Kripke Semantics for Martin-Löf's Extensional Type Theory

Steve Awodey[1] and Florian Rabe[2]

[1] Department of Philosophy, Carnegie Mellon University, Pittsburgh, USA,
awodey@andrew.cmu.edu
[2] School of Engineering and Science, Jacobs University Bremen, Germany,
f.rabe@jacobs-university.de *

Abstract. It is well-known that simple type theory is complete with respect to non-standard models. Completeness for standard models only holds when increasing the class of models, e.g., to cartesian closed categories. Similarly, dependent type theory is complete for locally cartesian closed categories. However, it is usually difficult to establish the coherence of interpretations of dependent type theory, i.e., to show that the interpretations of equal expressions are indeed equal. Several classes of models have been used to remedy this problem.
We contribute to this investigation by giving a semantics that is both coherent and sufficiently general for completeness while remaining relatively easy to compute with. Our models interpret types of Martin-Löf's extensional dependent type theory as sets indexed over posets or, equivalently, as fibrations over posets. This semantics can be seen as a generalization to dependent type theory of the interpretation of intuitionistic first-order logic in Kripke models. This yields a simple coherent model theory with respect to which simple and dependent type theory are sound and complete.

1 Introduction and Related Work

Martin-Löf's extensional type theory, MLTT, is a dependent type theory ([ML84]). The main characteristic is that there are type-valued function symbols that take terms as input and return types as output. This is enriched with further type constructors such as dependent sum and product. The syntax of dependent type theory is significantly more complex than that of simple type theory, because well-formed types and terms and both their equalities must be defined in a single joint induction.

The semantics of MLTT is similarly complicated. In [See84], the connection between MLTT and locally cartesian closed (LCC) categories was first established. LCC categories interpret contexts Γ as objects $[\![\Gamma]\!]$, types in context Γ as objects in the slice category over $[\![\Gamma]\!]$, substitution as pullback, and dependent sum and product as left and right adjoint to pullback. But there is a difficulty,

* The author was partially supported by a fellowship for Ph.D. research of the German Academic Exchange Service.

namely that these three operations are not independent: Substitution of terms into types is associative and commutes with sum and product formation, which is not necessarily the case for the choices of pullbacks and their adjoints. This is known as the coherence or strictness problem and has been studied extensively. In incoherent models, equal types are interpreted as isomorphic, but not necessarily equal objects such as in [Cur89]. In [Car86], coherent models for MLTT were given using categories with attributes. And in [Hof94], a category with attributes is constructed for every LCC category. Several other model classes and their coherence properties have been studied in, e.g., [Str91] and [Jac90,Jac99]. In [Pit00], an overview is given.

These model classes all have in common that they are rather abstract and have a more complicated structure than general LCC categories. It is clearly desirable to have simpler, more concrete models. But it is a hard problem to equip a given LCC category with choices for pullbacks and adjoints that are both natural and coherent. Our motivation is to find a simple concrete class of LCC categories for which such a choice can be made, and which is still general enough to be complete for MLTT.

Mathematically, our main results can be summarized very simply: Using a theorem from topos theory, it can be shown that MLTT is complete with respect to — not necessarily coherent — models in the LCC categories of the form $\mathcal{SET}^P$ for posets P. This is equivalent to using presheaves on posets as models, which are often called Kripke models. They were also studied in [Hof97]. For these rather simple models, a solution to the coherence problem can be given. $\mathcal{SET}$ can be equipped with a coherent choice of pullback functors, and hence the categories $\mathcal{SET}^P$ can be as well. Deviating subtly from the well-known constructions, we can also make coherent choices for the required adjoints to pullback. Finally, rather than working in the various slices $\mathcal{SET}^P/A$, we use the isomorphism $\mathcal{SET}^P/A \cong \mathcal{SET}^{\int_P A}$, where $\int_P A$ is the Grothendieck construction: Thus we can formulate the semantics of dependent types uniformly in terms of the simple categories of indexed sets $\mathcal{SET}^Q$ for various posets Q.

In addition to being easy to work with, this has the virtue of capturing the idea that a dependent type S in context Γ is in some sense a type-valued function on Γ: Our models interpret Γ as a poset $[\![\Gamma]\!]$ and S as an indexed set $[\![\Gamma|S]\!] : [\![\Gamma]\!] \to \mathcal{SET}$. We speak of Kripke models because these models are a natural extension of the well-known Kripke models for intuitionistic first-order logic ([Kri65]). Such models are based on a poset P of worlds, and the universe is given as a P-indexed set (possibly equipped with P-indexed structure). This can be seen as the special case of our semantics when there is only one base type.

In fact, our results are also interesting in the special case of simple type theory ([Chu40]). Contrary to Henkin models [Hen50,MS89], and the models given in [MM91], which like ours use indexed sets on posets, our models are standard: The interpretation $[\![\Gamma|S \to S']\!]$ of the function type is the exponential of $[\![\Gamma|S]\!]$ and $[\![\Gamma|S']\!]$. And contrary to the models in [Fri75,Sim95], our completeness result holds for theories with more than only base types and terms.

A different notion of Kripke-models for dependent type theory is given in [Lip92], which is related to [All87]. There, the MLTT types are translated into predicates in an untyped first-order language. The first-order language is then interpreted in a Kripke-model, i.e., there is one indexed universe of which all types are subsets. Such models correspond roughly to non-standard set-theoretical models.

We briefly review the syntax of MLTT in Sect. 2 and some categorical preliminaries in Sect. 3. Then we derive the coherent functor choices in Sect. 4 and use them to define the interpretation in Sect. 5. We give our main results regarding the interpretation of substitution, soundness, and completeness in Sect. 6. An extended version of this paper is available as [AR09].

2 Syntax

The basic syntax for MLTT expressions is given by the grammar in Fig. 1. The vocabulary of the syntax is declared in signatures and contexts: Signatures Σ declare globally accessible names c for constants of type S and names a for type-valued constants with a list Γ of argument types. Contexts Γ locally declare typed variables x. Substitutions γ translate from a context Γ to Γ' by providing terms in context Γ' for the variables in Γ. Thus, a substitution from Γ to Γ' can be applied to expressions in context Γ and yields expressions in context Γ'. Relative to a signature Σ and a context Γ, there are two syntactical classes: types and typed terms.

The base types are the application $a\ \gamma$ of a type-valued constant to a list of argument terms γ (which we write as a substitution for simplicity). The composed types are the unit type 1, the identity types $Id(s, s')$, the dependent product types $\Sigma_{x:S}\, T$, and the dependent function types $\Pi_{x:S}\, T$. Terms are constants c, variables x, the element $*$ of the unit type, the element $\mathit{refl}(s)$ of the type $Id(s, s)$, pairs $\langle s, s'\rangle$, projections $\pi_1(s)$ and $\pi_2(s)$, lambda abstractions $\lambda_{x:S}\, s$, and function applications $s\ s'$. We do not need equality axioms $s \equiv s'$ because they can be given as constants of type $Id(s, s')$. For simplicity, we omit equality axioms for types.

Signatures	Σ	::=	$\cdot \mid \Sigma, c\!:\!S \mid \Sigma, a\!:\!(\Gamma)\texttt{type}$
Contexts	Γ	::=	$\cdot \mid \Gamma, x\!:\!S$
Substitutions	γ	::=	$\cdot \mid \gamma, x/s$
Types	S	::=	$a\ \gamma \mid 1 \mid Id(s, s') \mid \Sigma_{x:S}\, S' \mid \Pi_{x:S}\, S'$
Terms	s	::=	$c \mid x \mid\ * \mid \mathit{refl}(s) \mid \langle s, s'\rangle \mid \pi_1(s) \mid \pi_2(s) \mid \lambda_{x:S}\, s \mid s\ s'$

Fig. 1. Basic Grammar

The judgments defining well-formed syntax are listed in Fig. 2. The typing rules for these judgments are well-known. Our formulation follows roughly

[See84], including the use of extensional identity types. The latter means that the equality judgment for the terms s and s' holds iff the type $Id(s, s')$ is inhabited. The equality of terms admits conversion rules such as β and η conversion for function terms. The full inference system can be found in [AR09].

Judgment	Intuition
$\vdash \Sigma$ `Sig`	Σ is a well-formed signature
$\vdash_\Sigma \Gamma$ `Ctx`	Γ is a well-formed context over Σ
$\vdash_\Sigma \gamma : \Gamma \to \Gamma'$	γ is a well-formed substitution over Σ from Γ to Γ'
$\Gamma \vdash_\Sigma S$: `type`	S is a well-formed type over Σ and Γ
$\Gamma \vdash_\Sigma S \equiv S'$	types S and S' are equal over Σ and Γ
$\Gamma \vdash_\Sigma s : S$	term s is well-formed with type S over Σ and Γ
$\Gamma \vdash_\Sigma s \equiv s'$	terms s and s' are equal over Σ and Γ

Fig. 2. Judgments

3 Categorical Preliminaries

In this section, we repeat some well-known definitions and results about indexed sets and fibrations over posets (see e.g., [Joh02]). We assume the basic notions of category theory (see, e.g., [Lan98]). We use a set-theoretical pairing function (a, b) and define tuples as left-associatively nested pairs, i.e., $(a_1, a_2, \ldots, a_n)$ abbreviates $(\ldots(a_1, a_2), \ldots, a_n)$.

Definition 1 (Indexed Sets). *$\mathcal{POSET}$ denotes the category of partially ordered sets. We treat posets as categories and write $p \leq p'$ for the uniquely determined morphism $p \to p'$. If P is a poset, $\mathcal{SET}^P$ denotes the category of functors $P \to \mathcal{SET}$ and natural transformations. These functors are also called P-*indexed sets.

We denote the constant P-indexed set that maps each $p \in P$ to $\{\varnothing\}$ by 1_P. It is often convenient to replace an indexed set A over P with a poset formed from the disjoint union of all sets $A(p)$ for $p \in P$. This is a special case of a construction by Mac Lane ([LM92]) usually called the Grothendieck construction.

Definition 2 (Grothendieck Construction). *For an indexed set A over P, we define a poset $\int_P A := \{(p, a) \mid p \in P, a \in A(p)\}$ with*

$$(p, a) \leq (p', a') \qquad \text{iff} \qquad p \leq p' \text{ and } A(p \leq p')(a) = a'.$$

We also write $\int A$ instead of $\int_P A$ if P is clear from the context.

Using the Grothendieck construction, we can work with sets indexed by indexed sets: We write $P|A$ if A is an indexed set over P, and $P|A|B$ if additionally B is an indexed set over $\int_P A$, etc.

Definition 3. *Assume $P|A|B$. We define an indexed set $P|(A \ltimes B)$ by*

$$(A \ltimes B)(p) = \{(a, b) \mid a \in A(p), b \in B(p, a)\}$$

and

$$(A \ltimes B)(p \leq p') : (a, b) \mapsto \big(a', B\big((p, a) \leq (p', a')\big)(b)\big) \qquad \text{for } a' = A(p \leq p')(a).$$

And we define a natural transformation $\pi_B : A \ltimes B \to A$ by

$$(\pi_B)_p : (a, b) \mapsto a.$$

The following definition introduces discrete opfibrations; for brevity, we will refer to them as "fibrations" in the sequel. Using the axiom of choice, these are necessarily split.

Definition 4 (Fibrations). *A* fibration *over a poset P is a functor $f : Q \to P$ with the following property: For all $p' \in P$ and $q \in Q$ such that $f(q) \leq p'$, there is a unique $q' \in Q$ such that $q \leq q'$ and $f(q') = p'$. We call f* normal *iff f is the first projection of $Q = \int_P A$ for some $P|A$.*

For every indexed set A over P, the first projection $\int_P A \to P$ is a (normal) fibration. Conversely, every fibration $f : Q \to P$ defines an indexed set over P by mapping $p \in P$ to its preimage $f^{-1}(p) \subseteq Q$ and $p \leq p'$ to the obvious function. This leads to a well-known equivalence of indexed sets and fibrations over P. If we only consider normal fibrations, we obtain an isomorphism as follows.

Lemma 1. *If we restrict the objects of $\mathcal{POSET}/P$ to be normal fibrations and the morphisms to be (arbitrary) fibrations, we obtain the full subcategory $Fib(P)$ of $\mathcal{POSET}/P$. There are isomorphisms*

$$F(-) : \mathcal{SET}^P \to Fib(P) \qquad \text{and} \qquad I(-) : Fib(P) \to \mathcal{SET}^P.$$

Proof. It is straightforward to show that $Fib(P)$ is a full subcategory.

For $A : P \to \mathcal{SET}$, we define the fibration $F(A) : \int_P A \to P$ by $(p, a) \mapsto p$. And for a natural transformation $\eta : A \to A'$, we define the fibration $F(\eta) : \int_P A \to \int_P A'$ satisfying $F(A) \circ F(\eta) = F(A')$ by $(p, a) \mapsto (p, \eta_p(a))$.

For $f : Q \to P$, we define an indexed set $I(f)$ by $I(f)(p) := \{a \mid f(p, a) = p\}$ and $I(f)(p \leq p') : a \mapsto a'$ where a' is the uniquely determined element such that $(p, a) \leq (p', a') \in Q$. And for a morphism φ between fibrations $f : Q \to P$ and $f' : Q' \to P$, we define a natural transformation $I(\varphi) : I(f) \to I(f')$ by $I(\varphi)_p : a \mapsto a'$ where a' is such that $\varphi(p, a) = (p, a')$.

Then it is easy to compute that I and F are mutually inverse functors. □

Definition 5 (Indexed Elements). *Assume $P|A$. The P-*indexed elements *of A are given by*

$$Elem(A) := \big\{\big(a_p \in A(p)\big)_{p \in P} \mid a_{p'} = A(p \leq p')(a_p) \text{ whenever } p \leq p'\big\}.$$

Then the indexed elements of A are in bijection with the natural transformations $1_P \to A$. For $a \in \mathit{Elem}(A)$, we will write $F(a)$ for the fibration $P \to \int A$ mapping p to (p, a_p).

Example 1. We exemplify the introduced notions by Fig. 3. P is a totally ordered set visualized as a horizontal line with two elements $p_1 \leq p_2 \in P$. For $P|A$, $\int A$ becomes a blob over P. The sets $A(p_i)$ correspond to the vertical lines in $\int A$, and $a_i \in A(p_i)$. The action of $A(p \leq p')$ and the poset structure of $\int A$ are horizontal: If we assume $A(p_1 \leq p_2) : a_1 \mapsto a_2$, then $(p_1, a_1) \leq (p_2, a_2)$ in $\int A$. In general, $A(p_1 \leq p_2)$ need not be injective or surjective. The action of $F(A)$ is vertical: $F(A)$ maps (p_i, a_i) to p_i.

For $P|A|B$, $\int B$ becomes a three-dimensional blob over $\int A$. The sets $B(p_i, a_i)$ correspond to the dotted lines, and $b_i \in B(p_i, a_i)$. The action of $B((p_1, a_1) \leq (p_2, a_2))$ and the poset structure of $\int B$ are horizontal, and $F(B)$ projects $\int B$ to $\int A$.

$\int_P (A \ltimes B)$ is isomorphic to $\int_{\int_P A} B$: Their elements differ only in the bracketing, i.e., $(p_i, (a_i, b_i))$ and $((p_i, a_i), b_i)$, respectively. We have $(a_i, b_i) \in (A \ltimes B)(p_i)$, and $(A \ltimes B)(p \leq p') : (a_1, b_1) \mapsto (a_2, b_2)$. Thus, the sets $(A \ltimes B)(p_i)$ correspond to the two-dimensional gray areas. Up to this isomorphism, the projection $F(A \ltimes B)$ is the composite $F(A) \circ F(B)$.

Indexed elements $a \in \mathit{Elem}(A)$ are families $(a_p)_{p \in P}$ and correspond to horizontal curves through $\int A$ such that $F(a)$ is a section of $F(A)$. Indexed elements of B correspond to two-dimensional vertical areas in $\int B$ (intersecting the dotted lines exactly once), and indexed elements of $A \ltimes B$ correspond to horizontal curves in $\int B$ (intersecting the gray areas exactly once).

We will use Lem. 1 frequently to switch between indexed sets and fibrations, as convenient. In particular, we will use the following two corollaries.

Lemma 2. *Assume $P|A$. Then*

$$\mathit{Elem}(A) \cong \mathit{Hom}_{\mathit{Fib}(P)}(\mathit{id}_P, F(A)) = \{f : P \to \textstyle\int_P A \mid F(A) \circ f = \mathit{id}_P\}.$$

and

$$\mathcal{SET}^P/A \cong \mathcal{SET}^{\int A}$$

Proof. Both claims follow from Lem. 1 by using $\mathit{Elem}(A) = \mathit{Hom}_{\mathcal{SET}^P}(1_P, A)$ as well as $\mathit{Fib}(P)/F(A) \cong \mathit{Fib}(\int_P A)$, respectively. □

Finally, as usual, we say that a category is *locally cartesian closed* (LCC) if it and all of its slice categories are cartesian closed (in particular, it has a terminal object). Then we have the following well-known result (see, e.g., [AR09]) .

Lemma 3. *$\mathcal{SET}^P$ is LCC.*

4 Operations on Indexed Sets

Because $\mathcal{SET}^P$ is LCC, we know that it has pullbacks and that the pullback along a fixed natural transformation has left and right adjoints (see, e.g., [Joh02]). However, these functors are only unique up to isomorphism, and it is non-trivial to pick coherent choices for them.

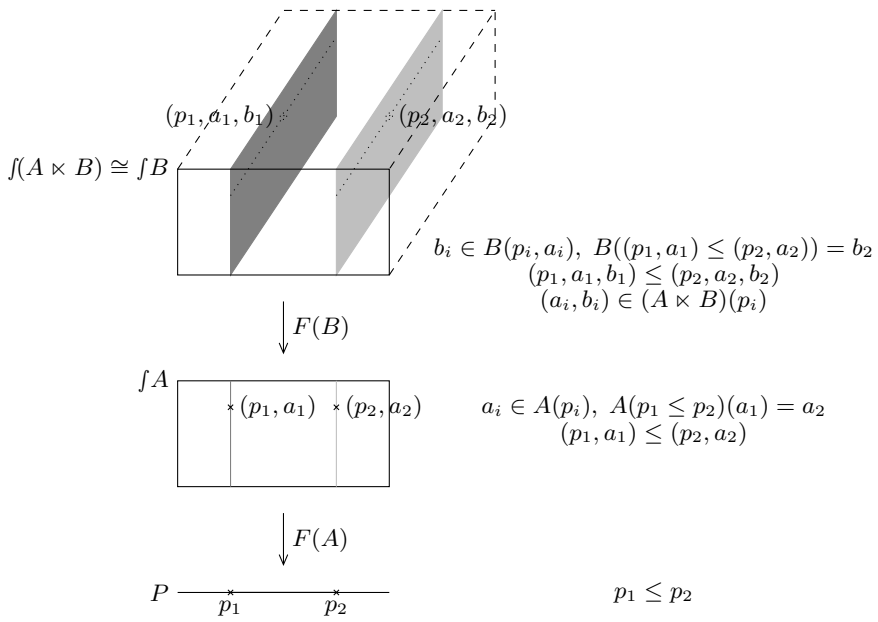


Fig. 3. Indexed Sets and Fibrations

Pullbacks Assume $P|A_1$ and $P|A_2$ and a natural transformation $h : A_2 \to A_1$. The pullback along h is a functor $\mathcal{SET}^P/A_1 \to \mathcal{SET}^P/A_2$. Using Lem. 2, we can avoid dealing with slice categories of $\mathcal{SET}^P$ and instead give a functor

$$h^* : \mathcal{SET}^{\int A_1} \to \mathcal{SET}^{\int A_2},$$

which we also call the pullback along h. The functor h^* is given by precomposition.

Definition 6. *Assume A_1 and A_2 indexed over P, and a natural transformation $h : A_2 \to A_1$. Then for $B \in \mathcal{SET}^{\int A_1}$, we put*

$$h^*B := B \circ F(h) \in \mathcal{SET}^{\int A_2},$$

where, as in Lem. 1, $F(h) : \int_P A_2 \to \int_P A_1$. The action of h^* on morphisms is defined similarly by precomposition with $F(h)$: $h^*(\beta : B \to B') = \beta \circ F(h)$. Finally, we define a natural transformation between P-indexed sets by

$$h \ltimes B : A_2 \ltimes h^*B \to A_1 \ltimes B, \qquad (h \ltimes B)_p : (a_2, b) \mapsto (h_p(a_2), b).$$

The application of $h \ltimes B$ is independent of B, which is only needed in the notation to determine the domain and codomain of $h \ltimes B$.

Lemma 4 (Pullbacks). *In the situation of Def. 6, the following is a pullback in $\mathcal{SET}^P$.*

$$\begin{array}{ccc} A_2 \ltimes h^*B & \xrightarrow{h \ltimes B} & A_1 \ltimes B \\ \downarrow{\scriptstyle \pi_{h^*B}} & & \downarrow{\scriptstyle \pi_B} \\ A_2 & \xrightarrow{h} & A_1 \end{array}$$

Furthermore, we have the following coherence properties for every natural transformation $g : A_3 \to A_2$:

$$id_{A_1}{}^*B = B, \qquad id_{A_1} \ltimes B = id_{A_1 \ltimes B},$$

$$(h \circ g)^*B = g^*(h^*B), \qquad (h \circ g) \ltimes B = (h \ltimes B) \circ (g \ltimes h^*B).$$

Proof. The following is a pullback in $\mathcal{POSET}$:

$$\begin{array}{ccc} \int A_2 \ltimes h^*B & \xrightarrow{F(h \ltimes B)} & \int A_1 \ltimes B \\ \downarrow{\scriptstyle F(\pi_{h^*B})} & & \downarrow{\scriptstyle F(\pi_B)} \\ \int A_2 & \xrightarrow{F(h)} & \int A_1 \end{array} \qquad \begin{array}{ccc} (p,(a_2,b)) & \xmapsto{F(h \ltimes B)} & (p,(h_p(a_2),b)) \\ \downarrow{\scriptstyle F(\pi_{h^*B})} & & \downarrow{\scriptstyle F(\pi_B)} \\ (p,a_2) & \xmapsto{F(h)} & (p,h_p(a_2)) \end{array}$$

If we turn this square into a cocone on P by adding the canonical projections $F(A_2)$ and $F(A_1)$, it becomes a pullback in $Fib(P)$. Then the result follows by Lem. 1. The coherence properties can be verified by simple computations. □

Equivalently, using the terminology of [Pit00], we can say that for every P the tuple

$$(\mathcal{SET}^P, \mathcal{SET}^{\int A}, A \ltimes B, \pi_B, h^*B, h \ltimes B)$$

forms a type category (where A, B, h indicate arbitrary arguments). Then giving coherent adjoints to the pullback means to show that this type category admits dependent sums and products.

Adjoints To interpret MLTT, the adjoints to h^*, where $h : A_2 \to A_1$, are only needed if h is a projection, i.e., $A_1 := A$, $A_2 := A \ltimes B$, and $h := \pi_B$ for some $P|A|B$. We only give adjoint functors for this special case because we use this restriction when defining the right adjoint. Thus, we give functors

$$\mathcal{L}_B, \mathcal{R}_B : \mathcal{SET}^{\int A \ltimes B} \to \mathcal{SET}^{\int A} \text{ such that } \mathcal{L}_B \dashv {\pi_B}^* \dashv \mathcal{R}_B.$$

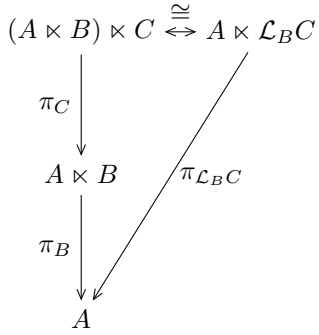
Definition 7. *We define the functor $\mathcal{L}_B$ as follows. For an object C, we put $\mathcal{L}_B C := B \ltimes (C \circ assoc)$ where assoc maps elements $((p,a),b) \in \int B$ to $(p,(a,b)) \in \int A \ltimes B$; and for a morphism, i.e., a natural transformation $\eta : C \to C'$, we put*

$$(\mathcal{L}_B \eta)_{(p,a)} : (b,c) \mapsto (b, \eta_{(p,(a,b))}(c)) \qquad \text{for } (p,a) \in \int A.$$

Lemma 5 (Left Adjoint). *$\mathcal{L}_B$ is left adjoint to ${\pi_B}^*$. Furthermore, for any natural transformation $g : A' \to A$, we have the following coherence property (the Beck-Chevalley condition)*

$$g^*(\mathcal{L}_B C) = \mathcal{L}_{g^* B}(g \ltimes B)^* C.$$

Proof. It is easy to show that $\mathcal{L}_B$ is isomorphic to composition along π_B, for which the adjointness is well-known. In particular, we have the following diagram in $\mathcal{SET}^P$:



The coherence can be verified by direct computation. □

The right adjoint is more complicated. Intuitively, $\mathcal{R}_B C$ must represent the dependent functions from B to C. The naive candidate for this is $Elem(C) \cong Hom(1_{\int B}, C)$ (i.e., $Hom(B, C)$ in the simply-typed case), but this is not a $\int A$-indexed set. There is a well-known construction to remedy this, but we use a subtle modification to achieve coherence, i.e., the analogue of the Beck-Chevalley condition. To do that, we need an auxiliary definition.

Definition 8. *Assume $P|A|B$, $P|A \ltimes B|C$, and an element $x := (p, a) \in \int A$. Let $y^x \in \mathcal{SET}^P$ and a natural transformation $i : y^x \to A$ be given by*

$$y^x(p') = \begin{cases} \{\varnothing\} & \text{if } p \leq p' \\ \varnothing & \text{otherwise} \end{cases} \qquad i_{p'} : \varnothing \mapsto A(p \leq p')(a).$$

Then we define indexed sets $P|y^x|B^x$ and $P|y^x \ltimes B^x|C^x$ by:

$$B^x := i^* B, \qquad C^x := (i \ltimes B)^* C$$

and put $d^x := \int y^x \ltimes B^x$ for the domain of C^x.

The left diagram in Fig. 4 shows the involved P-indexed sets, the right one gives the actions of the natural transformations for an element $p' \in P$ with $p \leq p'$. Below it will be crucial for coherence that B^x and C^x contain tuples in which a' is replaced with $\varnothing$.

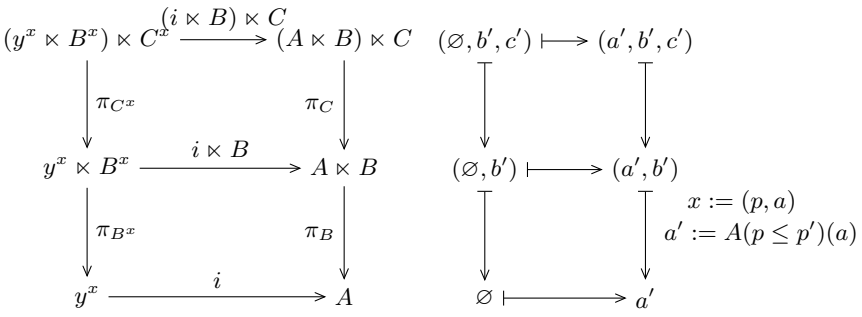


Fig. 4. The Situation of Def. 8

Definition 9. *Assume $P|A|B$. Then we define the functor $\mathcal{R}_B : \mathcal{SET}^{\int A \ltimes B} \to \mathcal{SET}^{\int A}$ as follows. Firstly, for an object C, we put for $x \in \int A$*

$$(\mathcal{R}_B C)(x) := \mathit{Elem}(C^x).$$

In particular, $f \in (\mathcal{R}_B C)(x)$ is a family $(f_y)_{y \in d^x}$ for $f_y \in C^x(y)$. For $x \leq x' \in \int A$, we have $d^x \supseteq d^{x'}$ and put

$$(\mathcal{R}_B C)(x \leq x') : (f_y)_{y \in d^x} \mapsto (f_y)_{y \in d^{x'}}.$$

Secondly, for a morphism, i.e., a natural transformation $\eta : C \to C'$, we define $\mathcal{R}_B \eta : \mathcal{R}_B C \to \mathcal{R}_B C'$ as follows: For $x := (p, a) \in \int A$ and $f \in (\mathcal{R}_B C)(x)$, we define $f' := (\mathcal{R}_B \eta)_x(f) \in (\mathcal{R}_B C')(x)$ by

$$f'_{(p',(\varnothing,b'))} := \eta_{(p',(a',b'))}(f_{(p',(\varnothing,b'))}) \qquad \text{for } (p', (\varnothing, b')) \in d^x \text{ and } a' := A(p \leq p')(a).$$

Lemma 6 (Right Adjoint). *$\mathcal{R}_B$ is right adjoint to ${\pi_B}^*$. Furthermore, for every natural transformation $g : A' \to A$, we have the following coherence property*

$$g^*(\mathcal{R}_B C) = \mathcal{R}_{g^* B}(g \ltimes B)^* C.$$

The proof can be found in [AR09].

The adjointness implies $\mathit{Elem}(\mathcal{R}_B C) \cong \mathit{Elem}(C)$. We spell out this isomorphism explicitly because we will use it later on.

Lemma 7. *Assume $P|A|B$ and $P|A \ltimes B|C$. For $t \in \mathit{Elem}(C)$ and $x := (p, a) \in \int A$, let $t^x \in \mathit{Elem}(C^x)$ be given by*

$$(t^x)_{(p',(\varnothing,b'))} = t_{(p',(a',b'))} \qquad \text{where } a' := A(p \leq p')(a).$$

And for $f \in \mathit{Elem}(\mathcal{R}_B C)$ and $x := (p, (a, b)) \in \int A \ltimes B$, we have $f_{(p,a)} \in \mathit{Elem}(C^x)$; thus, we can put

$$f^x := (f_{(p,a)})_{(p,(\varnothing,b))} \in C(p, (a, b)).$$

Then the sets $Elem(C)$ and $Elem(\mathcal{R}_B C)$ are in bijection via

$$Elem(C) \ni t \overset{sp(-)}{\longmapsto} (t^x)_{x \in \int A} \in Elem(\mathcal{R}_B C)$$

and

$$Elem(\mathcal{R}_B C) \ni f \overset{am(-)}{\longmapsto} (f^x)_{x \in \int A \ltimes B} \in Elem(C)$$

Proof. This follows from the right adjointness by easy computations. □

Intuitively, $sp(t)$ turns $t \in Elem(C)$ into a $\int A$-indexed set by splitting it into components. And $am(f)$ glues such a tuple of components back together. Syntactically, these operations correspond to currying and uncurrying, respectively.

Then we need one last notation. For $P|A$, indexed elements $a \in Elem(A)$ behave like mappings with domain P. We can precompose such indexed elements with fibrations $f : Q \to P$ to obtain Q-indexed elements of $Elem(A \circ f)$.

Definition 10. *Assume $P|A$, $f : Q \to P$, and $a \in Elem(A)$. $a * f \in Elem(A \circ f)$ is defined by: $(a * f)_q := a_{f(q)}$ for $q \in Q$.*

5 Semantics

Using the operations from Sect. 4, the definition of the semantics is straightforward. To demonstrate its simplicity, we will spell it out in an elementary way. The models are Kripke-models, i.e., a Σ-model I is based on a poset P^I of worlds, and provides interpretations $[\![c]\!]^I$ and $[\![a]\!]^I$ for all symbols declared in Σ. I extends to a function $[\![-]\!]^I$, which interprets all Σ-expressions. We will omit the index I if no confusion is possible. The interpretation is such that

- for a context Γ, $[\![\Gamma]\!]$ is a poset,
- for a substitution γ from Γ to Γ', $[\![\gamma]\!]$ is a monotone function from $[\![\Gamma']\!]$ to $[\![\Gamma]\!]$,
- for a type S, $[\![\Gamma|S]\!]$ is an indexed set on $[\![\Gamma]\!]$,
- for a term s of type S, $[\![\Gamma|s]\!]$ is an indexed element of $[\![\Gamma|S]\!]$.

If $\Gamma = x_1 : S_1, \ldots, x_n : S_n$, an element of $[\![\Gamma]\!]$ has the form $(p, (a_1, \ldots, a_n))$ where $p \in P$, $a_1 \in [\![\cdot|S_1]\!](p)$, …, $a_n \in [\![x_1 : S_1, \ldots, x_{n-1} : S_{n-1}|S_n]\!](p, (a_1, \ldots, a_{n-1}))$. Intuitively, a_i is an assignment to the variable x_i in world p. For a typed term $\Gamma \vdash_\Sigma s : S$, both $[\![\Gamma|s]\!]$ and $[\![\Gamma|S]\!]$ are indexed over $[\![\Gamma]\!]$. And if an assignment (p, α) is given, the interpretations of s and S satisfy $[\![\Gamma|s]\!]_{(p,\alpha)} \in [\![\Gamma|S]\!](p, \alpha)$. This is illustrated in the left diagram in Fig. 5.

If γ is a substitution $\Gamma \to \Gamma'$, then $[\![\gamma]\!]$ maps assignments $(p, \alpha') \in [\![\Gamma']\!]$ to assignments $(p, \alpha) \in [\![\Gamma]\!]$. And a substitution in types and terms is interpreted by pullback, i.e., composition. This is illustrated in the right diagram in Fig. 5; its commutativity expresses the coherence.

The poset P of worlds plays the same role as the various posets $[\![\Gamma]\!]$ — it interprets the empty context. In this way, P can be regarded as interpreting an implicit or relative context. This is in keeping with the practice of type theory

(and category theory), according to which closed expressions may be considered relative to some fixed but unspecified context (respectively, base category).

Sum types are interpreted naturally as the dependent sum of indexed sets given by the left adjoint. And pairing and projections have their natural semantics. Product types are interpreted as exponentials using the right adjoint. A lambda abstraction $\lambda_{x:S}\, t$ is interpreted by first interpreting t and then splitting it as in Lem. 7. And an application $f\ s$ is interpreted by amalgamating the interpretation of f as in Lem. 7 and using the composition from Def. 10.

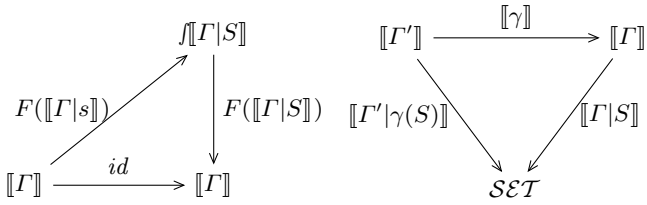


Fig. 5. Semantics of Terms, Types, and Substitution

Definition 11 (Models). *For a signature Σ, Σ-*models *are defined as follows:*

- *A model I for the empty signature $\cdot$ is a poset P^I.*
- *A model I for the signature $\Sigma, c\!:\!S$ consists of a Σ-model I_Σ and an indexed element $[\![c]\!]^I \in \mathit{Elem}([\![\cdot|S]\!]^{I_\Sigma})$.*
- *A model I for the signature $\Sigma, a\!:\!(\Gamma_0)\mathtt{type}$ consists of a Σ-model I_Σ and an indexed set $[\![a]\!]^I$ over $[\![\Gamma_0]\!]^{I_\Sigma}$.*

Definition 12 (Model Extension). *The extension of a model is defined by induction on the typing derivations. We assume in each case that all occurring expressions are well-formed. For example in the case for $[\![\Gamma|f\ s]\!]$, f has type $\Pi_{x:S}\, T$ and s has type S.*

- *Contexts: The context $x_1 : S_1, \ldots, x_n : S_n$ is interpreted as the poset whose elements are the tuples $(p, (a_1, \ldots, a_n))$ such that*

$$\begin{aligned} &p \in P \\ &a_1 \in [\![\cdot|S_1]\!](p, \varnothing) \\ &\vdots \\ &a_n \in [\![x_1\!:\!S_1, \ldots, x_{n-1}\!:\!S_{n-1}|S_n]\!](p, (a_1, \ldots, a_{n-1})) \end{aligned}$$

  The ordering of the poset is inherited from the n-times iterated Groethendieck construction, to which it is canonically isomorphic.
- *Substitutions $\gamma = x_1/s_1, \ldots, x_n/s_n$ from Γ to Γ':*

$$[\![\gamma]\!] : (p, \alpha') \mapsto \big(p, ([\![\Gamma'|s_1]\!]_{(p,\alpha')}, \ldots, [\![\Gamma'|s_n]\!]_{(p,\alpha')})\big) \qquad \text{for } (p, \alpha') \in [\![\Gamma']\!]$$

– *Base types:*

$$[\![\Gamma|a\ \gamma_0]\!] := [\![a]\!] \circ [\![\gamma_0]\!]$$

– *Composed types:*

$$\begin{aligned}
[\![\Gamma|1]\!](p,\alpha) &:= \{\varnothing\} \\
[\![\Gamma|Id(s,s')]\!](p,\alpha) &:= \begin{cases} \{\varnothing\} & \text{if } [\![\Gamma|s]\!]_{(p,\alpha)} = [\![\Gamma|s']\!]_{(p,\alpha)} \\ \varnothing & \text{otherwise} \end{cases} \\
[\![\Gamma|\Sigma_{x:S}\,T]\!] &:= \mathcal{L}_{[\![\Gamma|S]\!]}[\![\Gamma, x:S|T]\!] \\
[\![\Gamma|\Pi_{x:S}\,T]\!] &:= \mathcal{R}_{[\![\Gamma|S]\!]}[\![\Gamma, x:S|T]\!]
\end{aligned}$$

$[\![\Gamma|1]\!]$ *and* $[\![\Gamma|Id(s,s')]\!]$ *are only specified for objects; their extension to morphisms is uniquely determined.*

– *Elementary terms:*

$$[\![\Gamma|c]\!]_{(p,\alpha)} := [\![c]\!]_p, \qquad [\![x_1:S_1,\ldots,x_n:S_n|x_i]\!]_{(p,(a_1,\ldots,a_n))} := a_i$$

– *Composed terms:*

$$\begin{aligned}
[\![\Gamma|*]\!]_{(p,\alpha)} &:= \varnothing \\
[\![\Gamma|\mathit{refl}(s)]\!]_{(p,\alpha)} &:= \varnothing \\
[\![\Gamma|\langle s,s'\rangle]\!]_{(p,\alpha)} &:= ([\![\Gamma|s]\!]_{(p,\alpha)}, [\![\Gamma|s']\!]_{(p,\alpha)}) \\
[\![\Gamma|\pi_i(u)]\!]_{(p,\alpha)} &:= a_i \quad \text{where } [\![\Gamma|u]\!]_{(p,\alpha)} = (a_1,a_2) \\
[\![\Gamma|\lambda_{x:S}\,t]\!] &:= sp([\![\Gamma, x:S|t]\!]) \\
[\![\Gamma|f\ s]\!] &:= am([\![\Gamma|f]\!]) * (\mathit{assoc} \circ F([\![\Gamma|s]\!]))
\end{aligned}$$

Here assoc maps $((p,\alpha),a)$ *to* $(p,(\alpha,a))$.

Since the same expression may have more than one well-formedness derivation, the well-definedness of Def. 12 must be proved in a joint induction with the proof of Thm. 2 below (see also [Str91]). And because of the use of substitution, e.g., for application of function terms, the induction must be intertwined with the proof of Thm. 1 as well.

6 Main Results

Theorem 1 (Substitution). *Assume* $\vdash_\Sigma \gamma : \Gamma \to \Gamma'$. *Then:*

1. *for a substitution* $\vdash_\Sigma \gamma' : \Gamma' \to \Gamma''$: $\quad [\![\gamma' \circ \gamma]\!] = [\![\gamma]\!] \circ [\![\gamma']\!]$,
2. *for a type* $\Gamma \vdash_\Sigma S : \mathtt{type}$: $\quad [\![\Gamma'|\gamma(S)]\!] = [\![\Gamma|S]\!] \circ [\![\gamma]\!]$,
3. *for a term* $\Gamma \vdash_\Sigma s : S$: $\quad [\![\Gamma'|\gamma(s)]\!] = [\![\Gamma|s]\!] * [\![\gamma]\!]$.

Theorem 2 (Soundness). *Assume a signature* Σ, *and a context* Γ. *If* $\Gamma \vdash_\Sigma S \equiv S'$ *for two well-formed types* S, S', *then in every* Σ*-model:*

$$[\![\Gamma|S]\!] = [\![\Gamma|S']\!] \in \mathcal{SET}^{[\![\Gamma]\!]}.$$

And if $\Gamma \vdash_\Sigma s \equiv s'$ *for two well-formed terms* s, s' *of type* S, *then in every* Σ*-model:*

$$[\![\Gamma|s]\!] = [\![\Gamma|s']\!] \in \mathit{Elem}([\![\Gamma|S]\!]).$$

Thm. 1 and 2 must be proved in a joint induction over all expressions and their well-formedness derivations. The proofs can be found in [AR09].

According to the propositions-as-types interpretation — also known as the Curry-Howard correspondence — a type S holds in a model if its interpretation $[\![S]\!]$ is inhabited, i.e., the indexed set $[\![S]\!]$ has an indexed element. A type is valid if it holds in all models. Then soundness implies: If there is a term s of type S in context Γ, then in every Σ-model there is an indexed element of $[\![\Gamma|S]\!]$, namely $[\![\Gamma|s]\!]$. Conversely, we have:

Theorem 3 (Completeness). *Assume a signature Σ, a context Γ, and a well-formed type S. If $[\![\Gamma|S]\!]$ has an indexed element in every model, then there is a term s such that $\Gamma \vdash_\Sigma s : S$.*

The basic idea of the proof is to take the syntactical category, and then to construct a model out of it using topological embedding theorems. It can be found in [AR09].

Due to the presence of extensional identity types, Thm. 3 implies: For all well-formed terms s, s' of type S, if $[\![\Gamma|s]\!] = [\![\Gamma|s']\!]$ in all Σ-models, then $\Gamma \vdash_\Sigma s \equiv s'$. An analogous result for types is more complicated and remains future work.

7 Conclusion and Future Work

We have presented a concrete and intuitive semantics for MLTT in terms of indexed sets on posets. And we have shown soundness and completeness. Our semantics is essentially that proposed by Lawvere in [Law69] in the hyperdoctrine of posets, fibrations, and indexed sets on posets, but we have made particular choices for which the models are coherent. Our models use standard function spaces, and substitution has a very simple interpretation as composition. The same holds in the simply-typed case, which makes our models an interesting alternative to (non-standard) Henkin models. In both cases, we strengthen the existing completeness results by restricting the class of models.

We assume that the completeness result can still be strengthened somewhat further, e.g., to permit equality axioms between types. In addition, it is an open problem to find an elementary completeness proof, i.e., one that does not rely on topos-theoretical results. Going beyond the results presented here, we have developed a first-order logic on top of MLTT and extended the completeness result.

References

[All87] S. Allen. A Non-Type-Theoretic Definition of Martin-Löf's Types. In D. Gries, editor, *Proceedings of the Second Annual IEEE Symp. on Logic in Computer Science, LICS 1987*, pages 215–221. IEEE Computer Society Press, 1987.

[AR09] S. Awodey and F. Rabe. Kripke Semantics for Martin-Löf's Extensional Type Theory. See `http://kwarc.info/frabe/Research/LamKrip.pdf`, 2009.

[Car86] J. Cartmell. Generalized algebraic theories and contextual category. *Annals of Pure and Applied Logic*, 32:209–243, 1986.

[Chu40] A. Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5(1):56–68, 1940.

[Cur89] P. Curien. Alpha-Conversion, Conditions on Variables and Categorical Logic. *Studia Logica*, 48(3):319–360, 1989.

[Fri75] H. Friedman. Equality Between Functionals. In R. Parikh, editor, *Logic Colloquium*, pages 22–37. Springer, 1975.

[Hen50] L. Henkin. Completeness in the Theory of Types. *Journal of Symbolic Logic*, 15(2):81–91, 1950.

[Hof94] M. Hofmann. On the Interpretation of Type Theory in Locally Cartesian Closed Categories. In *CSL*, pages 427–441. Springer, 1994.

[Hof97] M. Hofmann. Syntax and Semantics of Dependent Types. In A. Pitts and P. Dybjer, editors, *Semantics and Logic of Computation*, pages 79–130. Cambridge University Press, 1997.

[Jac90] B. Jacobs. *Categorical Type Theory.* PhD thesis, Catholic University of the Netherlands, 1990.

[Jac99] B. Jacobs. *Categorical Logic and Type Theory.* Elsevier, 1999.

[Joh02] P. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium.* Oxford Science Publications, 2002.

[Kri65] S. Kripke. Semantical Analysis of Intuitionistic Logic I. In J. Crossley and M. Dummett, editors, *Formal Systems and Recursive Functions*, pages 92–130. North-Holland, 1965.

[Lan98] S. Mac Lane. *Categories for the working mathematician.* Springer, 1998.

[Law69] W. Lawvere. Adjointness in Foundations. *Dialectica*, 23(3–4):281–296, 1969.

[Lip92] J. Lipton. Kripke Semantics for Dependent Type Theory and Realizability Interpretations. In J. Myers and M. O'Donnell, editors, *Constructivity in Computer Science, Summer Symposium*, pages 22–32. Springer, 1992.

[LM92] S. Mac Lane and I. Moerdijk. *Sheaves in geometry and logic.* Lecture Notes in Mathematics. Springer, 1992.

[ML84] P. Martin-Löf. *Intuitionistic Type Theory.* Bibliopolis, 1984.

[MM91] J. Mitchell and E. Moggi. Kripke-style Models for Typed Lambda Calculus. *Annals of Pure and Applied Logic*, 51(1–2):99–124, 1991.

[MS89] J. Mitchell and P. Scott. Typed lambda calculus and cartesian closed categories. In *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 301–316. Amer. Math. Society, 1989.

[Pit00] A. Pitts. Categorical Logic. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 5. Algebraic and Logical Structures*, chapter 2, pages 39–128. Oxford University Press, 2000.

[See84] R. Seely. Locally cartesian closed categories and type theory. *Math. Proc. Cambridge Philos. Soc.*, 95:33–48, 1984.

[Sim95] A. Simpson. Categorical completeness results for the simply-typed lambda-calculus. In M. Dezani-Ciancaglini and G. Plotkin, editor, *Typed Lambda Calculi and Applications*, pages 414–427, 1995.

[Str91] T. Streicher. *Semantics of Type Theory.* Springer-Verlag, 1991.