M-Set Models

CHAD E. BROWN[1]

1 Introduction

In [1] Andrews studies elementary type theory, a form of Church's type theory [12] without extensionality, descriptions, choice, and infinity. Since most of the automated search procedures implemented in TPS [4] do not build in principles of extensionality, descriptions, choice or infinity, they are essentially searching for proofs in elementary type theory. In particular, search procedures based on Miller's expansion proofs correspond to proofs in elementary type theory extended with η-conversion. In [9] a model class $\mathfrak{M}_{\beta\eta}$ is defined and proven sound and complete with respect a natural deduction calculus corresponding to elementary type theory with η-conversion. One can add extensionality principles to automated search procedures [8, 10, 11] in order to target smaller (more restricted) model classes (as presented in [9]) which better approximate the class of standard models. Alternatively, one can construct interesting models in $\mathfrak{M}_{\beta\eta}$ which do not satisfy the full extensionality principles. One can then prove results about such models by proving theorems in the weaker logic. Suppose $\mathcal{M} \in \mathfrak{M}_{\beta\eta}$ and we want to know if some property P holds for $\mathcal{M}$. Suppose we can find a proposition $\mathbf{A}$ such that the property P holds if $\mathcal{M} \models \mathbf{A}$. We can conclude P holds if we prove the proposition $\mathbf{A}$ in elementary type theory with η-conversion.

Category theory can provide a Kripke-style semantics of intuitionistic higher-order logic [14, 15]. In particular, categories of presheaves are Cartesian closed (thus providing a semantics for simply-typed λ-calculus) and contain a subobject classifier (thus providing a semantics for intuitionistic logic). Since a one-object category is simply a monoid, a presheaf over a one-object category is simply a set with a monoid action (an M-set) [13, 15]. From these abstract considerations, we know that M-sets (for a fixed monoid M) provide a semantics for simply typed λ-calculus and intuitionistic higher-order logic.

In this article we consider M-sets as a semantics for simply typed λ-calculus and fragments of *classical* higher-order logic. We can start with

[1] This research has been funded by SFB 378 Project DIALOG.

any M-set of interest and use this to interpret a base type of individuals. Function types are interpreted using the presheaf exponent. This will provide a means for interpreting simply typed λ-terms in a way that respects $\beta\eta$-equality. However, the ξ extensionality principle may not hold in general. The type of truth values need not be interpreted as the topos subobject classifier. Instead, the type of truth values can be any M-set $\mathcal{D}_o$ with a function ν from $\mathcal{D}_o$ into a two-element set $\{\mathtt{T}, \mathtt{F}\}$. We will consider two choices for $\mathcal{D}_o$ and ν. Once we have such an M-set model of classical higher-order logic, we could use a classical theorem prover which does not build in extensionality principles (such as TPS) to prove properties of the M-set model. To demonstrate this idea, we use TPS to prove a simple fixed point theorem and then construct an M-set model in which the fixed point theorem is meaningful. In order to appeal to a wide audience, we will exclusively use set-theoretic rather than category-theoretic language.

2 Motivation: A Proof in TPS

The higher-order theorem prover TPS has been under development under the leadership of Peter B. Andrews for several decades [7, 6, 4, 5]. TPS supports both automated proof search and interactive proof construction. The automated search procedures combine mating search with higher-order unification. The search procedures in TPS written before 2003 did not build in extensionality reasoning (except η-conversion). When TPS proves a proposition using one of these search procedures, then the proposition is a theorem of elementary type theory with η.

The logic of TPS is based on simple type theory, as described briefly below. More details are given in other sources [3, 9, 11]. We take the set $\mathcal{T}$ of *simple types* to be the same as in [12]. There are two base types o (of truth values), ι (of individuals), and a type $(\alpha\beta)$ of functions from β to α for all types α and β. The set of well-formed formulas of a type α depend on given sets of variables, parameters and logical constants. Let us fix a set $\mathcal{V}$ of typed variables and a set $\mathcal{P}$ of typed parameters. For each type α, $\mathcal{V}_\alpha$ and $\mathcal{P}_\alpha$ denote the subset of $\mathcal{V}$ and $\mathcal{P}$ of type α (respectively). We assume each $\mathcal{V}_\alpha$ is countably infinite. The logical constants we consider are those in the set $\mathcal{S}_{all}$ defined by

$$\{\top_o, \bot_o, \neg_{oo}, \wedge_{ooo}, \vee_{ooo}, \Rightarrow_{ooo}, \equiv_{ooo}\}$$
$$\cup\{\Pi^\alpha_{o(o\alpha)} | \alpha \in \mathcal{T}\} \cup \{\Sigma^\alpha_{o(o\alpha)} | \alpha \in \mathcal{T}\} \cup \{=^\alpha_{o\alpha\alpha} | \alpha \in \mathcal{T}\}.$$

We will consider a signature $\mathcal{S}$ of typed (logical) constants which may vary throughout the paper. We will always assume $\mathcal{S}$ is a subset of $\mathcal{S}_{all}$. The set of *well-formed formulas* (or *terms*) of type α over a signature $\mathcal{S}$ is denoted by $\mathit{wff}_\alpha(\mathcal{S})$ (or wff_α when the signature $\mathcal{S}$ is clear in context). We define each such family of sets inductively as follows:

- $x_\alpha \in \mathit{wff}_\alpha(\mathcal{S})$ for each variable $x_\alpha \in \mathcal{V}_\alpha$.

- $W_\alpha \in \mathit{wff}_\alpha(\mathcal{S})$ for each parameter $W_\alpha \in \mathcal{P}_\alpha$.

- $c_\alpha \in \mathit{wff}_\alpha(\mathcal{S})$ for each constant $c_\alpha \in \mathcal{S}_\alpha$.

- $[\mathbf{F}_{\alpha\beta}\,\mathbf{B}_\beta] \in \mathit{wff}_\alpha(\mathcal{S})$ for each $\mathbf{F} \in \mathit{wff}_{\alpha\beta}(\mathcal{S})$ and $\mathbf{B} \in \mathit{wff}_\beta(\mathcal{S})$.

- $[\lambda x_\beta \mathbf{A}_\alpha] \in \mathit{wff}_{\alpha\beta}(\mathcal{S})$ for each variable $x_\beta \in \mathcal{V}_\beta$ and $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$.

The set $\mathbf{Free}(\mathbf{A}_\alpha) \subset \mathcal{V}$ of free variables in $\mathbf{A}$ is defined in the usual way. A term $\mathbf{A}_\alpha$ is *closed* if $\mathbf{Free}(\mathbf{A}_\alpha) = \emptyset$. Let $\mathit{cwff}_\alpha(\mathcal{S})$ (or cwff_α) be the set of all closed terms of type α. We use $\mathbf{A}^{\downarrow\beta}$ to refer to the β-normal form of $\mathbf{A}$ and $\mathbf{A}^{\downarrow}$ to refer to the $\beta\eta$-normal form of $\mathbf{A}$.

We now consider a simple example of a proof in Tps. Note that every individual is a fixed point of the identity function $[\lambda x_\iota\, x]$. For every individual i_ι, i is the unique fixed point of the identity function $[\lambda x_\iota\, i]$. If the only functions of type $\iota\iota$ are the constant functions and the identity function, then all such functions will have a fixed point. In fact, we can find a fixed point operator $Y_{\iota(\iota\iota)}$ taking each function $f_{\iota\iota}$ to a fixed point of f. The corresponding theorem can be proven formulated as

$$\forall P_{o(\iota\iota)}[\,P[\lambda\, x_\iota\, x] \wedge \forall\, i_\iota\, P[\lambda\, x\, i] \supset \forall\, f_{\iota\iota}\, P\, f] \supset \exists Y_{\iota(\iota\iota)} \forall\, f[\,f[\,Y\, f] \;=\; Y\, f] \quad (1)$$

This theorem can be proven automatically in Tps in less than a second. Tps also translates the proof into the natural deduction shown in Figure 1. The two nontrivial instantiations are shown in the justifications of lines (2) and (10) in Figure 1. In line (10) the fixed point operator Y is chosen to be the function taking f to $f\,u$ (where u is arbitrary). In order to prove $Y\,f$ is a fixed point, we use the hypothesis to prove $[f\,[f\,u]] = [f\,u]$. The corresponding instantiation for the predicate $P_{o\iota}$ is shown in line (2). Since this instantiation contains a logical symbol (equality at type ι), Tps must use a PRIMSUB (primitive substitution, see [2]) to prove the theorem automatically.

The conclusion of (1) may seem suspicious to many readers. Unless there is only one individual of type ι, there will of course be functions from individuals to individuals which do not have fixed points. On the other hand, the hypothesis of (1) is also very strong. In standard set-theoretic semantics

(1)	$1 \vdash$	$\forall P_{o(\iota\iota)}[P[\lambda x_\iota\, x] \land \forall i_\iota\, P[\lambda x\, i] \supset \forall f_{\iota\iota}\, P\, f]$	Hyp
(2)	$1 \vdash$	$[\lambda f_{\iota\iota}[f[f\, u_\iota] = f\, u]][\lambda x_\iota\, x] \land \forall i_\iota[\lambda f[f[f\, u] = f\, u]][\lambda x\, i]$ $\supset \forall f[\lambda f[f[f\, u] = f\, u]]\, f$	UI: $[\lambda f_{\iota\iota}[f[f\, u_\iota] = f\, u]]$ 1
(3)	$1 \vdash$	$u_\iota = u \land \forall i_\iota[i = i] \supset \forall f_{\iota\iota}[f[f\, u] = f\, u]$	Lambda: 2
(4)	$\vdash$	$u_\iota = u$	Assert REFL=
(5)	$\vdash$	$i_\iota = i$	Assert REFL=
(6)	$\vdash$	$\forall i_\iota[i = i]$	UGen: i_ι 5
(7)	$\vdash$	$u_\iota = u \land \forall i_\iota[i = i]$	RuleP: 4 6
(8)	$1 \vdash$	$\forall f_{\iota\iota}[f[f\, u_\iota] = f\, u]$	MP: 7 3
(9)	$1 \vdash$	$\forall f_{\iota\iota}[f[[\lambda f\, f\, u_\iota]\, f] = [\lambda f\, f\, u]\, f]$	Lambda: 8
(10)	$1 \vdash$	$\exists Y_{\iota(\iota\iota)} \forall f_{\iota\iota}[f[Y\, f] = Y\, f]$	EGen: $[\lambda f_{\iota\iota}\, f\, u_\iota]$ 9
(11)	$\vdash$	$\forall P_{o(\iota\iota)}[P[\lambda x_\iota\, x] \land \forall i_\iota\, P[\lambda x\, i] \supset \forall f_{\iota\iota}\, P\, f]$ $\supset \exists Y_{\iota(\iota\iota)} \forall f[f[Y\, f] = Y\, f]$	Deduct: 10

Figure 1. TPS Natural Deduction Proof of a Fixed Point Theorem

of type theory, both the hypothesis and conclusion of (1) can only be true if there is only one individual. However, TPS has proven (1) as a formal theorem of elementary type theory. Consequently, the theorem will be true in any model of elementary type theory. There are nontrivial models in which (1) is meaningful (in the sense that the hypothesis is valid in the model). In the next sections we will prove the existence of a class of M-set models. We will construct a particular M-set model in which (1) is meaningful in Section 6.

3 Semantics

We now summarize the semantic notions used in the paper. These notions are described in more detail in other sources [9, 11]. There are no new concepts introduced in this section.

An $\mathcal{S}$-model will be a tuple $\langle \mathcal{D}, @, \mathcal{E}, \nu \rangle$ where $\langle \mathcal{D}, @ \rangle$ is an applicative structure, $\mathcal{E}$ is an evaluation function interpreting terms in $\langle \mathcal{D}, @ \rangle$, and ν determines which members of $\mathcal{D}_o$ will be considered "true."

A (*typed*) *applicative structure* is a pair $\langle \mathcal{D}, @ \rangle$ where $\mathcal{D}$ is a typed family of nonempty sets and $@^{\alpha\beta} : \mathcal{D}_{\alpha\beta} \times \mathcal{D}_\beta \to \mathcal{D}_\alpha$ for each function type $(\alpha\beta)$. We write simply $\mathsf{f}@\mathsf{b}$ for $@^{\alpha\beta}(\mathsf{f}, \mathsf{b})$, leaving the types implicit. We call an applicative structure *functional* if for all types $\alpha, \beta \in \mathcal{T}$ and $\mathsf{f}, \mathsf{g} \in \mathcal{D}_{\alpha\beta}$, if $\mathsf{f}@\mathsf{b} = \mathsf{g}@\mathsf{b}$ for all $\mathsf{b} \in \mathcal{D}_\beta$, then $\mathsf{f} = \mathsf{g}$.

Let $\mathcal{D}$ be a typed family of nonempty sets. An *assignment* φ *into* $\mathcal{D}$ is a typed function $\varphi : \mathcal{V} \to \mathcal{D}$. $\varphi, [\mathsf{a}/x]$ denotes the assignment such that $(\varphi, [\mathsf{a}/x])(x) = \mathsf{a}$ and $(\varphi, [\mathsf{a}/x])(y) = \varphi(y)$ for variables y other than x.

Let $\langle \mathcal{D}, @\rangle$ be an applicative structure. An $\mathcal{S}$-*evaluation function* for $\mathcal{A}$ is a function $\mathcal{E}$ taking any assignment φ into $\mathcal{D}$ and term $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$ to $\mathcal{E}_\varphi(\mathbf{A}) \in \mathcal{D}_\alpha$ satisfying the following properties:

1. $\mathcal{E}_\varphi\big|_{\mathcal{V}} = \varphi$
2. $\mathcal{E}_\varphi([\mathbf{F}\,\mathbf{B}]) = \mathcal{E}_\varphi(\mathbf{F})@\mathcal{E}_\varphi(\mathbf{B})$ for any $\mathbf{F}$ in $\mathit{wff}_{\alpha\beta}(\mathcal{S})$ and $\mathbf{B}$ in $\mathit{wff}_\beta(\mathcal{S})$ and types α and β.
3. $\mathcal{E}_\varphi(\mathbf{A}) = \mathcal{E}_\psi(\mathbf{A})$ for any type α and $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$, whenever φ and ψ coincide on $\mathbf{Free}(\mathbf{A})$.
4. $\mathcal{E}_\varphi(\mathbf{A}) = \mathcal{E}_\varphi(\mathbf{A}^{\downarrow\beta})$ for all $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$.

The triple $\mathcal{J} = \langle \mathcal{D}, @, \mathcal{E}\rangle$ is an $\mathcal{S}$-*evaluation* if $\langle \mathcal{D}, @\rangle$ is an applicative structure and $\mathcal{E}$ is an $\mathcal{S}$-evaluation function for $\langle \mathcal{D}, @\rangle$. We say $\mathcal{J}$ is η-*functional* if

$$\mathcal{E}_\varphi(\mathbf{A}) = \mathcal{E}_\varphi(\mathbf{A}^{\downarrow})$$

for any type α, formula $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$, and assignment φ. We say $\mathcal{J}$ is ξ-*functional* if for all types $\alpha, \beta \in \mathcal{T}$, $\mathbf{M}, \mathbf{N} \in \mathit{wff}_\beta(\mathcal{S})$, assignments φ, and variables x_α,

$$\mathcal{E}_\varphi(\lambda x_\alpha \mathbf{M}) = \mathcal{E}_\varphi(\lambda x_\alpha \mathbf{N})$$

whenever $\mathcal{E}_{\varphi,[\mathsf{a}/x]}(\mathbf{M}) = \mathcal{E}_{\varphi,[\mathsf{a}/x]}(\mathbf{N})$ for every $\mathsf{a} \in \mathcal{D}_\alpha$. We say $\mathcal{J}$ is *functional* if the underlying applicative structure is functional. As proven in [9] (for a particular $\mathcal{S}$) and [11] (for a general $\mathcal{S}$), an evaluation is functional iff it is both η-functional and ξ-functional.

For the rest of the paper, we fix two distinct values $\mathtt{T} \neq \mathtt{F}$. Let $\mathcal{A} := \langle \mathcal{D}, @\rangle$ be an applicative structure and $\nu\colon \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$ be a function. For each logical constant c_α and element $\mathsf{a} \in \mathcal{D}_\alpha$, we define properties $\mathfrak{L}_c(\mathsf{a})$ with respect to ν in Table 1. Roughly speaking, the property $\mathfrak{L}_c(\mathsf{a})$ means a behaves like the logical constant c modulo ν.

Let $\mathcal{J} := \langle \mathcal{D}, @, \mathcal{E}\rangle$ be an evaluation. We call a function $\nu\colon \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$ an $\mathcal{S}$-*valuation* for $\mathcal{J}$ if for every logical constant $c \in \mathcal{S}$, $\mathfrak{L}_c(\mathcal{E}(c))$ holds with respect to ν. In such a case, we call $\mathcal{M} := \langle \mathcal{D}, @, \mathcal{E}, \nu\rangle$ an $\mathcal{S}$-*model* (or simply a *model* when $\mathcal{S}$ is clear in context). A model $\mathcal{M} := \langle \mathcal{D}, @, \mathcal{E}, \nu\rangle$ is called *functional* if the applicative structure $\langle \mathcal{D}, @\rangle$ is functional. We say $\mathcal{M}$ is η-*functional* [ξ-*functional*] if the evaluation $\langle \mathcal{D}, @, \mathcal{E}\rangle$ is η-functional [ξ-*functional*]. We define five properties a model $\mathcal{M}$ can satisfy in Table 2. Properties η, ξ, and $\mathfrak{f}$ are forms of functional extensionality. Property $\mathfrak{b}$ is a form of Boolean extensionality. Property $\mathfrak{q}$ is a requirement that the model realize equality at all types.

prop.	where	holds when			for all
$\mathfrak{L}_\top(\mathsf{a})$	$\mathsf{a} \in \mathcal{D}_o$	$\nu(\mathsf{a}) = \mathtt{T}$			
$\mathfrak{L}_\bot(\mathsf{b})$	$\mathsf{b} \in \mathcal{D}_o$	$\nu(\mathsf{b}) = \mathtt{F}$			
$\mathfrak{L}_\neg(\mathsf{n})$	$\mathsf{n} \in \mathcal{D}_{oo}$	$\nu(\mathsf{n}@\mathsf{a}) = \mathtt{T}$	iff	$\nu(\mathsf{a}) = \mathtt{F}$	$\mathsf{a} \in \mathcal{D}_o$
$\mathfrak{L}_\vee(\mathsf{d})$	$\mathsf{d} \in \mathcal{D}_{ooo}$	$\nu(\mathsf{d}@\mathsf{a}@\mathsf{b}) = \mathtt{T}$	iff	$\nu(\mathsf{a}) = \mathtt{T}$ or $\nu(\mathsf{b}) = \mathtt{T}$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$
$\mathfrak{L}_\wedge(\mathsf{c})$	$\mathsf{c} \in \mathcal{D}_{ooo}$	$\nu(\mathsf{c}@\mathsf{a}@\mathsf{b}) = \mathtt{T}$	iff	$\nu(\mathsf{a}) = \mathtt{T}$ and $\nu(\mathsf{b}) = \mathtt{T}$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$
$\mathfrak{L}_\Rightarrow(\mathsf{i})$	$\mathsf{i} \in \mathcal{D}_{ooo}$	$\nu(\mathsf{i}@\mathsf{a}@\mathsf{b}) = \mathtt{T}$	iff	$\nu(\mathsf{a}) = \mathtt{F}$ or $\nu(\mathsf{b}) = \mathtt{T}$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$
$\mathfrak{L}_\equiv(\mathsf{e})$	$\mathsf{e} \in \mathcal{D}_{ooo}$	$\nu(\mathsf{e}@\mathsf{a}@\mathsf{b}) = \mathtt{T}$	iff	$\nu(\mathsf{a}) = \nu(\mathsf{b})$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$
$\mathfrak{L}_{\Pi^\alpha}(\pi)$	$\pi \in \mathcal{D}_{o(o\alpha)}$	$\nu(\pi@\mathsf{f}) = \mathtt{T}$	iff	$\forall \mathsf{a} \in \mathcal{D}_\alpha\ \nu(\mathsf{f}@\mathsf{a}) = \mathtt{T}$	$\mathsf{f} \in \mathcal{D}_{o\alpha}$
$\mathfrak{L}_{\Sigma^\alpha}(\sigma)$	$\sigma \in \mathcal{D}_{o(o\alpha)}$	$\nu(\sigma@\mathsf{f}) = \mathtt{T}$	iff	$\exists \mathsf{a} \in \mathcal{D}_\alpha\ \nu(\mathsf{f}@\mathsf{a}) = \mathtt{T}$	$\mathsf{f} \in \mathcal{D}_{o\alpha}$
$\mathfrak{L}_{=^\alpha}(\mathsf{q})$	$\mathsf{q} \in \mathcal{D}_{o\alpha\alpha}$	$\nu(\mathsf{q}@\mathsf{a}@\mathsf{b}) = \mathtt{T}$	iff	$\mathsf{a} = \mathsf{b}$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha$

Table 1. Logical Properties of $\nu : \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$

$\mathcal{M}$ satisfies property	when
η	$\mathcal{M}$ is η-functional.
ξ	$\mathcal{M}$ is ξ-functional.
$\mathfrak{f}$	$\mathcal{M}$ is functional.
$\mathfrak{b}$	ν is injective.
$\mathfrak{q}$	for all types α there is a $\mathsf{q}^\alpha \in \mathcal{D}_{o\alpha\alpha}$ such that $\mathfrak{L}_{=^\alpha}(\mathsf{q}^\alpha)$ holds.

Table 2. Properties of Models

4 M-set Models

Let M be a monoid with identity e. An M-set is a set A with an *action* giving an element $\mathsf{a}m \in A$ for each $\mathsf{a} \in A$ and $m \in M$ such that

- $(\mathsf{a}m)n = \mathsf{a}(mn)$
- $\mathsf{a}e = \mathsf{a}$

for all $\mathsf{a} \in A$ and $m, n \in M$. Given two M-sets A and B, we can define the M-set exponent $A^{B,M}$ as the set

$$A^{B,M} := \{\mathsf{f} : M \times B \to A | \forall k, m \in M \forall \mathsf{b} \in B. \mathsf{f}(k, \mathsf{b})m = \mathsf{f}(km, \mathsf{b}m)\} \quad (2)$$

with action taking $\mathsf{f} \in A^{B,M}$ and $m \in M$ to the function $\mathsf{f}m : M \times B \to A$ defined by

$$\mathsf{f}m(k, \mathsf{b}) := \mathsf{f}(mk, \mathsf{b}). \quad (3)$$

Note that $\mathsf{f}m$ is in $A^{B,M}$ since for $k, n \in M$ and $\mathsf{b} \in B$

$$\mathsf{f}m(k, \mathsf{b})n = \mathsf{f}(mk, \mathsf{b})n = \mathsf{f}(mkn, \mathsf{b}n) = \mathsf{f}m(kn, \mathsf{b}n).$$

To check that $A^{B,M}$ is an M-set, we must ensure $(\mathsf{f}m)n = \mathsf{f}(mn)$ and $\mathsf{f}e = \mathsf{f}$. Both facts are easily verified:

$$((\mathsf{f}m)n)(k, \mathsf{b}) = (\mathsf{f}m)(nk, \mathsf{b}) = \mathsf{f}(m(nk), \mathsf{b}) = \mathsf{f}((mn)k, \mathsf{b}) = (\mathsf{f}(mn))(k, \mathsf{b})$$

$$(\mathsf{f}e)(k, \mathsf{b}) = \mathsf{f}(ek, \mathsf{b}) = \mathsf{f}(k, \mathsf{b})$$

DEFINITION 1. Let M be a monoid with identity e. An *M-set applicative structure* is a pair $\langle \mathcal{D}, @ \rangle$ where

- $\langle \mathcal{D}, @ \rangle$ is an applicative structure,
- $\mathcal{D}_\alpha$ is an M-set for each type $\alpha \in \mathcal{T}$,
- $\mathcal{D}_{\alpha\beta}$ is the M-set ${\mathcal{D}_\alpha}^{\mathcal{D}_\beta, M}$ for all types $\alpha, \beta \in \mathcal{T}$, and
- $(\mathsf{f}@\mathsf{b}) = \mathsf{f}(e, \mathsf{b})$ for $\mathsf{f} \in \mathcal{D}_{\alpha\beta}$ and $\mathsf{b} \in \mathcal{D}_\beta$ where $\alpha, \beta \in \mathcal{T}$.

We can specify an M-set applicative structure by giving two nonempty M-sets for the two base types.

THEOREM 2. *Let M be a monoid with identity $e \in M$. If A and B are nonempty M-sets, then there is a unique M-set applicative structure $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ such that $\mathcal{D}_\iota = A$ and $\mathcal{D}_o = B$.*

Proof. We define $\mathcal{D}_\alpha$ by induction on α as follows:

- $\mathcal{D}_\iota := A$
- $\mathcal{D}_o := B$
- $\mathcal{D}_{\alpha\beta} := {\mathcal{D}_\alpha}^{\mathcal{D}_\beta, M}$

Let @ be defined by $(\mathsf{f}@\mathsf{b}) := \mathsf{f}(e, \mathsf{b})$ for $\mathsf{f} \in \mathcal{D}_{\alpha\beta}$ and $\mathsf{b} \in \mathcal{D}_\beta$ where $\alpha, \beta \in \mathcal{T}$. We must verify that $\mathcal{A} := \langle \mathcal{D}, @ \rangle$ is an M-set applicative structure with $\mathcal{D}_\iota = A$ and $\mathcal{D}_o = B$. The only nontrivial property to check is that each $\mathcal{D}_\alpha$ is nonempty. We prove this by induction on types. We have assumed $\mathcal{D}_\beta$ is nonempty for base types $\beta \in \{\iota, o\}$. Assume $\mathcal{D}_\alpha$ is nonempty. Choose some $\mathsf{a} \in \mathcal{D}_\alpha$. Let $\mathsf{f} : M \times \mathcal{D}_\beta \to \mathcal{D}_\alpha$ be the function defined by $\mathsf{f}(k, \mathsf{b}) = \mathsf{a}k$. In order to conclude $\mathcal{D}_{\alpha\beta}$ is nonempty, we check $\mathsf{f} \in \mathcal{D}_{\alpha\beta}$. We must check $\mathsf{f}(k, \mathsf{b})m = \mathsf{f}(km, \mathsf{b}m)$ for $k, m \in M$ and $\mathsf{b} \in \mathcal{D}_\beta$. This is easy:

$$\mathsf{f}(k, \mathsf{b})m = (\mathsf{a}k)m = \mathsf{a}(km) = \mathsf{f}(km, \mathsf{b}m).$$

In order to show $\mathcal{A}$ is unique, suppose $\mathcal{A}' = \langle \mathcal{D}', @' \rangle$ is an M-set applicative structure such that $\mathcal{D}'_\iota = A$ and $\mathcal{D}'_o = B$. An easy induction on α proves each M-set $\mathcal{D}_\alpha$ is equal to $\mathcal{D}'_\alpha$. Given this fact, we know @ and @$'$ must coincide as well. ∎

We now define an action on the set of assignments in an obvious way.

DEFINITION 3. Let M be a monoid and $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure. For any assignment $\varphi : \mathcal{V} \to \mathcal{D}$ into $\mathcal{D}$ and $k \in M$, we let $\varphi k : \mathcal{V} \to \mathcal{D}$ denote the assignment given by $\varphi k(x) := \varphi(x)k$ for each variable x.

An evaluation function maps terms to values in an applicative structure. In order to obtain evaluation functions which respect the actions of an M-set applicative structure, we consider an M-indexed family of evaluation functions.

DEFINITION 4. Let M be a monoid and $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure. An *M-set family of $\mathcal{S}$-evaluation functions for $\mathcal{A}$* is a family $(\mathcal{E}^m)_{m \in M}$ of functions satisfying the following properties:

1. $\mathcal{E}^m_\varphi(x) = \varphi(x)$ for $x \in \mathcal{V}$.
2. $\mathcal{E}^m_\varphi([\mathbf{F}_{\alpha\beta}\,\mathbf{B}_\beta]) = \mathcal{E}^m_\varphi(\mathbf{F})@\mathcal{E}^m_\varphi(\mathbf{B})$.
3. $\mathcal{E}^m_\varphi(w) = \mathcal{E}^e_\psi(w)m$ for $w \in \mathcal{P} \cup \mathcal{S}$ and any assignments φ and ψ.

4. $\mathcal{E}^m_\varphi(\lambda x_\beta \mathbf{A}_\alpha) = \mathsf{f} \in \mathcal{D}_{\alpha\beta}$ where f is the function such that

$$\mathsf{f}(k, \mathsf{b}) = \mathcal{E}^{mk}_{\varphi k, [\mathsf{b}/x]}(\mathbf{A}).$$

Note that in Definition 4 we have not actually required each $\mathcal{E}^m$ to be an $\mathcal{S}$-evaluation function for $\mathcal{A}$. The fact that each $\mathcal{E}^m$ is such an $\mathcal{S}$-evaluation function follows from the conditions in Definition 4. The first two conditions in Definition 4 correspond directly to the first two conditions in the definition of evaluation functions. All four conditions in Definition 4 are used (together) to verify the remaining two conditions in the definition of evaluation functions.

THEOREM 5. *Let M be a monoid with identity e, $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure and $(\mathcal{E}^m)_{m \in M}$ be an M-set family of $\mathcal{S}$-evaluation functions for $\mathcal{A}$. For each $m \in M$, $\mathcal{E}^m$ is an η-functional $\mathcal{S}$-evaluation function for $\mathcal{A}$. Furthermore, for any $m, n \in M$, assignment φ, and term $\mathbf{A} \in wff_\alpha(\mathcal{S})$, we have*

$$\mathcal{E}^m_\varphi(\mathbf{A})n = \mathcal{E}^{mn}_{\varphi n}(\mathbf{A}).$$

Proof. See Appendix A. ■

Since such evaluation functions are η-functional, they will be ξ-functional iff the underlying applicative structure is functional. It is not difficult to show that the underlying applicative structure will be functional if M is a group. The more interesting case is when M is not a group. If M is not a group then using the theorems above we can construct evaluations in which η-functionality holds but ξ-functionality fails (see Example 11).

Just as we can specify an M-set applicative structure by giving nonempty M-sets for the two base types, we can specify an M-set family of $\mathcal{S}$-evaluation functions by interpreting the parameters and constants.

THEOREM 6. *Let M be a monoid with identity e, $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure and $\mathcal{I} : (\mathcal{P} \cup \mathcal{S}) \to \mathcal{D}$ be a typed function. There is a unique M-set family of evaluation functions $(\mathcal{E}^{\mathcal{I},m})_{m \in M}$ such that $\mathcal{E}^{\mathcal{I},e}_\varphi(w) = \mathcal{I}(w)$ for all $w \in \mathcal{P} \cup \mathcal{S}$ and assignments φ.*

Proof. See Appendix B. ■

An M-set model will be an M-set applicative structure with an evaluation function which is part of an M-set family of evaluation functions along with a function $\nu : \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$ which respects the interpretations of the logical constants in $\mathcal{S}$.

DEFINITION 7. Let M be a monoid with identity e. An *M-set $\mathcal{S}$-model* (or, *M-set model*) is an $\mathcal{S}$-model $\langle \mathcal{D}, @, \mathcal{E}, \nu \rangle$ where $\langle \mathcal{D}, @ \rangle$ is an M-set applicative structure and there is an M-set family of evaluation functions $(\mathcal{E}^m)_{m \in M}$ such that $\mathcal{E}_\varphi(\mathbf{A}) = \mathcal{E}^e_\varphi(\mathbf{A})$ for all terms $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$ and assignments φ.

We can specify an M-set $\mathcal{S}$-model by giving the M-set applicative structure, valuation ν, and an interpretation of parameters and logical constants which respects the properties of the logical constants.

THEOREM 8. *Let M be a monoid with identity e, $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure, $\mathcal{I} : (\mathcal{P} \cup \mathcal{S}) \to \mathcal{D}$ be a typed function and $\nu : \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$ be a function such that $\mathfrak{L}_c(\mathcal{I}(c))$ holds for all $c \in \mathcal{S}$. Let $(\mathcal{E}^{\mathcal{I},m})_{m \in M}$ be the M-set family of evaluation functions given by Theorem 6. Then $\mathcal{M} := \langle \mathcal{D}, @, \mathcal{E}^{\mathcal{I},e}, \nu \rangle$ is an M-set $\mathcal{S}$-model satisfying property η.*

Proof. This is an obvious consequence of Theorems 5 and 6. ■

5 Interpreting the Type of Truth Values

In order to apply Theorem 8, we must give an interpretation $\mathcal{I}$ for all parameters and logical constants. The interpretation $\mathcal{I}(c)$ of each logical constant $c \in \mathcal{S}$ must satisfy the corresponding logical property $\mathfrak{L}_c(\mathcal{I}(c))$ with respect to ν. Given an applicative structure $\mathcal{A}$ and function ν, it may be the case that no value a satisfies $\mathfrak{L}_c(\mathsf{a})$ with respect to ν. In such a case there are no M-set $\mathcal{S}$-models (where $c \in \mathcal{S}$) over this applicative structure using this valuation ν. If there *is* such an a, we will say $\mathcal{A}$ realizes c with respect to ν.

DEFINITION 9. Let $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an applicative structure, $\nu : \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$ be a function and c_α be a logical constant. We say $\mathcal{A}$ realizes c with respect to ν if there is some $\mathsf{a} \in \mathcal{D}_\alpha$ such that $\mathfrak{L}_c(\mathsf{a})$ holds with respect to ν.

There are several options one can choose for $\mathcal{D}_o$ and $\nu : \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$. This choice affects which logical constants will be realized. If we further want the model to satisfy property $\mathfrak{b}$ (Boolean extensionality), then the following choice is all but forced upon us:

- Set: $\mathcal{D}_o := \{\mathtt{T}, \mathtt{F}\}$
- Action: $\mathtt{T}m = \mathtt{T}$ and $\mathtt{F}m = \mathtt{F}$ (the *trivial action*).
- Valuation: ν is the identity – i.e., $\nu(\mathtt{T}) = \mathtt{T}$ and $\nu(\mathtt{F}) = \mathtt{F}$

In fact, using this simple choice we can realize all logical constants except equality. We can only realize equality in such a model if a certain cancellation law holds.

THEOREM 10. *Let M be a monoid with identity e and $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure. Suppose*

- $\mathcal{D}_o = \{\mathtt{T}, \mathtt{F}\}$,
- $\mathtt{T}m = \mathtt{T}$ *and* $\mathtt{F}m = \mathtt{F}$.

Let $\nu : \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$ be the identity function. Each logical constant in the set

$$\{\top_o, \bot_o, \neg_{oo}, \wedge_{ooo}, \vee_{ooo}, \Rightarrow_{ooo}, \equiv_{ooo}\}$$
$$\cup \{\Pi^\alpha_{o(o\alpha)} | \alpha \in \mathcal{T}\} \cup \{\Sigma^\alpha_{o(o\alpha)} | \alpha \in \mathcal{T}\}$$

is realized in $\mathcal{A}$ with respect to ν. Furthermore, for each $\alpha \in \mathcal{T}$, $=^\alpha$ is realized by $\mathcal{A}$ with respect to ν iff the following cancellation law holds:

$$\forall \mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha \, \forall m \in M \;\; \textit{if } \mathsf{a}m = \mathsf{b}m, \textit{ then } \mathsf{a} = \mathsf{b}.$$

Proof. Obviously $\top$ and $\bot$ are realized using $\mathtt{T}$ and $\mathtt{F}$, respectively. Let $\mathsf{n} : M \times \mathcal{D}_o \to \mathcal{D}_o$ be the function defined by

$$\mathsf{n}(m, \mathsf{b}) := \begin{cases} \mathtt{T} & \text{if } \mathsf{b} = \mathtt{F} \\ \mathtt{F} & \text{otherwise.} \end{cases}$$

We easily verify $\mathsf{n} \in \mathcal{D}_{oo}$:

$$\mathsf{n}(m, \mathsf{b})k = \mathsf{n}(m, \mathsf{b}) = \mathsf{n}(mk, \mathsf{b}) = \mathsf{n}(mk, \mathsf{b}k).$$

It is also clear that $\mathfrak{L}_\neg(\mathsf{n})$ holds with respect to ν:

$$\mathsf{n}@\mathsf{b} = \mathtt{T} \Leftrightarrow \mathsf{n}(e, \mathsf{b}) = \mathtt{T} \Leftrightarrow \mathsf{b} = \mathtt{F}.$$

Let $\mathsf{d} : M \times \mathcal{D}_o \to M \times \mathcal{D}_o \to \mathcal{D}_o$ be

$$\mathsf{d}(m, \mathsf{b})(n, \mathsf{c}) := \begin{cases} \mathtt{F} & \text{if } \mathsf{b} = \mathtt{F} = \mathsf{c} \\ \mathtt{T} & \text{otherwise.} \end{cases}$$

For any $k \in M$, $\mathsf{d}(m, \mathsf{b})(n, \mathsf{c})k = \mathtt{F}$ iff $\mathsf{b} = \mathtt{F} = \mathsf{c}$ iff $\mathsf{b} = \mathtt{F} = \mathsf{c}k$ iff $\mathsf{d}(m, \mathsf{b})(nk, \mathsf{c}k) = \mathtt{F}$. Hence $\mathsf{d}(m, \mathsf{b})(n, \mathsf{c})k = \mathsf{d}(m, \mathsf{b})(nk, \mathsf{c}k)$ and $\mathsf{d}(m, \mathsf{b}) \in \mathcal{D}_{oo}$. Similarly, for any $k \in M$, $\mathsf{d}(m, \mathsf{b})k(n, \mathsf{c}) = \mathtt{F}$ iff $\mathsf{d}(m, \mathsf{b})(kn, \mathsf{c}) = \mathtt{F}$ iff $\mathsf{b} = \mathtt{F} = \mathsf{c}$ iff $\mathsf{b}k = \mathtt{F} = \mathsf{c}$ iff $\mathsf{d}(mk, \mathsf{b}k)(n, \mathsf{c}) = \mathtt{F}$. Hence $\mathsf{d}(m, \mathsf{b})k = \mathsf{d}(mk, \mathsf{b}k)$ and $\mathsf{d} \in \mathcal{D}_{ooo}$. Clearly, $\mathfrak{L}_\vee(\mathsf{d})$ holds.

Since $\neg$ and $\vee$ are realized, we can conclude that $\wedge$, $\Rightarrow$ and $\equiv$ must also be realized. Similarly, to show that each Σ^α is realized, we can simply show each Π^α is realized.

Let $\pi : M \times \mathcal{D}_{o\alpha} \to \mathcal{D}_o$ be defined by

$$\pi(m, \mathsf{f}) := \begin{cases} \mathtt{T} & \text{if } \forall \mathsf{a} \in \mathcal{D}_\alpha\, \mathsf{f}(e, \mathsf{a}m) = \mathtt{T} \\ \mathtt{F} & \text{otherwise.} \end{cases}$$

To check $\pi \in \mathcal{D}_{o(o\alpha)}$, we must prove

$$\pi(m, \mathsf{f})k = \pi(mk, \mathsf{f}k)$$

for $m, k \in M$ and $\mathsf{f} \in \mathcal{D}_{o\alpha}$. Note that

$$\begin{aligned} \pi(m, \mathsf{f})k = \mathtt{T} \quad & \text{iff} \quad \pi(m, \mathsf{f}) = \mathtt{T} \\ & \text{iff} \quad \forall \mathsf{a} \in \mathcal{D}_\alpha\, \mathsf{f}(e, \mathsf{a}m) = \mathtt{T} \\ & \text{iff} \quad \forall \mathsf{a} \in \mathcal{D}_\alpha\, \mathsf{f}(e, \mathsf{a}m)k = \mathtt{T} \\ & \text{iff} \quad \forall \mathsf{a} \in \mathcal{D}_\alpha\, \mathsf{f}(k, \mathsf{a}mk) = \mathtt{T} \\ & \text{iff} \quad \forall \mathsf{a} \in \mathcal{D}_\alpha\, (\mathsf{f}k)(e, \mathsf{a}mk) = \mathtt{T} \\ & \text{iff} \quad \pi(mk, \mathsf{f}k) = \mathtt{T}. \end{aligned}$$

Hence $\pi(m, \mathsf{f})k = \pi(mk, \mathsf{f}k)$ and so $\pi \in \mathcal{D}_{o(o\alpha)}$. Note that $\pi@\mathsf{f} = \mathtt{T}$ iff $\forall \mathsf{a} \in \mathcal{D}_\alpha\, \mathsf{f}@\mathsf{a} = \mathsf{f}(e, \mathsf{a}) = \mathtt{T}$. Thus $\mathfrak{L}_{\Pi^\alpha}(\pi)$ holds.

Finally, we turn our attention to equality. First, suppose there is some $\mathsf{q} \in \mathcal{D}_{o\alpha\alpha}$ realizing $=^\alpha$, i.e., such that

$$\mathsf{q}@\mathsf{a}@\mathsf{b} = \mathtt{T} \Leftrightarrow \mathsf{a} = \mathsf{b}$$

for $\mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha$. Assume there is some $\mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha$ and $m \in M$ such that $\mathsf{a}m = \mathsf{b}m$ but $\mathsf{a} \neq \mathsf{b}$. We can compute $\mathtt{F} = \mathtt{T}$, a contradiction, as follows:

$$\begin{aligned} \mathtt{F} = \mathsf{q}@\mathsf{a}@\mathsf{b} = \mathsf{q}(e, \mathsf{a})(e, \mathsf{b}) = \mathsf{q}(e, \mathsf{a})(e, \mathsf{b})m &= \mathsf{q}(e, \mathsf{a})(m, \mathsf{b}m) \\ &= (\mathsf{q}(e, \mathsf{a})m)(e, \mathsf{b}m) = \mathsf{q}(m, \mathsf{a}m)(e, \mathsf{b}m) \\ &= \mathsf{q}(m, \mathsf{b}m)(e, \mathsf{b}m) = (\mathsf{q}(e, \mathsf{b})m)(e, \mathsf{b}m) \\ = \mathsf{q}(e, \mathsf{b})(m, \mathsf{b}m) = \mathsf{q}(e, \mathsf{b})(e, \mathsf{b})m &= \mathsf{q}(e, \mathsf{b})(e, \mathsf{b}) = \mathtt{T} \end{aligned}$$

Conversely, suppose that the cancellation law holds in $\mathcal{D}_\alpha$. We define $\mathsf{q} : M \times \mathcal{D}_\alpha \to M \times \mathcal{D}_\alpha \to \mathcal{D}_o$ (realizing $=^\alpha$) by

$$\mathsf{q}(m, \mathsf{a})(n, \mathsf{b}) = \begin{cases} \mathtt{T} & \text{if } \exists \mathsf{c} \in \mathcal{D}_\alpha \text{ such that } \mathsf{a} = \mathsf{c}m \text{ and } \mathsf{b} = \mathsf{c}mn \\ \mathtt{F} & \text{otherwise.} \end{cases}$$

Let $k \in M$ be given and suppose $\mathsf{q}(m, \mathsf{a})(n, \mathsf{b})k = \mathtt{T}$. Then $\mathsf{q}(m, \mathsf{a})(n, \mathsf{b}) = \mathtt{T}$ and there is some $\mathsf{c} \in \mathcal{D}_\alpha$ such that $\mathsf{a} = \mathsf{c}m$ and $\mathsf{b} = \mathsf{c}mn$. Since

$\mathsf{b}k = \mathsf{c}mnk$, c also witnesses $\mathsf{q}(m, \mathsf{a})(nk, \mathsf{b}k) = \mathtt{T}$. For the converse, suppose $\mathsf{q}(m, \mathsf{a})(nk, \mathsf{b}k) = \mathtt{T}$. Then there is some $\mathsf{c} \in \mathcal{D}_\alpha$ such that $\mathsf{a} = \mathsf{c}m$ and $\mathsf{b}k = \mathsf{c}mnk$. By the cancellation law, $\mathsf{b} = \mathsf{c}mn$ and so c witnesses $\mathsf{q}(m, \mathsf{a})(n, \mathsf{b}) = \mathtt{T}$. Hence $\mathsf{q}(m, \mathsf{a})(n, \mathsf{b})k = \mathsf{q}(m, \mathsf{a})(nk, \mathsf{b}k)$ and so $\mathsf{q}(m, \mathsf{a}) \in \mathcal{D}_{o\alpha}$. Next we prove $\mathsf{q} \in \mathcal{D}_{o\alpha\alpha}$ by proving $\mathsf{q}(m, \mathsf{a})k = \mathsf{q}(mk, \mathsf{a}k)$. Suppose $\mathsf{q}(m, \mathsf{a})k(n, \mathsf{b}) = \mathtt{T}$. That is, there is some $\mathsf{c} \in \mathcal{D}_\alpha$ such that $\mathsf{a} = \mathsf{c}m$ and $\mathsf{b} = \mathsf{c}mkn$. Since $\mathsf{a}k = \mathsf{c}mk$, this proves $\mathsf{q}(mk, \mathsf{a}k)(n, \mathsf{b}) = \mathtt{T}$. Finally, suppose $\mathsf{q}(mk, \mathsf{a}k)(n, \mathsf{b}) = \mathtt{T}$. Then for some $\mathsf{c} \in \mathcal{D}_\alpha$, $\mathsf{a}k = \mathsf{c}mk$ and $\mathsf{b} = \mathsf{c}mkn$. By the cancellation law, $\mathsf{a} = \mathsf{c}m$ and so $\mathsf{q}(m, \mathsf{a})(kn, \mathsf{b}) = \mathtt{T}$. Hence $\mathsf{q}(m, \mathsf{a})k = \mathsf{q}(mk, \mathsf{a}k)$ and so $\mathsf{q} \in \mathcal{D}_{o\alpha\alpha}$. Note that $\mathfrak{L}_{=^\alpha}(\mathsf{q})$ holds since $\mathsf{q}@\mathsf{a}@\mathsf{b} = \mathtt{T}$ iff $\exists \mathsf{c} \in \mathcal{D}_\alpha$ such that $\mathsf{a} = \mathsf{c} = \mathsf{b}$ iff $\mathsf{a} = \mathsf{b}$. ■

The fact that equality may not be realized by such a model is a problem, but can be overcome. Using Theorem 3.62 from [9] we can obtain a model which realizes equality by taking a quotient by the congruence relation induced by Leibniz equality. A different problem with this choice for $\mathcal{D}_o$ is that the sets $\mathcal{D}_{o\alpha}$ may be too small. In particular, as we show in the next example, the subsets of $\mathcal{D}_\iota$ represented in $\mathcal{D}_{o\iota}$ may be quite sparse.

EXAMPLE 11. Let M_2 be the monoid $\{0, 1\}$ under multiplication where 1 is the identity. Let $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be the M_2-set applicative structure given by Theorem 2 such that $\mathcal{D}_\iota$ is the M_2-set $\{0, 1\}$ with action by multiplication and $\mathcal{D}_o$ is the M_2-set $\{\mathtt{T}, \mathtt{F}\}$ with the trivial action. Let ν be the identity function. Note that the cancellation law does not hold in $\mathcal{D}_\iota$ since $1 \cdot 0 = 0 \cdot 0$ but $1 \neq 0$. By Theorem 10, $=^\iota$ is not realized in $\mathcal{A}$ with respect to ν. Let $\mathcal{S}$ be the set

$$\{\top_o, \bot_o, \neg_{oo}, \wedge_{ooo}, \vee_{ooo}, \Rightarrow_{ooo}, \equiv_{ooo}\}$$
$$\cup \{\Pi^\alpha_{o(o\alpha)} | \alpha \in \mathcal{T}\} \cup \{\Sigma^\alpha_{o(o\alpha)} | \alpha \in \mathcal{T}\}$$

By Theorem 10 each logical constant in $\mathcal{S}$ is realized in $\mathcal{A}$ with respect to ν. Let $\mathcal{I} : (\mathcal{P} \cup \mathcal{S}) \to \mathcal{D}$ be a function such that $\mathfrak{L}_c(\mathcal{I}(c))$ holds with respect to ν for every $c \in \mathcal{S}$. Let $(\mathcal{E}^{\mathcal{I},m})_{m \in M_2}$ be the M_2-set family of evaluation functions given by Theorem 6. By Theorem 8 $\mathcal{M} := \langle \mathcal{D}, @, \mathcal{E}^{\mathcal{I},e}, \nu \rangle$ is an M_2-set $\mathcal{S}$-model satisfying property η. Since ν is the identity function, $\mathcal{M}$ also satisfies property $\mathfrak{b}$. On the other hand, $\mathcal{M}$ does not satisfy property $\mathfrak{f}$ and hence does not satisfy property ξ. Consider the two functions $\mathsf{f}, \mathsf{g} \in \mathcal{D}_{\iota\iota}$ given by $\mathsf{f}(m, \mathsf{a}) := \mathsf{a}$ and $\mathsf{g}(m, \mathsf{a}) := m \cdot \mathsf{a}$. Clearly, $\mathsf{f}(1, \mathsf{a}) = \mathsf{g}(1, \mathsf{a})$ for all $\mathsf{a} \in \{0, 1\}$. However, $\mathsf{f} \neq \mathsf{g}$. Finally, we consider $\mathcal{D}_{o\iota}$. For all $\mathsf{p} \in \mathcal{D}_{o\iota}$, we must have $\mathsf{p}(1, 0) = \mathsf{p}(1, 0)0 = \mathsf{p}(0, 0) = \mathsf{p}(1, 1)0 = \mathsf{p}(1, 1)$. Thus, the only subsets of $\mathcal{D}_\iota$ represented by functions in $\mathcal{D}_{o\iota}$ are $\emptyset$ and $\{0, 1\}$. If we take the quotient of this model as in Theorem 3.62 from [9], then $\mathcal{D}_\iota$ will collapse to be a singleton.

If we are willing to accept models for which property $\mathfrak{b}$ fails, then we have much more flexibility in our choice of $\mathcal{D}_o$. We may want to interpret $\mathcal{D}_o$ to be an M-set so that all logical constants in $\mathcal{S}_{all}$ are realized and so that all subsets of $\mathcal{D}_\alpha$ are represented by a function in $\mathcal{D}_{o\alpha}$. We can obtain such an M-set by taking $\mathcal{D}_o$ to be the power set $\mathcal{P}(M)$ of M and defining an action taking Xm to $\{y \in M | my \in X\}$ for each $X \in \mathcal{P}(M)$ and $m \in M$. One can easily verify $(Xm)n = X(mn)$ and $Xe = X$ so that this is an M-set. We must also choose a function $\nu : \mathcal{P}(M) \to \{\mathtt{T}, \mathtt{F}\}$. A natural option to consider is taking $\mathcal{G}$ to be an ultrafilter on $\mathcal{P}(M)$ and then defining $\nu(X) := \mathtt{T}$ iff $X \in \mathcal{G}$. It turns out that we obtain a model with all the properties we want by taking $\mathcal{G}$ to be the principal ultrafilter with principal element $e \in M$. That is, we define $\nu(X) := \mathtt{T}$ iff $e \in X$.

THEOREM 12. *Let M be a monoid with identity e and $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure. Suppose*

- $\mathcal{D}_o = \mathcal{P}(M)$ *and*
- $Xm = \{y \in M | my \in X\}$.

Let $\nu : \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$ be defined by

$$\nu(X) := \begin{cases} \mathtt{T} & \textit{if } e \in X \\ \mathtt{F} & \textit{otherwise.} \end{cases}$$

Each logical constant in the set

$$\{\top_o, \bot_o, \neg_{oo}, \wedge_{ooo}, \vee_{ooo}, \Rightarrow_{ooo}, \equiv_{ooo}\}$$
$$\cup \{\Pi^\alpha_{o(o\alpha)} | \alpha \in \mathcal{T}\} \cup \{\Sigma^\alpha_{o(o\alpha)} | \alpha \in \mathcal{T}\} \cup \{=^\alpha_{o\alpha\alpha} | \alpha \in \mathcal{T}\}.$$

is realized in $\mathcal{A}$ with respect to ν. Furthermore, for all $S \subseteq \mathcal{D}_\alpha$ there is some $\mathsf{p}_S \in \mathcal{D}_{\alpha \to o}$ such that for all $\mathsf{a} \in \mathcal{D}_\alpha$ we have $\nu(\mathsf{p}_S @ \mathsf{a}) = \mathtt{T}$ iff $\mathsf{a} \in S$.

Proof. The constants $\top$ and $\bot$ can be realized using M and $\emptyset$, respectively.

To realize negation, we define $\mathsf{n} : M \times \mathcal{P}(M) \to \mathcal{P}(M)$ by $\mathsf{n}(m, X) := M \setminus X$. To check $\mathsf{n} \in \mathcal{D}_{oo}$, we must prove $\mathsf{n}(mk, Xk) = \mathsf{n}(m, X)k$ for all $X \in \mathcal{P}(M)$ and $m, k \in M$. This is true since $x \in \mathsf{n}(mk, Xk)$ iff $x \notin Xk$ iff $kx \notin X$ iff $x \in \mathsf{n}(m, X)k$. To prove $\mathfrak{L}_\neg(\mathsf{n})$, note that $\nu(\mathsf{n}@X) = \mathtt{T}$ iff $e \in \mathsf{n}(e, X)$ iff $e \notin X$ iff $\nu(X) = \mathtt{F}$.

Next we turn to disjunction. We define $\mathsf{d} : M \times \mathcal{P}(M) \to M \times \mathcal{P}(M) \to \mathcal{P}(M)$ by $\mathsf{d}(m, X)(n, Y) := (Xn) \cup Y$. We first prove $\mathsf{d}(m, X) \in \mathcal{D}_{oo}$ by proving $\mathsf{d}(m, X)(nk, Yk) = (\mathsf{d}(m, X)(n, Y))k$. This equation holds since $x \in \mathsf{d}(m, X)(nk, Yk)$ iff $x \in Xnk$ or $x \in Yk$ iff $kx \in Xn$ or $kx \in Y$ iff $kx \in \mathsf{d}(m, X)(n, Y)$ iff $x \in (\mathsf{d}(m, X)(n, Y))k$. We next prove $\mathsf{d} \in \mathcal{D}_{ooo}$

by proving $\mathsf{d}(mk, Xk) = \mathsf{d}(m, X)k$, which holds since $\mathsf{d}(mk, Xk)(n, Y) = (Xkn) \cup Y = \mathsf{d}(m, X)(kn, Y)$. To prove $\mathfrak{L}_\vee(\mathsf{d})$, note that $\nu(\mathsf{d}@X@Y) = \mathtt{T}$ iff $e \in X \cup Y$ iff either $\nu(X) = \mathtt{T}$ or $\nu(Y) = \mathtt{T}$. Since $\neg$ and $\vee$ are realized, so are $\wedge$, $\Rightarrow$ and $\equiv$.

Let α be a type. To prove Π^α is realized, let

$$\pi^\alpha(m, \mathsf{f}) := \{x \in M | \forall \mathsf{a} \in \mathcal{D}_\alpha \, e \in \mathsf{f}(x, \mathsf{a})\}$$

define $\pi^\alpha : M \times \mathcal{D}_{o\alpha} \to \mathcal{D}_o$. Now, $x \in \pi^\alpha(mk, \mathsf{f}k)$ iff $\forall \mathsf{a} \in \mathcal{D}_\alpha e \in (\mathsf{f}k)(x, \mathsf{a})$ iff $\forall \mathsf{a} \in \mathcal{D}_\alpha e \in \mathsf{f}(kx, \mathsf{a})$ iff $kx \in \pi^\alpha(m, \mathsf{f})$ iff $x \in \pi^\alpha(m, \mathsf{f})k$. Hence $\pi^\alpha(mk, \mathsf{f}k) = \pi^\alpha(m, \mathsf{f})k$ and so $\pi^\alpha \in \mathcal{D}_{o(o\alpha)}$. Also, $\mathfrak{L}_{\Pi^\alpha}(\pi^\alpha)$ holds since $\nu(\pi^\alpha@\mathsf{f}) = \mathtt{T}$ iff $\forall \mathsf{a} \in \mathcal{D}_\alpha e \in \mathsf{f}(e, \mathsf{a})$ iff $\forall \mathsf{a} \in \mathcal{D}_\alpha \nu(\mathsf{f}@\mathsf{a}) = \mathtt{T}$. Since $\neg$ and Π^α are realized, so is Σ^α.

To prove $=^\alpha$ is realized, let $\mathsf{q}^\alpha(m, \mathsf{a})(n, \mathsf{b}) := \{x \in M | \mathsf{a}nx = \mathsf{b}x\}$ define $\mathsf{q}^\alpha : M \times \mathcal{D}_\alpha \to M \times \mathcal{D}_\alpha \to \mathcal{P}(M)$. We have $x \in \mathsf{q}^\alpha(m, \mathsf{a})(nk, \mathsf{b}k)$ iff $\mathsf{a}nkx = \mathsf{b}kx$ iff $kx \in (\mathsf{q}^\alpha(m, \mathsf{a})(n, \mathsf{b})$ iff $x \in (\mathsf{q}^\alpha(m, \mathsf{a})(n, \mathsf{b}))k$. Hence $\mathsf{q}^\alpha(m, \mathsf{a})(nk, \mathsf{b}k) = (\mathsf{q}^\alpha(m, \mathsf{a})(n, \mathsf{b}))k$ and so $\mathsf{q}^\alpha(m, \mathsf{a}) \in \mathcal{D}_{o\alpha}$. Next, we compute $x \in \mathsf{q}^\alpha(mk, \mathsf{a}k)(n, \mathsf{b})$ iff $\mathsf{a}knx = \mathsf{b}x$ iff $x \in \mathsf{q}^\alpha(m, \mathsf{a})(kn, \mathsf{b})$. Consequently, $\mathsf{q}^\alpha(mk, \mathsf{a}k)(n, \mathsf{b}) = \mathsf{q}^\alpha(m, \mathsf{a})(kn, \mathsf{b})$ and so $\mathsf{q}^\alpha(mk, \mathsf{a}k) = \mathsf{q}^\alpha(m, \mathsf{a})k$. Therefore, $\mathsf{q}^\alpha \in \mathcal{D}_{o\alpha\alpha}$ as desired. The property $\mathfrak{L}_{=^\alpha}(\mathsf{q}^\alpha)$ holds since

$$\nu(\mathsf{q}^\alpha@\mathsf{a}@\mathsf{b}) = \mathtt{T} \text{ iff } e \in \{x \in M | \mathsf{a}ex = \mathsf{b}\} \text{ iff } \mathsf{a} = \mathsf{b}.$$

Finally, we verify that all subsets of $\mathcal{D}_\alpha$ are represented in $\mathcal{D}_{o\alpha}$. Let $S \subseteq \mathcal{D}_\alpha$ be given. We define $\mathsf{p}_S(m, \mathsf{a}) := \{x | \mathsf{a}x \in S\}$. We compute $x \in \mathsf{p}_S(mk, \mathsf{a}k)$ iff $\mathsf{a}kx \in S$ iff $kx \in \mathsf{p}_S(m, \mathsf{a})$ iff $x \in \mathsf{p}_S(m, \mathsf{a})k$. Hence $\mathsf{p}_S \in \mathcal{D}_{o\alpha}$. Note that $\nu(\mathsf{p}_S@\mathsf{a}) = \mathtt{T}$ iff $e \in \mathsf{p}_S(e, \mathsf{a})$ iff $\mathsf{a} \in S$, as desired. ■

We now use Theorem 12 to modify Example 11 and obtain an M-set model in $\mathfrak{M}_{\beta\eta}$. As in Example 11 we take M_2 to be the monoid $\{0, 1\}$ under multiplication where 1 is the identity. For the reasons discussed in Example 11 we will not choose $\mathcal{D}_o$ to be the two element set $\{\mathtt{T}, \mathtt{F}\}$. Instead we let $\mathcal{D}_o$ be $\mathcal{P}(\{0, 1\})$ with action and ν given as in Theorem 12. Using Theorems 2, 12 and 8 we know we can obtain an $\mathcal{S}_{all}$-model by specifying an M_2-set $\mathcal{D}_\iota$ and a value $\mathcal{I}(w_\alpha) \in \mathcal{D}_\alpha$ for each parameter w.

EXAMPLE 13. Let A be a set with an equivalence relation $\sim$. Suppose for each $\sim$-equivalence class we choose a canonical element. Let $C : A \to A$ be the function taking each a to the canonical element $C(a)$. Note that for all $a \in A$, $a \sim C(a)$ and for all $a, b \in A$, $a \sim b$ iff $C(a) = C(b)$. In particular, $C(C(a)) = C(a)$ for all $a \in A$. We can consider A an M_2-set by defining the action $a1 = a$ and $a0 := C(a)$. This is an M_2-set since C is idempotent. Let us take $\mathcal{D}_\iota$ to be this M_2-set and apply Theorem 2 to obtain an M_2-set applicative structure $\langle \mathcal{D}, @ \rangle$.

The equivalence relation $\sim$ on A and canonical form function C extends to all types. For each type α, we can define an idempotent function $C^\alpha : \mathcal{D}_\alpha \to \mathcal{D}_\alpha$ by $C^\alpha(\mathsf{a}) := \mathsf{a}0$. This clearly induces an equivalence relation $\sim^\alpha$ on each $\mathcal{D}_\alpha$ given by $\mathsf{a} \sim^\alpha \mathsf{b}$ iff $C^\alpha(\mathsf{a}) = C^\alpha(\mathsf{b})$. Only the functions $f : \mathcal{D}_\beta \to \mathcal{D}_\alpha$ which respect the equivalence relations are represented in $\mathcal{D}_{\alpha\beta}$. To see this, suppose $g \in \mathcal{D}_{\alpha\beta}$ and $(g@\mathsf{b}) = f(\mathsf{b})$ for all $\mathsf{b} \in \mathcal{D}_\beta$. If $\mathsf{a} \sim^\beta \mathsf{b}$, then

$$f(\mathsf{a})0 = g(1,\mathsf{a})0 = g(0,\mathsf{a}0) = g(0,\mathsf{b}0) = g(1,\mathsf{b})0 = f(\mathsf{b})0$$

and so $f(\mathsf{a}) \sim^\alpha f(\mathsf{b})$. Next, suppose $f : \mathcal{D}_\beta \to \mathcal{D}_\alpha$ is such that $f(\mathsf{a}) \sim^\alpha f(\mathsf{b})$ whenever $\mathsf{a} \sim^\beta \mathsf{b}$. Define $g : M \times \mathcal{D}_\beta \to \mathcal{D}_\alpha$ by $g(m,\mathsf{a}) := f(\mathsf{a})m$. Clearly, $g@\mathsf{a} = g(1,\mathsf{a}) = f(\mathsf{a})$. Since $g(m0,\mathsf{a}0) = f(\mathsf{a}0)0 = f(\mathsf{a})0 = g(m,\mathsf{a})0$ (using the fact that $\mathsf{a}0 \sim^\alpha \mathsf{a}$), we have $g \in \mathcal{D}_{\alpha\beta}$.

In order to apply Theorem 8 we need a typed function $\mathcal{I} : (\mathcal{P} \cup \mathcal{S}_{all}) \to \mathcal{D}$ interpreting parameters and logical constants. We know by Theorem 12 that for all $c \in \mathcal{S}_{all}$ there is some $\mathcal{I}(c)$ such that $\mathfrak{L}_c(\mathcal{I}(c))$ holds. Let $\mathcal{I}(w)$ be chosen arbitrarily for parameters. Using Theorem 8 we can conclude that $\mathcal{M} := \langle \mathcal{D}, @, \mathcal{E}^{\mathcal{I},e}, \nu \rangle$ is an M_2-set model satisfying property η. As in Example 11, property ξ fails in $\mathcal{M}$ since the applicative structure is not functional. To see this, the reader may consider $\mathsf{f}, \mathsf{g} \in \mathcal{D}_{oo}$ given by $\mathsf{f}(m,X) := X$ and $\mathsf{g}(m,X) := Xm$. Since $\mathcal{D}_o$ has four elements, property $\mathfrak{b}$ fails. Since the model realizes equality at all types, $\mathfrak{q}$ holds and so $\mathcal{M} \in \mathfrak{M}_{\beta\eta}$.

We know by Theorem 12 for all types α and sets $S \subseteq \mathcal{D}_\alpha$, there is a function in $\mathcal{D}_{o\alpha}$ representing S. In particular, $\mathcal{D}_{o\iota}$ is rich enough to represent all subsets of A. On the other hand, only the functions from A to A respecting $\sim$ are represented in $\mathcal{D}_{\iota\iota}$.

6 An Example Model

We now construct a concrete M-set model in which the formal theorem (1) from Section 2 is meaningful. Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ be the set of nonnegative integers. Let $M := \mathbb{N}^{\mathbb{N}}$ be the set of all functions from $\mathbb{N}$ into itself and let $e \in M$ be the identity function. For each $m, n \in M$ we define $mn \in M$ to be $(mn)(i) := n(m(i))$ (reverse composition). Clearly M is a monoid under this operation with identity e. We consider $\mathbb{N}$ as an M-set with action taking $a \in \mathbb{N}$ and $m \in M$ to $am := m(a)$.

By Theorem 2 there is an M-set applicative structure $\langle \mathcal{D}, @ \rangle$ such that $\mathcal{D}_\iota = \mathbb{N}$ (with action given above) and $\mathcal{D}_o = \mathcal{P}(M)$ (with action as in Theorem 12). Let $\nu : \mathcal{D}_o \to \{\mathtt{T}, \mathtt{F}\}$ be defined by

$$\nu(X) := \begin{cases} \mathtt{T} & \text{if } e \in X \\ \mathtt{F} & \text{otherwise.} \end{cases}$$

By Theorem 12 every logical constant in $\mathcal{S}_{all}$ is realized in the applicative structure with respect to ν. Let $\mathcal{I} : \mathcal{S}_{all} \cup \mathcal{P} \to \mathcal{D}$ be defined so that $\mathfrak{L}_c(\mathcal{I}(c))$ holds with respect to ν for each $c \in \mathcal{S}_{all}$. By Theorem 6 there is a unique M-set family of evaluation functions $(\mathcal{E}^{\mathcal{I},m})_{m\in M}$ such that $\mathcal{E}^{\mathcal{I},e}_{\varphi}(w) = \mathcal{I}(w)$ for $w \in \mathcal{S}_{all} \cup \mathcal{P}$ and assignments φ. By Theorem 8 $\mathcal{M} := \langle \mathcal{D}, @, \mathcal{E}^{\mathcal{I},e}, \nu\rangle$ is an M-set $\mathcal{S}$-model satisfying property η. Furthermore, since $\mathcal{S}_{all}$ includes equality at each type, property $\mathfrak{q}$ must hold and so $\mathcal{M} \in \mathfrak{M}_{\beta\eta}$.

Since $\mathcal{M}$ is a model of elementary type theory in η, the theorem (1) from Section 2

$$\forall P_{o(\iota\iota)} [\, P[\lambda\, x_\iota\, x] \wedge \forall\, i_\iota\, P[\lambda\, x\, i] \supset \forall\, f_{\iota\iota}\, P\, f] \supset \exists\, Y_{\iota(\iota\iota)} \forall\, f [\, f[\, Y\, f] \,=\, Y\, f]$$

must be true in the model. Consequently, if we can prove

$$\nu(\mathcal{E}^{\mathcal{I},e}(\forall\, P_{o(\iota\iota)}[\, P[\lambda\, x_\iota\, x] \wedge \forall\, i_\iota\, P[\lambda\, x\, i] \supset \forall\, f_{\iota\iota}\, P\, f])) = \mathtt{T} \tag{4}$$

then we can conclude

$$\nu(\exists\, Y_{\iota(\iota\iota)} \forall\, f [\, f[\, Y\, f] \,=\, Y\, f]) = \mathtt{T} \tag{5}$$

In order to interpret (4), fix an arbitrary assignment φ, let

$$\mathtt{I} := \mathcal{E}^{\mathcal{I},e}_{\varphi}(\lambda x_\iota\, x) \in \mathcal{D}_{\iota\iota}$$

and for each $i \in \mathbb{N}$ let

$$\mathtt{K}^i := \mathcal{E}^{\mathcal{I},e}_{\varphi,[i/y]}(\lambda x_\iota\, y) \in \mathcal{D}_{\iota\iota}.$$

We easily compute

$$\mathtt{I}(m,a) = \mathcal{E}^{\mathcal{I},m}_{\varphi m,[a/x]}(x) = a$$

and

$$\mathtt{K}^i(m,a) = \mathcal{E}^{\mathcal{I},m}_{\varphi m,[im/y],[a/x]}(y) = im = m(i).$$

Using the properties of the interpretations of logical constants in $\mathcal{M}$, we have (4) is valid in $\mathcal{M}$ if for all $\Phi \in \mathcal{D}_{o(\iota\iota)}$, $e \in \Phi(e,f)$ whenever $e \in \Phi(e,\mathtt{I})$ and $e \in \Phi(e,\mathtt{K}^i)$ for all $i \in \mathbb{N}$. This is the case if $\mathcal{D}_{\iota\iota} = \{\mathtt{I}\} \cup \{\mathtt{K}^i | i \in \mathbb{N}\}$. To verify this equation, it is enough to show every $f \in \mathcal{D}_{\iota\iota}$ is either $\mathtt{I}$ or $\mathtt{K}^i$ for some i.

Let $S \in M$ be the successor function. For each $m \in M$ and $a \in \mathbb{N}$, let $[a \cdot m] \in M$ be the function such that $[a \cdot m](0) := a$ and $[a \cdot m](i+1) := m(i)$ for $i \in \mathbb{N}$. Clearly $0[a \cdot m] = a$ and $S[a \cdot m] = m$.

Let $f \in \mathcal{D}_{\iota\iota}$ be given. We have

$$f(m,a) = f(S[a \cdot m], 0[a \cdot m]) = f(S,0)[a \cdot m] = [a \cdot m](f(S,0)).$$

If $f(S,0) = 0$, then

$$f(m,a) = [a \cdot m](0) = a$$

for all $a \in \mathbb{N}$ and so $f = \mathtt{I}$. If $f(S,0) = i+1$ for $i \in \mathbb{N}$, then

$$f(m,a) = [a \cdot m](i+1) = m(i)$$

and so $f = \mathtt{K}^i$.

Consequently, (4) is valid in $\mathcal{M}$. Since (1) is also valid in $\mathcal{M}$ (as a theorem of elementary type theory), the conclusion (5) must be valid in $\mathcal{M}$. We conclude the existence of a fixed point operator $Y \in \mathcal{D}_{\iota(\iota\iota)}$ such that $f(e, Y(e,f)) = Y(e,f)$ for all $f \in \mathcal{D}_{\iota\iota}$.

7 Conclusion and Future Work

We have used M-sets to construct models (in the sense of [9]) of fragments of classical higher-order logic. These models always satisfy property η, but may not satisfy property ξ. Property $\mathfrak{b}$ depends on the choice of $\mathcal{D}_o$ and ν. As we have demonstrated, there is a tradeoff between satisfying property $\mathfrak{b}$ and realizing equality (property $\mathfrak{q}$) in M-set models. We have used these abstract results to obtain a model in which $\mathcal{D}_{\iota\iota}$ is sparsely populated and there is a fixed point operator in $\mathcal{D}_{\iota(\iota\iota)}$. In future work we hope to consider more interesting choices for monoids M and M-sets. Such models may provide novel applications of nonextensional higher-order theorem proving.

BIBLIOGRAPHY

[1] Peter B. Andrews. Resolution in Type Theory. *Journal of Symbolic Logic*, 36:414–432, 1971.

[2] Peter B. Andrews. On Connections and Higher-Order Logic. *Journal of Automated Reasoning*, 5:257–291, 1989.

[3] Peter B. Andrews. Classical Type Theory. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume 2, chapter 15, pages 965–1007. Elsevier Science, 2001.

[4] Peter B. Andrews, Matthew Bishop, Sunil Issar, Dan Nesmith, Frank Pfenning, and Hongwei Xi. TPS: A Theorem Proving System for Classical Type Theory. *Journal of Automated Reasoning*, 16:321–353, 1996.

[5] Peter B. Andrews, Chad E. Brown, Frank Pfenning, Matthew Bishop, Sunil Issar, and Hongwei Xi. ETPS: A System to Help Students Write Formal Proofs. *Journal of Automated Reasoning*, 32:75–92, 2004. Available at http://journals.kluweronline.com/article.asp?PIPS=5264938.

[6] Peter B. Andrews, Sunil Issar, Dan Nesmith, and Frank Pfenning. The TPS Theorem Proving System. *Journal of Symbolic Logic*, 57:353–354, 1992. (abstract).

[7] Peter B. Andrews, Dale A. Miller, Eve Longini Cohen, and Frank Pfenning. Automating Higher-Order Logic. In W. W. Bledsoe and D. W. Loveland, editors, *Automated Theorem Proving: After 25 Years*, Contemporary Mathematics series, vol. 29, pages 169–192. American Mathematical Society, 1984. Proceedings of the Special Session on Automatic Theorem Proving, 89th Annual Meeting of the American Mathematical Society, held in Denver, Colorado, January 5-9, 1983.

[8] Christoph Benzmüller. *Equality and Extensionality in Automated Higher-Order Theorem Proving.* PhD thesis, Universität des Saarlandes, 1999.
[9] Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase. Higher Order Semantics and Extensionality. *Journal of Symbolic Logic*, 69:1027–1088, 2004.
[10] Chad E. Brown. *Set Comprehension in Church's Type Theory.* PhD thesis, Department of Mathematical Sciences, Carnegie Mellon University, 2004.
[11] Chad E. Brown. *Automated Reasoning in Higher-Order Logic: Set Comprehension and Extensionality in Church's Type Theory*, volume 10 of *Studies in Logic: Logic and Cognitive Systems.* College Publications, 2007.
[12] Alonzo Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–68, 1940.
[13] Robert Goldblatt. *Topoi: The Categorial Analysis of Logic.* Number 98 in Studies in Logic and the Foundations of Mathematics. North-Holland, 1979. Third edition, 1983.
[14] J. Lambek and P. Scott. *Introduction to Higher Order Categorial Logic.* Cambridge University Press, Cambridge, UK, 1986.
[15] Saunders MacLane and Ieke Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory.* Springer-Verlag, 1992.

A Proof of Theorem 5

Let M be a monoid with identity e, $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure, and $(\mathcal{E}^m)_{m\in M}$ be an M-set family of $\mathcal{S}$-evaluation functions for $\mathcal{A}$. We will prove a series of results allowing us to conclude Theorem 5.

LEMMA 14. *For any term $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$, $m, n \in M$, and assignment φ, we have*

$$\mathcal{E}^m_\varphi(\mathbf{A})n = \mathcal{E}^{mn}_{\varphi n}(\mathbf{A}).$$

Proof. This follows by induction on $\mathbf{A}$. The variable, parameter and constant cases are easy. The application case is verified by computing

$$\begin{array}{rcll} \mathcal{E}^m_\varphi(\mathbf{FB})n & = & (\mathcal{E}^m_\varphi(\mathbf{F})@\mathcal{E}^m_\varphi(\mathbf{B}))n & \text{by Definition 4(2)} \\ & = & \mathcal{E}^m_\varphi(\mathbf{F})(e, \mathcal{E}^m_\varphi(\mathbf{B}))n & \text{by Definition 1(1)} \\ & = & \mathcal{E}^m_\varphi(\mathbf{F})(n, \mathcal{E}^m_\varphi(\mathbf{B})n) & \text{by (2) and Definition 1(1)} \\ & = & (\mathcal{E}^m_\varphi(\mathbf{F})n)(e, \mathcal{E}^m_\varphi(\mathbf{B})n) & \text{by (3)} \\ & = & (\mathcal{E}^{mn}_{\varphi n}(\mathbf{F}))(e, \mathcal{E}^{mn}_{\varphi n}(\mathbf{B})) & \text{by the inductive hypothesis} \\ & = & \mathcal{E}^{mn}_{\varphi n}(\mathbf{F})@\mathcal{E}^{mn}_{\varphi n}(\mathbf{B}) & \text{by Definition 1(1).} \end{array}$$

For the abstraction case, we verify $\mathcal{E}^m_\varphi(\lambda x_\beta \mathbf{C}_\gamma)n = \mathcal{E}^{mn}_{\varphi n}(\lambda x_\beta \mathbf{C}_\gamma)$ by computing

$$\begin{array}{rcll} (\mathcal{E}^m_\varphi(\lambda x\mathbf{C})n)(k, \mathsf{b}) & = & \mathcal{E}^m_\varphi(\lambda x\mathbf{C})(nk, \mathsf{b}) & \text{by (3)} \\ & = & \mathcal{E}^{mnk}_{\varphi nk, [\mathsf{b}/x]}(\mathbf{C}) & \text{by the inductive hypothesis} \\ & = & \mathcal{E}^{mn}_{\varphi n}(\lambda x\mathbf{C})(k, \mathsf{b}) & \text{by Definition 4(4).} \end{array}$$

■

In the next lemma, we combine property (3) of Definition 4 with Lemma 14 to verify the third property of evaluation functions.

LEMMA 15. *For any term* $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$*,* $m \in M$*, and assignments* φ *and* ψ*, if* $\varphi(x) = \psi(x)$ *for all* $x \in \mathbf{Free}(\mathbf{A})$*, then* $\mathcal{E}^m_\varphi(\mathbf{A}) = \mathcal{E}^m_\psi(\mathbf{A})$*.*

Proof. The proof is by an easy induction on $\mathbf{A}$. Note that if $\mathbf{A}$ is a parameter or constant w, then we have $\mathcal{E}^m_\varphi(\mathbf{A}) = \mathcal{E}^e_\psi(\mathbf{A})m = \mathcal{E}^m_\psi(\mathbf{A})$ by using property (3) of Definition 4 twice. ■

We now prove a Substitution-Value Lemma similar to Lemma 3.20 in [9] for evaluations. In this case we must prove the result before we know $\mathcal{E}^m$ is an evaluation function.

LEMMA 16 (Substitution-Value Lemma). *For any term* $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$*,* $\mathbf{B} \in \mathit{wff}_\beta(\mathcal{S})$*,* $x \in \mathcal{V}_\beta$*,* $m \in M$*, and assignment* φ*,*

$$\mathcal{E}^m_{\varphi,[\mathcal{E}^m_\varphi(\mathbf{B})/x]}(\mathbf{A}) = \mathcal{E}^m_\varphi([\mathbf{B}/x]\mathbf{A}).$$

Proof. The proof is another straightforward induction on $\mathbf{A}$. If $\mathbf{A}$ is a parameter or constant w, we use Lemma 15 to conclude

$$\mathcal{E}^m_{\varphi,[\mathcal{E}^m_\varphi(\mathbf{B})/x]}(w) = \mathcal{E}^m_\varphi(w) = \mathcal{E}^m_\varphi([\mathbf{B}/x]w).$$

Suppose $\mathbf{A}$ is $[\lambda y_\delta \mathbf{C}_\gamma]$ where y and x are distinct and $y \notin \mathbf{Free}(\mathbf{B})$. In this case, we have

$$\begin{array}{lll}
 & \mathcal{E}^m_{\varphi,[\mathcal{E}^m_\varphi(\mathbf{B})/x]}([\lambda y_\delta \mathbf{C}])(k,\mathsf{d}) & \\
= & \mathcal{E}^{mk}_{\varphi k,[\mathcal{E}^m_\varphi(\mathbf{B})k/x],[\mathsf{d}/y]}(\mathbf{C}) & \text{by Definition 4(4)} \\
= & \mathcal{E}^{mk}_{\varphi k,[\mathcal{E}^{mk}_{\varphi k}(\mathbf{B})/x],[\mathsf{d}/y]}(\mathbf{C}) & \text{by Lemma 14} \\
= & \mathcal{E}^{mk}_{\varphi k,[\mathsf{d}/y],[\mathcal{E}^{mk}_{\varphi k,[\mathsf{d}/y]}(\mathbf{B})/x]}(\mathbf{C}) & \text{by Lemma 15 since } y \notin \mathbf{Free}(\mathbf{B}) \cup \{x\} \\
= & \mathcal{E}^{mk}_{\varphi k,[\mathsf{d}/y]}([\mathbf{B}/x]\mathbf{C}) & \text{by the inductive hypothesis} \\
= & \mathcal{E}^m_\varphi([\lambda y_\delta [\mathbf{B}/x]\mathbf{C}])(k,\mathsf{d}) & \text{by Definition 4(4)} \\
= & \mathcal{E}^m_\varphi([\mathbf{B}/x][\lambda y_\delta \mathbf{C}])(k,\mathsf{d}). &
\end{array}$$

The remaining cases are left to the reader. ■

We next check that $\mathcal{E}^m$ respects a single, top-level β-reduction.

LEMMA 17. *For any term* $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$, $\mathbf{B} \in \mathit{wff}_\beta(\mathcal{S})$, $x \in \mathcal{V}_\beta$, $m \in M$, *and assignment* φ,

$$\mathcal{E}^m_\varphi([[\lambda x\, \mathbf{A}]\, \mathbf{B}]) = \mathcal{E}^m_\varphi([\mathbf{B}/x]\mathbf{A}).$$

Proof. We compute

$$\mathcal{E}^m_\varphi([[\lambda x\, \mathbf{A}]\, \mathbf{B}]) = \mathcal{E}^m_\varphi([\lambda x\, \mathbf{A}])(e, \mathcal{E}^m_\varphi(\mathbf{B})) = \mathcal{E}^m_{\varphi,[\mathcal{E}^m_\varphi(\mathbf{B})/x]}(\mathbf{A}) = \mathcal{E}^m_\varphi([\mathbf{B}/x]\mathbf{A})$$

using Lemma 16. ■

We also check $\mathcal{E}^m$ respects a single, top-level η-reduction.

LEMMA 18. *For any term* $\mathbf{F} \in \mathit{wff}_{\alpha\beta}(\mathcal{S})$, $x \in \mathcal{V}_\beta$, $m \in M$, *and assignment* φ, *if* $x \notin \mathbf{Free}(\mathbf{F})$, *then*

$$\mathcal{E}^m_\varphi([\lambda x\, [\mathbf{F}\, x]]) = \mathcal{E}^m_\varphi(\mathbf{F}).$$

Proof. We verify this fact by computing

$$\begin{array}{lll}
 & \mathcal{E}^m_\varphi([\lambda x\, [\mathbf{F}\, x]])(k, \mathsf{b}) & \\
= & \mathcal{E}^{mk}_{\varphi k,[\mathsf{b}/x]}([\mathbf{F}\, x]) & \text{by Definition 4(4)} \\
= & \mathcal{E}^{mk}_{\varphi k,[\mathsf{b}/x]}(\mathbf{F})@\mathsf{b} & \text{by Definition 4(1 and 2)} \\
= & \mathcal{E}^{mk}_{\varphi k}(\mathbf{F})@\mathsf{b} & \text{by Lemma 15 since } x \notin \mathbf{Free}(\mathbf{F}) \\
= & \mathcal{E}^{mk}_{\varphi k}(\mathbf{F})(e, \mathsf{b}) & \text{by Definition 1(1)} \\
= & (\mathcal{E}^m_\varphi(\mathbf{F})k)(e, \mathsf{b}) & \text{by Lemma 14} \\
= & \mathcal{E}^m_\varphi(\mathbf{F})(k, \mathsf{b}) & \text{by (3).}
\end{array}$$

■

Using the previous two lemmas, we can prove $\mathcal{E}^m$ respects one step $\beta\eta$-reductions inside a term.

LEMMA 19. *For any terms* $\mathbf{A}, \mathbf{B} \in \mathit{wff}_\alpha(\mathcal{S})$, $m \in M$, *and assignment* φ, *if* $\mathbf{A}$ *$\beta\eta$-reduces to* $\mathbf{B}$ *in one step, then*

$$\mathcal{E}^m_\varphi(\mathbf{A}) = \mathcal{E}^m_\varphi(\mathbf{B}).$$

Proof. This follows by an induction on the position of the redex in $\mathbf{A}$ using Lemmas 17 and 18 for the base cases. ■

Finally, $\mathcal{E}^m$ respects $\beta\eta$-normalization. This establishes both the final property of evaluation functions and η-functionality.

LEMMA 20. *For any term* $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$, $m \in M$, *and assignment* φ, *we have*

$$\mathcal{E}^m_\varphi(\mathbf{A}) = \mathcal{E}^m_\varphi(\mathbf{A}^{\downarrow}).$$

Proof. The proof is by induction on the number of reductions using Lemma 19 at each step. ■

We now have the necessary results to conclude Theorem 5. By Definition 4, we know $\mathcal{E}^m_\varphi(x) = \varphi(x)$ for each variable x and we know $\mathcal{E}^m_\varphi([\mathbf{FB}]) = \mathcal{E}^m_\varphi(\mathbf{F})@\mathcal{E}^m_\varphi(\mathbf{B})$. By Lemma 15, we know $\mathcal{E}^m_\varphi(\mathbf{A}) = \mathcal{E}^m_\psi(\mathbf{A})$ whenever φ and ψ agree on $\mathbf{Free}(\mathbf{A})$. By Lemma 20 twice, we know

$$\mathcal{E}^m_\varphi(\mathbf{A}) = \mathcal{E}^m_\varphi(\mathbf{A}^{\downarrow}) = \mathcal{E}^m_\varphi(\mathbf{A}^{\downarrow\beta})$$

(since the $\beta\eta$-normal form of $\mathbf{A}^{\downarrow\beta}$ is $\mathbf{A}^{\downarrow}$). Hence $\mathcal{E}^m$ is an $\mathcal{S}$-evaluation function. Furthermore, $\mathcal{E}^m$ is η-functional by Lemma 20. Finally, we know

$$\mathcal{E}^m_\varphi(\mathbf{A})n = \mathcal{E}^{mn}_{\varphi n}(\mathbf{A})$$

holds by Lemma 14.

B Proof of Theorem 6

Let M be a monoid with identity e, $\mathcal{A} = \langle \mathcal{D}, @ \rangle$ be an M-set applicative structure and $\mathcal{I} : (\mathcal{P} \cup \mathcal{S}) \to \mathcal{D}$ be a typed function. If $(\mathcal{E}^{\mathcal{I},m})_{m\in M}$ and $(\mathcal{F}^{\mathcal{I},m})_{m\in M}$ are both M-set families of evaluation functions such that $\mathcal{E}^{\mathcal{I},e}(w) = \mathcal{I}(w) = \mathcal{F}^{\mathcal{I},e}(w)$ for all $w \in \mathcal{P} \cup \mathcal{S}$, then an easy induction on $\mathbf{A}$ proves

$$\forall \mathbf{A}, \varphi, m \text{ we have } \mathcal{E}^{\mathcal{I},m}_\varphi(\mathbf{A}) = \mathcal{F}^{\mathcal{I},m}_\varphi(\mathbf{A}).$$

Hence we have uniqueness and must only prove existence of such a family $(\mathcal{E}^{\mathcal{I},m})_{m\in M}$.

The construction of $(\mathcal{E}^{\mathcal{I},m})_{m\in M}$ for the proof of Theorem 6 requires some work. We could try to define $\mathcal{E}^{\mathcal{I},m}_\varphi(\mathbf{A})$ by induction on $\mathbf{A}$. Unfortunately, for the λ-abstraction case we must somehow know the function f where

$$\mathsf{f}(k, \mathsf{b}) = \mathcal{E}^{\mathcal{I},mk}_{\varphi k,[\mathsf{b}/x]}(\mathbf{A})$$

is in $\mathcal{D}_{\alpha\beta}$. Instead of defining $\mathcal{E}_\varphi^{\mathcal{I},m}(\mathbf{A}_\alpha) \in \mathcal{D}_\alpha$ by induction, we define set valued function $\mathbf{Eval}^{\mathcal{I}}(\mathbf{A}_\alpha, m, \varphi) \subseteq \mathcal{D}_\alpha$ by induction, prove each value of $\mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, m, \varphi)$ is a singleton, and define $\mathcal{E}_\varphi^{\mathcal{I},m}(\mathbf{A}_\alpha)$ to be the unique value in this singleton.

We define $\mathbf{Eval}^{\mathcal{I}}(\mathbf{A}_\alpha, m, \varphi) \subseteq \mathcal{D}_\alpha$ by induction as follows:

- $\mathbf{Eval}^{\mathcal{I}}(x_\alpha, m, \varphi) := \{\varphi(x)\}$ for $x \in \mathcal{V}$.
- $\mathbf{Eval}^{\mathcal{I}}(w_\alpha, m, \varphi) := \{\mathcal{I}(w)m\}$ for $w \in \mathcal{P} \cup \mathcal{S}$.
- $\mathbf{Eval}^{\mathcal{I}}([\mathbf{F}_{\alpha\beta}\,\mathbf{B}_\beta], m, \varphi) :=$
$$\{\mathsf{f}@\mathsf{b} | \mathsf{f} \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{F}, m, \varphi), \mathsf{b} \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{B}, m, \varphi)\}$$
for $\mathbf{F} \in \mathit{wff}_{\alpha\beta}(\mathcal{S})$ and $\mathbf{B} \in \mathit{wff}_\beta(\mathcal{S})$
- $\mathbf{Eval}^{\mathcal{I}}([\lambda x_\beta\,\mathbf{A}_\alpha], m, \varphi) :=$
$$\{\mathsf{f} \in \mathcal{D}_{\alpha\beta} | \forall k \in M, \mathsf{b} \in \mathcal{D}_\beta\, \mathsf{f}(k, \mathsf{b}) \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, mk, (\varphi k, [\mathsf{b}/x]))\}$$
for $x \in \mathcal{V}_\beta$ and $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$.

LEMMA 21. *For every $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$, assignment φ, $m, n \in M$ and $\mathsf{a} \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, m, \varphi)$, we have $\mathsf{a}n \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, mn, \varphi n)$.*

Proof. This follows by an easy induction on $\mathbf{A}$ using the definition of
$$\mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, m, \varphi).$$
If $\mathbf{A}$ is a variable x, then $\mathsf{a} = \varphi(x)$ and so
$$\mathsf{a}n = \varphi(x)n = (\varphi n)(x) \in \mathbf{Eval}^{\mathcal{I}}(x, mn, \varphi n).$$
If $\mathbf{A}$ is a parameter or constant w, then $\mathsf{a} = \mathcal{I}(w)m$ and so
$$\mathsf{a}n = \mathcal{I}(w)mn \in \mathbf{Eval}^{\mathcal{I}}(w, mn, \varphi n).$$
Suppose $\mathbf{A}$ is $[\mathbf{F}_{\alpha\beta}\mathbf{B}_\beta]$. Then a is $\mathsf{f}@\mathsf{b}$ for some $\mathsf{f} \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{F}, m, \varphi)$ and $\mathsf{b} \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{B}, m, \varphi)$. By the inductive hypothesis, $\mathsf{f}n \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{F}, mn, \varphi n)$ and $\mathsf{b}n \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{B}, mn, \varphi n)$. We compute
$$\mathsf{a}n = (\mathsf{f}@\mathsf{b})n = \mathsf{f}(e, \mathsf{b})n = \mathsf{f}(n, \mathsf{b}n) = (\mathsf{f}n)(e, \mathsf{b}n) = ((\mathsf{f}n)@(\mathsf{b}n))$$
and conclude $\mathsf{a}n \in \mathbf{Eval}^{\mathcal{I}}([\mathbf{F}\,\mathbf{B}], mn, \varphi n)$.

Suppose $\mathbf{A}$ is $[\lambda x_\beta \mathbf{C}_\gamma]$. Then $\mathsf{a} \in \mathcal{D}_{\gamma\beta}$ is a function such that $\mathsf{a}(k, \mathsf{b}) \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{C}, mk, (\varphi k, [\mathsf{b}/x]))$ for all $k \in M$ and $\mathsf{b} \in \mathcal{D}_\beta$. Note that $\mathsf{a}n(k, \mathsf{b}) = \mathsf{a}(nk, \mathsf{b})$. Hence $\mathsf{a}n(k, \mathsf{b}) \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{C}, mnk, (\varphi nk, [\mathsf{b}/x]))$. Thus
$$\mathsf{a}n \in \mathbf{Eval}^{\mathcal{I}}([\lambda x\,\mathbf{C}], mn, \varphi n)$$
as desired. ∎

LEMMA 22. *For every* $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$*, assignment* φ*, and* $m \in M$*, there is a unique* $\mathsf{a} \in \mathcal{D}_\alpha$ *such that* $\mathsf{a} \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, m, \varphi)$*.*

Proof. This also follows by induction on $\mathbf{A}$. The cases for variables, parameters, logical constants and applications are easy. We only show the λ-abstraction case. Suppose $\mathbf{A}$ is $[\lambda x_\beta \mathbf{C}_\gamma]$. By the inductive hypothesis, for all $k \in M$ and $\mathsf{b} \in \mathcal{D}_\beta$, there is a unique value in $\mathbf{Eval}^{\mathcal{I}}(\mathbf{C}, mk, (\varphi k, [\mathsf{b}/x]))$. Let $\mathsf{f} : M \times \mathcal{D}_\beta \to \mathcal{D}_\gamma$ be the function taking (k, b) to this unique value. In order to conclude $\mathsf{f} \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, m, \varphi)$, we must verify that $\mathsf{f} \in \mathcal{D}_{\gamma\beta}$. On the one hand, $\mathsf{f}(kn, \mathsf{b}n) \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{C}, mkn, (\varphi kn, [\mathsf{b}n/x]))$ by the choice of f. On the other hand, $\mathsf{f}(k, \mathsf{b})n \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{C}, mkn, (\varphi kn, [\mathsf{b}n/x]))$ by Lemma 21. Using the inductive hypothesis,

$$\mathbf{Eval}^{\mathcal{I}}(\mathbf{C}, mkn, (\varphi kn, [\mathsf{b}n/x]))$$

is a singleton and we must have

$$\mathsf{f}(k, \mathsf{b})n = \mathsf{f}(kn, \mathsf{b}n)$$

which verifies $\mathsf{f} \in \mathcal{D}_{\gamma\alpha}$. Finally, suppose $\mathsf{g} \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, m, \varphi)$. For any $k \in M$ and $\mathsf{b} \in \mathcal{D}_\beta$, we must have $\mathsf{g}(k, \mathsf{b}) \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{C}, mk, (\varphi k, [\mathsf{b}/x]))$ and so $\mathsf{g}(k, \mathsf{b}) = \mathsf{f}(k, \mathsf{b})$. Thus $\mathsf{g} = \mathsf{f}$ and $\mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, m, \varphi)$ is a singleton. ■

We can now prove Theorem 6. Let $\mathcal{E}_\varphi^{\mathcal{I},m}(\mathbf{A})$ be the unique member of $\mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, m, \varphi)$ for all $\mathbf{A} \in \mathit{wff}_\alpha(\mathcal{S})$, $m \in M$ and assignments φ into $\mathcal{D}$. Note that $\mathcal{E}_\varphi^{\mathcal{I},m}(w) = \mathcal{I}(w)m$ since $\mathbf{Eval}^{\mathcal{I}}(w, m, \varphi) = \{\mathcal{I}(w)m\}$ and so $\mathcal{E}_\varphi^{\mathcal{I},e}(w) = \mathcal{I}(w)$ for each parameter or constant w. It remains to check the conditions in Definition 4. For each variable x, $\mathcal{E}_\varphi^m(x) = \varphi(x)$ since $\mathbf{Eval}^{\mathcal{I}}(x, m, \varphi) = \{\varphi(x)\}$. For each parameter or constant w, we have $\mathcal{E}_\varphi^m(w) = \mathcal{I}(w)m = \mathcal{E}_\psi^e(w)m$. Since $\mathcal{E}_\varphi^m(\mathbf{F}) \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{F}, m, \varphi)$ and $\mathcal{E}_\varphi^m(\mathbf{B}) \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{B}, m, \varphi)$, we know $\mathcal{E}_\varphi^m(\mathbf{F})@\mathcal{E}_\varphi^m(\mathbf{B}) \in \mathbf{Eval}^{\mathcal{I}}([\mathbf{FB}], m, \varphi)$. Hence

$$\mathcal{E}_\varphi^m(\mathbf{FB}) = \mathcal{E}_\varphi^m(\mathbf{F})@\mathcal{E}_\varphi^m(\mathbf{B}).$$

Finally,

$$\mathcal{E}_\varphi^m(\lambda x_\beta \mathbf{A}_\alpha) = \mathsf{f} \in \mathbf{Eval}^{\mathcal{I}}([\lambda x_\beta \mathbf{A}_\alpha], m, \varphi) \subseteq \mathcal{D}_{\alpha\beta}$$

where $\mathsf{f}(k, \mathsf{b}) \in \mathbf{Eval}^{\mathcal{I}}(\mathbf{A}, mk, (\varphi k, [\mathsf{b}/x]))$, i.e., $\mathsf{f}(k, \mathsf{b}) = \mathcal{E}_{\varphi k, [\mathsf{b}/x]}^{\mathcal{I}, mk}(\mathbf{A})$. Thus all the conditions in Definition 4 hold and the proof is complete.