (B \Rightarrow C) \). In other words, \( e \in \omega \) is a map of the type shown under \( \rightarrow \) — application. Then define the code \( \Lambda e \) (of an \( F_0 \) — recursive function)

\[ \Lambda e = \Lambda z \cdot (e(\ll z_0, z_1 \gg))(\ll z_0, z_1 \gg). \]

Then, under \( \rightarrow \) — application, \( \Lambda e : A \times B \rightarrow C \) is easily checked.

Finally, observe that for \( r_o \) the root node of the Kripke model \( K \), the object

\[ Q_{r_o} = \{ < e, p \| p \geq r_o \} = N \times P \]

is weakly terminal: any object is included in it. \( \square \)

**Theorem 3.9.** Let \( K \) be a Kripke Model, formalized in \( N \) as described above, and let \( C_K \) be the category given in Definition 3.7. Then, for each node \( p \), and \( \mathcal{L} \)-sentence \( \varphi \)

\[ p \models \varphi \text{ iff there is an } e : Q_p \rightarrow |\varphi| \text{ in } C_K \]
**Proof:** First observe that \(< e, p > \in |\phi| \iff \Delta x \cdot e : Q_p \to |\phi|\) in \(C_K\). This is almost by definition: if \(< u, r > \in Q_p\), then \(r \geq p\). Thus

\[ (\Delta x \cdot e) \langle u, r \rangle \leq (\Delta x \cdot e) \langle u, r \rangle, r \geq < e, r >, \]

and \(< e, r > \in |\phi|\) if \(< e, p > \in |\phi|\).

Conversely, if \(\Delta x \cdot e : Q_p \to |\phi|\) then, since \(< 0, p > \in Q_p\), we have \(< e, p > \in |\phi|\).

Now, by Corollary 3.6

\[ p \models \neg \phi \iff \exists x (< x, p > \in |\phi|) \]

hence,

\[ p \models \neg \phi \iff \text{there is some } e \text{ such that } e : Q_p \to |\phi| \text{ in } C_K \]

In particular, we have

**Corollary 3.10.** \(\phi\) is true in the model \(K\) iff it is realizable in the category \(C_K\). 

To sum up: a straightforward generalization of the realizability categories one finds in ordinary practice gives a notion of realizability which is complete, and which can be obtained uniformly from a denumerable Kripke model. The interpretation can, of course be no more constructive than the Kripke model, but it will not be of any greater computational complexity.

If the model is “given” without any further information about how it was obtained, the realizability will be somewhat of a hybrid, since it must respect the truth-value structure imposed by the partial order of the original Kripke model. As can be seen in the Kripke models that naturally arise from realizability interpretations (e.g. for syntactic realizabilities, in Lipton [47], or, for Kreisel - Troelstra, and semantic Kleene realizability, in Hyland’s [33], or in [76]), the truth value structure imposed by realizability proper is essentially that of the degree of inhabitation of sets of, or predicates on the realizers. This idea survives in the category \(C_K\) just described in the sense that there is a morphism \(e : Q_p \to Q_p\) in \(C_K\) just in case \(p \geq r\) in the original model \(K\). Thus \(p \geq r\) corresponds to “\(Q_r\) is at least as inhabited as \(Q_p\).”

The realizers just constructed explicitly display more information than the nodes in the Kripke model do, in the sense that if \(< e, p > \in |\phi|\), we not only know that \(p \models \neg \phi\), but \(e\) tells us why. If \(\phi\) is a conjunction, say, each conjunct of which is forced by \(p\), the ordered pair structure of \(e\) will encode this. Of course the atomic information is simply copied from the diagram of \(K\).

If the Kripke structure itself has been constructively presented as is the case for Kripke models generated by Tableaux (as set forth in Fitting’s [21], and in Nerode’s [61]), the nodes in the category \(C_K\) can be replaced by descriptions of the algorithm that generates them. By so doing, one obtains a tableau refutation procedure, which produces “pure” realizability counterexamples. If a constructive metatheory is assumed, i.e. if it is not the case that for every \(p\) and \(\phi\) \(p \models \neg \phi\) or not \(p \models \neg \phi\), then the positive and negative information produced by the tableau are not complements. Separate notions of realizability are required for the predicates \(N(p, \phi)\) and \(F(p, \phi)\) representing negative and positive information about what is forced. We briefly sketch the necessary definitions.

**Realizability for Constructively Presented Kripke Models.**

**Definition 3.11.** The positive and negative satisfaction predicates, \(F(p, \phi)\) and \(N(p, \phi)\) are given by the following inductive definitions. We make use of the “truncated application” of nodes defined in (14) so as to build-in the applicative structure obtained in the preceding section. 

2Because the primary aim of these definitions is to develop realizability-tableaux, the induced applicative structure has been built into the satisfaction predicate. As can be seen from the use of the Skolem functions \(\nu\), the only “nodes” appearing in a tableau development are terms in \(\nu\), Gödel numbers of formulas, and the root node \(r_\nu\). Notice that for ease of notation we have dropped Gödel-number brackets, and “typing-predicates” such as \(D(p)\)
Axioms for \( N \)

We now define Definition 3.12.

\( F(p, \varphi) \defeq ax(p, \varphi) \quad \varphi \text{ atomic} \)

\( F(p, \varphi \land \psi) \defeq F(p, \varphi) \land F(p, \psi) \)

\( F(p, \varphi \lor \psi) \defeq [(w(p, \varphi \lor \psi) = 0 \& F(p, \varphi)) \lor (w(p, \varphi \lor \psi) = 1 \& F(p, \psi))] \)

\( F(p, \varphi \rightarrow \psi) \defeq \forall q[F(q, \varphi) \rightarrow F(pq, \psi)] \)

\( F(p, \exists x\theta(x)) \defeq D(p, w(p, \exists x\theta(x))) \land F(p, \theta(w(p, \exists x\theta(x)))) \)

\( F(p, \forall x\theta(x)) \defeq \forall q\forall z[D(q, z) \rightarrow F(pq, \theta(z))] \)

The negative satisfaction predicate \( N(p, \varphi) \), which can be thought of a negatively signed tableau entry: “\( p \) does not force \( \varphi \)” is given by:

\( N(p, \varphi) \defeq \neg ax(p, \varphi) \quad \varphi \text{ atomic} \)

\( N(p, \varphi \land \psi) \defeq [(w(p, \varphi \land \psi) = 0 \& N(p, \varphi)) \lor (w(p, \varphi \land \psi) = 1 \& N(p, \psi))] \)

\( N(p, \varphi \lor \psi) \defeq N(p, \varphi) \land N(p, \psi) \)

\( N(p, \varphi \rightarrow \psi) \defeq F(\nu(p, \varphi \rightarrow \psi), \varphi) \land N(\nu(p, \varphi \rightarrow \psi), \psi) \)

\( N(p, \exists x\theta(x)) \defeq \forall z[D(p, z) \rightarrow N(p, \theta(z))] \)

\( N(p, \forall x\theta(x)) \defeq D[\nu(p, \forall x\theta(x)), w(p, \forall x\theta(x))] \land N[\nu(p, \forall x\theta(x)), \theta(w(p, \forall x\theta(x)))] \)

where:

1. \( pq \) is the application of nodes: \( p \cdot q = \text{if } p \leq q \text{ then } q \text{ else } * \).
2. \( w(p, \varphi) \) is the witness function, that returns 1 or 0 for left or right component if \( \varphi \) is a disjunction (positive forcing) or a conjunction (negative forcing), and returns an individual \( c \) in \( D(p) \) if \( \varphi \) is existential, and positively forced. The negative forcing of implication and universal quantification require the introduction of a skolem function \( \nu(p, \varphi) \) producing nodes, which we now define.
3. \( \nu(p, \varphi) \) is a node \( q \) above \( p \) providing a witness to the failure of node \( p \) to force \( \varphi \) if it is an implication, or, if \( \varphi \) is \( \forall x\theta(x) \), of the failure of node \( p \) to force \( \theta(w(p, \forall x\theta(x))) \).

Axioms for \( N, P \) and the witnessing functions:

1. \( D(p, w(p, \exists x\theta(x))) \).
2. \( D(\nu(p, \forall x\theta(x)), w(p, \forall x\theta(x))) \).
3. \( p \cdot \nu(p, \varphi \rightarrow \psi) = \nu(p, \varphi \rightarrow \psi) \)
4. \( p \cdot \nu(p, \forall x\theta(x)) = \nu(p, \forall x\theta(x)) \)
5. \( N(p, \varphi) \land F(p, \varphi) \rightarrow \bot \)

Definition 3.12. We now define realizability of the satisfaction predicates of definition (3.11) as follows:
CONSTRUCTIVE KRIPEK SEMANTICS AND REALIZABILITY

Positive forcing:

\[ e \vDash F(p, \varphi) \overset{df}{=} \]
\[ \varphi \text{ atomic } \Rightarrow e_0 = 0 \& F(p, \varphi) \]
\[ \varphi \equiv \theta \& \psi \Rightarrow e_0 = 1 \& e_{10} \vDash F(p, \theta) \& e_{11} \not\vDash F(p, \psi) \]
\[ \varphi \equiv \theta \lor \psi \Rightarrow e_0 = 2 \& (e_{10} = 0 \& e_{11} \not\vDash F(p, \theta)) \lor \]
\[ (e_{10} = 1 \& e_{11} \vDash F(p, \psi)) \]
\[ \varphi \equiv \theta \rightarrow \psi \Rightarrow e_0 = 3 \land (\forall u)(\forall q)[u \vDash F(q, \theta) \rightarrow \]
\[ e_{1q} \downarrow \& (e_{1u}q) \downarrow \& (e_{1u}q) \vDash F(pq, \psi) \]
\[ \varphi \equiv \exists x\theta(x) \Rightarrow e_0 = 4 \& (\forall q)(\forall z)[D(p, z) \rightarrow e_{1q} \downarrow \& (e_{1q}z) \downarrow \& \]
\[ e_{1qz} \not\vDash F(pq, \theta(z))] \]
\[ \varphi \equiv \forall x\theta(x) \Rightarrow e_0 = 5 \& \forall q\forall z[D(p, z) \rightarrow e_{1q} \downarrow \& (e_{1q}z) \downarrow \& \]
\[ e_{1qz} \not\vDash N(pq, \theta(z))] \]

Negative forcing:

\[ e \vDash N(p, \varphi) \overset{df}{=} \]
\[ \varphi \text{ atomic } \Rightarrow e_0 = 0 \& N(p, \varphi) \]
\[ \varphi \equiv \theta \& \psi \Rightarrow e_0 = 1 \& (e_{10} = 0 \& e_{11} \not\vDash N(p, \theta)) \lor \]
\[ (e_{10} = 1 \& e_{11} \vDash N(p, \psi)) \]
\[ \varphi \equiv \theta \lor \psi \Rightarrow e_0 = 2 \& e_{10} \not\vDash N(p, \theta) \& e_{11} \vDash N(p, \psi) \]
\[ \varphi \equiv \theta \rightarrow \psi \Rightarrow e_0 = 3 \& P(e_{10}) \& e_{110} \not\vDash F(e_{10}, \theta) \& e_{111} \vDash N(e_{10}, \psi) \]
\[ \varphi \equiv \exists x\theta(x) \Rightarrow e_0 = 4 \& \forall q\forall z[D(p, z) \rightarrow e_{1q} \downarrow \& (e_{1q}z) \downarrow \& \]
\[ e_{1qz} \not\vDash N(p, \theta(z))] \]
\[ \varphi \equiv \forall x\theta(x) \Rightarrow e_0 = 5 \& P(e_{100}) \& \forall q\forall z[D(p, q) \rightarrow e_{1q} \downarrow \& (e_{1q}z) \downarrow \& \]
\[ e_{1qz} \not\vDash N(p, \theta(z))] \]

A similar proof to theorem 3.3 gives:

**Theorem 3.13.** *N and F are self-realizing*

With these definitions and a slight modification of the construction of the stratified realizability in definition (3.4) it is straightforward to give a realizability version of intuitionistic tableau proof development. This can be seen as a translation of Kripke-model based tableaux of, e.g., Fitting [21] or Nerode [60], along the lines of the preceding section.

From such realizers one is able to give highly effective counterexamples, when a tableau proof fails, and a Curry-Howard style proof term, if the tableau closes off. This constitutes an alternative constructive proof of the completeness theorem for intuitionistic logic. The details of these realizability tableaux are worked out in [49].
References
42. Lambek, J. and P. J. Scott [1986], Introduction to higher order categorical logic, Cambridge Studies in Advanced Mathematics 7, Cambridge.
64. Robinson, E. [1988], “How complete is PER?”, Technical Report 88-229, Queen’s University, Department of Computing and Information Science.
CONSTRUCTIVE Kripke Semantics and Realizability


This work was partially supported by a Sloan dissertation grant and by a fellowship from the Math Sciences Institute, Cornell
Dept. of Computer Science
Cornell University
and
Department of Mathematics
University of Pennsylvania

This paper appeared in Logic from Computer Science, MSRI publications 21, Y. N. Moschovakis, ed., Springer, 1992