A synthetic axiomatization of Map Theory[☆]

Chantal Berline[a], Klaus Grue[b,1,*]

[a] *CNRS, PPS, UMR 7126, Univ Paris Diderot, Sorbonne Paris Cité, F-75205 Paris, France*
[b] *DIKU, Universitetsparken 1, DK-2100 Copenhagen East, Denmark*

Abstract

This paper presents a subtantially simplified axiomatization of Map Theory and proves the consistency of this axiomatization in ZFC under the assumption that there exists an inaccessible ordinal.

Map Theory axiomatizes lambda calculus plus Hilbert's epsilon operator. All theorems of ZFC set theory including the axiom of foundation are provable in Map Theory, and if one omits Hilbert's epsilon operator from Map Theory then one is left with a computer programming language. Map Theory fulfills Church's original aim of introducing lambda calculus.

Map Theory is suited for reasoning about classical mathematics as well as computer programs. Furthermore, Map Theory is suited for eliminating the barrier between classical mathematics and computer science rather than just supporting the two fields side by side.

Map Theory axiomatizes a universe of "maps", some of which are "wellfounded". The class of wellfounded maps in Map Theory corresponds to the universe of sets in ZFC. Previous versions of Map Theory had axioms which populated the class of wellfounded maps, much like the power set axiom et.al. populates the universe of ZFC. The new axiomatization of Map Theory is "synthetic" in the sense that the class of wellfounded maps is *defined* inside Map Theory rather than being introduced through axioms.

Keywords: lambda calculus, foundation of mathematics, map theory

Contents

[☆] This document is a collaborative effort.

[*] Corresponding author

Email addresses: berline@pps.jussieu.fr (Chantal Berline), grue@diku.dk (Klaus Grue)

[1] Present address: Rovsing A/S, Dyregaardsvej 2, DK-2740 Skovlunde, Denmark

1. Introduction

Map theory is an axiomatic theory which consists of a computer programming language plus Hilbert's epsilon operator. All theorems of ZFC set theory including the axiom of foundation are provable in Map Theory, and if one omits Hilbert's epsilon operator from Map Theory then one is left with a computer programming language, c.f. Section 2.4.

Map Theory is suited for reasoning about classical mathematics as well as computer programs. Furthermore, Map Theory is suited for eliminating the barrier between classical mathematics and computer science rather than just supporting the two fields side by side.

The first version of Map Theory [8], which we will call MT_0 in this paper, had complex axioms and a complex model. [3] provided a simpler model. The present paper provides a simpler and more synthetic, while more powerful, axiomatization that we will call MT, and proves the consistency of the enhanced system starting from the *minimal* (or *canonical*) models of MT_0 built in [3].

Map Theory is an axiomatic system, but it does not rely on propositional and first order predicate calculus. Rather, it is an equational theory which relies on untyped lambda calculus. In particular, models of Map Theory are also models of untyped lambda calculus. We shall refer to elements of such models as *maps*. As for λ-calculus, programming is made possible in Map Theory by the adjunction of compatible reduction rules.

Map Theory generates quantifiers and first order calculus via a construct ε, whose semantics is that of Hilberts choice operator acting over a universe $\hat{\psi}$ of "wellfounded maps". ε is axiomatized throught the "quantification axioms" (four equations).

Apart from ε, MT and MT_0 have in common a few elementary constructs and related axioms which take care of the computational part of Map Theory, and simultaneously bear some other meanings [8]. Some "sugar" has also been added to MT, but this is inessential.

Apart from ε and the elementary constructs, MT_0 has only one construct ϕ, which is in spirit the characteristic function of $\hat{\psi}$. MT_0 has the power to embody ZFC because ϕ satisfies ten equations, each axiomatizing one, specific closure property of $\hat{\psi}$, and it also has one inference rule for wellfoundedness. Having ten axioms, even if most of them are very simple, was acceptable (after all ZFC also has many existence axioms) but not satisfactory in that all the closure properties are instances of a single, although non-axiomatizable, Generic Closure Property (GCP, [3]). GCP was one of the founding intuitions behind Map Theory (c.f. [8]), it was satisfied in our models of MT_0, and our desire was to reflect it at the level of syntax.

With the present MT not only do we solve this problem (whence "synthetic") but also eliminate the construct ϕ, the ten axioms, and the inference rule, replacing them by ... nothing (whence "simpler"). Moreover, the new system is stronger than MT_0. That one construct, ten axioms, and a rule can be replaced by nothing should be taken with three grains of salt as explained in the following.

The main trick is that we take $\hat{\psi}$ to be the smallest universe satisfying GCP, and the characteristic function ψ of that universe happens to be defininable from other MT constructs. The first grain of salt is that when we eliminate ϕ, we replace it by ψ in the quantification axioms. The second grain of salt is that for defining ψ we need to add a construct E ("pure existence") and its related axioms. However, and in contrast to ϕ, E is very simple to describe, to axiomatize, and to model, so the cost of that is small. The third grain of salt is that the definition of ψ also requires a minimal fixed point operator. Fixed point operators come for free with untyped λ-calculus, but forcing minimality at the level of syntax requires to axiomatize it w.r.t. some pertinent and MT-definable order. This too can be done, and at a rather low syntactic and semantic cost.

Finding the right MT was of course already a challenge, but proving its consistency was another one. Fortunately, the consistency of a system only has to be proved once, while hopefully the system will be used many times, so having a simpler system is a gain, even if its consistency proof is demanding.

To give an idea of the difficulty of finding an appropriate and consistent MT, it is worth to recall here that a first "synthetic" version of MT, called MT_c, conceived in the same spirit, was present in [9], that many proofs have been developed in it (which should be easy to translate to MT), but that the consistency of MT_c is still an open problem. We will soon come back to MT_c below.

The consistency proof of MT starts from a (canonical) model of MT_0, and then occupies several sections. Not all models of MT_0 can be enriched to a model of MT; in fact MT has necessarily much fewer models than MT_0.

Section 2 gives a preview of MT. Section 3 presents the semantics of MT informally. Section 4 presents the axioms and inference rules. Section 5 compares

MT to MT_0 and discusses the axioms. The remaining sections prove the consistency. Appendix A addresses the adequacy, soundness, and full abstraction of canonical pre-models with respect to the programming language underlying MT. Appendix B summarizes the axioms and rules of MT. Appendix C contains an index.

Relation to other systems

MT_0 was the first system fulfilling Church's original aim at the origin of the creation of (untyped) λ-calculus. Church's aim was to give a common and untyped foundation to mathematics and computation, based on functions (viewed as rules) and application, in place of sets and membership. As is well known, Church's general axiomatic system was soon proved inconsistent, but its computational part (the now usual untyped λ-calculus) had an immense impact on computer science. The various intuitions behind Map Theory, its very close links to Church's system, its advantages w.r.t. ZFC, including an integrated programming language, and a much richer expressive power (classes, operators, constructors, etc. also quite directly live in Map Theory), all this was developed in [3, 8].

For a comparison of Map Theory with other foundational+computational systems see [3, 8] and also Section 2.1 below. For a version of MT_0 with anti-foundation axioms a la Aczel [1], see [15, 16].

In the present paper we will hence concentrate on: giving enough of the intuitions on Map Theory to motivate the reader, to deepen the understanding of in which sense MT is an improvement over MT_0, and to support the technical developments that we will have to implement in order to prove the consistency of MT. Finally, we will also explore the computational properties of the simplest ("canonical") models of the equational theory MT, w.r.t. the computational rules which are behind it.

2. Preview of MT

2.1. Relation to ZFC

MT is a Hilbert style axiomatic system which comprises syntactic definitions of terms and well-formed formulas as well as axioms and inference rules.

MT has two terms T and F which denote truth and falsehood, respectively, and MT formulas have form $\mathcal{A} = \mathcal{B}$ where $\mathcal{A}$ and $\mathcal{B}$ are MT terms. We shall refer to such formulas as *equations.*

Set membership of ZFC is definable as a term $\textsc{e}$ of MT such that $\textsc{e}xy = \mathsf{T}$ iff the set represented by x belongs to the set represented by y. We shall use the infix notation $x \ddot{\in} y$ for $\textsc{e}xy$. Also definable in MT are universal quantification $\ddot{\forall}$, negation $\ddot{\neg}$, implication $\ddot{\Rightarrow}$, the empty set $\ddot{\emptyset}$, and so on.

In MT it is trivial to prove $\mathsf{T} = \ddot{\neg}\mathsf{F}$ and $\mathsf{F} = \ddot{\neg}\mathsf{T}$, and one cannot (Theorem 2.1.1) prove $\mathsf{T} = \mathsf{F}$. For suitable definitions of set membership and so on, each formula $\mathcal{A}$ of ZFC becomes a term $\ddot{\mathcal{A}}$ of MT. The general idea is that if $\mathcal{A}$ holds in ZFC then $\ddot{\mathcal{A}} = \mathsf{T}$ should hold in MT (Theorem 2.1.2). As an example,

$\forall x\colon x\notin\emptyset$ is a formula of ZFC, $\ddot{\forall} x\colon x \ddot{\notin} \ddot{\emptyset}$ is the corresponding term of MT, and $(\ddot{\forall} x\colon x \ddot{\notin} \ddot{\emptyset}) = \mathsf{T}$ holds in MT. The term $\ddot{\forall} x\colon x \ddot{\notin} \ddot{\emptyset}$ is shorthand for $\ddot{\forall}(\lambda x.\, \ddot{\neg}(x \ddot{\in} \ddot{\emptyset}))$.

We now make the statements above more precise. Let ZFC+SI denote ZFC extended with the assumption that there exists an inaccessible ordinal. Let σ be the smallest inaccessible. Let ¬SI be the assumption that there exist no inaccessible ordinals. Let V_σ be the usual model of ZFC+¬SI in ZFC+SI. Let $\mathcal{M}$ be the *canonical model* of MT in ZFC+SI built in Section 9. The present paper proves:

Theorem 2.1.1 (Main Theorem). *$\mathcal{M}$ satisfies all the axioms and inference rules of MT and does not satisfy $\mathsf{T} = \mathsf{F}$. This proves the consistency of MT.*

For arbitrary, closed formulas $\mathcal{A}$ of ZFC we have:

Theorem 2.1.2. *V_σ satisfies $\mathcal{A}$ iff $\mathcal{M}$ satisfies $\ddot{\mathcal{A}} = \mathsf{T}$.*

Theorem 2.1.2 follows easily from [3, Appendix A.4] and the fact that $\mathcal{M}$ builds on top of the model built in [3]. As a technicality, MT and MT_0 have slightly different syntax, but for closed formulas $\mathcal{A}$ of ZFC, $\ddot{\mathcal{A}}$ only uses constructs common to MT and MT_0, and Theorem 2.1.2 carries over from MT_0 to MT without changing the definition of $\ddot{\mathcal{A}}$.

Conjecture 2.1.3. *If $\mathcal{A}$ is provable in ZFC+¬SI then $\ddot{\mathcal{A}} = \mathsf{T}$ is provable in MT.*

Conjecture 2.1.3 is supported by the following:

Theorem 2.1.4 ([8]). *If $\mathcal{A}$ is provable in ZFC then $\ddot{\mathcal{A}} = \mathsf{T}$ is provable in* MT_0.

Theorem 2.1.5 ([9]). *If $\mathcal{A}$ is provable in ZFC then $\ddot{\mathcal{A}} = \mathsf{T}$ is provable in* MT_c *where* MT_c *is the version of Map Theory defined in [9].*

MT_c resembles MT, but all attempts at proving MT_c consistent have failed. A proof of $(\neg\mathrm{SI}) = \mathsf{T}$ in MT_c should be easy. To prove Conjecture 2.1.3 one has to prove $(\neg\mathrm{SI}) = \mathsf{T}$ in MT and to translate the proof of Theorem 2.1.5 to MT. This remains to be done.

It is not really intended that MT should prove $(\neg\mathrm{SI}) = \mathsf{T}$; it is rather a side effect. The original MT_0 was designed as "as flexible as ZFC", and is in particular consistent with $\mathrm{SI} = \mathsf{T}$ as well as $(\neg\mathrm{SI}) = \mathsf{T}$. The MT_0 system has a constant ϕ and a group of *ϕ-axioms* which comprises the three wellfoundedness axioms, the seven construction axioms, and the inference rule of transfinite induction of [8]. MT replaces the constant ϕ by a closed term ψ and omits the ϕ-axioms. That makes MT more rigid than MT_0 and in particular makes $(\neg\mathrm{SI}) = \mathsf{T}$ provable. The definition of ψ is somewhat analogous to the definition of the minimal V_σ in ZFC (σ the first inaccessible).

2.2. Recursion

MT has a number of advantages over ZFC. One is that it allows to combine unrestricted recursion with arbitrary set constructors. As an example, suppose $x \ddot{\cup} y$, $\dot{\bigcup} x$, $\dot{\{}x\dot{\}}$, and $\dot{\{}\mathcal{A} \mid x \ddot{\in} \mathcal{B}\dot{\}}$ are the binary union, unary union, unit set, and replacement set operators of ZFC, respectively, translated to MT. One may define the ordinal successor x' thus in MT:

$$x' \equiv x \ddot{\cup} \dot{\{}x\dot{\}}$$

And then one may define the set rank operator $\rho(x)$ thus:

$$\rho(x) \equiv \ddot{\bigcup} \dot{\{}\rho(y)' \mid y \ddot{\in} x\dot{\}}$$

In this paper, $\equiv$ is used for definitions. In MT, definitions are allowed to be recursive like the definition of ρ above where the defined concept ρ appears in the right hand side of its own definition. Recursive definitions in MT are shorthand for direct (i.e. non-recursive) definitions which involve the fixed point operator (c.f. Section 3.2).

ZFC only permits direct definitions and includes no fixed point operator. ZFC permits definition by transfinite induction, which resembles primitive recursion, but does not support unrestricted recursion like MT does.

The definition of ρ in MT above does not rely on ordinals or transfinite induction. Rather, in MT, one may define ρ as above and then use it to define the class Ord of ordinals:

$$\mathrm{Ord}(x) \equiv \ddot{\exists} y \colon x \ddot{\in} \rho(y)$$

As another example, in MT we may use Hilbert's choice operator ε recursively to well-order any set A:

$$\begin{array}{lcl} f(\alpha) & \equiv & \varepsilon x \colon x \ddot{\in} g(\alpha) \\ g(\alpha) & \equiv & A \ddot{\setminus} \dot{\{} f(\gamma) \mid \gamma \ddot{\in} \alpha \dot{\}} \\ x \prec y & \equiv & \ddot{\exists} \alpha \colon x \ddot{\in} A \ddot{\setminus} g(\alpha) \ddot{\wedge} y \ddot{\in} g(\alpha) \end{array}$$

Above, A is a wellfounded map of MT and $\prec$ well-orders the set represented by A. $\varepsilon x \colon \mathcal{B}$ chooses if possible a wellfounded x such that $\mathcal{B}$ is true, c.f. Section 4.5.

2.3. Russell's paradox

In naive set theory, define $S \equiv \{x | x \notin x\}$ and $\mathcal{R} \equiv S \in S$. We have $x \in S \Leftrightarrow x \notin x$ and $\mathcal{R} \Leftrightarrow S \in S \Leftrightarrow S \notin S \Leftrightarrow \neg\mathcal{R}$ which is Russell's paradox. The paradox states that negation has a fixed point, which is impossible in a consistent, two-valued logic.

In ZFC, the paradox is avoided by banning S, but that is not an option in MT which allows recursion. As an example, one may define Russell's paradoxical statement $\mathcal{R}$ thus in MT:

$$\mathcal{R} \equiv \ddot{\neg} \mathcal{R}$$

In MT, if $\mathcal{R} = \mathsf{T}$ then $\mathcal{R} = \ddot{\neg}\mathsf{T} = \mathsf{F}$ and if $\mathcal{R} = \mathsf{F}$ then $\mathcal{R} = \ddot{\neg}\mathsf{F} = \mathsf{T}$ so $\mathcal{R}$ equals neither T nor F. Indeed, MT has a fixed point operator Y and an element $\perp$ playing, among others, the role of the third logical value "undefinedness"; and it is indeed provable in MT that $\mathcal{R} = \mathsf{Y}\ddot{\neg} = \perp$.

2.4. Programming

Another advantage of MT over ZFC is that if one removes Hilbert's ε from the core syntax of MT then one is left with a Turing complete computer programming language. This language is a type free lambda calculus with uhr-elements and the programs are closed ε-free MT-terms.

The present paper is about MT as an equational axiomatic theory. That MT can be used for programming should be seen here as motivation only. When speaking of programming with MT it is understood that we have furthermore included compatible reduction rules (c.f. Section 3.3). We now elaborate on the programming motivations.

Having a computer programming language as a syntactical subset of the theory allows to reason about programs without having to model the programs mathematically. That simplifies the field of program semantics considerably. For a simple example of programming and reasoning in MT, see Example 4.2.1. Map Theory also provides good support for reasoning about languages different from its own.

Since MT contains a computer programming language, a programmer may ask questions like:

- Is is possible to implement arbitrary algorithms efficiently in the language?
- Is it possible to download compiler, linker, and runtime system for the language?
- Is it possible in the language e.g. to receive mouse clicks from a user, to write bytes to a disk, and to display graphics on a screen?

The answers to these questions are yes, c.f. `http://lox.la/`.

Sections 3.3–3.7 describe the computational aspects of MT. Appendix A proves some results on computational adequacy, soundness, and full abstraction. `http://lox.la/` elaborates on MT as a programming language.

3. Informal semantics

3.1. Introduction

To introduce ZFC one will typically give some examples of finite sets first. Actually, ZFC is nothing but the theory of finite sets extended by an infinite set ω. Likewise, MT is nothing but the theory of computable functions extended with Hilbert's non-computable epsilon operator.

3.2. Syntax

The syntax of variables, terms, and well-formed formulas of MT reads:

$$\begin{array}{lcl} \mathcal{V} & ::= & x \mid y \mid z \mid \cdots \\ \mathcal{T} & ::= & \mathcal{V} \mid \lambda\mathcal{V}.\mathcal{T} \mid \mathcal{T}\mathcal{T} \mid \mathsf{T} \mid \mathsf{if}[\mathcal{T},\mathcal{T},\mathcal{T}] \mid \bot \mid \mathsf{Y}\mathcal{T} \mid \mathcal{T}\|\mathcal{T} \mid \mathsf{E}\mathcal{T} \mid \varepsilon\mathcal{T} \\ \mathcal{W} & ::= & \mathcal{T} = \mathcal{T} \end{array}$$

$\lambda x.\mathcal{A}$ denotes lambda abstraction and fx denotes functional application. T denotes truth. Falsehood F is not included in the syntax; we define it by $\mathsf{F} \equiv \lambda x.\mathsf{T}$. if denotes selection; we have $\mathsf{if}[\mathsf{T}, b, c] = b$ and $\mathsf{if}[\lambda x.\mathcal{A}, b, c] = c$. $\bot$ denotes undefinedness or infinite looping. Y denotes a fixed point operator; we have $\mathsf{Y}f = f(\mathsf{Y}f)$ for all f. $\|$ denotes parallel or; $a \| b$ equals T if a or b or both equal T. E denotes pure existence; we have $\mathsf{E}p = \mathsf{T}$ iff $px = \mathsf{T}$ for some x. We say that x is *wellfounded* if $\psi x = \mathsf{T}$ where we define ψ in Section 4.6. ε denotes Hilbert's choice operator; under reasonable conditions, εp is a wellfounded x such that $px = \mathsf{T}$.

In Section 8 we introduce the notions of *standard quasimodels* of MT and the more restricted notion of *canonical models*, where the latter are unique up to the choice of a regular cardinal κ. Canonical models satisfy all axioms and inference rules of MT. Standard quasimodels satisfy some (possibly all) of the axioms and inference rules.

Now let $\mathsf{Y}_{\text{Curry}} = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ and $\bot_{\text{Curry}} = \mathsf{Y}_{\text{Curry}}\lambda x.x$. In canonical models of MT we will prove that

$$\begin{array}{lclcl} \mathsf{Y}f & = & \mathsf{Y}_{\text{Curry}}f & & \\ \bot & = & \bot_{\text{Curry}} & = & (\lambda x.\, xx)(\lambda x.\, xx) \end{array}$$

Thus, without loss of power and consistency, one might omit $\bot$ and Y from the syntax and use $\mathsf{Y}_{\text{Curry}}$ and $\bot_{\text{Curry}}$ instead. Doing so, however, would reduce the number of possible models of MT. Thus, we include $\bot$ and Y in the syntax and prove the two equations above as separate theorems which are only guaranteed in canonical models. Inclusion of $\bot$ and Y also simplifies the consistency proof since modelling of $\mathsf{Y}f$ and proving $\mathsf{Y}f = \mathsf{Y}_{\text{Curry}}f$ can be treated separately. We shall use MT_{def} to denote the version of MT where we omit $\bot$ and Y from the syntax.

Parallel or $\|$ is neither needed for developing ZFC in MT nor convenient when programming. Parallel or is merely included for the sake of a full abstraction result. We use full abstraction for explaining equality in Section 3.6.

3.3. Computation

The constructs $\lambda x.\mathcal{A}$, ab, T, and $\mathsf{if}[a, b, c]$ together with adequate reduction rules (defined below) form a computer programming language. The language is Turing complete in the sense that any recursive function can be expressed in it.

In the following, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ denote terms, a, b, c, and r denote closed terms, and x, y, and z denote variables. $\langle \mathcal{A} \mid x := \mathcal{B} \rangle$ denotes substitution with renaming of bound variables as needed.

From a theoretical point of view, and very remote from the implementation in [10], one can define the programming language by the smallest relation $\stackrel{1}{\to}$ which satisfies:

$$
\begin{array}{lcl}
\mathsf{T}\mathcal{B} & \stackrel{1}{\to} & \mathsf{T} \\
(\lambda x.\mathcal{A})\mathcal{B} & \stackrel{1}{\to} & \langle \mathcal{A} \mid x := \mathcal{B} \rangle \\
\mathsf{if}[\mathsf{T}, \mathcal{B}, \mathcal{C}] & \stackrel{1}{\to} & \mathcal{B} \\
\mathsf{if}[\lambda x.\mathcal{A}, \mathcal{B}, \mathcal{C}] & \stackrel{1}{\to} & \mathcal{C} \\
\mathcal{A} \stackrel{1}{\to} \mathcal{R} & \Rightarrow & \mathcal{A}\mathcal{B} \stackrel{1}{\to} \mathcal{R}\mathcal{B} \\
\mathcal{A} \stackrel{1}{\to} \mathcal{R} & \Rightarrow & \mathsf{if}[\mathcal{A}, \mathcal{B}, \mathcal{C}] \stackrel{1}{\to} \mathsf{if}[\mathcal{R}, \mathcal{B}, \mathcal{C}]
\end{array}
$$

As an example of a reduction, $\mathsf{if}[\lambda x.x, \lambda y.y, \lambda z.z]\mathsf{T}$ reduces to T:

$$\mathsf{if}[\lambda x.x, \lambda y.y, \lambda z.z]\mathsf{T} \stackrel{1}{\to} (\lambda z.z)\mathsf{T} \stackrel{1}{\to} \mathsf{T}$$

We have specified leftmost reduction order so that e.g. $\mathsf{if}[\mathsf{T}, \mathsf{T}, (\lambda x.xx)(\lambda x.xx)]$ reduces to T without $(\lambda x.xx)(\lambda x.xx)$ being reduced.

Suppose $a \stackrel{1}{\to} b$. Under this assumption, $a = b$ is provable in MT using only elementary axioms and inference rules. Hence, $a = b$ holds in all models of MT. Also, $a = b$ holds in all standard quasimodels, even those which do not model all of MT (c.f. Section 5.4). That holds for the definition of $a \stackrel{1}{\to} b$ given above as well as for the extensions given in the following.

We use $\lambda xy.\mathcal{A}$ to denote $\lambda x.\lambda y.\mathcal{A}$. Furthermore, application $\mathcal{A}\mathcal{B}$ is left associative and has higher priority than $\lambda x.\mathcal{A}$ so $\lambda xy.\mathcal{A}\mathcal{B}\mathcal{C}$ means $\lambda x.\lambda y.((\mathcal{A}\mathcal{B})\mathcal{C})$.

We shall say that a term is a *root normal term* if it has form T or $\lambda x.\mathcal{A}$. Reduction stops when a root normal term is reached. As an example,

$$(\lambda xy.x)((\lambda x.xx)(\lambda x.xx))$$

reduces to

$$\lambda y.(\lambda x.xx)(\lambda x.xx)$$

which is not reduced further. We shall refer to terms of form T and $\lambda x.\mathcal{A}$ as *true* and *function* normal terms, respectively.

3.4. Further constructs

One may extend the programming language by the constructs $\bot$, $\mathsf{Y}a$, $a \parallel b$, and $\mathsf{E}a$. One cannot extend the language by εa because ε cannot be seen as computable.

In the following, a, b, c, f, and r denote closed, epsilon free terms.

The constructs $\bot$ and Y may be defined or may be included in the syntax. If they are defined, they need no reduction rules. If they are included in the syntax, their reduction rules read:

$$
\begin{array}{lcl}
\bot & \stackrel{1}{\to} & \bot \\
\mathsf{Y}f & \stackrel{1}{\to} & f(\mathsf{Y}f)
\end{array}
$$

The construct $a \parallel b$ can be computed thus: Reduce a and b in parallel. If one of them reduces to T, halt the other reduction and return T. If both reduce to function normal terms, return $\lambda x.\mathsf{T}$.

The construct $\mathsf{E}a$ can be computed thus: Reduce ab for all closed terms b in parallel. If ab reduces to T for some b, halt all reductions and return T. Otherwise, proceed computing indefinitely.

The construct $\mathsf{E}a$ is not very useful in computer programs since $\mathsf{E}a$ either loops indefinitely or returns T. The construct $a \parallel b$ is slightly more useful since it has two possible return values, T and $\lambda x.\mathsf{T}$, but it is still not a popular programming construct, and few programming languages support it. The implementation in [10] supports neither $\mathsf{E}a$ nor $a \parallel b$.

The construct $\mathsf{E}a$ is needed for defining ψ and so is indirectly needed for axiomatizing Hilbert's choice operator ε. The construct $a \parallel b$ is included for the sake of full abstraction.

Reduction rules for $a \parallel b$ read:

$$\begin{array}{lll} \mathsf{T} \parallel b & \stackrel{1}{\rightarrow} & \mathsf{T} \\ (\lambda x.\mathcal{A}) \parallel b & \stackrel{1}{\rightarrow} & \mathsf{if}[b, \mathsf{T}, \lambda x.\mathsf{T}] \\ a \stackrel{1}{\rightarrow} r & \Rightarrow & (a \parallel b) \stackrel{1}{\rightarrow} (b \parallel r) \end{array}$$

A reduction rule for $\mathsf{E}a$ is more complicated. To reduce $\mathsf{E}a$ we need to reduce ab for all closed terms b in parallel. Now define

$$\begin{array}{lllll} \mathsf{S} & \equiv & \mathsf{C}_1 & \equiv & \lambda xyz.xz(yz) \\ \mathsf{K} & \equiv & \mathsf{C}_2 & \equiv & \lambda xy.x \\ & & \mathsf{C}_3 & \equiv & \mathsf{T} \\ & & \mathsf{C}_4 & \equiv & \lambda xyz.\mathsf{if}[x, y, z] \\ & & \mathsf{C}_5 & \equiv & \bot \\ & & \mathsf{C}_6 & \equiv & \lambda x.\mathsf{Y}x \\ & & \mathsf{C}_7 & \equiv & \lambda xy.(x \parallel y) \\ & & \mathsf{C}_8 & \equiv & \lambda x.\mathsf{E}x \end{array}$$

We shall refer to terms built up from the eight combinators above plus functional application as *combinator terms*. Every closed, epsilon free term of MT is computationally equivalent to a combinator term. Thus, we may compute $\mathsf{E}a$ by applying a to all combinator terms b:

$$\begin{array}{lll} \mathsf{E}a & \stackrel{1}{\rightarrow} & a\mathsf{C}_1 \parallel \cdots \parallel a\mathsf{C}_8 \parallel \mathsf{E}x.\mathsf{E}y.a(xy) \\ a \stackrel{1}{\rightarrow} r & \Rightarrow & \mathsf{E}a \stackrel{1}{\rightarrow} \mathsf{E}r \end{array}$$

Above, $\mathsf{E}x.\mathcal{A}$ denotes $\mathsf{E}(\lambda x.\mathcal{A})$. To see how E works, first note that $\mathsf{E}a$ by definition reduces to

$$a\mathsf{C}_1 \parallel \cdots \parallel a\mathsf{C}_8 \parallel \mathsf{E}x.\mathsf{E}y.a(xy)$$

Second note that the last factor $\mathsf{E}x.\mathsf{E}y.a(xy)$ in turn reduces to

$$\mathsf{E}y.a(\mathsf{C}_1 y) \parallel \cdots \parallel \mathsf{E}y.a(\mathsf{C}_8 y) \parallel \mathsf{E}u.\mathsf{E}v.\mathsf{E}y.a((uv)y)$$

Third note that the first factor $\mathsf{E}y.a(\mathsf{C}_1 y)$ in turn reduces to

$$a(\mathsf{C}_1\mathsf{C}_1) \parallel \cdots \parallel a(\mathsf{C}_1\mathsf{C}_8) \parallel \mathsf{E}u.\mathsf{E}v.a(\mathsf{C}_1(uv))$$

The penultimate factor $a(\mathsf{C}_1\mathsf{C}_8)$ shows that a, among other, is applied to the combinator term $\mathsf{C}_1\mathsf{C}_8$. In general, reduction of $\mathsf{E}a$ causes a to be applied to all combinator terms in parallel.

We have now given reduction rules for reducing arbitrary closed, epsilon free terms. We shall give no reduction rules for εa since, as mentioned, it is not computable.

3.5. Programs

We shall refer to closed, ε-free MT terms as *MT programs*. Likewise, we shall refer to closed, ε-free $\mathrm{MT}_{\mathrm{def}}$ terms as *MT_{def} programs* and to closed, ε- and ϕ-free MT_0 terms as *MT_0 programs*.

The programs of each of the theories are exactly the closed terms which are reducible by machine. Here we do not require reduction to terminate: a machine is supposed to loop indefinitely when reducing e.g. $\perp$, and $\perp$ is counted among the programs.

3.6. Equality

Wellformed formulas of MT have form $a = b$ where a and b are terms. We now present some of the intuition behind equality.

Let $\mathcal{N}_t$ be the set of MT programs that reduce to T, let $\mathcal{N}_f$ be the set of MT programs that reduce to function normal form, and let $\mathcal{N}_\perp$ be the set of the remaining MT programs. For MT programs a and b we define *root equivalence* $a \sim b$ thus:

$$a \sim b \Leftrightarrow (a \in \mathcal{N}_t \Leftrightarrow b \in \mathcal{N}_t) \wedge (a \in \mathcal{N}_f \Leftrightarrow b \in \mathcal{N}_f) \wedge (a \in \mathcal{N}_\perp \Leftrightarrow b \in \mathcal{N}_\perp)$$

In the definition above, each conjunct follows from the other two, so one could omit one of the conjuncts. We say that the MT programs a and b are *observationally equal*, written $a =_{\mathrm{obs}} b$, if $ca \sim cb$ for all MT programs c.

Intuitively, equality of MT is observational equality. Technically, matters are a bit more complicated:

Let $\mathcal{M}_\kappa$ be the MT canonical κ-model as defined in Sections 7–9. Modelling ε requires $\sigma < \kappa$ for an inaccessible σ, but modelling the other constructs just requires $\kappa \geq \omega$. Now let $a =_\kappa b$ denote $\mathcal{M}_\kappa \models a = b$. We have:

Theorem 3.6.1 (Full Abstraction of $\mathcal{M}_\omega$). *$a =_{\mathrm{obs}} b \Leftrightarrow a =_\omega b$ for MT programs a and b.*

See Appendix A for a proof and for related positive and negative results. Full abstraction may help understanding MT minus Hilbert's epsilon operator.

Now for $a, b \in \mathcal{M}_\kappa$ define

$$a \sim_\kappa b \Leftrightarrow (a =_\kappa \mathsf{T} \Leftrightarrow b =_\kappa \mathsf{T}) \wedge (a =_\kappa \perp \Leftrightarrow b =_\kappa \perp)$$

Let $a =^\kappa_{\mathrm{obs}} b$ denote $\forall c{\in}\mathcal{M}_\kappa\colon ca \sim_\kappa cb$. The closest one can get to full abstraction in the general case reads:

Fact 3.6.2. *$a =^{\kappa}_{\mathrm{obs}} b \Leftrightarrow a =_\kappa b$ for $a, b \in \mathcal{M}_\kappa$, $\kappa \geq \omega$, κ regular.*

The fact follows trivially from the definition of $\mathcal{M}_\kappa$, c.f. Section 9.7.

3.7. Extensionality

Two MT programs a and b happen to be observationally equivalent iff

$$ay_1 \cdots y_n \sim by_1 \cdots y_n$$

for all $n \geq 0$ and all MT programs $y_1, \ldots, y_n$. That follows from Theorem 10.1.2 and Appendix A and provides another intuitive description of equality. We also have:

Fact 3.7.1. *Let $a, b \in \mathcal{M}_\kappa$. We have $ca \sim_\kappa cb$ for all $c \in \mathcal{M}_\kappa$ iff $ay_1 \cdots y_n \sim_\kappa by_1 \cdots y_n$ for all $n \geq 0$ and all $y_1, \ldots, y_n \in \mathcal{M}_\kappa$.*

Fact 3.7.1 follows from Theorem 10.1.2. The ZFC equivalent of Fact 3.7.1 reads:

$$a \in c \Leftrightarrow b \in c \text{ for all sets } c \text{ iff } y \in a \Leftrightarrow y \in b \text{ for all sets } y$$

We shall refer to Fact 3.7.1 as *semantic extensionality*; we express it axiomatically in Section 4.3.

3.8. Hilbert's choice operator

To explain ε, we shall resort to a standard model $\mathcal{M}$ as presented in Section 8.3 (in particular, we may use the canonical model $\mathcal{M}_\kappa$, κ regular, $\kappa > \sigma$, σ inaccessible). We shall refer to elements of $\mathcal{M}$ as *maps*.

For all closed MT terms $\mathcal{A}$ let $\underline{\mathcal{A}}$ denote the element of $\mathcal{M}$ denoted by $\mathcal{A}$. Let $\hat{\psi} = \{x{\in}\mathcal{M} \mid \underline{\psi}x = \underline{\mathsf{T}}\}$ where we define ψ in Section 4.6. We shall refer to elements of $\hat{\psi}$ as *wellfounded* maps.

As before, let V_σ be the usual model of ZFC inside ZFC+SI. The set $\hat{\psi}$ is a big set in the sense that there exists a surjective function $Z\colon \hat{\psi} \rightarrow V_\sigma$ which allows to represent all sets of V_σ by wellfounded maps.

We shall say that $p \in \mathcal{M}$ is *total*, written Total(p), if $\forall x{\in}\hat{\psi}\colon px \neq \underline{\perp}$.

We shall use $\underline{\varepsilon}$ to denote the intended interpretation of Hilbert's choice operator εp. $\underline{\varepsilon}$ is a function of type $\underline{\varepsilon}\colon \mathcal{M} \rightarrow \mathcal{M}$. For all $p \in \mathcal{M}$, $\underline{\varepsilon}$ has the following properties:

$$\begin{array}{lcll}
\underline{\varepsilon}(p) & = & \underline{\perp} & \text{if } \neg\mathrm{Total}(p) \\
\underline{\varepsilon}(p) & \in & \hat{\psi} & \text{if } \mathrm{Total}(p) \\
p(\underline{\varepsilon}(p)) & = & \underline{\mathsf{T}} & \text{if } \mathrm{Total}(p) \wedge \exists x{\in}\hat{\psi}\colon px = \underline{\mathsf{T}} \\
\underline{\varepsilon}(p) & = & \underline{\varepsilon}q & \text{if } \mathrm{Total}(p) \wedge \mathrm{Total}(q) \wedge \forall x{\in}\hat{\psi}\colon (px = \underline{\mathsf{T}} \Leftrightarrow qx = \underline{\mathsf{T}})
\end{array}$$

In other words, $\underline{\varepsilon}$ is a Hilbert choice operator over $\hat{\psi}$. The last property above is Ackermann's axiom.

The strictness requirement that $\varepsilon p = \perp$ if $\neg$Total(p) has two motivations. First, MT includes an inference rule which implies that application is monotonic for a certain order $p \preceq q$ so ε must be monotonic in the sense that $p \preceq q$ must imply $\varepsilon p \preceq \varepsilon q$. Strictness together with Ackermann's axiom and the definition of $p \preceq q$ given later is sufficient to ensure monotonicity of ε. Second, the strictness requirement simplifies the quantification axioms stated later.

When comparing with [3] note that the present ψ is analogous to the ϕ of [3] and the present $\hat{\psi}$ is analogous to the Φ of [3]. The Ψ of [3] is unrelated to ψ and $\hat{\psi}$.

3.9. Pure existence revisited

Let $\mathcal{M}$ and $\mathcal{M}_\kappa$ be as above. Pure existence E is designed to satisfy in $\mathcal{M}$ that $\mathsf{E}p = \mathsf{T}$ if $px = \mathsf{T}$ for some x and $\mathsf{E}p = \perp$ otherwise (c.f. Section 4.4). So $\mathsf{E}p = \mathsf{T}$ in $\mathcal{M}$ iff $px = \mathsf{T}$ for some $x \in \mathcal{M}$ while the reduction rule for $\mathsf{E}p$ given in Section 3.4 gives that $\mathsf{E}p = \mathsf{T}$ iff $px = \mathsf{T}$ for some program x. We now compare these two notions of existential quantification. Define

$$\begin{array}{lcl} \mathsf{E}_{\text{semantic}} & \equiv & \lambda p.\, \mathsf{E}p \\ \mathsf{E}_{\text{syntactic}} & \equiv & \lambda p.\, [\, p\mathsf{C}_1 \parallel \cdots \parallel p\mathsf{C}_8 \parallel \mathsf{E}_{\text{syntactic}}\, \lambda u.\, \mathsf{E}_{\text{syntactic}}\, \lambda v.\, p(uv)] \end{array}$$

We have

$$\begin{array}{lclll} \mathsf{E}_{\text{semantic}}\, p & = & \mathsf{T} & \text{iff} & px = \mathsf{T} \text{ for some map } x \\ \mathsf{E}_{\text{syntactic}}\, p & = & \mathsf{T} & \text{iff} & px = \mathsf{T} \text{ for some program } x \end{array}$$

$\mathcal{M}_\omega$ happens to be a simple and very pertinent model for modelling the computational and elementary part of MT even if $\mathcal{M}_\omega$ is not a model of the full theory. We will see this later on, and we will prove in Appendix A that, among other nice properties, $\mathcal{M}_\omega$ satisfies $\mathsf{E}_{\text{semantic}} = \mathsf{E}_{\text{syntactic}}$ (c.f. Lemma A.4.1). Now, this equation can be proved to be false in $\mathcal{M}_\kappa, \kappa > \omega$ (c.f. Theorem A.8.1 and its proof), and more generally should be false in all the models of MT built from standard premodels, for a similar reason (these models are in a sense "too big").

The E of MT is the semantic one. The computational intuition behind E that we provided at the end of Section 3.4 is valid in $\mathcal{M}_\omega$ but does not hold in full MT.

4. Axioms and inference rules

MT has five groups of axioms and inference rules:

1. Elementary axioms and inference rules
2. Monotonicity and minimality
3. Extensionality
4. Axioms on E
5. Quantification axioms

4.1. Elementary axioms and inference rules

Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ be (possibly open) terms and let x and y be variables. Let $\mathcal{A} \stackrel{\alpha}{=} \mathcal{B}$ denote that $\mathcal{A}$ and $\mathcal{B}$ are identical except for naming of bound variables. Let $\mathbf{1} \equiv \lambda xy.xy$. The first set of axioms and inference rules of MT reads:

Trans	$\mathcal{A} = \mathcal{B}; \mathcal{A} = \mathcal{C} \vdash \mathcal{B} = \mathcal{C}$
Sub	$\mathcal{B} = \mathcal{C} \vdash \mathcal{AB} = \mathcal{AC}$
Gen	$\mathcal{A} = \mathcal{B} \vdash \lambda x.\mathcal{A} = \lambda x.\mathcal{B}$
A1	$\mathsf{T}\mathcal{B} = \mathsf{T}$
A2 (β)	$(\lambda x.\mathcal{A})\mathcal{B} = \mathcal{C} \quad \text{if } \mathcal{C} \stackrel{\alpha}{=} \langle \mathcal{A} \mid x := \mathcal{B} \rangle$
A3	$\bot\mathcal{B} = \bot$
I1	$\mathsf{if}[\mathsf{T}, \mathcal{B}, \mathcal{C}] = \mathcal{B}$
I2	$\mathsf{if}[\lambda x.\mathcal{A}, \mathcal{B}, \mathcal{C}] = \mathcal{C}$
I3	$\mathsf{if}[\bot, \mathcal{B}, \mathcal{C}] = \bot$
QND	$\mathcal{A}\mathsf{T} = \mathcal{B}\mathsf{T}; \mathcal{A}(\mathbf{1}\mathcal{C}) = \mathcal{B}(\mathbf{1}\mathcal{C}); \mathcal{A}\bot = \mathcal{B}\bot \vdash \mathcal{AC} = \mathcal{BC}$
P1	$\mathsf{T} \parallel \mathcal{B} = \mathsf{T}$
P2	$\mathcal{A} \parallel \mathsf{T} = \mathsf{T}$
P3	$\lambda x.\mathcal{A} \parallel \lambda y.\mathcal{B} = \lambda z.\mathsf{T}$
Y	$\mathsf{Y}\mathcal{A} = \mathcal{A}(\mathsf{Y}\mathcal{A})$

Quartum Non Datur (QND) approximates that every map x satisfies $x = \mathsf{T}$ or $x = \bot$ or $x = \mathbf{1}x$, there is no fourth possibility.

Example 4.1.1. As an example of use of QND, define

$$\begin{array}{lcl} \mathsf{F} & \equiv & \lambda x.\mathsf{T} \\ \approx x & \equiv & \mathsf{if}[x, \mathsf{T}, \mathsf{F}] \\ x \wedge y & \equiv & \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \mathsf{F}], \mathsf{if}[y, \mathsf{F}, \mathsf{F}]] \end{array}$$

Using the definitions above, QND allows to prove the following:

$$\begin{array}{rcl} x \wedge y & = & y \wedge x \\ (x \wedge y) \wedge z & = & x \wedge (y \wedge z) \\ x \wedge x & = & \approx x \end{array}$$

4.2. Monotonicity and Minimality

Define:

$$\begin{array}{rcl} x \downarrow y & \equiv & \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \bot], \mathsf{if}[y, \bot, \lambda z.(xz) \downarrow (yz)]] \\ x \preceq y & \equiv & x = x \downarrow y \end{array}$$

The recursive definition of $x \downarrow y$ is shorthand for:

$$x \downarrow y \equiv (\mathsf{Y}\lambda fxy.\mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \bot], \mathsf{if}[y, \bot, \lambda z.f(xz)(yz)]])xy$$

In canonical models, $x \preceq y$ coincides with the order of the model and $x \downarrow y$ is the greatest lower bound of x and y.

The rules of Monotonicity and Minimality read:

Mono	$\mathcal{B} \preceq \mathcal{C} \vdash \mathcal{AB} \preceq \mathcal{AC}$
Min	$\mathcal{AB} \preceq \mathcal{B} \vdash \mathsf{Y}\mathcal{A} \preceq \mathcal{B}$

Example 4.2.1. We now introduce a primitive representation of natural numbers. We first do so semantically. Let $\mathcal{M}$ be a model of MT. We shall refer to elements of $\mathcal{M}$ as maps.

We shall say that a map x is *wellfounded w.r.t. a set G of maps* if, for all $y_1, y_2, \ldots \in G$ there exists a natural number n such that $xy_1 \cdots y_n = \mathsf{T}$. We shall say that a map x is a *natural number map* if it is wellfounded w.r.t. $\{\mathsf{T}\}$. Thus, x is a natural number map if

$$x \overbrace{\mathsf{T}\mathsf{T} \cdots \mathsf{T}}^{n} = \mathsf{T}$$

for some natural number n. As examples, $\lambda xyz.\mathsf{T}$ is a natural number map and $\lambda xyz.\bot$ is not. We shall say that a natural number map x represents the smallest n which satisfies the equation above so $\lambda xyz.\mathsf{T}$ represents 'three'.

We now formalize natural numbers in MT in the sense that we give a number of syntactic definitions which allow to reason formally about natural numbers in MT. The definitions read:

$$\begin{array}{lcl}
0 & \equiv & \mathsf{T} \\
\mathsf{K} & \equiv & \lambda xy.\, x \\
x' & \equiv & \mathsf{K}x \\
\varpi & \equiv & \lambda fx.\, \mathsf{if}[x, \mathsf{T}, f(x\mathsf{T})] \\
\omega & \equiv & \mathsf{Y}\varpi \\
x \stackrel{\omega}{=} y & \equiv & \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \mathsf{F}], \mathsf{if}[y, \mathsf{F}, x\mathsf{T} \stackrel{\omega}{=} y\mathsf{T}]] \\
\mathcal{E} & \equiv & \lambda x.\, x \stackrel{\omega}{=} x
\end{array}$$

As an example, $0''$ denotes one among many maps which represents 'two'. The semantics of ω in the model $\mathcal{M}$ is that $\omega x = \mathsf{T}$ if x is a natural number map and $\omega x = \bot$ otherwise. For that reason we shall refer to ω as the characteristic map of the set of natural number maps (c.f. Definition 4.6.1). For natural number maps x and y we have $(x \stackrel{\omega}{=} y) = \mathsf{T}$ iff x and y represent the same number.

In MT, we can prove $\varpi\mathcal{E} = \lambda x.\, \mathsf{if}[x, \mathsf{T}, x\mathsf{T} \stackrel{\omega}{=} x\mathsf{T}]$. Furthermore, we can prove $\mathcal{E} = (\lambda x.\, x \stackrel{\omega}{=} x) = \lambda x.\, \mathsf{if}[x, \mathsf{if}[x, \mathsf{T}, \mathsf{F}], \mathsf{if}[x, \mathsf{F}, x\mathsf{T} \stackrel{\omega}{=} x\mathsf{T}]] = \lambda x.\, \mathsf{if}[x, \mathsf{T}, x\mathsf{T} \stackrel{\omega}{=} x\mathsf{T}]$ where the latter equality requires QND. Hence, we can prove $\varpi\mathcal{E} = \mathcal{E}$ so $\varpi\mathcal{E} \preceq \mathcal{E}$ (c.f. Example 4.3.2). Hence, we can prove $\omega \preceq \mathcal{E}$ by Min.

Semantically, $\omega \preceq \mathcal{E}$ expresses that $(x \stackrel{\omega}{=} x) = \mathsf{T}$ for all natural number maps: For each natural number map x we have $\omega x = \mathsf{T}$ and $\omega x \preceq \mathcal{E}x$ which shows $\mathcal{E}x = \mathsf{T}$.

Thus, the syntactic statement $\omega \preceq \mathcal{E}$ formalizes the semantic statement that every natural number equals itself and the syntactic statement $\omega \preceq \mathcal{E}$ has a formal proof in MT.

From a program correctness point of view we have now defined a program $x \overset{\omega}{=} y$ inside MT and then proved $\omega \preceq \mathcal{E}$ which expresses that $(x \overset{\omega}{=} x) = \mathsf{T}$ for all natural number maps x. While this is a very simple example and even though space does not even permit to write out proofs in details, this still gives a first, small example of the fact that MT allows programming and reasoning inside the same framework. For a continuation of the present example which uses quantifiers see Example 4.5.2.

Note that the definition of $x \overset{\omega}{=} y$ only uses the four constructs $\lambda x.\mathcal{A}$, ab, T, and $\mathsf{if}[\,a\,,\,b\,,\,c\,]$, so one can compile the program $x \overset{\omega}{=} y$ and run it on arguments x and y using the system described in [10].

In other logical frameworks than MT, given a recursive program like $x \overset{\omega}{=} y$, proofs of theorems like $(x \overset{\omega}{=} x) = \mathsf{T}$ for all natural numbers x usually requires some sort of Peano induction. In MT, induction is expressed by Min. Above, we applied Min to the characteristic function ω of natural number maps to get something equivalent to Peano induction (c.f. [9, Section 7.13]). Applying Min to the characteristic map ψ defined in Section 4.6 yields an induction scheme which resembles but is stronger than transfinite induction (c.f. [9, Section 9.13]).

4.3. Extensionality

Recall $\approx x \equiv \mathsf{if}[\,x\,,\,\mathsf{T}\,,\,\mathsf{F}\,]$ from Example 4.1.1. For all (possibly open) terms $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ (possibly containing epsilon), the inference rule of extensionality reads:

Ext	For variables x, y not free in $\mathcal{A}$ and $\mathcal{B}$ we have $\approx(\mathcal{A}x) = \approx(\mathcal{B}x); \mathcal{A}xy{=}\mathcal{A}\mathcal{C}; \mathcal{B}xy{=}\mathcal{B}\mathcal{C} \vdash \mathcal{A}x = \mathcal{B}x$

Note that if the premises of Ext hold and if $c = \lambda xy.\mathcal{C}$ then we have e.g.

$$\begin{aligned}\approx(\mathcal{A}xy_1y_2) = \approx(\mathcal{A}(cxy_1)y_2) = \approx(\mathcal{A}(c(cxy_1)y_2)) = \\ \approx(\mathcal{B}(c(cxy_1)y_2)) = \approx(\mathcal{B}(cxy_1)y_2) = \approx(\mathcal{B}xy_1y_2)\end{aligned}$$

More generally, we have $\approx(\mathcal{A}xy_1\cdots y_n) = \approx(\mathcal{A}xy_1\cdots y_n)$. Now, canonical models $\mathcal{M}$ have the semantic extensionality property that if $a, b \in \mathcal{M}$ and if

$$\approx(ay_1\cdots y_n) = \approx(by_1\cdots y_n)$$

for all natural numbers n and all $y_1, \ldots, y_n \in \mathcal{M}$ then $a = b$. Rule Ext is a syntactical approximation of that fact which works in those cases where one can find a $\mathcal{C}$ for which one can prove the premises of Ext. It is typically rather difficult to find a witness $\mathcal{C}$ but it is possible more often than one should expect.

The relation between Ext and semantic extensionality is: The premises of Ext entail $\approx(\mathcal{A}xy_1\cdots y_n) = \approx(\mathcal{B}xy_1\cdots y_n)$ which by semantic extensionality entail $\mathcal{A}x = \mathcal{B}x$ which is exactly the conclusion of Ext.

Extensionality in MT corresponds to extensionality in set theory, where the latter says that if $y \in a \Leftrightarrow y \in b$ then $a = b$. Here, $\mathcal{P} \Leftrightarrow \mathcal{Q}$ of set theory corresponds to $\approx\mathcal{P} = \approx\mathcal{Q}$ in MT and $y \in a$ of set theory corresponds to $ay_1 \cdots y_n$ in MT.

Example 4.3.1. Let $i \equiv \lambda x.\,\mathsf{if}[\,x\,,\,\mathsf{T}\,,\,\lambda y.\,i(xy)\,]$ and $I \equiv \lambda x.\,x$. To prove $ix = Ix$ by Ext take $\mathcal{C}$ to be xy and prove $\approx(ix) = \approx(Ix)$, $ixy = i(xy)$, and $Ixy = I(xy)$. The two first statements above can be proved using QND and the third is trivial.

Example 4.3.2. Ext allows to prove $x \downarrow x = x$, $x \downarrow y = y \downarrow x$, and $x \downarrow (y \downarrow z) = (x \downarrow y) \downarrow z$. Those results are useful since they entail $x \preceq x$, $x \preceq y; y \preceq x \vdash x = y$, and $x \preceq y; y \preceq z \vdash x \preceq z$. (For proofs, see [9]).

When developing ZFC in MT, Ext plays a marginal but essential role [9]. In Example 4.2.1, Min replaced usual Peano induction and Min was used in the essential step in proving $(x \stackrel{\omega}{=} x) = \mathsf{T}$, but Ext was also in play for proving $\varpi\mathcal{E} \preceq \mathcal{E}$ from $\varpi\mathcal{E} = \mathcal{E}$. Likewise, when developing ZFC, the results listed in Example 4.3.2 are used in many places. Among other, it is used for proving the MT version of transfinite induction which in turn is used for proving most of the proper axioms of ZFC. Concerning Ext, the development of ZFC only depends on the results listed in Example 4.3.2 and does not make other use of Ext.

Example 4.3.3. Ext also allows to prove $F_2 = F_3$ where

$$\begin{array}{lcl} F_2 & \equiv & \lambda x.\lambda y.F_2 \\ F_3 & \equiv & \lambda x.\lambda y.\lambda z.F_3 \end{array}$$

F_2 and F_3 both denote $\lambda x_1.\lambda x_2.\lambda x_3.\cdots$ and we have $F_2 =_{\mathrm{obs}} F_3$. Thus, F_2 and F_3 provide an example of two pure lambda terms which are provably equal in MT and observationally equal from the point of view of a computer, but not beta equivalent in lambda calculus.

4.4. Axioms on E

Pure existence E is designed to satisfy $\mathsf{E}x = \mathsf{T}$ if $xy = \mathsf{T}$ for some y and $\mathsf{E}x = \bot$ if $xy = \mathsf{T}$ for no y in standard models. Its axiomatization is a syntactical approximation of this. Now define:

$$\begin{array}{lcl} x \circ y & \equiv & \lambda z.x(yz) \\ \chi & \equiv & \lambda x.\lambda z.\mathsf{if}[xz, \mathsf{T}, \bot] \\ x \to y & \equiv & \mathsf{if}[x, y, \mathsf{F}] = \mathsf{if}[x, \mathsf{T}, \mathsf{F}] \end{array}$$

The axioms on E read:

ET	$\mathsf{ET} = \mathsf{T}$
EB	$\mathsf{E}\bot = \bot$
EX	$\mathsf{E}x = \mathsf{E}(\chi x)$
EC	$\mathsf{E}(x \circ y) \to \mathsf{E}x$

Axioms ET and EB are natural since $\mathsf{T}x = \mathsf{T}$ and $\bot x = \bot$ are axioms of MT. EX says that $\mathsf{E}x$ does not care about the value of xy if $xy \neq \mathsf{T}$. EC says that if $x(yz) = \mathsf{T}$ for some z then $xw = \mathsf{T}$ for some w.

4.5. Quantification axioms

Define:

$$
\begin{array}{lcl}
!x & \equiv & \mathsf{if}[x, \mathsf{T}, \mathsf{T}] \\
\ddot{\neg} x & \equiv & \mathsf{if}[x, \mathsf{F}, \mathsf{T}] \\
\ddot{\exists} p & \equiv & \approx(p(\varepsilon p)) \\
\ddot{\exists} x\colon \mathcal{A} & \equiv & \ddot{\exists} \lambda x.\mathcal{A} \\
\ddot{\forall} x\colon \mathcal{A} & \equiv & \ddot{\neg}\ddot{\exists} x\colon \ddot{\neg}\mathcal{A} \\
\varepsilon x\colon \mathcal{A} & \equiv & \varepsilon \lambda x.\mathcal{A} \\
\ddot{\forall} p & \equiv & \ddot{\forall} x\colon px
\end{array}
$$

Note that $\forall$, $\exists$, and $\neg$ are part of the syntax of ZFC+SI whereas $\ddot{\forall}$, $\ddot{\exists}$, and $\ddot{\neg}$ are terms of MT. The quantifier axioms depend on a term ψ which will be defined in Section 4.6. Recall $\hat{\psi} \equiv \{x \in \mathcal{M} \mid \psi x = \mathsf{T}\}$. In canonical models, for maps p, we have

$$
\begin{array}{lcll}
\ddot{\forall} p & = & \mathsf{T} & \text{if } \forall x {\in} \hat{\psi}\colon px = \mathsf{T} \\
\ddot{\forall} p & = & \bot & \text{if } \exists x {\in} \hat{\psi}\colon px = \bot \\
\ddot{\forall} p & = & \mathsf{F} & \text{otherwise}
\end{array}
$$

Hence, $\ddot{\forall}$ expresses universal quantification over $\hat{\psi}$. Likewise, $\ddot{\exists}$ expresses existential quantification over $\hat{\psi}$. The quantification axioms read:

Elim	$(\ddot{\forall} x\colon px) \wedge \psi y \rightarrow py$
Ackermann	$\varepsilon x\colon px = \varepsilon x\colon (\psi x \wedge px)$
StrictE	$\psi(\varepsilon x\colon px) = \ddot{\forall} x\colon !(px)$
StrictA	$!(\ddot{\forall} x\colon px) = \ddot{\forall} x\colon !(px)$

Above, p, x, and y are variables of MT.

Example 4.5.1. As we shall see, elements of $\hat{\psi}$ are wellfounded w.r.t. $\hat{\psi}$ (see Example 4.2.1 for the definition of wellfoundedness with respect to a set). This allows to use elements of $\hat{\psi}$ to represent sets of ZFC. We define the set represented by $x \in \hat{\psi}$ thus:

$$
\begin{array}{lcll}
Z[\mathsf{T}] & \equiv & \emptyset & \\
Z[x] & \equiv & \{Z[xz] \mid z \in \hat{\psi}\} & \text{if } x \neq \mathsf{T}
\end{array}
$$

For the usual model V_σ of ZFC in ZFC+SI and canonical models $\mathcal{M}$ of MT we have $V_\sigma = \{Z[x] \mid x \in \hat{\psi}\}$ (c.f. [3, Appendix A.4]) so all sets of ZFC are representable by wellfounded maps $x \in \hat{\psi}$. Now define:

$$
\begin{array}{lcl}
x \ddot{\Rightarrow} y & \equiv & \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \mathsf{F}], \mathsf{if}[y, \mathsf{T}, \mathsf{T}]] \\
x \ddot{\Leftrightarrow} y & \equiv & \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \mathsf{F}], \mathsf{if}[y, \mathsf{F}, \mathsf{T}]] \\
x \ddot{\in} y & \equiv & \mathsf{if}[y, \mathsf{F}, \ddot{\exists} z\colon x \ddot{=} yz] \\
x \ddot{=} y & \equiv & \ddot{\forall} z\colon (z \ddot{\in} x \ddot{\Leftrightarrow} z \ddot{\in} y)
\end{array}
$$

For $x, y \in \hat{\psi}$ we have $(x \ddot{\in} y) = \mathsf{T}$ iff $Z[x] \in Z[y]$ and $(x \ddot{=} y) = \mathsf{T}$ iff $Z[x] = Z[y]$. An equivalent definition of $\ddot{=}$ reads:

$$x \ddot{=} y \quad \equiv \quad \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \mathsf{F}], \mathsf{if}[y, \mathsf{F}, (\ddot{\forall} u \ddot{\exists} v\colon xu \ddot{=} yv) \wedge (\ddot{\forall} v \ddot{\exists} u\colon xu \ddot{=} yv)]]$$

The latter formulation of $\ddot{=}$ resembles that of $\stackrel{\omega}{=}$ in Example 4.2.1.

Using $\ddot{\in}$, $\ddot{\neg}$, $\ddot{\Rightarrow}$, and $\ddot{\forall}$ we may now express all wellformed formulas of ZFC in MT. All closed theorems of ZFC are satisfied by the standard model $\mathcal{M}$ of MT (Theorem 2.1.2). As a conjecture (Conjecture 2.1.3), closed theorems of ZFC+¬SI are provable in MT.

Example 4.5.2. As a continuation of Example 4.2.1, define

$$x + y \quad \equiv \quad \mathsf{if}[x, y, (x\mathsf{T}) + y']$$

Having a quantifier in MT allows to prove in MT e.g. that the term

$$\ddot{\forall} x, y\colon x + y \stackrel{\omega}{=} y + x$$

equals T. The proof involves a proof of $\ddot{\forall} y\colon x + y \stackrel{\omega}{=} y + x$ by induction in x (or, more precisely, a proof of $\omega \preceq \lambda x. \ddot{\forall} y\colon x + y \stackrel{\omega}{=} y + x$ by Min). The proof requires the ability to apply induction to a statement which contains both a quantifier ($\ddot{\forall}$) and recursive programs ($+$ and $\stackrel{\omega}{=}$) and thus requires the ability to mix recursive programs and quantifiers as is possible in MT.

4.6. The definition of ψ

We conclude the presentation of axioms by defining ψ. Like in Section 3.8 let $\mathcal{M}$ be any standard model of MT. We first define some auxiliary concepts.

Definition 4.6.1.
(a) $a \in \mathcal{M}$ is a *characteristic map* if $a \in \mathcal{F}$ and $ax \in \{\mathsf{T}, \bot\}$ for all $x \in \mathcal{M}$.
(b) $\mathrm{Dom}[a] = \{x \in \mathcal{M} \mid ax = \mathsf{T}\}$
(c) $a \in \mathcal{M}$ is a *characteristic map of $S \subseteq \mathcal{M}$* if a is a characteristic map and $S = \mathrm{Dom}[a]$.

In Example 4.2.1 we referred to ω as "the characteristic map of the set of natural number maps".

Definition 4.6.2.
(a) $\sqcup \equiv \lambda f y.\, \mathsf{E}x.\, fxy$
(b) $x \,!\, y \equiv \mathsf{if}[\, x \,,\, y \,,\, \bot \,]$
(c) $\mathsf{D} \equiv \lambda x.\, \mathsf{if}[\, x \,,\, \mathsf{T} \,,\, \mathsf{T} \,]$
(d) $f \,/\, g \equiv \mathsf{if}[\, f \,,\, \mathsf{T} \,,\, \lambda x.\, gx \,!\, (fx \,/\, g) \,]$

The intuitions behind Definition 4.6.2 are as follows. $\sqcup$ satisfies $\mathrm{Dom}[\sqcup a] = \cup_{x \in \mathcal{M}} \mathrm{Dom}[ax]$. $x \,!\, y$ is y guarded by x in the sense that if $x = \mathsf{T}$ then $x \,!\, y = y$ and if $x \neq \mathsf{T}$ then $x \,!\, y = \bot$. $\mathsf{D}x$ is true if x is "defined", i.e. if $x \neq \bot$.

$f\ /\ g$ is a kind of "transitive restriction" of the function f to the domain $G = \mathrm{Dom}[g]$ in the following sense: Suppose $x_1, \ldots, x_n \in G$ and $fx_1 \cdots x_n \in \mathcal{F}$ then

$$(f\ /\ g)x_1 \cdots x_n y \sim \begin{cases} fx_1 \cdots x_n y & \text{if } y \in G \\ \perp & \text{otherwise} \end{cases}$$

where root equality $u \sim v$ was defined in Section 3.6. One may also think of $f\ /\ g$ as a projection in the sense that $(f\ /\ g)\ /\ g = f\ /\ g \preceq_{\mathcal{M}} f$. The intuitions given above hold in all standard models $\mathcal{M}$, c.f. Section 11.7 and Fact 13.3.2(b). $f\ /\ g$ equals $\Downarrow_G f$ of [3].

In Section 3.8 we defined $\hat{\psi} = \{x{\in}\mathcal{M} \mid \psi x = \mathsf{T}\}$. We now go on to define ψ. To do so we need to define a number of auxiliary terms. In $\mathcal{M}$, the terms ψ, s, S, $\bar{S}$, P, Q, R, R_1, and R_0 will have the following properties:

$$\begin{array}{lll}
y \in \hat{\psi} & \text{iff} & \psi y = \mathsf{T} & \text{by definition of } \hat{\psi} \\
y \in \hat{\psi} & \text{iff} & say = \mathsf{T} & \text{for some } a \\
y \in \hat{\psi} & \text{iff} & Ssay = \mathsf{T} & \text{for some } a \\
y \in \hat{\psi} & \text{iff} & \bar{S}s\psi ay = \mathsf{T} & \text{for some } a \\
y \in \hat{\psi} & \text{if} & Py = \mathsf{T} & \\
y \in \hat{\psi} & \text{if} & Q(sa)y = \mathsf{T} & \text{for some } a \\
y \in \hat{\psi} & \text{if} & Rs\psi bcy = \mathsf{T} & \text{for some } b,\ c \\
y \in \hat{\psi} & \text{if} & \psi c = \mathsf{T} \text{ and } R_1 s\psi bc = \mathsf{T} \text{ and } R_0 s\psi bcy = \mathsf{T} & \text{for some } b,\ c
\end{array}$$

For all a, b, and c, we will have that ψ, sa, Ssa, $\bar{S}s\psi a$, P, $Q(sa)$, $Rs\psi bc$, and $R_0 s\psi bc$ are characteristic maps. For all a, $\mathrm{Dom}[sa]$ will be *essentially σ-small* in the sense that there exists a set $A \subseteq \hat{\psi}$ of cardinality less than σ such that $\mathrm{Dom}[sa] = \{w \in \hat{\psi} \mid \exists u{\in}A\colon u \preceq w\}$. The function Q mentioned above will have the property that if $\mathrm{Dom}[v]$ is essentially σ-small and if $Qvy = \mathsf{T}$ then y is wellfounded. See Sections 11–13 for proofs.

The definition of ψ and the auxiliary terms reads:

Definition 4.6.3.
(a) $\psi \equiv \sqcup s$
(b) $s \equiv \mathsf{Y} S$
(c) $S \equiv \lambda f.\, \bar{S} f(\sqcup f)$
(d) $\bar{S} \equiv \lambda f \theta a.\, \mathsf{if}[\, a\, ,\, P\, ,\, \mathsf{if}[\, a\mathsf{T}\, ,\, Q(f(a\mathsf{F}))\, ,\, Rf\theta(a\mathsf{T})(a\mathsf{F})\,]\,]$
(e) $P \equiv \lambda y.\, \mathsf{if}[\, y\, ,\, \mathsf{T}\, ,\, \perp\,]$
(f) $Q \equiv \lambda v.\, \mathsf{D}v\ !\ \lambda y.\, \ddot{\forall} z.\, v(y(z\ /\ v))$
(g) $R \equiv \lambda f \theta bc.\, \theta c\ !\ R_1 f\theta bc\ !\ R_0 f \theta bc$
(h) $R_1 \equiv \lambda f\theta bc.\, \ddot{\forall} z.\, \mathsf{D}(f(b(cz\ /\ \theta)))$
(i) $R_0 \equiv \lambda f \theta bcy.\, \mathsf{E}z.\, (\theta z\ !\ f(b(cz\ /\ \theta))y)$

The definition of ψ replaces the ϕ-axioms of MT_0 (ten axioms and one inference rule). The definition of ψ in MT corresponds to the following in ZFC: the null set axiom, the pair set axiom, the power set axiom, the union set axiom, the axiom of replacement, the axiom of infinity, and the axiom of restriction.

Note that $s = \mathsf{Y}S = S(\mathsf{Y}S) = Ss = \bar{S}s(\sqcup s) = \bar{S}s\psi$. Hence, in (d-i) above one may think of f and θ as s and ψ, respectively.

We have $\psi y = \sqcup sy = \mathsf{E}a.\, say$. So, in $\mathcal{M}$, y is wellfounded iff $say = \mathsf{T}$ for some a.

Example 4.6.4. We now prove that T is wellfounded by proving $s\mathsf{TT} = \mathsf{T}$. To do so we first prove $say = \bar{S}s\psi ay$ as follows: $say = (\mathsf{Y}S)ay = S(\mathsf{Y}S)ay = Ssay = \bar{S}s(\sqcup s)ay = \bar{S}s\psi ay$. Then we note that $s\mathsf{TT} = \bar{S}s\psi\mathsf{TT} = P\mathsf{T} = \mathsf{T}$.

We now prove that $\lambda u.\mathsf{T}$ is wellfounded by proving that $s(\mathsf{T}::\mathsf{T})(\lambda u.\mathsf{T}) = \mathsf{T}$: Define $b::c \equiv \lambda z.\, \mathsf{if}[\, z\, ,\, b\, ,\, c\,]$. We have $(b::c)\mathsf{T} = b$ and $(b::c)\mathsf{F} = c$. Hence, if $\mathsf{D}(sc) = \mathsf{T}$ then $s(\mathsf{T}::c)y = \bar{S}s\psi(\mathsf{T}::c)y = Q(sc)y = \ddot{\forall} z.\, sc(y(z\ /\ sc))$. Hence, $s(\mathsf{T}::\mathsf{T})(\lambda u.\mathsf{T}) = \ddot{\forall} z.\, s\mathsf{T}((\lambda u.\mathsf{T})(z\ /\ s\mathsf{T})) = s\mathsf{TT} = \mathsf{T}$.

Recall that we defined $0 = \mathsf{T}$, $1 = \lambda u.\mathsf{T}$, $2 = \lambda uv.\,\mathsf{T}$, and so on in Example 4.2.1. We have now proved that 0 and 1 are wellfounded. Furthermore, $s(\mathsf{T}::(\mathsf{T}::\mathsf{T}))2 = \mathsf{T}$ proves that 2 is wellfounded. We may go on and prove that 3 is wellfounded and so on into the transfinite. For the complete development, see the proof of Theorem 11.1.3 in Section 13.

The ability of MT to model ZFC stems from several sources. First, the quantification axioms reference ψ in a way which forces MT quantifiers to quantify over the universe $\hat{\psi} = \{x{\in}\mathcal{M} \mid \psi x = \mathsf{T}\}$. Second, as shown in Example 4.6.4, recursive use of $s = \mathsf{Y}S$ populates $\hat{\psi}$, putting a lower bound on the size of the universe. Third, the minimality of Y permits a kind of transfinite induction over $\hat{\psi}$, putting an upper bound on the size of the universe. Fourth, Ext plays a marginal but essential role in that it forces $\preceq$ to be a partial order.

When modelling ZFC in MT, one may define $\ddot{\in}$, $\ddot{\neg}$, $\ddot{\Rightarrow}$, and $\ddot{\forall}$ as in Section 4.5. Then, to prove e.g. the power set axiom one may find an MT term $\mathcal{P}(x)$ such that $\mathcal{P}(x)$ represents the power set of the set represented by x. Then one may prove $\ddot{\forall} x, y\colon (y\ddot{\in}\mathcal{P}(x) \ddot{\Leftrightarrow} \ddot{\forall} z\colon (z\ddot{\in}y \ddot{\Rightarrow} z\ddot{\in}x))$ and $\ddot{\forall} x\colon \psi(\mathcal{P}(x))$ from which the power set axiom is easy to prove. Proving $\ddot{\forall} x\colon \psi(\mathcal{P}(x))$ makes use of the second point above by using that ψ makes the universe big enough to contain $\mathcal{P}(x)$. But it also uses the third point above because the proof requires a kind of transfinite induction in x and thereby uses that the universe is so small that all sets have powersets.

5. Comparing MT and MT_0

We now deepen the comparison between MT and MT_0 which was only hinted at in the introduction.

5.1. Axioms and rules of MT_0

MT_0 consisted of only three groups of axioms, using constructors T, $\perp$, if, ε, and ϕ (the innocuous $\|$ could have been added as well):

- Elementary axioms and rules (same as for MT).

- Quantification axioms.
- The ϕ-axioms

The ϕ-axioms comprise the three wellfoundedness axioms, the seven construction axioms, and the inference rule of transfinite induction of [8]. The role of the ϕ-axioms was to force ϕ to behave as the characteristic function of the universe Φ of wellfounded sets. Most of the ϕ-axioms were easy (e.g. $\phi\mathsf{T} = \mathsf{T}$), two were highly non-obvious, and the rule was just expressing wellfoundation. But in fact all were (independent) instances of a single recursive set-theoretical equation $\Phi = \mathcal{F}_\sigma(\Phi)$ on Φ which involves an inaccessible cardinal σ and will be recalled in Definition 8.3.2. This recursive equation was the formalization in $\mathcal{M}$ of the main intuition behind map theory, which was, as for Church ([4, 5]) to have a universe whose primitives were the notions of "functions and application" instead of "sets and membership".

5.2. Axioms and rules of MT

The next intended step was hence to succeed to reflect the equation on Φ at the axiomatic level. This resulted in the present MT, where ϕ is now replaced by the MT-term ψ, whose definition requires the supplementary constructs Y and E. E is easy to axiomatize and to model, so the real technical cost is the addition of the extra rules Ext, Mono, and Min, and the (hard) proof that ψ truly represents Φ. Mono and Min force the constant Y to behave, at the level of the syntax, as a fixed point operator which is minimal w.r.t. the syntactic order $\preceq$.

MT consists of five groups of axioms and rules, and uses constructors T, $\perp$, if, $\|$, Y, E, and ε:

- Elementary axioms and rules plus Y is a fixed point operator.
- Axioms on E.
- Axioms of Monotonicity (Mono) and Minimality (Min).
- The inference rule of Extensionality (Ext).
- Quantification axioms

The quantification axioms are the same[2] as for MT_0 except that the ϕ of MT_0 is replaced by ψ in MT; but the proof that these axioms can be satisfied is much harder and very different, c.f. Section 8.4.

[2]This variant, containing 4 axioms, already appears in [3, Appendix C], and is equivalent to the original set of 5+1 axioms where the five ones were stated in [8] and the sixth one, as pointed out by Thierry Vallée, was used but not stated in [8]

5.3. Proof theoretical strength

MT_0 can prove neither $\mathrm{SI} = \mathsf{T}$ nor $(\neg\mathrm{SI}) = \mathsf{T}$ since it can be consistently extended by either one. In contrast, $(\neg\mathrm{SI}) = \mathsf{T}$ is conjectured to be provable in MT (Conjecture 2.1.3). Furthermore, MT can prove more pure lambda terms equivalent (e.g. $F_2 = F_3$, c.f. Example 4.3.3, which we conjecture is not provable in MT_0). MT is very likely stronger than MT_0:

Conjecture 5.3.1. *If $\mathcal{A} = \mathcal{B}$ is provable in* MT_0 *and if $\mathcal{A}'$ and $\mathcal{B}'$ arise from $\mathcal{A}$ and $\mathcal{B}$, respectively, by replacing all occurrences of ϕ by ψ, then $\mathcal{A}' = \mathcal{B}'$ is provable in MT.*

If $(\neg\mathrm{SI}) = \mathsf{T}$ is provable in MT then Conjecture 2.1.3 follows from Conjecture 5.3.1 and Theorem 2.1.4. Conjecture 5.3.1 is true if the ϕ-axioms of MT_0 are provable in MT. Less support exists for Conjecture 5.3.1 than for Conjecture 2.1.3.

5.4. Models of MT *versus models of* MT_0

Now let $\kappa \geq \omega$ be a regular cardinal. Modelling ε requires an inaccessible $\sigma < \kappa$, while modelling the other constructs only requires $\kappa \geq \omega$. Sections 7–9 recall the notions of *κ-Scott semantics* and *κ-continuity* and introduce a number of concepts which have the following names and forms:

Underlying set	$\mathcal{M}^0$
κ-Scott domain	$\mathcal{M}^1 = (\mathcal{M}^0, \leq)$
κ-premodel	$\mathcal{M}^2 = (\mathcal{M}^1, A, \lambda)$
Canonical κ-premodel	A particular κ-premodel
MT_0 standard κ-σ-quasimodel	$\mathcal{M}^3_0 = (\mathcal{M}^2, \mathsf{T}, \mathsf{if}, \varepsilon, \phi)$
MT_0 canonical κ-σ-quasimodel	same as above where $\mathcal{M}^2$ is canonical
MT standard κ-σ-quasimodel	$\mathcal{M}^3 = (\mathcal{M}^2, \mathsf{T}, \mathsf{if}, \varepsilon, \bot, \mathsf{Y}, \Vert, \mathsf{E})$
MT canonical κ-σ-quasimodel	same as above where $\mathcal{M}^2$ is canonical

We use $\mathcal{M}$ to denote any one of $\mathcal{M}^0$, $\mathcal{M}^1$, $\mathcal{M}^2$, $\mathcal{M}^3_0$, and $\mathcal{M}^3$, depending on context. For MT_0 (MT) κ-σ-quasimodels we drop "quasi" when the quasimodel satisfies MT_0 (MT), and we drop σ when σ is understood.

As stated in Theorem 8.3.5, when σ is inaccessible, all MT_0 standard κ-σ-quasimodels satisfy MT_0. The main result of the present paper is that when σ is the first inaccessible, all MT canonical κ-σ-quasimodels satisfy MT. Thus, satisfying MT is harder than satisfying MT_0: One both needs canonicity and needs that σ is the *first* inaccessible.

5.5. Levels of difficulty of the groups of axioms

We now move on to consider the "difficulty" of the group of elementary axioms, the group of E-axioms, and so on. "Difficulty" is a multi-dimensional notion. When looking at groups of axioms it is natural to ask oneself the following questions:

- *Naturality.* Are the axioms intuitive or "natural" in some sense, i.e. is there a natural or simple or motivated intuition behind?

- *Strength*, here in the following sense: where is the existence of an inaccessible cardinal σ required? Which axioms can we model at not cost? meaning here that $\kappa = \omega$ would be enough, and/or that they can be modeled in all premodels?

- *Conceptual hardness.* Do we need to introduce original and/or high level tools for modeling them?

- *Technical hardness.* Do we need difficult computations?

The Elementary Axioms are natural (if one is used to λ-calculus) and can be modelled at not cost (i.e. in any κ-premodel, $\kappa \geq \omega$); the four E-axioms are at first look purely technical, in fact they are easy from all the above points of view, the reason being that they are just four instances of a single, simple intuition, which allows us to model them easily and at "no cost".

Of course, all the axioms of MT are natural in some sense, since they were designed from semantical and computational intuitions (c.f. [8]), but this naturality can be lost when approximating the ideas through formalization.

Mono and Min are semantically natural (syntactically a little less because of the definition of $\preceq$), and can be modelled at no cost in terms of strength ($\kappa \geq \omega$), but fixing a syntactic definition of the order induces a technical cost which drastically reduces the class of possible models, c.f. Section 7.5.

Concerning the quantifier axioms, it is interesting to note that replacing ϕ of MT_0 by ψ in MT induces no change in strength in the sense that an inaccessible is used (and apparently needed) for modelling MT_0 as well as MT, but that they are conceptually somewhat harder for MT (because they refer to the defined ψ which incorporates the ϕ-axioms) and technically much harder (c.f. Sections 11–13).

We pursue and summarize the comparison between MT and MT_0 in the following section.

5.6. Models of subsystems of MT

Recall from Section 3.2 that $\mathrm{MT}_{\mathrm{def}}$ is the version of MT where Y and $\perp$ are omitted and are replaced by $\mathsf{Y}_{\mathrm{Curry}}$ and $\perp_{\mathrm{Curry}}$, respectively. This concerns mainly $\mathsf{A3}$, $\mathsf{I3}$, QND, Mono, Min, Elim, $\mathsf{Ackermann}$, and $\mathsf{StrictE}$ where Y or $\perp$ appear explicitly or implicitly.

We now introduce two subsystems of MT for which we keep the same syntax and which hence have the same terms. First, MT^{-} is MT from which the quantifier axioms are removed. Second, MT^{--} is MT^{-} from which Ext, Mono, and Min are removed. We have:

- Modeling MT^{--} can be done from *any* κ-premodel, $\kappa \geq \omega$.

- Modeling MT^{-} can be done from any *canonical* κ-premodel, $\kappa \geq \omega$

- Modeling MT and $\mathrm{MT}_{\mathrm{def}}$ can be done from any *canonical* κ-premodel, $\kappa > \sigma$, using the *first* inaccessible σ.
- Modelling MT_0 can be done from *any* κ-premodel, $\kappa > \sigma$, using *any* inaccessible σ.

6. Approach

The aim of the rest of this paper is to show that some of the models of MT_0 in [3] (the canonical ones) can be adjusted into models of MT.

Models have to be adjusted because MT_0 and MT have different syntax. Among others, MT_0 does not have the E of MT and MT does not have the ϕ of MT_0. To go from a model of MT_0 to a model of MT one must first delete the interpretation of ϕ and then add interpretations of $\perp$, Y, $\|$, and E. One then has to check that all axioms and inference rules of MT are satisfied under adequate hypotheses on the model of MT_0 one started from. The most difficult part will be to show that the quantifier axioms can be satisfied when replacing the construct ϕ of MT_0 by the defined term ψ of MT.

We delay as far as possible the specialization to canonical models. Working like this first increases our conceptual understanding, but will moreover facilitate for the future the design of consistent variations of Map Theory that users might wish to introduce.

We work in ZFC+SI where SI asserts the existence of an inaccessible ordinal.

κ-premodels

For all regular cardinals $\kappa \geq \omega$, [3] defines the notion of *κ-premodels* of map theory. The notion of a κ-premodel reflects the basic intuitions which were behind map theory. In particular, all the constructors of MT and MT_0, namely $\perp$, T, if, $\|$, Y, E, ε, and ϕ have natural interpretations as functions in all premodels, as we will see.

The definition of a κ-premodel (c.f. Section 8.2) involves Scott's semantics, which is the most classical mathematical way for modeling type-free λ-calculus. Looking for a model as powerful as a model of ZFC, we will have to use generalizations (weakenings) of Scott's semantics, called here κ-Scott semantics (where Scott semantics is the case $\kappa = \omega$). Until specified otherwise, κ is any regular cardinal such that $\kappa \geq \omega$. κ-Scott semantics will be treated in more detail in Section 7.

A κ-premodel has a domain which is a partially ordered set plus some structure which indicates how to interpret lambda abstraction, functional application, T, $\perp$, and arbitrary κ-continuous constructs. A κ-premodel is a reflexive object in κ-Scott semantics, which furthermore satisfies a simple domain equation Eq_κ (c.f. Section 8.1). As explained in [3, Section 3], the role of Eq_κ is to force standard models to satisfy the elementary axioms and rules of MT_0 (or MT).

Canonical κ-premodels

For all regular cardinals $\kappa \geq \omega$, [3, Section 8] constructs a *canonical κ-premodel of Map Theory.* This was done in [3] in order to prove the existence of κ-premodels, which was the first step of the consistency proof of MT_0. The next step in [3] was to prove that all κ-premodels, $\kappa > \sigma$, σ inaccessible, could be enriched to MT_0 models.

For each regular $\kappa \geq \omega$ there are many κ-premodels but only one canonical one. The canonical κ-premodel is the minimal solution to Eq_κ. It is computationally adequate for the computational part of MT_0 in the sense described in Section A.5.

In the present paper, canonical κ-premodels play an even more crucial role than in [3], since it is only the canonical κ-premodels, $\kappa > \sigma$, σ the first inaccessible, that we prove can be enriched to MT models.

Canonical κ-premodels admit an elementary and direct construction which we will recall in the sequel. By "elementary" we mean in particular that the construction uses no category theory which is crucial for being able to work in practice with the model.[3]

MT_0 *standard κ-models*

Given an inaccessible ordinal σ and a regular cardinal $\kappa > \sigma$, [3, Sections 4 and 7] defines a method for enriching any κ-premodel to a model of MT_0. The method adds interpretations of T, if, ε, and ϕ to the κ-premodel in such a way that all the axioms and inference rules of MT_0 are satisfied. We shall refer to models constructed this way as *MT_0 standard models.*

The function interpreting if is κ-continuous for all regular $\kappa \geq \omega$. The function interpreting ε is only κ-continuous if $\kappa > \sigma$ for some inaccessible ordinal σ.

MT *standard κ-quasimodels*

Later, we adjust MT_0 standard models by deleting the interpretation of ϕ and adding interpretations of $\perp$, Y, $\|$, and E by κ-continuous functions ($\kappa \geq \omega$) in an obvious way. We shall refer to the result of that as MT *standard quasimodels.* Such MT standard quasimodels need not satisfy all axioms and inference rules of MT but they do satisfy some of them.

The MT *canonical κ-model $\mathcal{M}_\kappa$*

Given an inaccessible ordinal σ and a regular cardinal $\kappa > \sigma$, one may enrich the canonical κ-premodel into an MT standard quasimodel. If σ is the first inaccessible ordinal then the quasimodel can be proved to satisfy all axioms and inference rules of MT. Hence, we shall refer to the model constructed this way as the MT canonical κ-model $\mathcal{M}_\kappa$.

[3] A classical but far less feasible alternative would have been to build the canonical model as an inverse limit of an ordinal sequence, similar to what Scott did with his first model $\mathcal{D}_\infty$ in the case $\kappa = \omega$.

Summary

To summarize, the notions of κ-premodels and canonical κ-premodels are the same for MT and MT_0. The notions of models of MT and MT_0 differ slightly due to differences in syntax. Furthermore, for canonical models of MT we assume that σ is the *first* inaccessible.

7. The κ-Scott semantics

7.1. Notation

Let ω denote the set of finite ordinals (i.e. the set of natural numbers).

For all sets G let $G^{<\omega}$ denote the set of tuples (i.e. finite sequences) of elements of G. Let $\langle\,\rangle$ denote the empty tuple.

For all sets G let G^{ω} denote the set of infinite sequences of elements of G.

Let $f\colon G \to H$ denote that f is a total function from G to H.

Given any partially ordered or preordered set $(R, \leq)$ and $S \subseteq R$, we let $\uparrow S$ and $\downarrow S$ be respectively the upward and downward closure of S for $\leq$ in R.

We shall say that a set G is *κ-small* if G has cardinality smaller than κ. Let $\mathcal{P}(G)$ denote the power set of G and let $\mathcal{P}^{\kappa}(G)$ denote the set of κ-small subsets of G.

7.2. κ-Scott semantics

The *κ-Scott category* is the Cartesian closed category whose objects are the κ-Scott domains and morphisms the κ-continuous functions. As κ grows there are more and more κ-Scott domains and κ-continuous functions.

The theory of Scott domains (case $\kappa = \omega$) is well known, and its κ-analogue was developed in full details in [3]. For the reader familiar with Scott domain theory, passing from Scott to κ-Scott is straightforward and just amounts to changing everywhere "finite" by "κ-small". The regularity of κ is essential. We recall some key definitions and results in the following.

κ-Scott semantics was first used around 1987-89 in [6, 7] and was used independently in [3], but Scott was aware of the notion from the beginning, and κ-Scott semantics appeared in German lecture notes by Scott which are probably lost now.

7.3. κ-Scott domains

Let $(\mathcal{D}, \leq)$ be a partially ordered set (p.o. for short). A subset S of $\mathcal{D}$ is κ-directed if all its κ-small subsets have an upper bound in S. The p.o. $(\mathcal{D}, \leq)$ is a *κ-Scott domain* if it has a least (or bottom) element, is such that all κ-directed and all upper-bounded subsets have sups (suprema), and finally if it is κ-algebraic as defined below. As κ grows there are more and more κ-Scott domains. The simplest example of κ-Scott domains is that of the full powerset $(\mathcal{P}(D), \subseteq)$ of some set D, which is a κ-Scott domain for all κ. The domain underlying the canonical model $\mathcal{M}$ of MT_0 will not be a full power set, but will still be a set of sets, ordered by inclusion.

An element u of $\mathcal{D}$ is *compact* (resp. *prime*) if, whenever $u \leq \sup(S)$ for some κ-directed (upper bounded) set S, then $u \leq v$ for some $v \in S$. $\mathcal{D}$ is *κ-algebraic* if for every $u \in \mathcal{D}$ the set of compact elements below u is κ-directed, and has u as its sup. A κ-Scott domain is *prime-algebraic* if each element of $\mathcal{D}$ is the sup of the primes below it.

Definition 7.3.1. *$\mathcal{K}(\mathcal{D})$ is the set of compact elements of the κ-Scott domain $\mathcal{D}$.*

$(\mathcal{P}(D), \subseteq)$ and $\mathcal{M}$ are prime algebraic κ-Scott domains. The compact elements of $(\mathcal{P}(D), \subseteq)$ are the κ-small subsets of D and its primes are the singletons. In the case of $\mathcal{M}$, compact elements are downward closures of adequate κ-small subsets of D, while primes are downward closures of singletons.

7.4. κ-continuous functions

A function between two κ-Scott domains is *κ-continuous* if it is monotone and commutes with all sups of non-empty κ-directed sets.

Given κ-Scott domains $\mathcal{D}, \mathcal{D}'$ we use $[\mathcal{D} \rightarrow_\kappa \mathcal{D}']$ to denote the κ-Scott domain whose carrier set is the set of κ-continuous functions from $\mathcal{D}$ to $\mathcal{D}'$ ordered pointwise. As κ grows there are more and more κ-continuous functions.

7.5. Syntactic monotonicity

Monotonicity, which was part of the founding intuitions behind map theory (c.f. [8]) comes for free in models living in Scott's semantics, but of course only relative to the order $x \preceq_{\mathcal{M}} y$ of the underlying domain.

In MT, monotonicity is explicitly required in the axiomatisation (by Mono), but necessarily for a syntactic order. We shall prove (Theorem 8.5.2) that the syntactic order $x \preceq y$ coincides with the model order $x \preceq_{\mathcal{M}} y$ in canonical κ-quasimodels and we shall conclude (Corollary 8.5.3) that the canonical κ-quasimodel satisfies Mono.

The Mono rule was not part of MT_0.

7.6. κ-open sets

$G \subseteq \mathcal{D}$ is κ-open if $G = \uparrow K$ for some set $K \subseteq \mathcal{K}(\mathcal{D})$. Equivalently, G is κ-open if $G = \uparrow G$ and whenever G contains $\sup(S)$ for some directed set S then it contains some element of S. This defines a topology, the *κ-Scott topology*, and the κ-continuous functions, as defined above, are exactly the functions which are continuous with respect to this topology. Finally, it is easy to check, and crucial to note, that the intersection of a κ-small family of κ-open sets is still κ-open.

$G \subseteq \mathcal{D}$ is *essentially κ-small* if $V \subseteq G \subseteq \uparrow V$ for some κ-small V. It follows that G is an essentially κ-small open set if and only if $G = \uparrow K$ for some κ-small $K \subseteq \mathcal{K}(\mathcal{D})$.

7.7. Reflexive objects and models of pure λ-calculus

By definition a *reflexive object* of the κ-category is a triple $(\mathcal{M}, A, \lambda)$ where $\mathcal{M}$ is a κ-Scott domain and $A: \mathcal{M} \to [\mathcal{M} \to_\kappa \mathcal{M}]$ and $\lambda: [\mathcal{M} \to_\kappa \mathcal{M}] \to \mathcal{M}$ are two morphisms such that $A \circ \lambda$ is the identity. This gives a model of untyped λ-calculus, i.e. of rules Trans, Sub, Gen, and A2, when we use A and λ to interpret the pure λ-terms, in the standard way (c.f. Section 9.6).

Most of the time $A(u)(v)$ will be abbreviated as uv, and $uv_1 \cdots v_n$ will mean $((\cdots((u)v_1)\cdots)v_n)$. Furthermore, $u\bar{w} \equiv uw_1 \cdots w_n$ if $\bar{w} = w_1 \cdots w_n$ $(n \geq 0)$.

All n-ary κ-continuous functions, $n \in \omega$, can be internalized in $\mathcal{M}$: for any such f there is an element $v \in \mathcal{M}$ such that $f(u_1, \ldots, u_n) = vu_1 \cdots u_n$ for all $u_1, \ldots, u_n \in \mathcal{M}$. In the case $n = 1$ we can take $v = \lambda(f)$.

7.8. Tarski's minimal fixed point operators

Let $\mathcal{D}$ be a κ-Scott domain and let $f \in [\mathcal{D} \to_\kappa \mathcal{D}]$. If $\kappa = \omega$ then f has a fixed point and even has a minimal such. That does not always hold for $\kappa > \omega$. As an example, $(\omega, \leq)$ is a κ-Scott domain for all regular $\kappa > \omega$ but the successor function has no fixed point.

We now turn to sufficient conditions for the existence of fixed points. For all $f \in [\mathcal{D} \to_\kappa \mathcal{D}]$, $x \in \mathcal{D}$, and ordinals α define

$$f^\alpha(x) = \sup\{f(f^\beta(x)) \mid \beta \in \alpha\}$$

whenever the sup exists. Furthermore, define

$$\mathcal{Y}_{\mathrm{Tarski}}(f) \equiv f^\kappa(\bot)$$

We shall say that v is a pre-fixed point of f if $f(v) \preceq_\mathcal{M} v$.

Lemma 7.8.1. *If $f^\kappa(\bot)$ is defined then $f^\alpha(\bot)$ is defined for all α, $f^\alpha(\bot) = f^\kappa(\bot)$ for all $\alpha > \kappa$, f has a fixed (and pre-fixed) point, it has a unique minimal fixed (and pre-fixed) point, and $\mathcal{Y}_{\mathrm{Tarski}}(f) = f^\kappa(\bot)$ is that minimal fixed point.*

Proof of 7.8.1 Easy and classical.

Lemma 7.8.2.
(a) If $\kappa = \omega$ then $\mathcal{Y}_{\mathrm{Tarski}} \in [\mathcal{D} \to_\kappa \mathcal{D}] \to \mathcal{D}$ is total.
(b) If f has a fixed point then $\mathcal{Y}_{\mathrm{Tarski}}(f)$ is defined.
(c) If there are A, λ making $(\mathcal{D}, A, \lambda)$ a reflexive object then $\mathcal{Y}_{\mathrm{Tarski}}$ is total and κ-continuous.

Proof of 7.8.2
(a) Easy.
(b) Note that if f has a fixed point x then x is an upper bound for each $\{f(f^\beta(\bot)) \mid \beta \in \alpha\}$ which thus has a sup because $\mathcal{D}$ is κ-Scott.
(c) Totality follows from (b) because $\mathsf{Y}_{\mathrm{Curry}}\lambda(f)$ is a fixed point where $\mathsf{Y}_{\mathrm{Curry}}$ is definable when A and λ exist. Continuity can be proved by a rather standard proof (which can be found e.g. in [15]).

Now suppose $\mathcal{M} = (\mathcal{D}, A, \lambda)$ is a reflexive object and define $\mathsf{Y}_{\text{Tarski}} \in \mathcal{M}$ by

$$\mathsf{Y}_{\text{Tarski}} \equiv \lambda(\mathcal{Y}_{\text{Tarski}} \circ A)$$

where $\circ$ is composition.

Corollary 7.8.3.
(a) $\mathsf{Y}_{\text{Tarski}} u = u(\mathsf{Y}_{\text{Tarski}} u)$ (Y)
(b) $uv \preceq_{\mathcal{M}} v \Rightarrow \mathsf{Y}_{\text{Tarski}} u \preceq_{\mathcal{M}} v$ (Min)

Proof of 7.8.3 First note that $A \circ \lambda$ is the identity since $\mathcal{M}$ is reflexive so $\mathsf{Y}_{\text{Tarski}} u \equiv A(\lambda(\mathcal{Y}_{\text{Tarski}} \circ A))(u) = (\mathcal{Y}_{\text{Tarski}} \circ A)(u)$. Then (a) and (b) follow from the fact that $\mathcal{Y}_{\text{Tarski}}$ is the minimal fixed and pre-fixed operator.

$\mathsf{Y}_{\text{Tarski}}$ would hence be a good candidate for interpreting Y, provided the syntactic order $x \preceq y$ and the model order $x \preceq_{\mathcal{M}} y$ coincide, as they do when $\mathcal{M}$ is canonical.

8. Premodels of Map Theory

8.1. The domain equation Eq_κ

Given a κ-Scott domain $\mathcal{D}'$ we denote by $\mathcal{D}' \oplus_{\perp'} \{\mathsf{T}'\}$ the κ-Scott domain obtained by adding to $\mathcal{D}'$ an element T' which we decide to be incomparable to all the elements of $\mathcal{D}'$, and a bottom element $\perp'$ which we decide to be below T' and all the elements of $\mathcal{D}'$.

Definition 8.1.1. *Eq_κ is the domain equation $\mathcal{D} \simeq [\mathcal{D} \to_\kappa \mathcal{D}] \oplus_{\perp'} \{\mathsf{T}'\}$.*

Eq_κ asserts that the two sides of $\simeq$ are order isomorphic κ-Scott domains. It is the most natural semantic counterpart of rule QND, and the heart of the notion of a premodel. Proving the existence of solutions of Eq_κ within Scott's semantics is a well mastered technique, and passing from ω to κ is straightforward. Eq_κ admits moreover a minimal solution, which will be re-built in Section 9.

8.2. Premodels

In this subsection κ is any regular cardinal, and we do not need any σ.

Given a solution $\mathcal{M}$ of Eq_κ and an order isomorphism λ from $[\mathcal{M} \to_\kappa \mathcal{M}] \oplus_{\perp'} \{\mathsf{T}'\}$ to $\mathcal{M}$, let T and $\perp$ denote $\lambda(\mathsf{T}')$ and $\lambda(\perp')$, respectively. Thus, $\perp$ is the bottom element of $\mathcal{M}$ while T only compares to $\perp$. The following theorem is easy to prove and the details can be found in [3, Section 3.1]:

Theorem 8.2.1. *Let $\mathcal{M}$ be a solution of Eq_κ and let λ be an order isomorphism from $[\mathcal{M} \to_\kappa \mathcal{M}] \oplus_{\perp'} \{\mathsf{T}'\}$ to $\mathcal{M}$. There exists a reflexive object $(\mathcal{M}, A, \lambda)$ such that*
(a) for all $u \in \mathcal{M}$ we have $A(\mathsf{T})(u) = \mathsf{T}$ and $A(\perp)(u) = \perp$.
(b) $\mathcal{F} \equiv \{\lambda(f) \mid f \in [\mathcal{M} \to \mathcal{M}]\} = \{u {\in} \mathcal{M} \mid u = \mathbf{1}u\} = \mathcal{M} \setminus \{\perp, \mathsf{T}\}$
(c) $\mathcal{F}$ and $\{\mathsf{T}\}$ are disjoint κ-open subsets of $\mathcal{M}$.

Note that in (b) above, only the last equation uses that $\mathcal{M}$ is a solution to Eq_κ. The first equation is classic.

In fact $\mathcal{F}$ is the isomorphic image of $[\mathcal{M} \to_\kappa \mathcal{M}]$ under λ, and $\lambda x.\bot \equiv \lambda(x \mapsto \bot)$ is the bottom element of $\mathcal{F}$.

Conversely, any object $(\mathcal{M}, A, \lambda)$ satisfying the above theorem can easily be turned into a solution of Eq_κ.

Definition 8.2.2. A *κ-premodel* $\mathcal{M}^2$ is a triple $(\mathcal{M}^1, A, \lambda)$ satisfying the above requirements.

Definition 8.2.3. The *canonical* κ-premodel is the premodel associated to the minimal solution to Eq_κ.

Definition 8.2.3 will become more concrete in Section 9.6.

Theorem 8.2.4. *Given any κ-premodel $\mathcal{M}$ ($\kappa \geq \omega$), there are elements* if*,* $\|$*, and* E *in $\mathcal{M}$ such that $\mathcal{M}$ satisfies the elementary axioms and rules of MT and the axioms on* E *when the syntactic constructs* $\bot$*,* T *,* if*,* $\|$*, and* E *are interpreted by* $\bot$*,* T*,* if*,* $\|$*, and* E*, respectively.*

Proof. $\mathcal{M}$ satisfies rules Trans, Sub, Gen, and A2 since it is a reflexive object of a Cartesian closed category, $\mathcal{M}$ satisfies A1 and A3 and rule QND by Theorem 8.2.1. We now turn to the axioms concerning if, $\|$, and E. The argument is the same for the three constructors, and the case of if was already treated in [3].

The function If defined by $\mathsf{If}(u, v, w) = v$ if $u = \mathsf{T}$, w if $u \in \mathcal{F}$, and $\bot$ if $u = \bot$ is clearly κ-continuous. Hence, there is some element $\mathsf{if} \in \mathcal{M}$ such that $\mathsf{if}uvw = \mathsf{If}(u, v, w)$ for all $u, v, w \in \mathcal{M}$. Then it is easy to see that $\mathcal{M}$ satisfies Axioms I1, I2, and I3.

The function Paror defined by $\mathsf{Paror}(u, v) = \mathsf{T}$ if u or v is T, $\lambda x.\mathsf{T}$ if $u, v \in \mathcal{F}$, and $\bot$ otherwise is clearly κ-continuous. Hence, there is some element $\| \in \mathcal{M}$ such that $\|uv = \mathsf{Paror}(u, v)$ for all $u, v \in \mathcal{M}$. Then it is easy to see that $\mathcal{M}$ satisfies Axioms P1, P2, and P3.

The function Exist defined by $\mathsf{Exist}(u) = \mathsf{T}$ if $uv = \mathsf{T}$ for some $v \in \mathcal{M}$ and $\bot$ otherwise is clearly κ-continuous. Hence, there is some element $\mathsf{E} \in \mathcal{M}$ such that $\mathsf{E}u = \mathsf{Exist}(u)$ for all $u \in \mathcal{M}$. Then it is easy to see that $\mathcal{M}$ satisfies the four axioms on E. $\Box$

We now turn to Y and to the monotonicity and minimality axioms Mono and Min.

Theorem 8.2.5. *Given any κ-premodel $\mathcal{M}$ ($\kappa \geq \omega$), when the syntactic construct* Y *is interpreted by* $\mathsf{Y}_{\mathrm{Tarski}}$*, $\mathcal{M}$ satisfies the Monotonicity and the Minimality axioms for the model order $\preceq_{\mathcal{M}}$ (but possibly not for the syntactic order $\preceq$).*

Proof. Monotonicity is for free when $\mathcal{M}$ lives in Scott's semantics and the rest follows from Corollary 7.8.3. $\Box$

We now turn to the quantifier axioms. These axioms were easy to model in MT_0 (the difficulty was carried by some of the ϕ-axioms), but in MT they become very difficult to model since the constant ϕ of MT_0 is replaced here by a complex term ψ, whose definition involves ε and Y. Our trick will be to use that they hold for the characteristic function ϕ of Φ, and to prove later on (Sections 12 and 13) that, provided σ is the first inaccessible cardinal, ψ and ϕ coincide in all premodels when Y is interpreted as in the proof above (i.e. by $\mathsf{Y}_{\mathrm{Tarski}}$).

Recall $\mathrm{Dom}[w] \equiv \{u{\in}\mathcal{M} \mid wu = \mathsf{T}\}$ from Definition 4.6.1 and define:

Definition 8.2.6. *For $U \subseteq \mathcal{M}$ and $w \in \mathcal{M}$ we let:*
(a) $wU \equiv \{wu \mid u \in U\}$
(b) $\chi_U{:}\,\mathcal{M} \to \mathcal{M}$ *is defined by* $\chi_U(x) = \mathsf{T}$ *if* $x \in U$ *and* $\chi_U(x) = \perp$ *otherwise.*

Remark 8.2.7.
(a) $\mathrm{Dom}[w]$ *is a κ-open set for all $w \in \mathcal{M}$*
(b) χ_U *is κ-continuous iff U is κ-open*

Theorem 8.2.8 ([3]). *Let $\mathcal{M}$ be a κ-premodel ($\kappa \geq \omega$), and let $\Phi \subseteq \mathcal{M}$ be such that $\Phi = \uparrow U$ for some κ-small set U such that $\mathsf{T} \in U$ and $\perp \notin U$. Then there is an $\varepsilon \in \mathcal{M}$ such that, when the syntactical ε is interpreted by this ε, $\mathcal{M}$ satisfies the quantifier axioms,* ***but*** *for $\phi \equiv \lambda(\chi_\Phi)$ instead of the MT-term ψ.*

Proof. Sketch of proof (details in [3, Section 4.1]): Let ξ be a choice function on Φ, i.e. a function $\xi{:}\,\mathcal{P}(\Phi) \to \Phi$ such that $\xi(V) \in V$ for all non-empty $V \subseteq \Phi$. Let $e{:}\,\mathcal{M} \to \Phi{\cup}\{\perp\}$ be defined by: $e(u) = \perp$ if $\perp \in u\Phi$, $e(u) = \mathsf{T}$ if $u\Phi \subseteq \mathcal{F}$, and $e(u) = \xi\{x{\in}\Phi \mid ux = \mathsf{T}\}$ otherwise. Then e is κ-continuous (it is already clear that $u \preceq_{\mathcal{M}} v$ implies $e(u) = \perp$ or $e(u) = e(v)$) and can hence be internalized by an element $\varepsilon \in \mathcal{M}$ which has the required properties. □

8.3. Standard and canonical models of MT_0

We suppose now that there is some inaccessible σ such that $\sigma < \kappa$. And we recall the method which allows us to enrich any κ-premodel into a model of MT_0, under this hypothesis.

We define *σ-small* sets and *essentially σ-small* sets as was done for κ (c.f. Section 7.1 and 7.6), and we note that a κ-open set O is essentially σ-small if and only if $O = \uparrow K$ for some σ-small set of compact elements of $\mathcal{M}$.

Definition 8.3.1. *[3] For any $U, V, H \subseteq \mathcal{M}$ where H is open we let:*
(a) $\mathcal{O}^\sigma(U)$ *be the set of all essentially σ-small open subsets of U*
(b) $U \to V \equiv \{x{\in}\mathcal{M} \mid xU \subseteq V\}$
(c) $U^\circ \equiv \{x{\in}\mathcal{M} \mid \forall u_1, \ldots, u_n, \ldots {\in} U^\omega \exists n{\in}\omega{:}\, xu_1 \cdots u_n = \mathsf{T}\}$
(d) $\mathcal{F}_\sigma(H) \equiv \{\mathsf{T}\} \cup \bigcup\{G^\circ \to G \mid G \in \mathcal{O}^\sigma(H)\}$

Definition 8.3.2. $\Phi \subseteq \mathcal{M}$ *is the smallest solution of the equation* $\Phi = \mathcal{F}_\sigma(\Phi)$.

Definition 8.3.2 is equivalent to the definition used in [3] and several other definitions as studied in [3]. The property $\Phi = \mathcal{F}_\sigma(\Phi)$ is called the Generic Closure Property (GCP) in [3].

Theorem 8.3.3 ([3]). *Φ is an essentially κ-small open subset of $\mathcal{M}$.*

Definition 8.3.4. *An* MT_0 *standard κ-σ-quasimodel is a tuple $(\mathcal{M}, \mathsf{T}, \mathsf{if}, \varepsilon, \phi)$ where $\mathcal{M}$ is a κ-premodel, T, if, and ε are defined as in the previous section, and $\phi = \lambda(\chi_\Phi)$.*

Theorem 8.3.5 ([3]). *If σ is inaccessible and $\kappa > \sigma$ then any* MT_0 *standard κ-σ-quasimodel satisfies* MT_0.

Thus, we can drop "quasi" for σ inaccessible, $\kappa > \sigma$. For each κ, MT_0 has many κ-models but of course only one canonical κ-model: the one corresponding to the minimal solution to Eq_κ. We now proceed to MT models.

8.4. Towards modelling the quantification axioms

Now define $\hat{\psi} = \mathrm{Dom}[\psi]$. To model the quantification axioms of MT it remains to show that, if σ is the first inaccessible cardinal, then $\hat{\psi} = \Phi$. This is by far the most difficult proof of the paper, and it is split into two parts, called the Upper Bound Theorem (UBT) and the Lower Bound Theorem (LBT).

UBT says $\hat{\psi} \subseteq \Phi$. It puts an upper bound on $\hat{\psi}$ and is proved in Section 12. The proof uses the existence of an inaccessible σ (actually, the mere definition of Φ needs it). The proof also uses that the construct Y (which is part of the definition of ψ) is interpreted by $\mathsf{Y}_{\mathrm{Tarski}}$.

LBT says $\Phi \subseteq \hat{\psi}$. It puts a lower bound on $\hat{\psi}$ and is proved in Section 13. The proof of LBT uses UBT and also uses the assumption that σ is the *first* inaccessible ordinal (the proof of UBT does not use it). UBT and LBT together entail the following:

Theorem 8.4.1. *If σ is the first inaccessible and $\kappa > \sigma$, then:*
(a) Any MT *standard κ-σ-quasimodel $\mathcal{M}$ satisfies the quantifier axioms of* MT.
(b) If $\mathcal{M}$ is furthermore canonical then it satisfies the quantifier axioms of $\mathrm{MT}_{\mathrm{def}}$.

As already noticed in the introduction, $\mathrm{MT}_{\mathrm{def}}$ is a priori more difficult to model than MT. Fortunately, canonical models $\mathcal{M}$ are suited for it. Indeed, (b) above follows from (a) plus the results stated in Section 8.5.

Remark 8.4.2. *We avoid using Ψ for* $\mathrm{Dom}[\psi]$ *because Ψ has another meaning in [3].*

8.5. Towards modelling of Mono*,* Min*, and* Ext

Modelling Mono, Min, and Ext can be done in ZFC (no inaccessible is needed), but canonicity is crucial here.

Theorem 8.5.1. *For all $\kappa \geq \omega$ the canonical κ-quasimodel $\mathcal{M}$ satisfies* Ext.

Proof. Section 10.2 □

Now recall that we define the syntactic order $\preceq$ and the syntactic infimum $\downarrow$ in a roundabout way in that we define $\downarrow$ first and then define $\preceq$ as the order induced by $\downarrow$. Theorem 8.5.2 is equally roundabout:

Theorem 8.5.2. *For all $\kappa \geq \omega$ the canonical κ-quasimodel $\mathcal{M}$ satisfies that infimum in the model coincides with the syntactic infimum $\downarrow$ and, as a corollary, the model order $\preceq_{\mathcal{M}}$ coincides with the syntactic order $\preceq$.*

Proof. Section 10.3 □

Now recall that Y is interpreted by $\mathsf{Y}_{\mathrm{Tarski}}$.

Corollary 8.5.3. *$\mathcal{M}$ satisfies* Mono *and* Min *of MT.*

Theorem 8.5.4. *For all $\kappa \geq \omega$, the canonical κ-quasimodel $\mathcal{M}$ satisfies*

$$\mathsf{Y}_{\mathrm{Tarski}} = \mathsf{Y}_{\mathrm{Curry}}$$

Proof. Section 10.4 □

Corollary 8.5.5. *$\mathcal{M}$ satisfies $\mathsf{Y}f = \mathsf{Y}_{\mathrm{Curry}}f$ and $\perp = \perp_{\mathrm{Curry}}$.*

Corollary 8.5.6. *$\mathcal{M}$ satisfies* Mono *and* Min *of* $\mathrm{MT}_{\mathrm{def}}$.

9. Building the canonical κ-premodel

The aim is here to recall the elementary construction we gave in [3, Section 8] of the canonical premodel, i.e. of the minimal solution of the domain equation Eq_κ (which proves in passing the existence of such a solution). This premodel is a *webbed model* in the sense that it is built as an (enriched) powerset of some lower level structure, called its *web*, which here can be taken to be a *reflexive pcs*. The terminology of "webbed model" was introduced in [2] and pcs's are defined below.

9.1. Pcs's

A *pre-pcs* is a tuple $\mathbf{D} = \langle D_{\mathbf{D}}, \leq_{\mathbf{D}}, \frown_{\mathbf{D}} \rangle$ for which $\leq_{\mathbf{D}}$ and $\frown_{\mathbf{D}}$ are binary relations on $D_{\mathbf{D}}$.

A *pcs* is a pre-pcs $\mathbf{D} = \langle D, \leq, \frown \rangle$ with the following properties:

Partial order	$\leq$ is reflexive and transitive.
Coherence	$\frown$ is reflexive and symmetric.
Compatibility	$x \leq x' \wedge y \leq y' \wedge x' \frown y' \Rightarrow x \frown y$.

From now on, $\mathbf{D} = \langle D, \leq, \frown \rangle$ and $\mathbf{D}' = \langle D', \leq', \frown' \rangle$ denote pre-pcs's.

$\mathbf{D}$ is a *sub-pcs* of $\mathbf{D}'$, written $\mathbf{D} \sqsubseteq \mathbf{D}'$, if the following hold:

$$\begin{array}{rccc} & D & \subseteq & D' \\ \forall x, y \in D: & x \leq y & \Leftrightarrow & x \leq' y \\ \forall x, y \in D: & x \frown y & \Leftrightarrow & x \frown' y \end{array}$$

A set S of pre-pcs's is a *chain* if $\forall \mathbf{D}, \mathbf{D}' {\in} S\colon \mathbf{D} \sqsubseteq \mathbf{D}' \vee \mathbf{D}' \sqsubseteq \mathbf{D}$. For all pre-pcs's $\mathbf{D}$, all $u, v \subseteq \mathbf{D}$, and $p \in \mathbf{D}$ define

$$
\begin{array}{lcl}
u \leq^*_{\mathbf{D}} v & \Leftrightarrow & \forall x{\in}u \exists y{\in}v\colon x \leq_{\mathbf{D}} y \\
u \frown^*_{\mathbf{D}} v & \Leftrightarrow & \forall x{\in}u \forall y{\in}v\colon x \frown_{\mathbf{D}} y \\
\mathrm{Coh}_{\mathbf{D}} u & \Leftrightarrow & u \frown^*_{\mathbf{D}} u \\
\downarrow_{\mathbf{D}} u & = & \{y \in \mathbf{D} \mid \exists x{\in}u\colon y \leq x\} \\
\downarrow_{\mathbf{D}} p & = & \downarrow_{\mathbf{D}} \{p\} \\
\mathcal{I}(\mathbf{D}) & = & \{\downarrow_{\mathbf{D}} u \mid u \subseteq \mathbf{D} \wedge \mathrm{Coh}_{\mathbf{D}} u\}
\end{array}
$$

Above $\mathcal{I}(\mathbf{D})$ denotes the set of coherent, initial segments of $\mathbf{D}$.

Fact 9.1.1. *For all pcs's $\mathbf{D}$, $(\mathcal{I}(\mathbf{D}), \subseteq)$ is a prime algebraic κ-Scott domain whose sets of prime and compact elements are $\{\downarrow_{\mathbf{D}} p \mid p \in \mathbf{D}\}$ and $\{\downarrow_{\mathbf{D}} u \mid u \in \mathcal{P}^\kappa(D) \wedge \mathrm{Coh}_{\mathbf{D}} u\}$, respectively.*

The goal of Section 9.2–9.3 is to define a pcs $\mathbf{P}$ such that $(\mathcal{I}(\mathbf{P}), \subseteq)$ satisfies Eq_κ.

9.2. Pcs generators

Let $D \uplus D'$ denote disjoint union (i.e. $D \cup D'$ when $D \cap D' = \emptyset$ and undefined otherwise). Let $\mathbf{D}, \mathbf{D}'$ be pre-pcs's and let S be a set of such structures. We now define the pre-pcs's $\mathrm{U}(t)$, $\mathbf{D}_f$, $\cup S$, $\mathbf{D} \oplus \mathbf{D}'$, $\mathbf{D} \to \mathbf{D}'$, and $\mathcal{P}^\kappa_{\mathrm{coh}}(\mathbf{D})$:

R	D_R	$x \leq_R y$	$x \frown_R y$
$\mathrm{U}(t)$	$\{t\}$	*true*	*true*
$\mathbf{D}_f$	$D \uplus \{f\}$	$x{=}f \vee x{\leq}y$	$x{=}f \vee y{=}f \vee x{\frown}y$
$\cup S$	$\cup_{\mathbf{D} \in S} D_{\mathbf{D}}$	$\exists \mathbf{D}{\in}S\colon x{\leq_{\mathbf{D}}}y$	$\exists \mathbf{D}{\in}S\colon x{\frown_{\mathbf{D}}}y$
$\mathbf{D} \oplus \mathbf{D}'$	$D \uplus D'$	$x{\leq}y \vee x{\leq'}y$	$x{\frown}y \vee x{\frown'}y$
$\mathbf{D} \to \mathbf{D}'$	$D \times D'$	$y_1{\leq}x_1 \wedge x_2{\leq'}y_2$	$x_1{\not\frown}y_1 \vee x_2{\frown'}y_2$
$\mathcal{P}^\kappa_{\mathrm{coh}}(\mathbf{D})$	$\{a{\in}\mathcal{P}^\kappa(D) \mid \mathrm{Coh}_{\mathbf{D}} a\}$	$x{\preceq^*}y$	$x{\frown^*}y$

As an example of reading the table, the fourth line says that $\mathbf{D} \oplus \mathbf{D}'$ is the unique pre-pcs R for which

$$
\begin{array}{lcl}
D_R & = & D \cup D' \\
x {\leq_R} y & \Leftrightarrow & x, y \in D_R \wedge (x{\leq}y \vee x{\leq'}y) \\
x {\frown_R} y & \Leftrightarrow & x, y \in D_R \wedge (x{\frown}y \vee x{\frown'}y)
\end{array}
$$

In the line defining $\mathbf{D} \to \mathbf{D}'$, we define $D \times D' = \{\langle x_1, x_2 \rangle \mid x_1 \in D \wedge x_2 \in D'\}$. For all $x \in D \times D'$ we define x_1 and x_2 to be the first and second component, respectively, of the tuple x.

Under reasonable conditions, the above pre-pcs's are pcs's:

Fact 9.2.1. $\mathrm{U}(t)$ *is a pcs for all objects t (of ZFC).*

Fact 9.2.2. *If $\mathbf{D}$ is a pcs and $f \notin D$ then $\mathbf{D}_f$ is a pcs.*

Fact 9.2.3. *If S is a chain of pcs's then $\cup S$ is a pcs.*

Fact 9.2.4. *If $\mathbf{D}$ and $\mathbf{D}'$ are pcs's and D and D' are disjoint, then $\mathbf{D} \oplus \mathbf{D}'$ is a pcs.*

Fact 9.2.5. *If $\mathbf{D}$ and $\mathbf{D}'$ are pcs's then $\mathbf{D} \to \mathbf{D}'$ is a pcs.*

Fact 9.2.6. *If $\mathbf{D}$ is a pcs and κ is a cardinal, then $\mathcal{P}^{\kappa}_{\mathrm{coh}}(\mathbf{D})$ is a pcs.*

9.3. The web of the canonical κ-premodel

Now let κ be a regular cardinal greater than σ and, for all pre-pcs's $\mathbf{D}$, define

$$\mathbf{H}(\mathbf{D}) = (\mathcal{P}^{\kappa}_{\mathrm{coh}}(\mathbf{D}) \to \mathbf{D})_f \oplus \mathrm{U}(t)$$

Furthermore let Eq'_κ be the equation

$$\mathbf{H}(\mathbf{P}) = \mathbf{P}$$

Fact 9.3.1. *If a pcs $\mathbf{P}$ satisfies Eq'_κ then $(\mathcal{I}(\mathbf{P}), \subseteq)$ satisfies Eq_κ.*

Now define

$$\begin{array}{lcl} \mathbf{P}_0 & = & \langle \emptyset, \emptyset, \emptyset \rangle \\ \mathbf{P}_{\beta+1} & = & \mathbf{H}(\mathbf{P}_\beta) \\ \mathbf{P}_\delta & = & \cup\{\mathbf{P}_\beta \mid \beta \in \delta\} \\ \mathbf{P} & = & \mathbf{P}_\kappa \end{array}$$

It is easy to prove by transfinite induction that $\mathbf{P}_\beta$ is a pcs, that $\{\mathbf{P}_\beta \mid \beta \in \delta\}$ is a chain of pcs's, and that the pcs $\mathbf{P}$ is the minimal solution of Eq'_κ. Note that $\mathbf{P}_1 = \{t, f\}$.

We define the rank $rk(p)$ of $p \in \mathbf{P}$ as the smallest ordinal α for which $p \in \mathbf{P}_\alpha$. Recall that $\mathbf{P}_0 = \emptyset$ and $\mathbf{P}_1 = \{t, f\}$ (as sets).

9.4. Some properties of the web

From now on $\downarrow$ means $\downarrow_{\mathbf{P}}$. Define $\mathbf{C} \equiv \mathcal{P}^{\kappa}_{\mathrm{coh}}(\mathbf{P})$. For $p \in \mathbf{P}$ and $\bar{a} = \langle a_1, \ldots, a_n \rangle \in \mathbf{C}^{<\omega}$ let $\ell(\bar{a})$ denote n (i.e. the length of $\bar{a}$) and define

$$\langle \bar{a}, p \rangle \equiv \langle a_1, \ldots \langle a_n, p \rangle \ldots \rangle$$

In particular, $\langle \bar{a}, p \rangle = p$ if $\ell(\bar{a}) = 0$. Using that there are no decreasing infinite sequences of ordinals we easily get:

Lemma 9.4.1 ([3]). *For each $p \in \mathbf{P}$ there is a unique decomposition of p as $p = \langle \bar{a}, t \rangle$ or $p = \langle \bar{a}, f \rangle$ where $\bar{a} \in \mathbf{C}^{<\omega}$.*

For $p = \langle \bar{a}, t \rangle$ ($p = \langle \bar{a}, f \rangle$) we define $\ell(p) = \ell(\bar{a}) + 1$ and refer to t (f) as the head of p.

Remark 9.4.2.
$\langle a, p \rangle \leq r \in \mathbf{P}$ implies $r = \langle b, q \rangle$ for some b, q.
$\langle a, p \rangle \leq \langle b, q \rangle$ iff $b \subseteq \downarrow a$ and $p \leq q$.

9.5. The domain of the canonical κ-premodel

The κ-Scott domain $\mathcal{M}$ of the canonical κ-premodel is defined by

$$\mathcal{M} \equiv (\mathcal{I}(\mathbf{P}), \subseteq)$$

We have:

Fact 9.5.1.
$\mathcal{M}_p = \{\downarrow p \mid p \in \mathbf{P}\}$ *is the set of prime maps of* $\mathcal{M}$.
$\mathcal{M}_c = \{\downarrow a \mid a \in \mathbf{C}\}$ *is the set of compact maps of* $\mathcal{M}$.
In $\mathcal{M}$*, sups are unions and infs are intersections.*

$\mathcal{M}_c$ was called $\mathcal{K}(\mathcal{M})$ in Section 7.3.

9.6. The canonical κ-premodel

Let T' and $\perp'$ be arbitrary objects such that $\mathsf{T}' \neq \perp'$ and $\mathsf{T}', \perp' \notin [\mathcal{M} \to_\kappa \mathcal{M}]$. Then define λ, T, $\perp$, and A by

$$\begin{array}{lllll}
\lambda(\mathsf{T}') & \equiv & \mathsf{T} \quad \equiv \quad \{t\} & \\
\lambda(h) & \equiv & \{f\} \cup \{\langle a, p\rangle \in \mathbf{C} \times \mathbf{P} \mid p \in h(\downarrow a)\} & \text{for } h \in [\mathcal{M} \to_\kappa \mathcal{M}] \\
\lambda(\perp') & \equiv & \perp \quad \equiv \quad \emptyset & \\
A(\mathsf{T})(v) & = & \mathsf{T} & \text{for } v \in \mathcal{M} \\
A(u)(v) & = & \{p{\in}\mathbf{P} \mid \exists a{\subseteq}v \colon \langle a, p\rangle \in u\} & \text{for } u \in \mathcal{F},\, v \in \mathcal{M} \\
A(\perp)(v) & = & \perp & \text{for } v \in \mathcal{M}
\end{array}$$

We have:

Fact 9.6.1.
(a) $\lambda : [\mathcal{M} \to_\kappa \mathcal{M}] \oplus_{\perp'} \{\mathsf{T}'\} \to \mathcal{M}$ is an order isomorphism.
(b) $\mathcal{M}$ is the minimal solution to Eq_κ.
(c) $(\mathcal{M}, A, \lambda)$ is a reflexive object.

Definition 9.6.2. *The canonical κ-premodel is the triple $(\mathcal{M}, A, \lambda)$ with λ and A defined as above. $\mathcal{M}_\kappa$ is the associated* MT *canonical κ-quasimodel (c.f. Section 5.4).*

Note that we have $\mathsf{T} = \{t\}$, $\perp = \emptyset$, and $\mathcal{F} = \mathcal{M} \setminus \{\mathsf{T}, \perp\}$ with $\mathcal{F}$ defined as for Theorem 8.2.1. We have:

Fact 9.6.3.
(a) $u \in \mathcal{F}$ iff $u \in \mathcal{M}$ and $f \in u$.
(b) $\perp, \mathsf{T}, \{f\} \in \mathcal{M}_c$ and $\{f\}$ models $\lambda x.\perp$.

9.7. Tying up a loose end

Now recall the definitions of $a \sim_\kappa b$, $a =^\kappa_{\text{obs}} b$, and $a =_\kappa b$ from Section 3.6. Note that if $\forall c{\in}\mathcal{M}_\kappa \colon ca \sim_\kappa cb$ then, in particular, $(\downarrow\langle\{p\}, t\rangle)a = \mathsf{T} \Leftrightarrow (\downarrow\langle\{p\}, t\rangle)b = \mathsf{T}$ so $p \in a \Leftrightarrow p \in b$. Thus, $a =^\kappa_{\text{obs}} b \Rightarrow a =_\kappa b$ which is the non-trivial direction of Fact 3.6.2.

10. Canonical premodels satisfy Mono, Min, and Ext

In Section 10 we only suppose $\kappa \geq \omega$, and that $\mathcal{M}$ is the canonical κ-premodel described above; in particular its domain is the minimal solution of Eq_κ.

We prove that $\mathcal{M}$ satisfies Mono, Min, and Ext, that the model order $\subseteq$ coincides with the syntactic order $\preceq$, and that we can eliminate the constant Y in favor of Curry's paradoxical combinator.

Monotonicity of application w.r.t. $\subseteq$ will be used constantly, most often without mention.

10.1. A characterization of the order of $\mathcal{M}$ via application

The following applicative characterization of the model order $\subseteq$ of $\mathcal{M}$ is the key for proving later on that the model order coincides with the syntactical order $\preceq$ and that $\mathcal{M}$ satisfies Ext.

Definition 10.1.1. *Let $r = \lambda u.\, \mathsf{if}[u, \mathsf{T}, \lambda x.\bot]$.*

Thus in $\mathcal{M}$ we have that $ru = \mathsf{T}$ if $u = \mathsf{T}$, $ru = \emptyset$ if $u = \emptyset$, and $ru = \{f\}$ if $u \in \mathcal{F}$.

Theorem 10.1.2. *For all $u, v \in \mathcal{M}$ the following are equivalent:*
- *(i)* $u \subseteq v$
- *(ii)* *For all $\bar{w} \in \mathcal{M}^{<\omega}$ we have $r(u\bar{w}) \subseteq r(v\bar{w})$*

Proof. (i) $\Rightarrow$ (ii) because application is monotone.

(ii) $\Rightarrow$ (i). Let $U = \mathcal{M} \setminus \{\emptyset, \mathsf{T}, \{f\}\}$. The only non-trivial case is when $u \in U$. Let V be the set of triples (u, v, p) such that $u \in U$, u and v satisfy (ii), and $p \in u \setminus v$. Note that $v \in \mathcal{F}$ and $p \neq t, f$. Suppose $V \neq \emptyset$, choose a triple $(u, v, p) \in V$ such that $\ell(p)$ is minimal. Since $\ell(p) \geq 2$ let $(c, q) \in \mathbf{C} \times \mathbf{P}$ be such that $p = \langle c, q \rangle$, which implies $\ell(q) < \ell(p)$ and $q \in ({\downarrow}p)w \subseteq uw$, where $w \equiv {\downarrow}c$. Since the pair (uw, vw) satisfies (ii), by the minimality hypothesis we have that $q \in vw$. Now, by definition of application in $\mathcal{M}$, and since $v \neq \mathsf{T}$, there is a $c' \subseteq {\downarrow}c \equiv w$ such that $\langle c', q \rangle \in v$ and $p = \langle c, q \rangle \leq \langle c', q' \rangle \in v$, thus $p \in v$. A contradiction which proves $V = \emptyset$, i.e. (ii) $\Rightarrow$ (i). $\square$

Corollary 10.1.3. *For all $u, v \in \mathcal{M}$ we have*
- *(i)* $u \subseteq v$ *iff* $r(u) \subseteq r(v)$ *and* $\forall w \colon (uw \subseteq vw)$
- *(ii)* $u = v$ *iff* $r(u) = r(v)$ *and* $\forall w \colon (uw = vw)$

Proof. (i) is an immediate consequence of the theorem, from which (ii) follows. In fact both are also direct consequences of the fact that $\mathcal{M}$ was a premodel ($\mathcal{M}$ is not required to be canonical for the corollary). $\square$

10.2. Ext

Theorem 10.2.1. $\mathcal{M} \models \mathsf{Ext}$

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be two MT-terms that do not contain x and y free and suppose there is an MT-term $\mathcal{C}[x,y]$ such that (for all assignments of values to free variables) $\mathcal{M} \models \forall w \forall v\colon (\mathcal{A}wv = \mathcal{AC}[w/x, v/y] \wedge \mathcal{B}wv = \mathcal{BC}[w/x, v/y])$. The task is to prove that $\mathcal{M} \models \forall w\colon (\mathcal{A}w = \mathcal{B}w)$ under the hypothesis that $\mathcal{M} \models \forall w\colon (r(\mathcal{A}w) = r(\mathcal{B}w))$. Now, the hypothesis on $\mathcal{A}$ and $\mathcal{B}$ obviously imply that, given $w \in \mathcal{M}$, the elements $\mathcal{A}w$ and $\mathcal{B}w$ satisfy point (ii) of Theorem 10.1.2; by (i) we hence have $\mathcal{A}w \subseteq \mathcal{B}w$. Similarly, $\mathcal{B}w \subseteq \mathcal{A}w$ so $\mathcal{A}w = \mathcal{B}w$. $\square$

10.3. λ-definability of the order of $\mathcal{M}$

Theorem 10.3.1. $\mathcal{M} \models \forall u \forall v\colon (u \downarrow v = u \cap v)$.

Proof. If one of u, v is equal to $\bot$ or $\top$ or $\{f\}$ then it is easy to check that the equality holds. Let V be the set of triples (u, v, p) such that p falsifies $u \downarrow v = u \cap v$ (i.e. p is in one of $u \downarrow v$ and $u \cap v$, but not both). Suppose $V \neq \emptyset$ and choose $(u, v, p) \in V$ such that $\ell(p)$ is minimal. Thanks to the preceding remarks, u, v are different from $\top$, $\bot$, and $\{f\}$, so $p \neq t$, $p \neq f$, and $u \downarrow v = \lambda z.(uz \downarrow vz)$. Let $(c, q) \in \mathbf{C} \times \mathbf{P}$ be such that $p = \langle c, q \rangle$, and $w = {\downarrow}c$. Then $q \in ({\downarrow}p)w$.

Suppose $p \in u \downarrow v$. Then $q \in (u \downarrow v)w = (uw) \downarrow (vw)$. By the minimality hypothesis $q \in (uw) \cap (vw)$, and, since $u \neq \top$ and $v \neq \top$ there are $c' \subseteq w$ and $c'' \subseteq w$ such that $\langle c', q \rangle \in u$ and $\langle c'', q \rangle \in v$. Since $w \equiv {\downarrow}c$ we have $p \leq \langle c', q \rangle \in u$ and $p \leq \langle c'', q \rangle \in v$, hence $p \in u \cap v$.

Suppose now that $p \in u \cap v$. Then $q \in (u \cap v)w \subseteq (uw) \cap (vw)$. By the minimality hypothesis $q \in (uw) \downarrow (vw) = (u \downarrow v)w$. Hence there is a $c' \subseteq w$ such that $\langle c', q \rangle \in (u \downarrow v)$. Since $w \equiv {\downarrow}c$ we have $p \leq \langle c', q \rangle \in u \downarrow v$.

We have reached a contradiction, hence $V = \emptyset$. $\square$

Corollary 10.3.2. $\mathcal{M} \models \forall u \forall v\colon (u \preceq v \Leftrightarrow u \subseteq v)$.

Corollary 10.3.3. *In $\mathcal{M}$ the binary κ-continuous function inf is definable by a λ-term (using* if*, $\bot$, and $\top$), and hence the model order $\subseteq$ is equationally definable.*

Remark 10.3.4. *It is interesting to compare this last result (which only applies to canonical premodels of MT) to the following one, which deserves to be known: the order of a reflexive Scott domain is always definable by a first order formula using only application (and which is the same for all these domains). This result, proved by Plotkin in 1972, and only published twenty years later in [14], was rediscovered independently by Kerth [11], who proved that it also holds in Berry's and Girard's stable semantics, and Ehrhard's strongly stable semantics [12] (with different formulas).*

10.4. Y *and minimality*

We now show that $\mathcal{M}$ interprets Curry's fixed point combinator as $\mathsf{Y}_{\mathrm{Tarski}}$. A first proof was worked out by Thierry Vallée (private communication, 2002), the present one is slightly more direct.

Definition 10.4.1. *For all $u \in \mathcal{M}$ and ordinals α let $u_\alpha \equiv {\downarrow}(u \cap \mathbf{P}_\alpha) \in \mathcal{M}$.*

Lemma 10.4.2. *For all $u, v \in \mathcal{M}$ we have:*
- *(i)* $u_0 = \emptyset$ *and* $u_\kappa = u$.
- *(ii)* $u_\delta v = \cup_{\beta<\delta}(u_\beta v)$ *for all limit ordinals* δ.
- *(iii)* $u_{\beta+1} v = u_{\beta+1} v_\beta$ *for all ordinals* β.

Proof. (ii) $\cup_{\beta<\delta}(u_\beta v) \subseteq u_\delta v$ by monotonicity. Now assume $p \in u_\delta v$. Choose $a \subseteq v$ such that $\langle a, p\rangle \in u_\delta \equiv {\downarrow}(u \cap \mathbf{P}_\delta)$. Choose $q \in u \cap \mathbf{P}_\delta$ such that $\langle a, p\rangle \leq q$. Choose $\beta < \delta$ such that $q \in \mathbf{P}_\beta$. Choose a', p' such that $q = \langle a', p'\rangle$. We have $p \leq p'$ and $a' \subseteq {\downarrow}a$, c.f. Remark 9.4.2. Now $p \leq p' \in ({\downarrow}q)({\downarrow}a') \subseteq u_\beta v$ so $p \in u_\beta v$.

(iii) $u_{\beta+1} v_\beta \subseteq u_{\beta+1} v$ by monotonicity. Now assume $p \in u_{\beta+1} v$. Choose $a \subseteq v$ such that $\langle a, p\rangle \in u_{\beta+1} \equiv {\downarrow}(u \cap \mathbf{P}_{\beta+1})$. Choose $q \in u \cap \mathbf{P}_{\beta+1}$ such that $\langle a, p\rangle \leq q$. Choose a', p' such that $q = \langle a', p'\rangle$. We have $p \leq p'$ and $a' \subseteq {\downarrow}a$, c.f. Remark 9.4.2. Furthermore, $q \in \mathbf{P}_{\beta+1}$ implies $a' \subseteq \mathbf{P}_\beta$. Now $p \leq p' \in ({\downarrow}q)({\downarrow}a') \subseteq u_{\beta+1} v_\beta$ so $p \in u_{\beta+1} v_\beta$. $\square$

Theorem 10.4.3. $\mathcal{M} \models \mathsf{Y}_{\mathrm{Curry}} = \mathsf{Y}_{\mathrm{Tarski}}$.

Proof. Since $\mathsf{Y}_{\mathrm{Tarski}}$ acts as the least fixed point operator on $\mathcal{M}$ it is enough to prove that, for all $u \in \mathcal{M}$, we have $ww \subseteq \mathsf{Y}_{\mathrm{Tarski}} u$, where $w \equiv \lambda x.u(xx)$. We prove $w_\alpha w \subseteq \mathsf{Y}_{\mathrm{Tarski}} u$ by induction on $\alpha \leq \kappa$. The case $\alpha = 0$ is clear and the limit case is by Lemma 10.4.2(ii). If $\alpha = \beta + 1$ we have $w_{\beta+1} w = w_{\beta+1} w_\beta \subseteq w w_\beta = u(w_\beta w_\beta) \subseteq u(w_\beta w) \subseteq u(\mathsf{Y}_{\mathrm{Tarski}} u) = \mathsf{Y}_{\mathrm{Tarski}} u$, the first equality coming from Lemma 10.4.2(iii) and the last inclusion by induction hypothesis. $\square$

Remark 10.4.4. *Most usual models are stratified, in the sense (very roughly speaking) that it is possible to find a way of decomposing them in such a way that each u is the inf of an increasing sequence u_α, $\alpha \in \kappa$ (usually $\kappa = \omega$) satisfying all the properties listed in Lemma 10.4.2 except $u_1 v = u_1 v_0$. This last equation is really the key point here as for, say, Scott's first model $\mathcal{D}_\infty$, for which it holds; it is false for Park's variant of $\mathcal{D}_\infty$, which does not satisfy* Min.

11. Concepts for proving UBT and LBT

11.1. Main theorem

In the following, σ denotes the smallest inaccessible ordinal (in [3], σ denotes an arbitrary, inaccessible ordinal). Let κ be a regular cardinal greater than σ. We work in a standard κ-model $\mathcal{M}$. We refer to elements of $\mathcal{M}$ as *maps*. Unless otherwise noted, variables range over $\mathcal{M}$.

Define Φ and ϕ as in [3]. ϕ satisfies

$$\phi x = \begin{cases} \mathsf{T} & \text{when } x \in \Phi \\ \bot & \text{otherwise} \end{cases}$$

Recall Definition 4.6.1 and 4.6.3 and define $\hat{\psi} = \mathrm{Dom}[\psi]$. The elements of Φ and $\hat{\psi}$ are the wellfounded maps of MT_0 and MT, respectively. We now prove that the two notions of wellfoundedness coincide:

Theorem 11.1.1 (Main Theorem). $\hat{\psi} = \Phi$ *(or, equivalently,* $\psi = \phi$*)*

To do so, we prove that Φ is both an upper and a lower bound of $\hat{\psi}$:

Theorem 11.1.2 (Upper Bound Theorem/UBT). $\hat{\psi} \subseteq \Phi$.

Theorem 11.1.3 (Lower Bound Theorem/LBT). $\Phi \subseteq \hat{\psi}$.

We prove UBT and LBT in Sections 12 and 13, respectively. Section 11 analyzes ψ and Φ.

The proof of LBT uses UBT (e.g. Fact 13.6.2(c) and Lemma 13.8.3(e)).

The proof of LBT uses that σ is the *smallest* inaccessible ordinal (in Lemma 13.7.3).

11.2. Elementary observations

Fact 11.2.1.
(a) $\bot \preceq_{\mathcal{M}} y$
(b) $x \preceq_{\mathcal{M}} y \wedge x = \mathsf{T} \Rightarrow y = \mathsf{T}$
(c) $x \preceq_{\mathcal{M}} y \wedge x \in \mathcal{F} \Rightarrow y \in \mathcal{F}$

Fact 11.2.2.
(a) $(\mathsf{E}x.\,\mathcal{A}) = \mathsf{T} \Leftrightarrow \exists x{\in}\mathcal{M}\colon (\mathcal{A} = \mathsf{T})$
(b) $(\ddot{\forall} x.\,\mathcal{A}) = \mathsf{T} \Leftrightarrow \forall x{\in}\Phi\colon (\mathcal{A} = \mathsf{T})$
(c) $\phi x = \mathsf{T} \Leftrightarrow x \in \Phi$

Fact 11.2.3.
(a) $(x \,!\, y) \neq \bot \Leftrightarrow x = \mathsf{T} \wedge y \neq \bot$
(b) $(x \,!\, y) \neq \bot \Rightarrow x \,!\, y = y$
(c) $(x \,!\, y) \,!\, z = x \,!\, (y \,!\, z)$
(d) $(x \,!\, y \,!\, z) \neq \bot \Leftrightarrow x = \mathsf{T} \wedge y = \mathsf{T} \wedge z \neq \bot$
(e) $(x \,!\, y \,!\, z) \neq \bot \Rightarrow x \,!\, y \,!\, z = z$

Fact 11.2.4.
(a) $\mathsf{D}x = \mathsf{T} \Leftrightarrow x \neq \bot$

Fact 11.2.5.
(a) $\mathrm{Dom}[\sqcup f] = \cup_{x \in \mathcal{M}} \mathrm{Dom}[fx]$

11.3. Duals, boundaries, closure, and functions

Definition 11.3.1. Let $G, H \subseteq \mathcal{M}$

(a) $G^\circ \equiv \{g{\in}\mathcal{M} \mid \forall x_0, x_1, \ldots{\in}G \exists n{\in}\omega\colon g x_0 \cdots x_n = \mathsf{T}\}$ for $G \neq \emptyset$
(b) $\emptyset^\circ \equiv \mathcal{M} \setminus \{\bot\}$
(c) $G^\delta \equiv \{g{\in}G \mid \forall f{\in}G\colon (f \preceq_{\mathcal{M}} g \Rightarrow f = g)\}$
(d) $G^\uparrow \equiv \{h{\in}\mathcal{M} \mid \exists g{\in}G\colon g \preceq_{\mathcal{M}} h\}$
(e) $G \to H = \{f{\in}\mathcal{M} \mid \forall x{\in}G\colon fx \in H\}$

We shall refer to G°, G^δ, and $G^\uparrow$ as the *dual*, *boundary*, and *upward closure*, respectively, of G. Definition (e) above repeats Definition 8.3.1(b). Definition (a) above repeats Definition 8.3.1(c). Definition (b) above makes explicit how to understand $\emptyset^\circ$.

Fact 11.3.2.

(a) $G \subseteq H \Rightarrow H^\circ \subseteq G^\circ$
(b) $G \subseteq H \Rightarrow G^{\circ\circ} \subseteq H^{\circ\circ}$
(c) $G' \subseteq G \wedge H \subseteq H' \Rightarrow G \to H \subseteq G' \to H'$
(d) $G \subseteq H \Rightarrow G^\circ \to G \subseteq H^\circ \to H$
(e) $G \neq \emptyset \Rightarrow G^\circ = G \to G^\circ$
(f) $G \subseteq H \subseteq H^{\circ\circ} \Rightarrow G^\circ \to G \subseteq H^\circ \to H^{\circ\circ} = H^{\circ\circ}$

Lemma 11.3.3. *$G^{\circ\delta\uparrow} = G^\circ$ for all open G.*

Proof of 11.3.3 Trivial for $G = \emptyset$. Follows from [3, Theorem 6.1.11] for $G \neq \emptyset$.

11.4. Cardinality

We shall use $G <_c H$ to denote that G has smaller cardinality than H and define $G =_c H$ and $G \leq_c H$ likewise. For $G \subseteq \mathcal{M}$ recall from Sections 7.1, 7.6, and 8.3 that G is *essentially σ-small* if there exists a σ-small V such that $V \subseteq G \subseteq V^\uparrow$. Recall from Definition 8.3.1(a) that $\mathcal{O}^\sigma(G)$ denotes the set of essentially σ-small open subsets of G. We only use this notation for open G.

Lemma 11.4.1.

(a) If $G \in \mathcal{O}^\sigma(\mathcal{M})$ then $G^\circ \in \mathcal{O}^\sigma(\mathcal{M})$.
(b) If $G \in \mathcal{O}^\sigma(\Phi)$ then $G^\circ \to G \in \mathcal{O}^\sigma(\Phi)$.

Proof of 11.4.1

(a) [3, Theorem 6.1.11].
(b) This follows directly from the definition of Φ (Definition 8.3.2).

11.5. Hierarchies

Definition 11.5.1.

(a) $\Phi_0 = \{\mathsf{T}\}$
(b) $\Phi_{\alpha'} = {\Phi_\alpha}^\circ \to \Phi_\alpha$
(c) $\Phi_\delta = \cup_{\beta\in\delta} \Phi_\beta$ for limit ordinals δ.
(d) $\mathcal{H}_0 = \{\mathsf{T}\}$

(e) $\mathcal{H}_{\alpha'} = \mathcal{H}_\alpha{}^{\circ\circ}$
(f) $\mathcal{H}_\delta = \cup_{\beta\in\delta}\mathcal{H}_\beta$ for limit ordinals δ.

Lemma 11.5.2.
(a) $\alpha \in \beta \Rightarrow \Phi_\alpha \subseteq \Phi_\beta$
(b) $\alpha \in \beta \Rightarrow \mathcal{H}_\alpha \subseteq \mathcal{H}_\beta$
(c) $\Phi_\alpha \subseteq \mathcal{H}_\alpha$
(d) $\Phi = \Phi_\sigma = \mathcal{H}_\sigma$
(e) $\forall G{\in}\mathcal{O}^\sigma(\Phi)\exists\alpha{\in}\sigma{:}\, G \subseteq \Phi_\alpha \subseteq \mathcal{H}_\alpha$
(f) $\Phi \subseteq \Phi^\circ$
(g) $\Phi_\alpha \subseteq \mathcal{H}_\alpha \subseteq \Phi \subseteq \Phi^\circ \subseteq \mathcal{H}_\alpha^\circ \subseteq \Phi_\alpha^\circ$

Proof of 11.5.2
(a) By transfinite induction using Fact 11.3.2(d).
(b) By transfinite induction using Fact 11.3.2(b).
(c) We have $\mathcal{H}_\alpha \subseteq \mathcal{H}_{\alpha'} = \mathcal{H}_\alpha{}^{\circ\circ}$ by Lemma 11.5.2(b) and Definition 11.5.1(e). Hence, $\Phi_\alpha \subseteq \mathcal{H}_\alpha$ by transfinite induction using Fact 11.3.2(f).
(d) For $\Phi = \Phi_\sigma$ see the proof of [3, Lemma A.1.1]. For $\mathcal{H}_\sigma \subseteq \Phi$ see [3, Theorem A.2.1] and its proof. $\Phi_\sigma \subseteq \mathcal{H}_\sigma$ is given by (c).
(e) For each $g \in G$ let $\rho(g)$ be the smallest ordinal for which $g \in \Phi_{\rho(g)}$. Take $\alpha = \cup_{g\in G}\rho(g)$.
(f) [3, Theorem 7.1.1].
(g) Follows trivially from (a-d,f) and Fact 11.3.2(a).

11.6. Self-extensionality

We now recall the definition of *self-extensionality* plus some auxiliary concepts from [3, Appendix A.2]. First recall $r = \lambda u.\, \mathsf{if}[u, \mathsf{T}, \lambda x.\bot]$ from Definition 10.1.1. Then recall the definition of $x =_G y$ from [3]:

Definition 11.6.1. $x =_G y$ *iff* $\forall \bar{z}{\in}G^{<\omega} : r(x\bar{z}) = r(y\bar{z})$

Note that $x = y$ iff $x =_\mathcal{M} y$ according to Theorem 10.1.2. Now the definition of self-extensionality reads:

Definition 11.6.2. $G \subseteq \mathcal{M}$ is *self-extensional* if
(a) $\emptyset \neq G \in \mathcal{O}^\sigma(\Phi)$
(b) $G \subseteq G^{\circ\circ}$
(c) $x =_G y \Rightarrow x \downarrow y \in G$ for all $x, y \in G$

The name "self-extensionality" refers to the property $x =_G y \Rightarrow x =_\Phi y$ which follows from property (c) above.

Note that $G^{\circ\circ} = G^\circ \to G^{\circ\circ}$ for all G [3, Fact 6.2.1].

Lemma 11.6.3.
(a) $\mathcal{H}_\alpha$ is self-extensional for all $\alpha \in \sigma$.
(b) For all $G \in \mathcal{O}^\sigma(\Phi)$ there exists a self-extensional H such that $G \subseteq H$.
(c) $\mathcal{H}_\alpha \in \mathcal{O}^\sigma(\Phi)$ for all $\alpha \in \sigma$.

(d) $\forall G{\in}\mathcal{O}^\sigma(\Phi){:}\, G^{\circ\circ} \in \mathcal{O}^\sigma(\Phi)$.

Proof of 11.6.3

(a) By transfinite induction in α using [3, Corollary 6.1.6 and Theorem A.2.1].
(b) Follows from (a) and Lemma 11.5.2(e).
(c) Follows from (a) and the definition of self-extensionality.
(d) Choose $\alpha \in \sigma$ such that $G \subseteq \mathcal{H}_\alpha$. We have $G \subseteq \mathcal{H}_\alpha \Rightarrow G^{\circ\circ} \subseteq \mathcal{H}_\alpha{}^{\circ\circ} = \mathcal{H}_{\alpha'} \subseteq \Phi$. Furthermore, $G^{\circ\circ}$ is open and essentially σ-small according to Lemma 11.4.1(a).

11.7. Restriction

Recall that $f \;/\; g \equiv \mathsf{if}[\, f\, ,\, \mathsf{T}\, ,\, \lambda x.\, gx \;!\; (fx \;/\; g)\,]$ (Definition 4.6.2(d)). If $G = \mathrm{Dom}[g]$ then $f \;/\; g$ equals $\Downarrow_G f$ of [3].

Two maps $x, y \in \mathcal{M}$ are said to be *incompatible* if they have no upper bound in $\mathcal{M}$ w.r.t. $\preceq_\mathcal{M}$. We have:

Lemma 11.7.1. Let $G = \mathrm{Dom}[g] \neq \emptyset$.
(a) $G^{\circ\delta} = \{f/g \mid f \in G^\circ\}$
(b) $G^{\circ\delta}$ is a set of pairwise incompatible elements.

Proof of 11.7.1 [3, Theorem 6.1.11].

Lemma 11.7.2. Let $G = \mathrm{Dom}[g]$ and $f \in G^\circ$.
(a) $f \;/\; g \preceq_\mathcal{M} f$
(b) $f \preceq_\mathcal{M} f' \Rightarrow f \;/\; g = f' \;/\; g$

Proof of 11.7.2

(a) By lemma [3, Lemma 6.1.9].
(b) Assume $f \preceq_\mathcal{M} f'$. Now $f \;/\; g \preceq_\mathcal{M} f' \;/\; g$ by monotonicity. Furthermore, $f' \in G^\circ$ because G° is upward closed. Thus, $f \;/\; g = f' \;/\; g$ by Lemma 11.7.1.

Lemma 11.7.3. Let $G = \mathrm{Dom}[g] \subseteq \Phi$.
(a) $\{f \;/\; g \mid f \in \Phi\} \subseteq G^{\circ\delta}$
(b) $\{f \;/\; g \mid f \in \Phi\} = G^{\circ\delta}$ if G is self-extensional.
(c) $\{az \;/\; \phi \mid z \in \Phi\} \in \mathcal{P}^\sigma(\Phi^{\circ\delta})$ if $a \in \Phi$.

Proof of 11.7.3

(a) $G \subseteq \Phi$ gives $\Phi \subseteq \Phi^\circ \subseteq G^\circ$ by Lemma 11.5.2(f) and Fact 11.3.2(a). We conclude using Lemma 11.7.1.
(b) We now prove the reverse inclusion of (a). Let $x \in G^{\circ\delta}$ and take $y = \Uparrow^*_G x$ where $\Uparrow^*_G$ is defined in [3, Section A.2]. According to [3, Lemma A.2.4] we have $y \in G^{\circ\circ}$ and $x \preceq_\mathcal{M} y$. Lemma 11.6.3(d) and $y \in G^{\circ\circ}$ gives $y \in \Phi$. From Lemma 11.7.2 we have $y/g \preceq_\mathcal{M} y$. From (a) we have $y/g \in G^{\circ\delta}$. According to Lemma 11.7.1, $G^{\circ\delta}$ is a set of incompatible maps. Hence, $x \in G^{\circ\delta}$, $y/g \in G^{\circ\delta}$, $x \preceq_\mathcal{M} y$, and $y/g \preceq_\mathcal{M} y$ gives $x = y/g \in \{f \;/\; g \mid f \in \Phi\}$.

(c) Choose $\alpha < \sigma$ such that $a \in \Phi_{\alpha'} = \Phi_\alpha^\circ \to \Phi_\alpha$. Since $\Phi \subseteq \Phi_\alpha^\circ$ (c.f. Lemma 11.5.2) we have $a\Phi \subseteq \Phi_\alpha \subseteq \Phi$. Let $A = \{az \;/\; \phi \mid z \in \Phi\} = \{x \;/\; \phi \mid x \in a\Phi\}$. By (a) we have $A \subseteq \Phi^{\circ\delta}$. From Lemma 11.6.3(a) we have $\mathcal{H}_\alpha \in \mathcal{O}^\sigma(\Phi)$. Choose $K <_c \sigma$ such that $\mathcal{H}_\alpha = K^\uparrow$. We have $A \subseteq \{x \;/\; \phi \mid x \in \Phi_\alpha\} \subseteq \{x \;/\; \phi \mid x \in \mathcal{H}_\alpha\} = \{x \;/\; \phi \mid x \in K\} <_c \sigma$.G

Lemmas 11.7.3 (a) and (b) are central; they are used thus:
11.7.3(b) $\Rightarrow$ 11.8.1(c,d) $\Rightarrow$ 12.2.1(b,e) $\Rightarrow$ 12.3.5 $\Rightarrow$ 12.3.6 $\Rightarrow$ UBT.
11.7.3(a) $\Rightarrow$ 11.8.1(b) $\Rightarrow$ 13.5.2(c) $\Rightarrow$ 13.6.3 $\Rightarrow$ 13.8.4 $\Rightarrow$ LBT.

11.8. Closure properties of $\mathcal{O}^\sigma(\Phi)$

For $\theta \in \mathcal{M}$ let $\hat\theta$ denote Dom$[\theta]$. This is consistent with $\hat\psi = \mathrm{Dom}[\psi]$.

We use below the definitions of P, Q, and R given in Definition 4.6.3 and $\Phi = \mathrm{Dom}[\phi]$. We have:

Lemma 11.8.1.

(a) $Qv \neq \bot \Rightarrow \mathrm{Dom}[Qv] = K \to \mathrm{Dom}[v]$ where $K = \{z \;/\; v \mid z \in \Phi\}$.
(b) $\mathrm{Dom}[v] \in \mathcal{O}^\sigma(\Phi) \wedge Qv \neq \bot \Rightarrow \mathrm{Dom}[Qv] \supseteq \mathrm{Dom}[v]^\circ \to \mathrm{Dom}[v]$.
(c) $\mathrm{Dom}[v] \in \mathcal{O}^\sigma(\Phi) \wedge Qv \neq \bot \Rightarrow \mathrm{Dom}[Qv] = \mathrm{Dom}[v]^\circ \to \mathrm{Dom}[v]$ if Dom$[v]$ is self-extensional.
(d) $\mathrm{Dom}[v] \in \mathcal{O}^\sigma(\Phi) \Rightarrow \mathrm{Dom}[Qv] \in \mathcal{O}^\sigma(\Phi)$.
(e) $Rf\theta bc \neq \bot \Rightarrow \mathrm{Dom}[Rf\theta bc] = \cup_{z\in\hat\theta}\mathrm{Dom}[f(b(cz \;/\; \theta))]$
(f) $\mathrm{Dom}[P] = \{\mathsf{T}\}$

Proof of 11.8.1

(a) $Qv \neq \bot$ and the definition of Q gives $\mathsf{D}v = \mathsf{T}$. We have

$y \in \mathrm{Dom}[Qv]$	
$\Leftrightarrow Qvy = \mathsf{T}$	Definition of Dom
$\Leftrightarrow \ddot\forall z.\, v(y(z \;/\; v)) = \mathsf{T}$	$\mathsf{D}v = \mathsf{T}$ and the definition of Q
$\Leftrightarrow \forall z{\in}\Phi\colon v(y(z \;/\; v)) = \mathsf{T}$	Properties of $\ddot\forall$
$\Leftrightarrow \forall z{\in}\Phi\colon y(z \;/\; v) \in \mathrm{Dom}[v]$	Definition of Dom
$\Leftrightarrow y \in K \to \mathrm{Dom}[v]$	Definition of K and $\to$

(b) Follows from (a) and Lemma 11.7.3(a)
(c) Follows from (a) and Lemma 11.7.3(b).
(d) Using Lemma 11.6.3(b), choose $H \in \mathcal{O}^\sigma(\Phi)$ such that H is open, self-extensional, and contains Dom$[v]$ as a subset. Let w be the characteristic map of H. Now $v \preceq_\mathcal{M} w$ so $Qv \preceq_\mathcal{M} Qw$ by monotonicity. Hence, $\mathrm{Dom}[Qv] \subseteq \mathrm{Dom}[Qw] = \mathrm{Dom}[w]^\circ \to \mathrm{Dom}[w] \in \mathcal{O}^\sigma(\Phi)$ by 11.8.1(c) and 11.4.1(b). Thus, $\mathrm{Dom}[Qv] \subseteq \Phi$. It remains to prove $\mathrm{Dom}[Qv] \in \mathcal{O}^\sigma(\mathcal{M})$.
If $v = \bot$ then $\mathrm{Dom}[Qv] = \emptyset \in \mathcal{O}^\sigma(\mathcal{M})$. Now assume $v \neq \bot$. Let $G = \mathrm{Dom}[v]$. Assume $G \in \mathcal{O}^\sigma(\Phi)$. Choose a σ-small $H \subseteq \mathcal{M}_c$ such that $G = {\uparrow}H$ and let K be as in (a). Now K and $K \to H$ are σ-small and $\mathrm{Dom}[Qv] = K \to {\uparrow}H = {\uparrow}(K \to H) \in \mathcal{O}^\sigma(\mathcal{M})$.
(e) $Rf\theta bc \neq \bot$ and the definition of R gives $\theta c = \mathsf{T}$ and $R_1 f\theta bc = \mathsf{T}$. Now:

$y \in \mathrm{Dom}[Rf\theta bc]$

$\Leftrightarrow Rf\theta bcy = \mathsf{T}$	Definition of Dom
$\Leftrightarrow R_0 f\theta bcy = \mathsf{T}$	$\theta c = \mathsf{T}$, $R_1 f\theta bc = \mathsf{T}$, and definition of R.
$\Leftrightarrow \mathsf{E}z.\,(\theta z \,!\, f(b(cz \,/\, \theta))y) = \mathsf{T}$	Definition of R_0
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon \theta z \,!\, f(b(cz \,/\, \theta))y = \mathsf{T}$	Properties of E
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon \theta z = \mathsf{T} \wedge f(b(cz \,/\, \theta))y = \mathsf{T}$	Properties of guards
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon z \in \hat{\theta} \wedge f(b(cz \,/\, \theta))y = \mathsf{T}$	Definition of $\hat{\theta}$
$\Leftrightarrow \exists z{\in}\hat{\theta}\colon f(b(cz \,/\, \theta))y = \mathsf{T}$	Trivial
$\Leftrightarrow \exists z{\in}\hat{\theta}\colon y \in \mathrm{Dom}[f(b(cz \,/\, \theta))]$	Definition of Dom
$\Leftrightarrow y \in \cup_{z\in\hat{\theta}} \mathrm{Dom}[f(b(cz \,/\, \theta)))]$	Trivial

(f) Trivial: $x \in \mathrm{Dom}[P] \Leftrightarrow Px = \mathsf{T} \Leftrightarrow \mathsf{if}[\,x\,,\,\mathsf{T}\,,\,\bot\,] = \mathsf{T} \Leftrightarrow x = \mathsf{T}$.

12. Proof of the Upper Bound Theorem (UBT)

Recall that UBT states that $\hat{\psi} \subseteq \Phi$. In this section we only need that σ is inaccessible, that $\mathcal{M}$ is an MT standard κ-quasimodel where $\kappa > \sigma$ is regular, that Y acts as $\mathsf{Y}_{\mathrm{Tarski}}$, the least fixed point operator of $\mathcal{M}$ w.r.t. the model order $\preceq_{\mathcal{M}}$, and that application is monotonic w.r.t. $\preceq_{\mathcal{M}}$.

12.1. Flat order

For $u, v \in \mathcal{M}$ define the "flat order" $u \leq_{\bot} v$ thus:

Definition 12.1.1. $u \leq_{\bot} v \Leftrightarrow u = \bot \vee u = v$

We have $u \leq_{\bot} v \Rightarrow u \preceq_{\mathcal{M}} v$.

Lemma 12.1.2. *Assume*
(1) $a \preceq_{\mathcal{M}} b$
(2) $\forall c, d{\in}\mathcal{M}\colon (c \preceq_{\mathcal{M}} d \Rightarrow fc \leq_{\bot} gd)$
(3) $\theta \preceq_{\mathcal{M}} \phi$
We have
(4) $\bar{S} f\theta a \leq_{\bot} \bar{S} g\theta b$

Proof of 12.1.2 The lemma is trivial if $\bar{S} f\theta a = \bot$. Now assume
(5) $\bar{S} f\theta a \neq \bot$
From (5) and the definition of $\bar{S}$ we have that (6), (7), or (8) holds:
(6) $a = \mathsf{T}$
(7) $a \in \mathcal{F} \wedge a\mathsf{T} = \mathsf{T}$
(8) $a \in \mathcal{F} \wedge a\mathsf{T} \in \mathcal{F}$
We proceed by a proof by cases

Case 1. Assume (6). From (1) and (6) we have $b = \mathsf{T}$. By the definition of $\bar{S}$ we then have $\bar{S} f\theta a = P = \bar{S} g\theta b$ which proves $\bar{S} f\theta a \leq_{\bot} \bar{S} g\theta b$.

Case 2. Assume (7). We have

(9)	$b \in \mathcal{F} \wedge b\mathsf{T} = \mathsf{T}$	by 1, 7
(10)	$\bar{S} f\theta a = Q(f(a\mathsf{F}))$	by 7, def. of $\bar{S}$

(11)	$\bar{S}g\theta b = Q(g(b\mathsf{F}))$	by 9, def. of $\bar{S}$
(12)	$Q(f(a\mathsf{F})) \neq \bot$	by 5, 10
(13)	$f(a\mathsf{F}) \neq \bot$	by 12, def. of Q
(14)	$f(a\mathsf{F}) \leq_\bot g(b\mathsf{F})$	by 1, 2
(15)	$f(a\mathsf{F}) = g(b\mathsf{F})$	by 13, 14
(16)	$\bar{S}f\theta a = \bar{S}g\theta b$	by 10, 11, 15
(17)	$\bar{S}f\theta a \leq_\bot \bar{S}g\theta b$	by 16

Case 3. Assume (8). We have

(18)	$b \in \mathcal{F} \wedge b\mathsf{T} \in \mathcal{F}$	by 1, 8
(19)	$\bar{S}f\theta a = \theta(a\mathsf{F}) \mathbin{!} R_1 f\theta(a\mathsf{T})(a\mathsf{F}) \mathbin{!} R_0 f\theta(a\mathsf{T})(a\mathsf{F})$	by 8, def. of $\bar{S}$
(20)	$\bar{S}g\theta b = \theta(b\mathsf{F}) \mathbin{!} R_1 g\theta(b\mathsf{T})(b\mathsf{F}) \mathbin{!} R_0 g\theta(b\mathsf{T})(b\mathsf{F})$	by 18, def. of $\bar{S}$
(21)	$\theta(a\mathsf{F}) = \mathsf{T}$	by 5, 19
(22)	$\theta(a\mathsf{F}) = \theta(b\mathsf{F})$	by 1, 21
(23)	$R_1 f\theta(a\mathsf{T})(a\mathsf{F}) = \mathsf{T}$	by 5, 19
(24)	$\ddot{\forall} z.\, \mathsf{D}(f(a\mathsf{T}(a\mathsf{F}z \mathbin{/} \theta))) = \mathsf{T}$	by 23, def. of R_1
(25)	$\forall z{\in}\Phi\colon f(a\mathsf{T}(a\mathsf{F}z \mathbin{/} \theta)) \neq \bot$	by 24
(26)	$\forall z{\in}\hat{\theta}\colon f(a\mathsf{T}(a\mathsf{F}z \mathbin{/} \theta)) \neq \bot$	by 3, 25
(27)	$\forall z{\in}\hat{\theta}\colon f(a\mathsf{T}(a\mathsf{F}z \mathbin{/} \theta)) = g(b\mathsf{T}(b\mathsf{F}z \mathbin{/} \theta))$	by 1, 2, 26
(28)	$R_1 f\theta(a\mathsf{T})(a\mathsf{F}) = R_1 g\theta(b\mathsf{T})(b\mathsf{F})$	by 27, def. of R_1
(29)	$R_0 f\theta(a\mathsf{T})(a\mathsf{F}) = R_0 g\theta(b\mathsf{T})(b\mathsf{F})$	by 27, def. of R_0
(30)	$\bar{S}f\theta a = \bar{S}g\theta b$	by 19, 20, 22, 28, 29
(31)	$\bar{S}f\theta a \leq_\bot \bar{S}g\theta b$	by 30

12.2. Limited size

Lemma 12.2.1. Assume $f, a, b, c, v, \theta \in \mathcal{M}$, $\theta \preceq_{\mathcal{M}} \phi$, and $\forall x{\in}\mathcal{M}\colon \mathrm{Dom}[fx] \in \mathcal{O}^\sigma(\Phi)$. We have:

(a) $\mathrm{Dom}[P] \in \mathcal{O}^\sigma(\Phi)$
(b) $\mathrm{Dom}[Q(fv)] \in \mathcal{O}^\sigma(\Phi)$
(c) $\mathrm{Dom}[Rf\theta bc] \in \mathcal{O}^\sigma(\Phi)$
(d) $\bar{S}f\theta a \in \{\bot, P, Q(f(a\mathsf{F})), Rf\theta(a\mathsf{T})(a\mathsf{F})\}$
(e) $\mathrm{Dom}[\bar{S}f\theta a] \in \mathcal{O}^\sigma(\Phi)$

Proof of 12.2.1

(a) Follows from 11.8.1(f)
(b) Follows from 11.8.1(d).
(c) If $Rf\theta bc = \bot$ then $\mathrm{Dom}[Rf\theta bc] = \emptyset \in \mathcal{O}^\sigma(\Phi)$. Now assume $Rf\theta bc \neq \bot$. From the definition of R we have $\theta c = \mathsf{T}$ so $c \in \Phi$. Hence, $\{cz \mathbin{/} \phi \mid z \in \Phi\} \subseteq \Phi^{\circ\delta}$ is σ-small by Lemma 11.7.3(c). Thus, $\{b(cz \mathbin{/} \phi) \mid z \in \Phi\}$ is σ-small so $\{b(cz \mathbin{/} \phi) \mid z \in \hat{\theta}\}$ is also σ-small. Hence, $\cup_{z\in\hat{\theta}}\mathrm{Dom}[f(b(cz \mathbin{/} \phi))] \in \mathcal{O}^\sigma(\Phi)$. Combined with 11.8.1(e) this gives $\mathrm{Dom}[Rf\theta bc] \in \mathcal{O}^\sigma(\Phi)$.
(d) $\bar{S}f\theta ay$

$= \mathsf{if}[\, a\,,\, P\,,\, \mathsf{if}[\, a\mathsf{T}\,,\, Q(f(a\mathsf{F}))\,,\, Rf\theta(a\mathsf{T})(a\mathsf{F})\,]\,]y$ — Definition of $\bar{S}$

$\in \{\bot, P, Q(f(a\mathsf{F})), Rf\theta(a\mathsf{T})(a\mathsf{F})\}$ — Properties of if

(e) Follows from 12.2.1(a-d).

12.3. Proof of UBT

For UBT we need the minimality of Y w.r.t. $\preceq_{\mathcal{M}}$, i.e. that Y is interpreted as $\mathsf{Y}_{\mathrm{Tarski}}$.

Definition 12.3.1. Define $s_\alpha \in \mathcal{M}$ by:
(a) $s_0 = \bot$
(b) $s_{\alpha'} = \bar{S}s_\alpha(\sqcup s_\alpha)$
(c) $s_\delta = \sup_{\alpha\in\delta} s_\alpha$ for limit ordinals δ

Note that $\sqcup s_\alpha$ denotes the map $\sqcup$ applied to the map s_α. Above, α is a free variable in $\sqcup s_\alpha$ but a bound variable in $\sup_{\alpha\in\delta} s_\alpha$.

Fact 12.3.2.
(a) $\alpha \le \beta \Rightarrow s_\alpha \preceq_{\mathcal{M}} s_\beta$
(b) $\sup_{\alpha\in\delta} s_\alpha$ exists
(c) $s = \mathsf{Y}S = s_\kappa$
(d) $\psi = \sqcup s_\kappa$

We now define $\bar{\theta}_\alpha$ for all ordinals α and all $\theta \in \mathcal{M}$. We shall use $\bar{\theta}_\alpha$ for $\theta = \phi$ and $\theta = \psi$. In the case $\theta = \phi$ we are going to have $s_\alpha \preceq_{\mathcal{M}} \bar{\phi}_\alpha$ and $\sqcup\bar{\phi}_\kappa \preceq_{\mathcal{M}} \phi$ so that $\psi = \sqcup s_\kappa \preceq_{\mathcal{M}} \sqcup\bar{\phi}_\kappa \preceq_{\mathcal{M}} \phi$ which is the essence of UBT.

Definition 12.3.3. For ordinals α and for $\theta \in \mathcal{M}$ define $\bar{\theta}_\alpha$ thus:
(a) $\bar{\theta}_0 = \bot$
(b) $\bar{\theta}_{\alpha'} = \bar{S}\bar{\theta}_\alpha\theta$
(c) $\bar{\theta}_\delta = \sup_{\alpha\in\delta} \bar{\theta}_\alpha$ for limit ordinals δ

Fact 12.3.4.
(a) $\alpha \le \beta \Rightarrow \bar{\theta}_\alpha \preceq_{\mathcal{M}} \bar{\theta}_\beta$
(b) $\sup_{\alpha\in\delta} \bar{\theta}_\alpha$ exists.
(c) $s = \bar{\psi}_\kappa$.
(d) $\psi = \sqcup\bar{\psi}_\kappa$.

Lemma 12.3.5. For $\theta \preceq_{\mathcal{M}} \phi$ and for all ordinals β we have
(a) $\forall u{\in}\mathcal{M}\colon \mathrm{Dom}[\bar{\theta}_\beta u] \in \mathcal{O}^\sigma(\Phi)$
(b) $\forall\gamma{\in}\beta\forall u,v{\in}\mathcal{M}\colon (u \preceq_{\mathcal{M}} v \Rightarrow \bar{\theta}_\gamma u \le_\bot \bar{\theta}_\beta v)$

Proof of 12.3.5 We now prove (a) and (b) together using complete induction. The proof has six parts: We prove (a) and (b) for $\beta = 0$, we prove (b) and (a) for $\beta = \delta$ assuming (a) and (b) for $\beta < \delta$ for limit ordinals δ, and we prove (a) and (b) for $\beta = \alpha'$ assuming (a) and (b) for $\beta \le \alpha$. The proof of (b) for $\beta = \alpha'$ in turn uses induction in γ.

Recall that $\mathrm{Dom}[u]$ is open for all $u \in \mathcal{M}$, c.f. Remark 8.2.7.

Proof of (a) for $\beta = 0$: $\mathrm{Dom}[\bar{\theta}_\beta u] = \mathrm{Dom}[\bot] = \emptyset \in \mathcal{O}^\sigma(\Phi)$.

Proof of (b) for $\beta = 0$: Trivial.

Proof of (b) for $\beta = \delta$: Assume $\gamma \in \delta$, $u, v \in \mathcal{M}$, and $u \preceq_{\mathcal{M}} v$. The proof is trivial for $\bar{\theta}_\gamma u = \bot$. Now assume $\bar{\theta}_\gamma u \ne \bot$. (b) for $\beta < \delta$ gives $\bar{\theta}_\gamma u \le_\bot \bar{\theta}_\beta v$ if $\gamma \le \beta < \delta$ so $\bar{\theta}_\gamma u = \bar{\theta}_\beta v$ if $\gamma \le \beta < \delta$. Thus, $\bar{\theta}_\gamma u = \bar{\theta}_\delta v$ so $\bar{\theta}_\gamma u \le_\bot \bar{\theta}_\delta v$.

Proof of (a) for $\beta = \delta$: Assume $u \in \mathcal{M}$. If $\bar{\theta}_\beta u = \bot$ for all $\beta < \delta$ then $\bar{\theta}_\delta = \bot$ so $\mathrm{Dom}[\bar{\theta}_\delta u] = \emptyset \in \mathcal{O}^\sigma(\Phi)$. If $\bar{\theta}_\gamma u \neq \bot$ for some $\gamma < \delta$ then $\bar{\theta}_\gamma u = \bar{\theta}_\beta u$ for $\gamma \leq \beta < \delta$ as above so $\bar{\theta}_\gamma u = \bar{\theta}_\delta u$. (a) for $\beta < \delta$ gives $\bar{\theta}_\gamma u \in \mathcal{O}^\sigma(\Phi)$ so $\bar{\theta}_\delta u \in \mathcal{O}^\sigma(\Phi)$.

Proof of (a) for $\beta = \alpha'$: From (a) for $\beta = \alpha$ we have $\forall u {\in} \mathcal{M}\colon \mathrm{Dom}[\bar{\theta}_\alpha u] \in \mathcal{O}^\sigma(\Phi)$. Thus, $\mathrm{Dom}[\bar{\theta}_{\alpha'} u] = \mathrm{Dom}[\bar{S}\bar{\theta}_\alpha \theta u] \in \mathcal{O}^\sigma(\Phi)$ by 12.2.1(e).

Proof of (b) for $\beta = \alpha'$: The proof is by induction in γ, but we only use the inductive hypothesis on γ in the limit case. In the successor case we use the inductive hypothesis on β.

For $\gamma = 0$ we have $\bar{\theta}_\gamma u = \bot \leq_\bot \bar{\theta}_{\alpha'} v$.

If $\gamma \in \beta = \alpha'$ is a successor ordinal then let $\tilde{\gamma}$ be the predecessor of γ. Now $\tilde{\gamma}' = \gamma \in \beta = \alpha'$ so $\tilde{\gamma} \in \alpha$. From (b) applied to $\tilde{\gamma}$ and α we have $\forall c, d {\in} \mathcal{M}\colon (c \preceq_{\mathcal{M}} d \Rightarrow \bar{\theta}_{\tilde{\gamma}} c \leq_\bot \bar{\theta}_\alpha d)$. Thus, by 12.1.2 we have $\bar{\theta}_\gamma u = \bar{S}\bar{\theta}_{\tilde{\gamma}} \theta u \leq_\bot \bar{S}\bar{\theta}_\alpha \theta v = \bar{\theta}_{\alpha'} v$.

Now let ϵ be a limit ordinal and assume

(c) $\qquad \forall u, v {\in} \mathcal{M}\colon (u \preceq_{\mathcal{M}} v \Rightarrow \bar{\theta}_\gamma u \leq_\bot \bar{\theta}_{\alpha'} v)$

for all $\gamma \in \epsilon$. Assume $u, v \in \mathcal{M}$ and $u \preceq_{\mathcal{M}} v$. If $\bar{\theta}_\gamma u = \bot$ for all $\gamma \in \epsilon$ then $\bar{\theta}_\epsilon u = \bot \leq_\bot \bar{\theta}_{\alpha'} v$. If $\bar{\theta}_\gamma u \neq \bot$ for some $\gamma \in \epsilon$ then $\bar{\theta}_\epsilon u = \bar{\theta}_\gamma u \leq_\bot \bar{\theta}_{\alpha'} v$ by (c).

Lemma 12.3.6. For ordinals α and $a \in \mathcal{M}$ we have
- (a) $\sqcup \bar{\phi}_\alpha \preceq_{\mathcal{M}} \phi$
- (b) $s_\alpha \preceq_{\mathcal{M}} \bar{\phi}_\alpha$
- (c) $\psi \preceq_{\mathcal{M}} \phi$
- (d) $\mathrm{Dom}[sa] \in \mathcal{O}^\sigma(\Phi)$

Proof of 12.3.6
- (a) From Lemma 12.3.5(a) we have $\mathrm{Dom}[\bar{\phi}_\alpha u] \in \mathcal{O}^\sigma(\Phi)$. Hence, $\bar{\phi}_\alpha uv = \mathsf{T} \Rightarrow v \in \Phi$. Using the properties of E we get $(\mathsf{E}u.\bar{\phi}_\alpha uv) = \mathsf{T} \Rightarrow v \in \Phi$ and $(\mathsf{E}u.\bar{\phi}_\alpha uv) \in \{\bot, \mathsf{T}\}$. Hence, $\sqcup \bar{\phi}_\alpha \preceq_{\mathcal{M}} \phi$ using the definition of $\sqcup$ (Definition 4.6.2(a)).
- (b) The proof is by transfinite induction in α. The cases where α is zero or a limit ordinal are trivial. We now assume $s_\alpha \preceq_{\mathcal{M}} \bar{\phi}_\alpha$ and prove $s_{\alpha'} \preceq_{\mathcal{M}} \bar{\phi}_{\alpha'}$. From Lemma 12.3.6(a) we have $\sqcup \bar{\phi}_\alpha \preceq_{\mathcal{M}} \phi$. From $s_\alpha \preceq_{\mathcal{M}} \bar{\phi}_\alpha$ and monotonicity we have $\sqcup s_\alpha \preceq_{\mathcal{M}} \sqcup \bar{\phi}_\alpha$. Hence, $\sqcup s_\alpha \preceq_{\mathcal{M}} \phi$ and $s_{\alpha'} = \bar{S} s_\alpha (\sqcup s_\alpha) \preceq_{\mathcal{M}} \bar{S} \bar{\phi}_\alpha \phi = \bar{\phi}_{\alpha'}$.
- (c) $\psi = \sqcup s_\kappa \preceq_{\mathcal{M}} \sqcup \bar{\phi}_\kappa \preceq_{\mathcal{M}} \phi$ by Fact 12.3.2(d), Lemma 12.3.6(b), and 12.3.6(a).
- (d) We have $\psi \preceq_{\mathcal{M}} \phi$ by (c) so $\mathrm{Dom}[\bar{\psi}_\kappa a] \in \mathcal{O}^\sigma(\Phi)$ by Lemma 12.3.5(a). Thus, $\mathrm{Dom}[sa] \in \mathcal{O}^\sigma(\Phi)$ since $s = \bar{\psi}_\kappa$ by Fact 12.3.4(c).

The Upper Bound Theorem (Theorem 11.1.2) follows trivially from Lemma 12.3.6(c) above. Lemma 12.3.6(d) is a strengthening of UBT which we shall need for proving LBT.

13. Proof of the Lower Bound Theorem (LBT)

Having proved UBT, we proceed by proving LBT, thus completing the proof of the Main Theorem. As already mentioned, the proof of LBT uses UBT (e.g.

Fact 13.6.2(c) and Lemma 13.8.3(e)) and that σ is the *smallest* inaccessible (only used in Lemma 13.7.3).

13.1. A countable collection of maps

Definition 13.1.1.

(a) $\mathsf{T}_0 \equiv \mathsf{T}$
(b) $\mathsf{T}_{n+1} \equiv \lambda x.\, \mathsf{T}_n$
(c) $N \equiv \{\mathsf{T}_0, \mathsf{T}_1, \mathsf{T}_2, \ldots\}$

13.2. Characteristic maps

Recall that we refer to elements of $\chi \equiv (\mathcal{M} \to \{\mathsf{T}, \perp\}) \cap \mathcal{F}$ as *characteristic maps.* For $G \subseteq \mathcal{M}$ we have $\mathrm{Dom}[g] = G$ for at most one $g \in \chi$. We shall refer to that g, if any, as the *characteristic map of* G. Define $\chi_\perp \equiv \chi \cup \{\perp\}$. Also recall the following facts:

Fact 13.2.1.

(a) For $g, h \in \chi$ we have $g \preceq_{\mathcal{M}} h \Leftrightarrow \mathrm{Dom}[g] \subseteq \mathrm{Dom}[h]$
(b) ϕ is the characteristic map of Φ
(c) ψ is the characteristic map of $\hat{\psi}$
(d) $h \in \chi_\perp \wedge g \preceq_{\mathcal{M}} h \Rightarrow g \in \chi_\perp$

Lemma 13.2.2.

(a) $sa = \bar{S}s\psi a$
(b) $sa \in \chi_\perp$
(c) $sa \preceq_{\mathcal{M}} \psi$
(d) If $\mathrm{Dom}[sa] \neq \emptyset$ then $sa \preceq_{\mathcal{M}} sb \Leftrightarrow \mathrm{Dom}[sa] \subseteq \mathrm{Dom}[sb]$.

Proof of 13.2.2

(a)

sa	
$= \mathsf{Y}Sa$	Definition of s
$= S(\mathsf{Y}S)a$	Property of Y
$= Ssa$	Definition of s
$= \bar{S}s(\sqcup s)a$	Definition of S
$= \bar{S}s\psi a$	Definition of ψ

(b) From $\mathsf{if}[\, y\,,\, \mathsf{T}\,,\, \perp\,] \preceq_{\mathcal{M}} \mathsf{T}$ we have $P = \lambda y.\, \mathsf{if}[\, y\,,\, \mathsf{T}\,,\, \perp\,] \preceq_{\mathcal{M}} \lambda y.\, \mathsf{T} = \mathsf{T}_1$.
From $Q\mathsf{T}_1 = \mathsf{D}\mathsf{T}_1 \,!\, \lambda y.\, \ddot{\forall} z.\, \mathsf{T}_1(y(z \,/\, v)) = \lambda y.\, \mathsf{T} = \mathsf{T}$ we have $Q\mathsf{T}_1 \preceq_{\mathcal{M}} \mathsf{T}_1$.
From $\mathsf{E}z.\cdots \preceq_{\mathcal{M}} \mathsf{T}$ we have $R\mathsf{T}_2(\sqcup\mathsf{T}_2)a = \lambda y.\, \mathsf{E}z.\cdots \preceq_{\mathcal{M}} \lambda y.\, \mathsf{T}$.
From $\bar{S}\mathsf{T}_2(\sqcup\mathsf{T}_2)a = \mathsf{if}[\, a\,,\, P\,,\, \mathsf{if}[\, a\mathsf{T}\,,\, Q(\mathsf{T}_2(aF))\,,\, R\mathsf{T}_2(\sqcup\mathsf{T}_2)a\,]\,]$ we have $\bar{S}\mathsf{T}_2(\sqcup\mathsf{T}_2)a \in \{\perp, P, Q\mathsf{T}_1, RT_2(\sqcup\mathsf{T}_2)a\}$ and $\bar{S}\mathsf{T}_2(\sqcup\mathsf{T}_2)a \preceq_{\mathcal{M}} \mathsf{T}_1$.
Hence, $S\mathsf{T}_2 = \lambda a.\, \bar{S}\mathsf{T}_2(\sqcup\mathsf{T}_2)a \preceq_{\mathcal{M}} \lambda a.\, \mathsf{T}_1 = \mathsf{T}_2$ so, by the minimality of Y we have $s = \mathsf{Y}S \preceq_{\mathcal{M}} \mathsf{T}_2$ and $sa \preceq_{\mathcal{M}} \mathsf{T}_2 a = \mathsf{T}_1$. $sa \preceq_{\mathcal{M}} \mathsf{T}_1$ combined with $\mathsf{T}_1 \in \chi$ gives $sa \in \chi_\perp$.

(c) From $\psi = \sqcup s$ we have $\mathrm{Dom}[\psi] = \sqcup_{a \in \mathcal{M}} \mathrm{Dom}[sa]$ so $\mathrm{Dom}[sa] \subseteq \mathrm{Dom}[\psi]$ which proves $sa \preceq_{\mathcal{M}} \psi$.

(d) If $sa \neq \bot$ and $sb \neq \bot$ then the lemma follows from 13.2.1(a) and 13.2.2(b). The lemma is trivially true if $sa = \bot$. Actually, $sa \preceq_{\mathcal{M}} sb \Leftrightarrow \mathrm{Dom}[sa] \subseteq \mathrm{Dom}[sb]$ only fails for $(sa = \lambda x.\bot) \wedge (sb = \bot)$, which is prevented by $\mathrm{Dom}[sa] \neq \emptyset$.

13.3. Cardinality

Definition 13.3.1. For all $g, h \in \mathcal{M}$ define $f \mathbin{//} g \equiv \{x \mathbin{/} g \mid x \in \mathrm{Dom}[f]\}$.

Fact 13.3.2. Assume $\mathrm{Dom}[g] \subseteq \mathrm{Dom}[h]$

(a) $gy \mathbin{!} z = gy \mathbin{!} (hy \mathbin{!} z)$

(b) $f \mathbin{/} g = (f \mathbin{/} h) \mathbin{/} g$

Recall from Section 11.4 that $G <_c H$ if G has smaller cardinality than H.

Lemma 13.3.3.

(a) $\mathrm{Dom}[h] \subseteq \mathrm{Dom}[h'] \Rightarrow g \mathbin{//} h \leq_c g \mathbin{//} h'$

(b) $\mathrm{Dom}[g] \subseteq \mathrm{Dom}[g'] \wedge \mathrm{Dom}[h] \subseteq \mathrm{Dom}[h'] \Rightarrow g \mathbin{//} h \leq_c g' \mathbin{//} h'$

Proof of 13.3.3

(a) Define $k(x) = x \mathbin{/} h$. We prove the lemma by proving that k is a surjective function from $g \mathbin{//} h'$ to $g \mathbin{//} h$: Suppose $y \in g \mathbin{//} h = \{x \mathbin{/} h \mid x \in \mathrm{Dom}[g]\}$. Select $x \in \mathrm{Dom}[g]$ such that $y = x \mathbin{/} h$. Define $z = x \mathbin{/} h'$. We have $z \in \{x \mathbin{/} h' \mid x \in \mathrm{Dom}[g]\} = g \mathbin{/} h'$ and $k(z) = (x \mathbin{/} h') \mathbin{/} h = x \mathbin{/} h = y$.

(b) $\mathrm{Dom}[g] \subseteq \mathrm{Dom}[g']$ trivially gives $g \mathbin{//} h \leq_c g' \mathbin{//} h$ which combined with 13.3.3(a) gives $g \mathbin{//} h \leq_c g' \mathbin{//} h'$.

13.4. Pairs

Define $x{::}y \equiv \lambda z.\, \mathsf{if}[\, z \,,\, x \,,\, y \,]$.

Fact 13.4.1.

(a) $(x{::}y) \in \mathcal{F}$

(b) $(x{::}y)\mathsf{T} = x$

(c) $(x{::}y)\mathsf{F} = y$

13.5. Analysis of s

Lemma 13.5.1.

(a) $s\mathsf{T} = P$

(b) $\mathrm{Dom}[s\mathsf{T}] = \{\mathsf{T}\}$

Proof of 13.5.1

(a)
$$\begin{array}{lll} & s\mathsf{T} & \\ = & \bar{S}s\psi a & \text{13.2.2(a)} \\ = & P & \text{Definition of } \bar{S} \end{array}$$

(b)
$$\begin{array}{lll} & \mathrm{Dom}[s\mathsf{T}] & \\ = & \mathrm{Dom}[P] & \text{13.5.1(a)} \\ = & \{T\} & \text{11.8.1(f)} \end{array}$$

Lemma 13.5.2.

(a) $s(\mathsf{T}{::}a) = Q(sa)$

(b) $sa \neq \perp \Rightarrow Q(sa) \neq \perp$

(c) $sa \neq \perp \Rightarrow \mathrm{Dom}[s(\mathsf{T}{::}a)] \supseteq \mathrm{Dom}[sa]^\circ \to \mathrm{Dom}[sa]$

Proof of 13.5.2

(a)
$$\begin{array}{lll} & s(\mathsf{T}{::}a) & \\ = & \bar{S}s\psi(\mathsf{T}{::}a) & \text{13.2.2(a)} \\ = & Q(s((\mathsf{T}{::}a)\mathsf{F})) & \text{Definition of } \bar{S} \\ = & Q(sa) & \text{13.4.1(c)} \end{array}$$

(b) From $sa \neq \perp$ we have $\mathsf{D}(sa) = \mathsf{T}$ so
$$\begin{array}{lll} & Q(sa) & \\ = & \mathsf{D}(sa)\ !\ \lambda y. \cdots & \text{Definition of } Q \\ = & \mathsf{T}\ !\ \lambda y. \cdots & \text{From the assumption} \\ = & \lambda y. \cdots & \text{Definition of guards} \\ \neq & \perp & \text{Trivial} \end{array}$$

(c) We have $\mathrm{Dom}[sa] \in \mathcal{O}^\sigma(\Phi)$ by 12.3.6(d) and $Q(sa) \neq \perp$ by 13.5.2(b). Hence,
$$\begin{array}{lll} & \mathrm{Dom}[s(\mathsf{T}{::}a)] & \\ = & Q(sa) & \text{13.5.2(a)} \\ \supseteq & \mathrm{Dom}[sa]^\circ \to \mathrm{Dom}[sa] & \text{11.8.1(b)} \end{array}$$

Lemma 13.5.3. Assume $f \in \mathcal{F}$, $\psi a = \mathsf{T}$, and $\forall z{\in}\Phi\colon s(f(az \;/\; \psi)) \neq \perp$

(a) $R_1 s\psi(f{::}a) = \mathsf{T}$

(b) $s(f{::}a) = \lambda y.\, \mathsf{E}z.\, (\psi z\ !\ s(f(az \;/\; \psi))y)$

(c) $s(f{::}a) \neq \perp$

(d) $\mathrm{Dom}[s(f{::}a)] = \cup_{z \in \hat{\psi}} \mathrm{Dom}[s(f(az \;/\; \psi))]$

Proof of 13.5.3

(a)
$$\begin{array}{lll} & R_1 s\psi(f{::}a) & \\ = & \ddot{\forall} z.\, \mathsf{D}(s((f{::}a)\mathsf{T}((f{::}a)\mathsf{F}z \;/\; \psi))) = \mathsf{T} & \text{Definition of } R_1 \\ = & \ddot{\forall} z.\, \mathsf{D}(s(f(az \;/\; \psi))) = \mathsf{T} & \text{13.4.1} \\ = & \mathsf{T} & \text{Third assumption} \end{array}$$

(b)
$$\begin{array}{lll} & s(f{::}a) & \\ = & \bar{S}s\psi(f{::}a) & \text{13.2.2(a)} \\ = & Rs\psi(f{::}a) & \text{Definition of } \bar{S} \\ = & \psi((f{::}a)\mathsf{F})\ !\ R_1 s\psi(f{::}a)\ !\ R_0 s\psi(f{::}a) & \text{Definition of } R \\ = & \psi a\ !\ R_1 s\psi(f{::}a)\ !\ R_0 s\psi(f{::}a) & \text{13.4.1} \\ = & R_1 s\psi(f{::}a)\ !\ R_0 s\psi(f{::}a) & \text{Second assumption} \\ = & R_0 s\psi(f{::}a) & \text{13.5.3(a)} \\ = & \lambda y.\, \mathsf{E}z.\, (\theta z\ !\ s((f{::}a)\mathsf{T}((f{::}a)\mathsf{F}z \;/\; \theta))y) & \text{Definition of } R_0 \\ = & \lambda y.\, \mathsf{E}z.\, (\theta z\ !\ s(f(az \;/\; \theta))y) & \text{13.4.1} \end{array}$$

(c) Follows from 13.5.3(b)

(d) $\quad y \in \mathrm{Dom}[s(f{::}a)]$

$\Leftrightarrow s(f{::}a)y = \mathsf{T}$	Definition of Dom
$\Leftrightarrow \mathsf{E}z.\,(\psi z \,!\, s(f(az \,/\, \psi))y) = \mathsf{T}$	13.5.3(b)
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon (\psi z \,!\, s(f(az \,/\, \psi))y) = \mathsf{T}$	Properties of E
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon (\psi z = \mathsf{T} \wedge s(f(az \,/\, \psi))y) = \mathsf{T}$	Properties of guards
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon (z \in \mathrm{Dom}[\psi] \wedge s(f(az \,/\, \psi))y) = \mathsf{T}$	Definition of Dom
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon (z \in \hat{\psi} \wedge s(f(az \,/\, \psi))y) = \mathsf{T}$	Definition of $\hat{\psi}$
$\Leftrightarrow \exists z{\in}\hat{\psi}\colon s(f(az \,/\, \psi))y = \mathsf{T}$	Trivial
$\Leftrightarrow \exists z{\in}\hat{\psi}\colon y \in \mathrm{Dom}[s(f(az \,/\, \psi))]$	Definition of Dom
$\Leftrightarrow y \in \cup_{z\in\hat{\psi}} \mathrm{Dom}[s(f(az \,/\, \psi))]$	Definition of Dom

13.6. Lower bounds

Recall $f \,/\!/\, g \equiv \{x \,/\, g \mid x \in \mathrm{Dom}[f]\}$ from Definition 13.3.1. Now define:

Definition 13.6.1. $\mathrm{self}[\,g\,] \equiv g \,/\!/\, g$

Fact 13.6.2. Suppose $x \in \Phi$ and $G = \mathrm{Dom}[g] \subseteq \Phi$ and let b be any map.

(a) $x \,/\, g \in G^{\circ\delta}$ (Lemma 11.7.3(a))
(b) $\mathrm{self}[\,g\,] \subseteq G^{\circ\delta}$
(c) $\mathrm{self}[\,sb\,] \subseteq \mathrm{Dom}[sb]^{\circ\delta}$ (because of UBT)

A key to proving LBT is to prove that $\mathrm{self}[\,sb\,]$ can have arbitrarily large cardinality below σ which uses both that $\mathrm{self}[\,sb\,]$ can be large and that σ is the smallest inaccessible.

Lemma 13.6.3.

(a) $\mathrm{self}[\,s\mathsf{T}\,] = \{\mathsf{T}\}$
(b) $\mathrm{self}[\,sb\,] \neq \emptyset \Rightarrow sb \neq \bot$
(c) $\mathsf{T} \in \mathrm{self}[\,sb\,] \Rightarrow 2 \leq_c \mathrm{self}[\,s(\mathsf{T}{::}b)\,]$
(d) $2 \leq_c \mathrm{self}[\,sb\,] \Rightarrow \omega \leq_c s(\mathsf{T}{::}b) \,/\!/\, sb$
(e) $2 \leq_c \mathrm{self}[\,sb\,] \Rightarrow \mathcal{P}(\mathrm{self}[\,sb\,]) \leq_c s(\mathsf{T}{::}b) \,/\!/\, sb$

Proof of 13.6.3

(a) $\quad \mathrm{self}[\,s\mathsf{T}\,]$

$= s\mathsf{T} \,/\!/\, s\mathsf{T}$	Definition of $\mathrm{self}[\,x\,]$
$= \{x \,/\, s\mathsf{T} \mid x \in \mathrm{Dom}[s\mathsf{T}]\}$	Definition of $x \,/\!/\, y$
$= \{x \,/\, s\mathsf{T} \mid x \in \mathsf{T}\}$	13.5.1(b)
$= \{\mathsf{T} \,/\, s\mathsf{T}\}$	Trivial
$= \{\mathsf{T}\}$	Definition of $x \,/\, y$

(b) We have $\mathrm{self}[\,\bot\,] = \bot \,/\!/\, \bot = \{x \,/\, \bot \mid x \in \mathrm{Dom}[\bot]\} = \{x \,/\, \bot \mid x \in \emptyset\} = \emptyset$. Hence, $\mathrm{self}[\,x\,] \neq \emptyset \Rightarrow x \neq \bot \Rightarrow x \neq \bot$.

(c) Assume $\mathsf{T} \in \mathrm{self}[\,sb\,]$. Let $G = \mathrm{Dom}[sb]$. We have $\mathsf{T} \in \mathrm{self}[\,sb\,] = sb \,/\!/\, sb = \{x \,/\, sb \mid x \in \mathrm{Dom}[sb]\}$ so $\exists x{\in}\mathrm{Dom}[sb]\colon x \,/\, sb = \mathsf{T}$. Since $x \,/\, sb = \mathsf{T}$ holds only for $x = \mathsf{T}$ this proves $\mathsf{T} \in \mathrm{Dom}[sb] = G$. From $\mathsf{T} \in \mathrm{self}[\,sb\,]$ and 13.6.3(b) we have $sb \neq \bot$ so $\mathrm{Dom}[s(\mathsf{T}{::}b)] \supseteq G^{\circ} \to G$ by 13.5.2(c). Since $\mathsf{T} \in G$ we have $\{\mathsf{T}, \lambda x.\,\mathsf{T}\} \subseteq G^{\circ} \to G \subseteq \mathrm{Dom}[s(\mathsf{T}{::}b)]$. Now let

$H = \mathrm{self}[\, s(\mathsf{T}{::}b)\,]$ and $u = (\lambda x.\, \mathsf{T}) \;/\; s(\mathsf{T}{::}b)$. We have $(\lambda x.\, \mathsf{T}) \;/\; s(\mathsf{T}{::}b) \in \mathcal{F}$ so $\{\mathsf{T}, u\}$ has two, distinct elements. Furthermore, $\mathrm{self}[\, s(\mathsf{T}{::}b)\,] = s(\mathsf{T}{::}b) \;//\; s(\mathsf{T}{::}b) = \{x \;/\; s(\mathsf{T}{::}b) \mid x \in \mathrm{Dom}[s(\mathsf{T}{::}b)]\} \supseteq \{x \;/\; s(\mathsf{T}{::}b) \mid x \in \{\mathsf{T}, \lambda x.\, \mathsf{T}\}\} = \{\mathsf{T} \;/\; s(\mathsf{T}{::}b), (\lambda x.\, \mathsf{T}) \;/\; s(\mathsf{T}{::}b)\} = \{\mathsf{T}, u\}$ proving $\mathrm{self}[\, s(\mathsf{T}{::}b)\,] \geq_c 2$.

(d) Let $G = \mathrm{Dom}[sb]$ and $V = \mathrm{self}[\, sb\,]$. Assume $2 \leq_c \mathrm{self}[\, sb\,] = V$. Choose $u, v \in V$ such that $u \neq v$. We have $\{u, v\} \subseteq \mathrm{self}[\, sb\,] = sb \;//\; sb = \{x \;/\; sb \mid x \in \mathrm{Dom}[sb]\} = \{x \;/\; sb \mid x \in G\}$. Choose $p, q \in G$ such that $p \;/\; sb = u$ and $q \;/\; sb = v$. From $\mathrm{self}[\, sb\,] \neq \emptyset$ we have $sb \neq \perp$ by 13.6.3(b) so $\mathrm{Dom}[s(\mathsf{T}{::}b)] \supseteq G^\circ \to G$. Now define

$$f_n = \lambda x.\mathsf{if}[\, x \overbrace{uu \cdots u}^{n},\, p\,,\, q\,]$$

We have $f_n \in G^\circ \to G \subseteq \mathrm{Dom}[s(\mathsf{T}{::}b)]$ for all $n \in \omega$. Now define

$$g_n = \lambda x.\mathsf{if}[\, x \overbrace{uu \cdots u}^{n},\, u\,,\, v\,]$$

We have $g_n = f_n \;/\; sb \in s(\mathsf{T}{::}b) \;//\; sb$ and all the g_n are distinct, proving $\omega \leq_c s(\mathsf{T}{::}b) \;//\; sb$.

(e) Define G, V, u, v, p, and q as above. Assume $2 \leq_c \mathrm{self}[\, sb\,] = V$. According to [3], $V \subseteq G^{\circ\delta}$ is a set of incompatible, compact maps. For all $W \subseteq V$ let f_W be the unique element of $\mathcal{F}$ for which

$$f_W x = \begin{cases} p & \text{if } x \in W^\uparrow \\ q & \text{if } x \in (G^{\circ\delta} \setminus W)^\uparrow \\ \perp & \text{otherwise} \end{cases}$$

We have $f_W \in G^\circ \to G \subseteq \mathrm{Dom}[s(\mathsf{T}{::}b)]$ for all $W \subseteq G$. Now define

$$g_W x = \begin{cases} u & \text{if } x \in W^\uparrow \\ v & \text{if } x \in (G^{\circ\delta} \setminus W)^\uparrow \\ \perp & \text{otherwise} \end{cases}$$

We have $g_W = f_W \;/\; sb \in s(\mathsf{T}{::}b) \;//\; sb$ and all the g_W are distinct, proving $\mathcal{P}(V) \leq_c s(\mathsf{T}{::}b) \;//\; sb$.

13.7. Beth numbers

For all ordinals α we now give a non-standard definition of the Beth number/cardinal B_α. The definition is non-standard in that B_0 and B_1 are non-standard and B_α is shifted two places for finite α.

We use $\mathrm{card}(S)$ to denote the cardinality of S (i.e. the smallest ordinal equinumerous to S).

Definition 13.7.1.

(a) $B_0 \equiv 1$
(b) $B_1 \equiv 2$
(c) $B_2 \equiv \omega$
(d) $B_{\alpha'} = \mathrm{card}(\mathcal{P}(B_\alpha))$ for $\alpha \geq 2$
(e) $B_\delta \equiv \cup_{\alpha\in\delta} B_\alpha$ for limit ordinals δ

We proceed by giving some definitions related to co-finality.

Definition 13.7.2.

(a) $\mathrm{cf}(\alpha)$ denotes the co-finality of α, i.e. $\mathrm{cf}(\alpha)$ is the smallest ordinal such that there exists an $f\colon \mathrm{cf}(\alpha) \rightarrow \alpha$ which is unlimited in α.
(b) $\mathrm{ccf}(\alpha)$ denotes the smallest ordinal β such that $\mathrm{cf}(\alpha) \leq_c B_\beta$.

Thus $\mathrm{cf}(\alpha)$ is always a cardinal and $\leq_c$ could be replaced by $\leq$ in (b) above. $\mathrm{ccf}(\alpha)$ does not need to be a cardinal.

We note that $\mathrm{cf}(\alpha) = 1$ if α is a successor ordinal and that $\mathrm{cf}(\alpha) = \alpha$ if, among others, α is 0, 1, ω, or an infinite successor cardinal. We recall that an ordinal α is a *regular cardinal* if, by definition, $\mathrm{cf}(\alpha) = \alpha \geq \omega$. Finally recall that σ is *inaccessible* if $\sigma > \omega$ and σ is a regular cardinal and $\gamma < \sigma \Rightarrow \mathcal{P}(\gamma) <_c \sigma$.

Lemma 13.7.3.
(a) $\alpha \leq_c B_\alpha$
(b) $\mathrm{ccf}(\alpha) \leq \mathrm{cf}(\alpha) \leq \alpha$
(c) $\mathrm{ccf}(\alpha) < \alpha$ or $\alpha = 0$ or α is inaccessible.
(d) $\mathrm{ccf}(\alpha) < \alpha$ if σ is the first inaccessible and $0 < \alpha < \sigma$.

Proof of 13.7.3
(a) By induction in α
(b) By (a), $\mathrm{cf}(\alpha) \leq_c B_{\mathrm{cf}(\alpha)}$. Hence, $\mathrm{ccf}(\alpha) \leq \mathrm{cf}(\alpha)$ by the definition of $\mathrm{ccf}(\alpha)$. $\mathrm{cf}(\alpha) \leq \alpha$ follows from the definition of $\mathrm{cf}(\alpha)$.
(c) Assume $\mathrm{ccf}(\alpha) \not< \alpha$ and $\alpha \neq 0$. We now prove that α is inaccessible. We have $\alpha = \mathrm{cf}(\alpha) = \mathrm{ccf}(\alpha)$ from (b). From $\mathrm{ccf}(1) = 0$, $\mathrm{ccf}(\omega) = 2$, and $\mathrm{ccf}(\gamma) = 1$ for $1 < \gamma < \omega$ we have $\alpha > \omega$. Since $\mathrm{cf}(\alpha) = \alpha > \omega$ it remains to prove $\gamma < \alpha \Rightarrow \mathcal{P}(\gamma) <_c \alpha$. Assume $\gamma < \alpha$. Since $\alpha = \mathrm{cf}(\alpha)$ is a limit ordinal we have $\gamma' < \alpha$. From $\gamma' < \alpha = \mathrm{ccf}(\alpha)$ and the definition of $\mathrm{ccf}(\alpha)$ we have $\mathrm{cf}(\alpha) \not\leq_c B_{\gamma'}$. Thus $\mathcal{P}(\gamma) \leq_c \mathcal{P}(B_\gamma) =_c B_{\gamma'} <_c \mathrm{cf}(\alpha) = \alpha$.
(d) Follows trivially from (c).

13.8. Growth lemma

We now define b_α such that $\mathrm{Dom}[sb_\alpha]$ and $\mathrm{self}[\,sb_\alpha\,]$ are growing in α and such that $\forall\gamma{\in}\sigma\exists\beta{\in}\sigma\colon \gamma \leq_c \mathrm{self}[\,sb_\beta\,]$.

For defining b_α we will use two auxiliary maps c_α and d_α defined in Definition 13.8.2 and three auxiliary functions f_α, g_α, and h_α defined as follows:

Definition 13.8.1.
(a) Choose $f_\alpha\colon \mathrm{cf}(\alpha) \rightarrow \alpha$ such that f_α is unlimited in α (this is possible by the definition of $\mathrm{cf}(\alpha)$).
(b) Choose $g_\alpha\colon \mathrm{self}[\,sb_{\mathrm{ccf}(\alpha)}\,] \rightarrow \mathrm{cf}(\alpha)$ such that g_α is surjective if cardinality permits and non-surjective otherwise.
(c) Choose $h_\alpha\colon \mathrm{Dom}[sb_\alpha]^{\circ\delta} \rightarrow \mathrm{Dom}[sb_\alpha]$ such that $h_\alpha y/sb_\alpha = y$ when $y \in \mathrm{self}[\,sb_\alpha\,]$. We define h_α only if $\mathrm{Dom}[sb_\alpha] \neq \emptyset$. From Fact 13.6.2(c) recall $\mathrm{self}[\,sb_\alpha\,] \subseteq \mathrm{Dom}[sb_\alpha]^{\circ\delta}$. Recall that $\mathrm{self}[\,sb_\alpha\,] = \{x \;/\; sb_\alpha \mid x \in \mathrm{Dom}[sb_\alpha]\}$ by definition so that the choice of h_α is possible.

Definition 13.8.2.

(a) $b_0 \equiv \mathsf{T}$
(b) $b_{\alpha'} \equiv \mathsf{T}{::}b_\alpha$
(c) $b_\delta \equiv c_\delta{::}d_{\mathrm{ccf}(\delta)}$ for limit ordinals δ
(d) Let c_α be the unique element of $\mathcal{F}$ for which

$$c_\alpha x \equiv \begin{cases} b_{f_\alpha(g_\alpha(x/sb_{\mathrm{ccf}(\alpha)}))} & \text{if } x \mathbin{/} sb_{\mathrm{ccf}(\alpha)} \in \mathrm{self}[\, sb_{\mathrm{ccf}(\alpha)}\,] \\ \bot & \text{otherwise} \end{cases}$$

(e) If $\mathrm{Dom}[sb_\alpha] \neq \emptyset$ let d_α be the unique element of $\mathcal{F}$ for which

$$d_\alpha x \equiv \begin{cases} h_\alpha(x \mathbin{/} sb_\alpha) & \text{if } x \mathbin{/} sb_\alpha \in \mathrm{Dom}[sb_\alpha]^{\circ\delta} \\ \bot & \text{otherwise} \end{cases}$$

Lemma 13.8.3. For (b)-(f) assume $\mathrm{Dom}[sb_\alpha] \neq \emptyset$
(a) $c_\alpha x = c_\alpha(x \mathbin{/} sb_{\mathrm{ccf}(\alpha)})$
(b) $\{d_\alpha x \mathbin{/} sb_\alpha \mid x \in \mathrm{Dom}[sb_\alpha]\} = \mathrm{self}[\, sb_\alpha\,]$
(c) $d_\alpha \in \mathrm{Dom}[sb_\alpha]^\circ \to \mathrm{Dom}[sb_\alpha]$
(d) $\psi d_\alpha = \mathsf{T}$
(e) $\{d_\alpha x \mathbin{/} sb_\alpha \mid x \in \Phi\} = \mathrm{self}[\, sb_\alpha\,]$
(f) $\{d_\alpha x \mathbin{/} sb_\alpha \mid x \in \hat{\psi}\} = \mathrm{self}[\, sb_\alpha\,]$

Proof of 13.8.3
(a) From Fact 13.3.2 we have $f \mathbin{/} g = (f \mathbin{/} g) \mathbin{/} g$. The lemma follows from $x \mathbin{/} sb_{\mathrm{ccf}(\alpha)} = (x \mathbin{/} sb_{\mathrm{ccf}(\alpha)}) \mathbin{/} sb_{\mathrm{ccf}(\alpha)}$ and the definition of c_α.
(b) Assume $\mathrm{Dom}[sb_\alpha] \neq \emptyset$. Assume $x \in \mathrm{Dom}[sb_\alpha]$. We have $\mathrm{Dom}[sb_\alpha] \subseteq \Phi$ by UBT so $x \mathbin{/} sb_\alpha \in \mathrm{self}[\, sb_\alpha\,] \subseteq \mathrm{Dom}[sb_\alpha]^{\circ\delta}$ by Fact 13.6.2(a). The definition of d_α gives $d_\alpha x = h_\alpha(x \mathbin{/} sb_\alpha)$. The definition of h_α gives $h_\alpha y \mathbin{/} sb_\alpha = y$ if $y \in \mathrm{self}[\, sb_\alpha\,]$, so $h_\alpha(x \mathbin{/} sb_\alpha) \mathbin{/} sb_\alpha = x \mathbin{/} sb_\alpha$. Thus, $d_\alpha x \mathbin{/} sb_\alpha = h_\alpha(x \mathbin{/} sb_\alpha) \mathbin{/} sb_\alpha = x \mathbin{/} sb_\alpha$. Hence, $\{d_\alpha x \mathbin{/} sb_\alpha \mid x \in \mathrm{Dom}[sb_\alpha]\} = \{x \mathbin{/} sb_\alpha \mid x \in \mathrm{Dom}[sb_\alpha]\} = \mathrm{self}[\, sb_\alpha\,]$.
(c) Assume $\mathrm{Dom}[sb_\alpha] \neq \emptyset$. If $x \in \mathrm{Dom}[sb_\alpha]^\circ$ then $x \mathbin{/} sb_\alpha \in \mathrm{Dom}[sb_\alpha]^{\circ\delta}$ by Lemma 11.7.1(a). Thus, $d_\alpha x = h_\alpha(x \mathbin{/} sb_\alpha) \in \mathrm{Dom}[sb_\alpha]$ by the definitions of d_α and h_α.
(d) Assume $\mathrm{Dom}[sb_\alpha] \neq \emptyset$. Now $sb_\alpha \neq \bot$. $d_\alpha \in \mathrm{Dom}[sb_\alpha]^\circ \to \mathrm{Dom}[sb_\alpha] \subseteq \mathrm{Dom}[s(\mathsf{T}{::}b_\alpha)]$ follows from $sb_\alpha \neq \bot$, (c), and Lemma 13.5.2(c), proving $\psi d_\alpha = \mathsf{T}$.
(e) Proof of $\supseteq$: Follows from (b) and $\Phi \supseteq \mathrm{Dom}[sb_\alpha]$. Proof of $\subseteq$: Assume $z \in \{d_\alpha x \mathbin{/} sb_\alpha \mid x \in \Phi\}$. Choose $x \in \Phi$ such that $z = d_\alpha x \mathbin{/} sb_\alpha$. Like in the proof of (b) we have $x \mathbin{/} sb_\alpha \in \mathrm{Dom}[sb_\alpha]^{\circ\delta}$ and $d_\alpha x = h_\alpha(x \mathbin{/} sb_\alpha)$. The definition of h_α and $x \mathbin{/} sb_\alpha \in \mathrm{Dom}[sb_\alpha]^{\circ\delta}$ gives $h_\alpha(x \mathbin{/} sb_\alpha) \in \mathrm{Dom}[sb_\alpha]$. Thus, $z = d_\alpha x \mathbin{/} sb_\alpha = h_\alpha(x \mathbin{/} sb_\alpha) \mathbin{/} sb_\alpha \in \mathrm{self}[\, sb_\alpha\,]$.
(f) From $\psi = \sqcup s$ we have $\mathrm{Dom}[sb_\alpha] \subseteq \hat{\psi}$. The lemma follows from (b), (e), and $\mathrm{Dom}[sb_\alpha] \subseteq \hat{\psi} \subseteq \Phi$.

Lemma 13.8.4. For all $\alpha \in \sigma$ we have:
(a) $B_\alpha \leq_c \mathrm{self}[\, sb_\alpha\,]$
(b) $\forall \beta{\in}\alpha'' \forall \gamma{\in}\beta \colon \mathrm{Dom}[sb_\gamma] \subseteq \mathrm{Dom}[sb_\beta]$

Proof of 13.8.4 We prove the conjunction of (a) and (b) by transfinite induction up to σ.

Base case. Suppose $\alpha = 0$. We now prove (a).

$$\begin{array}{ll} B_0 & \\ = 1 & \text{Definition of } B \\ \le_c \{\mathsf{T}\} & \text{Trivial} \\ = \mathrm{self}[\, s\mathsf{T}\,] & \text{13.6.3(a)} \\ = \mathrm{self}[\, sb_0\,] & \text{Definition of } b \\ = \mathrm{self}[\, sb_\alpha\,] & \alpha = 0 \end{array}$$

To prove (b) for $\alpha = 0$ we merely have to prove $\mathrm{Dom}[sb_0] \subseteq \mathrm{Dom}[sb_1]$. From 13.5.1(a) we have $s\mathsf{T} = P$ which gives $s\mathsf{T} \neq \bot$. Hence, by 13.5.2(b) we have $\mathrm{Dom}[s(\mathsf{T}::\mathsf{T})] = \mathrm{Dom}[s\mathsf{T}]^\circ \to \mathrm{Dom}[s\mathsf{T}]$. $\mathsf{T} \in \mathrm{Dom}[s\mathsf{T}]$ by 13.5.1(b) combined with the definition of $X \to Y$ gives $\mathsf{T} \in \mathrm{Dom}[s\mathsf{T}]^\circ \to \mathrm{Dom}[s\mathsf{T}]$ so $\mathsf{T} \in \mathrm{Dom}[s(\mathsf{T}::\mathsf{T})]$ which proves $\{\mathsf{T}\} \subseteq \mathrm{Dom}[s(\mathsf{T}::\mathsf{T})]$. Hence,

$$\begin{array}{ll} \mathrm{Dom}[sb_0] & \\ = \mathrm{Dom}[s\mathsf{T}] & \text{Definition of } b \\ = \{\mathsf{T}\} & \text{13.5.1(b)} \\ \subseteq \mathrm{Dom}[s(\mathsf{T}::\mathsf{T})] & \text{See above} \\ = \mathrm{Dom}[s(\mathsf{T}::b_0)] & \text{Definition of } b \\ = \mathrm{Dom}[sb_1] & \text{Definition of } b \end{array}$$

Induction step. Assume

(1) $B_\alpha \le_c \mathrm{self}[\, sb_\alpha\,]$

(2) $\forall\beta\in\alpha''\forall\gamma\in\beta\colon \mathrm{Dom}[sb_\gamma] \subseteq \mathrm{Dom}[sb_\beta]$

To prove (2) in which α is replaced by α' it is sufficient to prove $\mathrm{Dom}[sb_{\alpha'}] \subseteq \mathrm{Dom}[sb_{\alpha''}]$. From (1) we have $\mathrm{Dom}[sb_\alpha] \neq \emptyset$. Then, from (2) and 13.2.2(d) we have $sb_\alpha \preceq_{\mathcal{M}} sb_{\alpha'}$. Hence,

$$\begin{array}{ll} sb_{\alpha'} & \\ = s(\mathsf{T}::b_\alpha) & \text{Definition of } b \\ = Q(sb_\alpha) & \text{13.5.2(a)} \\ \preceq_{\mathcal{M}} Q(sb_{\alpha'}) & \text{Monotonicity} \\ = s(\mathsf{T}::b_{\alpha'}) & \text{13.5.2(a)} \\ = sb_{\alpha''} & \text{Definition of } b \end{array}$$

From $sb_{\alpha'} \preceq_{\mathcal{M}} sb_{\alpha''}$ and 13.2.2(d) we have $\mathrm{Dom}[sb_{\alpha'}] \subseteq \mathrm{Dom}[sb_{\alpha''}]$ as required.

We now prove

(3) $B_{\alpha'} \le_c \mathrm{self}[\, sb_{\alpha'}\,]$

From 13.6.3(a) we have $\mathsf{T} \in \mathrm{self}[\, s\mathsf{T}\,]$. From 13.6.3(c) we have $B_1 = 2 \le_c \mathrm{self}[\, s(\mathsf{T}::\mathsf{T})\,] = \mathrm{self}[\, sb_1\,]$ so (3) holds for $\alpha = 0$. For $\alpha = 1$ we have $\mathrm{Dom}[sb_1] \subseteq \mathrm{Dom}[sb_2]$ from (2). Hence, we may prove (3) for $\alpha = 1$ as follows:

$$\begin{array}{ll} B_2 & \\ = \omega & \text{Definition of } B \\ \le_c s(\mathsf{T}::b_1) \mathbin{//} sb_1 & \text{13.6.3(d)} \\ = sb_2 \mathbin{//} sb_1 & \text{Definition of } b \\ \le_c sb_2 \mathbin{//} sb_2 & \text{13.3.3(a) and } \mathrm{Dom}[sb_1] \subseteq \mathrm{Dom}[sb_2] \\ = \mathrm{self}[\, b_2\,] & \text{Definition of self} \end{array}$$

Now assume $\alpha \ge 2$. From $2 \le_c \mathrm{self}[\, b_1\,]$, (2), and 13.3.3(b) we have $2 \le_c \mathrm{self}[\, b_\alpha\,]$

$$
\begin{array}{lll}
& B_{\alpha'} & \\
= & \mathcal{P}(B_\alpha) & \text{Definition of } B \\
\leq_c & \mathcal{P}(\mathrm{self}[\,sb_\alpha\,]) & (1) \\
\leq_c & s(\mathsf{T}{::}b_\alpha) \mathbin{//} sb_\alpha & 13.6.3(\mathrm{e}) \\
= & sb_{\alpha'} \mathbin{//} sb_\alpha & \text{Definition of } b \\
\leq_c & sb_{\alpha'} \mathbin{//} sb_{\alpha'} & 13.3.3(\mathrm{a}) \text{ and } (2) \\
= & \mathrm{self}[\,sb_{\alpha'}\,] & \text{Definition of self}
\end{array}
$$

Limit case. Suppose $\delta \in \sigma$ is a limit ordinal. For all $\alpha \in \delta$ assume

(4) $\quad B_\alpha \leq_c \mathrm{self}[\,sb_\alpha\,]$

(5) $\quad \forall\beta{\in}\alpha''\forall\gamma{\in}\beta\colon \mathrm{Dom}[sb_\gamma] \subseteq \mathrm{Dom}[sb_\beta]$

From 13.7.3(d) we have $\mathrm{ccf}(\delta) \in \delta$. Hence, by the definition of ccf and (4) we have $\mathrm{cf}(\delta) \leq_c B_{\mathrm{ccf}(\delta)} \leq_c \mathrm{self}[\,sb_{\mathrm{ccf}(\delta)}\,]$ so $g_\delta\colon \mathrm{self}[\,sb_{\mathrm{ccf}(\delta)}\,] \to \mathrm{cf}(\delta)$ is surjective according to 13.8.1(b).

From $\mathrm{ccf}(\delta) \in \delta$ and (4) we have $\mathrm{Dom}[sb_{\mathrm{ccf}(\delta)}] \neq \emptyset$ so the conditions in 13.8.1(c), 13.8.2(e), and 13.8.3 (b)-(f) are satisfied. We have

$$
\begin{array}{lll}
& \forall z{\in}\Phi\colon s(c_\delta(d_{\mathrm{ccf}(\delta)}z \,/\, \psi)) \neq \bot & \\
\Leftrightarrow & \forall z{\in}\Phi\colon s(c_\delta(d_{\mathrm{ccf}(\delta)}z \,/\, \psi \,/\, sb_{\mathrm{ccf}(\delta)})) \neq \bot & 13.8.3(\mathrm{a}) \\
\Leftrightarrow & \forall z{\in}\Phi\colon s(c_\delta(d_{\mathrm{ccf}(\delta)}z \,/\, sb_{\mathrm{ccf}(\delta)})) \neq \bot & 13.3.2(\mathrm{b}) \text{ and } 13.2.2(\mathrm{c}) \\
\Leftrightarrow & \forall z{\in}\mathrm{self}[\,sb_{\mathrm{ccf}(\delta)}\,]\colon s(c_\delta z) \neq \bot & 13.8.3(\mathrm{e}) \\
\Leftrightarrow & \forall z{\in}\mathrm{self}[\,sb_{\mathrm{ccf}(\delta)}\,]\colon sb_{f_\delta(g_\delta(z))} \neq \bot & 13.8.2(\mathrm{d}) \\
\Leftrightarrow & \forall z{\in}\mathrm{cf}(\delta)\colon sb_{f_\delta(z)} \neq \bot & 13.8.1(\mathrm{b}) \\
\Leftrightarrow & \mathbf{true} & (4)
\end{array}
$$

Hence, we have

(6) $\quad \forall z{\in}\Phi\colon s(c_\delta(d_{\mathrm{ccf}(\delta)}z \,/\, \psi)) \neq \bot$

Furthermore,

$$
\begin{array}{lll}
& \mathrm{Dom}[sb_\delta] & \\
= & \mathrm{Dom}[s(c_\delta{::}d_{\mathrm{ccf}(\delta)})] & \text{Definition of } b \\
= & \cup_{z\in\hat\psi}\mathrm{Dom}[s(c_\delta(d_{\mathrm{ccf}(\delta)}z \,/\, \psi))] & 13.5.3(\mathrm{d}),\ 13.8.3(\mathrm{d}), \text{ and } (6) \\
= & \cup_{z\in\hat\psi}\mathrm{Dom}[s(c_\delta(d_{\mathrm{ccf}(\delta)}z \,/\, \psi \,/\, sb_{\mathrm{ccf}(\delta)}))] & 13.8.3(\mathrm{a}) \\
= & \cup_{z\in\hat\psi}\mathrm{Dom}[s(c_\delta(d_{\mathrm{ccf}(\delta)}z \,/\, sb_{\mathrm{ccf}(\delta)}))] & 13.3.2(\mathrm{b}) \text{ and } 13.2.2(\mathrm{c}) \\
= & \cup_{y\in\{d_{\mathrm{ccf}(\delta)}z/sb_{\mathrm{ccf}(\delta)}|z\in\hat\psi\}}\mathrm{Dom}[s(c_{\mathrm{ccf}(\delta)}y)] & \text{Trivial} \\
= & \cup_{y\in\mathrm{self}[\,sb_{\mathrm{ccf}(\delta)}\,]}\mathrm{Dom}[s(c_\delta y)] & 13.8.3(\mathrm{f}) \\
= & \cup_{y\in\mathrm{self}[\,sb_{\mathrm{ccf}(\delta)}\,]}\mathrm{Dom}[sb_{f_\delta(g_\delta(y))}] & 13.8.2(\mathrm{d}) \\
= & \cup_{\gamma\in\mathrm{cf}(\delta)}\mathrm{Dom}[sb_{f_\delta(\gamma)}] & 13.8.1(\mathrm{b}) \\
= & \cup_{\gamma\in\delta}\mathrm{Dom}[sb_\gamma] & 13.7.2(\mathrm{a}) \text{ and } (5)
\end{array}
$$

Hence, we have $\forall\gamma{\in}\delta\colon \mathrm{Dom}[sb_\gamma] \subseteq \mathrm{Dom}[sb_\delta]$ and $\forall\gamma{\in}\delta\colon sb_\gamma \preceq_{\mathcal{M}} sb_\delta$. Like in the deduction step, the latter implies $\forall\gamma{\in}\delta\colon sb_{\gamma'} \preceq_{\mathcal{M}} sb_{\delta'}$ and $\forall\gamma{\in}\delta\colon \mathrm{Dom}[sb_{\gamma'}] \subseteq \mathrm{Dom}[sb_{\delta'}]$. The latter implies $\mathrm{Dom}[sb_\delta] = \cup_{\gamma\in\delta}\mathrm{Dom}[sb_\gamma] \subseteq \mathrm{Dom}[sb_{\delta'}]$. Hence, we have proved (2) in which α is replaced by δ.

From $\forall\gamma{\in}\delta\colon \mathrm{Dom}[sb_\gamma] \subseteq \mathrm{Dom}[sb_\delta]$, 13.3.3(b), and (4) we have $\forall\gamma{\in}\delta\colon B_\gamma \leq_c \mathrm{self}[\,sb_\gamma\,] \leq_c \mathrm{self}[\,sb_\delta\,]$. Hence, $B_\delta = \cup_{\gamma\in\delta}B_\gamma \leq_c \mathrm{self}[\,sb_\delta\,]$ which completes the proof.

13.9. Proof of the lower bound theorem

To prove the lower bound theorem (Theorem 11.1.3) it is sufficient to prove

$$\forall G{\in}\mathcal{O}^\sigma(\hat{\psi}){:}\, G^\circ \to G \subseteq \hat{\psi}$$

To do so, assume $G \in \mathcal{O}^\sigma(\hat{\psi})$. Choose $\alpha \in \delta$ such that $G \leq_c \mathrm{self}[\, sb_\alpha\,]$. Let $h{:}\,\mathrm{self}[\, sb_\alpha\,] \to G$ be surjective. For all $x \in G$ choose $i_x \in \mathcal{M}$ such that $x \in \mathrm{Dom}[si_x]$. Let c be the unique element of $\mathcal{F}$ for which

$$cx \equiv \begin{cases} i_{h(x/sb_\alpha)} & \text{if } x \,/\, sb_\alpha \in \mathrm{self}[\, sb_\alpha\,] \\ \perp & \text{otherwise} \end{cases}$$

By a proof similar to the limit case of 13.8.4 we have $G \subseteq \mathrm{Dom}[s(c{::}d_\alpha)]$. Hence, $G^\circ \to G \subseteq \mathrm{Dom}[s(\mathsf{T}{::}c{::}d_\alpha)]$ by 13.5.2(c) so $G^\circ \to G \subseteq \hat{\psi}$ as required.

A. Computational properties of canonical pre-models

We now proceed to compare the observational, computational behavior of programs with their semantics as defined by the canonical models.

Let $\mathcal{M}_\kappa$ be the MT canonical κ-model. Modelling ε requires $\kappa > \sigma$ for an inaccessible σ, but modelling the other constructs just requires $\kappa \geq \omega$. Now assume $\kappa \geq \omega$. For all MT, $\mathrm{MT_{def}}$, and MT_0 programs d let $\underline{d}$ denote the interpretation of d in $\mathcal{M}_\kappa$.

A.1. Introduction of $\mathcal{T}_c$ and auxiliary concepts

Let $\mathbf{C}_\omega = \mathcal{P}^\omega_{\mathrm{coh}}(\mathbf{P}_\omega)$. Recall from Section 9.5 that if $p \in \mathbf{P}_\omega \subseteq \mathbf{P}$ then $\downarrow\! p \in \mathcal{M}$ is a prime map and if $c \in \mathbf{C}_\omega \subseteq \mathbf{C}$ then $\downarrow\! c \in \mathcal{M}$ is a compact map. For $p \in \mathbf{P}_\omega$ and $c \in \mathbf{C}_\omega$ we now proceed to define $\mathrm{MT_{def}}$ programs $\mathcal{T}_p$, $\mathcal{T}_c$, χ_p, and χ_c which satisfy:

$$\begin{array}{lcl} \underline{\mathcal{T}_p} & = & \downarrow\! p \\ \underline{\mathcal{T}_c} & = & \downarrow\! c \\ \underline{\chi_p} x & = & \begin{cases} \mathsf{T} & \text{if } \downarrow\! p \preceq_{\mathcal{M}} x \\ \perp & \text{otherwise} \end{cases} \\ \underline{\chi_c} x & = & \begin{cases} \mathsf{T} & \text{if } \downarrow\! c \preceq_{\mathcal{M}} x \\ \perp & \text{otherwise} \end{cases} \end{array}$$

To define the terms above, we also define a number of auxiliary concepts. For $n \in \omega$ and for n-tuples $\langle c_1, \ldots, c_n \rangle$ and $\langle c'_1, \ldots, c'_n \rangle$ in $\mathbf{C}^n_\omega$ we define

$$\langle c_1, \ldots, c_n \rangle \frown \langle c'_1, \ldots, c'_n \rangle \Leftrightarrow c_1 \frown c'_1 \wedge \cdots \wedge c_n \frown c'_n$$

and

$$\downarrow\! \langle c_1, \ldots, c_n \rangle = \langle \downarrow\! c_1, \ldots, \downarrow\! c_n \rangle$$

For $\langle m_1, \ldots, m_n \rangle$ and $\langle m'_1, \ldots, m'_n \rangle$ in $\mathcal{M}^n$ we define

$$\langle m_1, \ldots, m_n \rangle \preceq_{\mathcal{M}} \langle m'_1, \ldots, m'_n \rangle \Leftrightarrow m_1 \preceq_{\mathcal{M}} m'_1 \wedge \cdots \wedge m_n \preceq_{\mathcal{M}} m'_n$$

For sets of n-tuples $s, s' \in \mathcal{P}^\omega(\mathbf{C}^n_\omega)$ we define

$$s \frown s' \Leftrightarrow \exists \bar{c} {\in} s \exists \bar{c}' {\in} s' \colon \bar{c} \frown \bar{c}'$$

For $p, p' \in \mathbf{P}_\omega$, $c, c' \in \mathbf{C}_\omega$, $\bar{c}, \bar{c}' \in \mathbf{C}^n_\omega$, and $s, s' \in \mathcal{P}^\omega(\mathbf{C}^n_\omega)$ for which $p \not\frown p'$, $c \not\frown c'$, $\bar{c} \not\frown \bar{c}'$, and $s \not\frown s'$ we are going to define $\mathrm{MT_{def}}$ programs $\delta_{pp'}$, $\delta_{cc'}$, $\delta_{\bar{c}\bar{c}'}$, and $\delta_{ss'}$ which satisfy:

$$\begin{array}{lcll}
\underline{\delta_{pp'}} x & = & \mathsf{T} & \text{if } {\downarrow} p \preceq_{\mathcal{M}} x \\
\underline{\delta_{pp'}} x & = & \mathsf{F} & \text{if } {\downarrow} p' \preceq_{\mathcal{M}} x \\
\underline{\delta_{cc'}} x & = & \mathsf{T} & \text{if } {\downarrow} c \preceq_{\mathcal{M}} x \\
\underline{\delta_{cc'}} x & = & \mathsf{F} & \text{if } {\downarrow} c' \preceq_{\mathcal{M}} x \\
\underline{\delta_{\bar{c}\bar{c}'}} x_1 \cdots x_n & = & \mathsf{T} & \text{if } {\downarrow} \bar{c} \preceq_{\mathcal{M}} \langle x_1, \ldots, x_n \rangle \\
\underline{\delta_{\bar{c}\bar{c}'}} x_1 \cdots x_n & = & \mathsf{F} & \text{if } {\downarrow} \bar{c}' \preceq_{\mathcal{M}} \langle x_1, \ldots, x_n \rangle \\
\underline{\delta_{ss'}} x_1 \cdots x_n & = & \mathsf{T} & \text{if } \exists \bar{c} {\in} s \colon {\downarrow} \bar{c} \preceq_{\mathcal{M}} \langle x_1, \ldots, x_n \rangle \\
\underline{\delta_{ss'}} x_1 \cdots x_n & = & \mathsf{F} & \text{if } \exists \bar{c}' {\in} s' \colon {\downarrow} \bar{c}' \preceq_{\mathcal{M}} \langle x_1, \ldots, x_n \rangle
\end{array}$$

Finally, for $\bar{c} \in \mathbf{C}^n_\omega$, and $s \in \mathcal{P}^\omega(\mathbf{C}^n_\omega)$ we are going to define $\mathrm{MT_{def}}$ programs $\chi_{\bar{c}}$, and χ_s which satisfy:

$$\begin{array}{lcl}
\underline{\chi_{\bar{c}}} x_1 \cdots x_n & = & \begin{cases} \mathsf{T} & \text{if } {\downarrow} \bar{c} \preceq_{\mathcal{M}} \langle x_1, \ldots, x_n \rangle \\ \perp & \text{otherwise} \end{cases} \\
\underline{\chi_s} x_1 \cdots x_n & = & \begin{cases} \mathsf{T} & \text{if } \exists \bar{c} \in s \colon {\downarrow} \bar{c} \preceq_{\mathcal{M}} \langle x_1, \ldots, x_n \rangle \\ \perp & \text{otherwise} \end{cases}
\end{array}$$

A.2. Parallel constructs

As a supplement to parallel or define parallel and:

$$x \,\&\, y = \neg(\neg x \parallel \neg y)$$

Let ϵ be a choice function over $\mathbf{P}_\omega \cup \mathbf{C}_\omega \cup \mathbf{C}^n_\omega \cup \mathcal{P}^\omega(\mathbf{C}^n_\omega)$. For finite subsets Z of $\mathbf{P}_\omega$, $\mathbf{C}_\omega$, $\mathbf{C}^n_\omega$, or $\mathcal{P}^\omega(\mathbf{C}^n_\omega)$ and for $\mathrm{MT_{def}}$ programs a_z, $z \in Z$, define the $\mathrm{MT_{def}}$ programs $\sum_{z \in Z} a_z$ and $\prod_{z \in Z} a_z$ thus:

$$\begin{array}{lcl}
\sum_{z \in Z} a_z & = & \begin{cases} \mathsf{F} & \text{if } Z = \emptyset \\ a_{\epsilon Z} \parallel \sum_{z \in Z \setminus \{\epsilon Z\}} a_z & \text{otherwise} \end{cases} \\
\prod_{z \in Z} a_z & = & \begin{cases} \mathsf{T} & \text{if } Z = \emptyset \\ a_{\epsilon Z} \,\&\, \prod_{z \in Z \setminus \{\epsilon Z\}} a_z & \text{otherwise} \end{cases}
\end{array}$$

A.3. Definition of $\mathcal{T}_p$ and $\mathcal{T}_c$

For $p, p' \in \mathbf{P}_\omega$, $c, c' \in \mathbf{C}_\omega$, $n \in \omega$, $\bar{c}, \bar{c}' \in \mathbf{C}^n_\omega$, and $s, s' \in \mathcal{P}^\omega(\mathbf{C}^n_\omega)$ we define the following $\mathrm{MT}_{\mathrm{def}}$ programs by induction in the set rank of $p, p', c, c', \bar{c}, \bar{c}', s, s'$:

$$\begin{array}{lcll}
\mathcal{T}_t & = & \mathsf{T} & \\
\mathcal{T}_f & = & \lambda x.\, \bot_{\mathrm{Curry}} & \\
\mathcal{T}_{\langle c,p\rangle} & = & \lambda x.\, \mathsf{if}[\, \chi_c x\, ,\, \mathcal{T}_p\, ,\, \bot_{\mathrm{Curry}}\,] & \\
\chi_t & = & \lambda x.\, \mathsf{if}[\, x\, ,\, \mathsf{T}\, ,\, \bot_{\mathrm{Curry}}\,] & \\
\chi_f & = & \lambda x.\, \mathsf{if}[\, x\, ,\, \bot_{\mathrm{Curry}}\, ,\, \mathsf{T}\,] & \\
\chi_{\langle c,p\rangle} & = & \lambda x.\, \chi_p(x\mathcal{T}_c) & \\
\chi_c & = & \lambda x.\, \mathsf{if}[\, \prod_{p\in c} \chi_p x\, ,\, \mathsf{T}\, ,\, \bot_{\mathrm{Curry}}\,] & \\
\chi_{\langle c_1,\cdots,c_n\rangle} & = & \lambda x_1 \cdots x_n.\, \chi_{c_1} x_1 \,\&\, \cdots \,\&\, \chi_{c_n} x_n & \\
\chi_s & = & \lambda x_1 \cdots x_n.\, \sum_{\bar{c}\in s} \chi_{\bar{c}} x_1 \cdots x_n & \\
\delta_{tp} & = & \lambda x.\, \mathsf{if}[\, x\, ,\, \mathsf{T}\, ,\, \mathsf{F}\,] & \text{if } p \neq t \\
\delta_{pt} & = & \lambda x.\, \mathsf{if}[\, x\, ,\, \mathsf{F}\, ,\, \mathsf{T}\,] & \text{if } p \neq t \\
\delta_{c,c'} & = & \lambda x.\, \prod_{p\in c} \sum_{p'\in c'} \delta_{pp'} x & \text{if } c \not\supset c' \\
\delta_{s,s'} & = & \lambda x_1 \cdots x_n.\, \sum_{\bar{c}\in s} \prod_{\bar{c}'\in s'} \delta_{\bar{c}\bar{c}'} x_1 \cdots x_n & \text{if } s \not\supset s'
\end{array}$$

Above, the definitions of $\delta_{\langle c,p\rangle\langle c',p'\rangle}$, $\delta_{\bar{c}\bar{c}'}$, and $\mathcal{T}_c$ are missing. For $\langle c,p\rangle \not\supset \langle c',p'\rangle$ define

$$\delta_{\langle c,p\rangle\langle c',p'\rangle} = \lambda x.\, \delta_{pp'}(x\mathcal{T}_{c\cup c'})$$

In the definition above note that $\langle c,p\rangle \not\supset \langle c',p'\rangle$ implies $c \supset\!\!\!\subset c'$ and $p \not\supset p'$. From $c \supset\!\!\!\subset c'$ we have $c \cup c' \in \mathbf{C}$ and the set rank of $c \cup c'$ is the larger of the set ranks of c and c'. Thus, the set rank of $c\cup c'$ is smaller than one of the set ranks of $\langle c,p\rangle$ and $\langle c',p'\rangle$ which makes it legal to use $\mathcal{T}_{c\cup c'}$ in the recursive definition.

For $\langle c_1, \cdots, c_n\rangle \not\supset \langle c'_1, \ldots, c'_n\rangle$ define

$$\delta_{\langle c_1,\cdots,c_n\rangle\langle c'_1,\ldots,c'_n\rangle} = \lambda x_1 \cdots x_n.\, \delta_{c_i c'_i} x_i$$

where $i \in \{1, \ldots, n\}$ is the smallest index for which $c_i \not\supset c'_i$.

To define $\mathcal{T}_c$, recall the definition of $\langle \bar{c}, p\rangle$ from Section 9.4 and define

$$\begin{array}{lcl}
\mathsf{def}(n,c) & = & \{\bar{c} \in \mathbf{C}^n_\omega \mid \exists p{\in}\mathbf{P}\colon \langle \bar{c}, p\rangle \in c\} \\
\mathsf{true}(n,c) & = & \{\bar{c} \in \mathbf{C}^n_\omega \mid \langle \bar{c}, t\rangle \in c\} \\
\mathsf{false}(n,c) & = & \mathsf{def}(n,c) \setminus \mathsf{true}(n,c)
\end{array}$$

Now let ℓ be the smallest natural number for which $\mathsf{def}(\ell, c)$ is empty and then define the monstrous $\mathrm{MT}_{\mathrm{def}}$ program $\mathcal{T}_c$ thus:

$$\begin{array}{lcl}
\mathcal{T}_c & = & \mathsf{if}[\, \delta_{\mathsf{true}(0,c)\mathsf{false}(0,c)}\, ,\, \chi_{\mathsf{true}(0,c)}\, ,\, \chi_{\mathsf{false}(0,c)}\, !\, \lambda x_1. \\
& & \mathsf{if}[\, \delta_{\mathsf{true}(1,c)\mathsf{false}(1,c)} x_1\, ,\, \chi_{\mathsf{true}(1,c)} x_1\, ,\, \chi_{\mathsf{false}(1,c)} x_1\, !\, \lambda x_2. \\
& & \mathsf{if}[\, \delta_{\mathsf{true}(2,c)\mathsf{false}(2,c)} x_1 x_2\, ,\, \chi_{\mathsf{true}(2,c)} x_1 x_2\, ,\, \chi_{\mathsf{false}(2,c)} x_1 x_2\, !\, \lambda x_3. \\
& & \vdots \\
& & \mathsf{if}[\, \delta_{\mathsf{true}(\ell,c)\mathsf{false}(\ell,c)} x_1 \cdots x_\ell\, ,\, \chi_{\mathsf{true}(1,c) x_1 \cdots x_\ell}\, ,\, \bot_{\mathrm{Curry}}\,] \cdots]]]
\end{array}$$

In the definition above, $\delta_{\mathsf{true}(\ell,c)\mathsf{false}(\ell,c)} x_1 \cdots x_\ell = \delta_{\emptyset\emptyset} x_1 \cdots x_\ell = \mathsf{F}$.

Theorem A.3.1. *Let $p, p' \in \mathbf{P}_\omega$, $c, c' \in \mathbf{C}_\omega$, $\bar{c}, \bar{c}' \in \mathbf{C}_\omega^n$, and $s, s' \in \mathcal{P}^\omega(\mathbf{C}_\omega^n)$ satisfy $p \not\supset p'$, $c \not\supset c'$, $\bar{c} \not\supset \bar{c}'$, and $s \not\supset s'$. Under these conditions, $\mathcal{T}_p$, $\mathcal{T}_c$, χ_p, χ_c, $\chi_{\bar{c}}$, χ_s, $\delta_{pp'}$, $\delta_{cc'}$, $\delta_{\bar{c}\bar{c}'}$, and $\delta_{ss'}$ have the properties stated in Section A.1.*

Proof. By induction in α we have that the theorem holds for all p, p', c, c', $\bar{c}$, $\bar{c}'$, s, and s' of set rank less than α. $\square$

Corollary A.3.2. *For $p \in \mathbf{P}_\omega$ and $c \in \mathbf{C}_\omega$ the* $\mathrm{MT}_{\mathrm{def}}$ *programs $\mathcal{T}_p$ and $\mathcal{T}_c$ satisfy $\underline{\mathcal{T}_p} = \downarrow p$ and $\underline{\mathcal{T}_c} = \downarrow c$.*

Now recall the combinators $\mathsf{C}_1, \ldots, \mathsf{C}_8$, defined in Section 3.4 where C_1 and C_2 are the usual S and K combinators, respectively. We shall refer to terms built up from these combinators and functional application as MT combinator programs. We refer to C_5- and C_6-free MT combinator terms as $\mathrm{MT}_{\mathrm{def}}$ combinator programs, where C_5 and C_6 are the combinators corresponding to $\perp$ and $\mathsf{Y}f$, respectively.

For all $c \in \mathbf{C}_\omega$ let $\mathcal{T}'_c$ denote the result of applying abstraction elimination using S and K to $\mathcal{T}_c$. Thus, the $\mathrm{MT}_{\mathrm{def}}$ combinator program $\mathcal{T}'_c$ satisfies $\underline{\mathcal{T}'_c} = \underline{\mathcal{T}_c}$, so we have:

Corollary A.3.3. *For $p \in \mathbf{P}_\omega$ and $c \in \mathbf{C}_\omega$ the* $\mathrm{MT}_{\mathrm{def}}$ *combinator programs $\mathcal{T}'_p$ and $\mathcal{T}'_c$ satisfy $\underline{\mathcal{T}'_p} = \downarrow p$ and $\underline{\mathcal{T}'_c} = \downarrow c$.*

Corollaries A.3.2 and A.3.3 of course also hold for MT. They do not hold for MT_0 because parallel or is missing in MT_0.

A.4. Semantic and syntactic existence

As promised in Section 3.9:

Lemma A.4.1. $\mathcal{M}_\omega \models \mathsf{E}_{\text{semantic}} = \mathsf{E}_{\text{syntactic}}$

Proof of A.4.1 Both $\mathsf{E}_{\text{semantic}}$ and $\mathsf{E}_{\text{syntactic}}$ are characteristic functions. They satisfy

$$\begin{array}{rcll} \mathsf{E}_{\text{semantic}}\, p & = & \mathsf{T} & \text{iff} \quad px = \mathsf{T} \text{ for some map } x \\ \mathsf{E}_{\text{syntactic}}\, p & = & \mathsf{T} & \text{iff} \quad px = \mathsf{T} \text{ for some program } x \end{array}$$

Thus we need to prove

$$px = \mathsf{T} \text{ for some map } x \quad \text{iff} \quad px = \mathsf{T} \text{ for some program } x$$

The direction $\Leftarrow$ is trivial. To see $\Rightarrow$ note that if $px = \mathsf{T}$ for some map x then $py = \mathsf{T}$ for some $y \in \mathbf{C}_\omega$ so $p\mathcal{T}_y = \mathsf{T}$.

Corollary A.4.2. $\mathcal{M}_\omega \models \mathsf{E}a = \mathsf{E}_{\text{semantic}}\, a = \mathsf{E}_{\text{syntactic}}\, a$.

A.5. Computational adequacy

Recall the notions of $\mathcal{N}_t$, $\mathcal{N}_f$, and $\mathcal{N}_\perp$ from Section 3.6.

Definition A.5.1. $\mathcal{M}$ *is* computationally adequate *for a set* $\mathcal{T}$ *of* MT_0, $\mathrm{MT}_{\mathrm{def}}$, *or* MT *programs if*

$$\begin{array}{lcl} a \in \mathcal{N}_t & \Leftrightarrow & \mathcal{M} \models a = \mathsf{T} \\ a \in \mathcal{N}_f & \Leftrightarrow & \mathcal{M} \models a = \lambda x.\, ax \\ a \in \mathcal{N}_\perp & \Leftrightarrow & \mathcal{M} \models a = \perp \end{array}$$

for all a *in* $\mathcal{T}$*, where* $\mathcal{N}_t$*,* $\mathcal{N}_f$*, and* $\mathcal{N}_\perp$ *are defined using the reduction rules of* MT_0*,* $\mathrm{MT}_{\mathrm{def}}$*, and* MT*, respectively.*

As we shall see in a moment, $\mathcal{M}_\kappa$ is computationally adequate for MT_0 programs, for E-free $\mathrm{MT}_{\mathrm{def}}$ programs, and for E-free MT programs.

Any term a satisfies one of $a \in \mathcal{N}_t$, $a \in \mathcal{N}_f$, and $a \in \mathcal{N}_\perp$, and one of $\mathcal{M} \models a = \mathsf{T}$, $\mathcal{M} \models a = \lambda x.ax$, and $\mathcal{M} \models a = \perp$ (c.f. Section 8.2), so each of the three statements of Definition A.5.1 follows from the two other ones.

Each statement has a trivial direction:

$$\begin{array}{lcl} a \in \mathcal{N}_t & \Rightarrow & \mathcal{M} \models a = \mathsf{T} \\ a \in \mathcal{N}_f & \Rightarrow & \mathcal{M} \models a = \lambda x.\, ax \\ a \in \mathcal{N}_\perp & \Leftarrow & \mathcal{M} \models a = \perp \end{array}$$

Furthermore, if

$$a \in \mathcal{N}_\perp \quad \Rightarrow \quad \mathcal{M} \models a = \perp$$

then

$$\begin{array}{lcl} a \in \mathcal{N}_t & \Leftarrow & \mathcal{M} \models a = \mathsf{T} \\ a \in \mathcal{N}_f & \Leftarrow & \mathcal{M} \models a = \lambda x.\, ax \end{array}$$

follows trivially. The notion of computational adequacy of a model, as well as the notion of full abstraction, were introduced by Plotkin in [13] (for a paradigmatic simply typed lambda calculus called PCF). The definition of computational adequacy given above is equivalent to the one in [13] which merely requires $a \in \mathcal{N}_\perp \Leftrightarrow \mathcal{M} \models a = \perp$. However, MT is an untyped lambda-calculus, which, for the problems treated in this appendix, considerably increases the technicality of the proofs.

Recall that $\mathcal{M}_\kappa$ denotes the canonical κ-model. Theorem B.0.2 of [3] states:

Theorem A.5.2. $\mathcal{M}_\kappa$ *is computationally adequate for* MT_0 *programs.*

Likewise, we have:

Theorem A.5.3. $\mathcal{M}_\kappa$ *is computationally adequate for* E*-free* $\mathrm{MT}_{\mathrm{def}}$ *programs.*

The proof of Theorem A.5.3 is the same as the proof in [3] of Theorem A.5.2 above with the following modifications: First, one has to include parallel or the relevant places. Second, the proof of Lemma B.0.4 of [3], which is by structural induction, has one more case, namely one for parallel or.

Finally, we have:

Theorem A.5.4. *$\mathcal{M}_\kappa$ is computationally adequate for E-free MT programs.*

Proof of A.5.4 The theorem follows trivially from

$$(\mathcal{M}_\kappa \models a \neq \bot) \Rightarrow a \in \mathcal{N}_t \cup \mathcal{N}_f$$

which we prove in the following. Let $\tilde{\bot}$ be the term

$$(\lambda x.\, xx)(\lambda x.\, xx)$$

Thus, $\tilde{\bot}$ is $\bot_{\text{Curry}}$ (c.f. Section 3.2). For terms f let $\tilde{\mathsf{Y}}\{f\}$ be the term

$$(\lambda x.\, f(xx))(\lambda x.\, f(xx))$$

where x is chosen such that x is not free in f. Since $\mathcal{M}_\kappa$ is canonical we have $\mathcal{M}_\kappa \models \bot = \tilde{\bot}$ and $\mathcal{M}_\kappa \models \mathsf{Y}f = \tilde{\mathsf{Y}}\{f\}$.

For all terms b of MT we define the $\bot\mathsf{Y}$-less transform $[b]$ of b to be the term which results when replacing all occurrences of $\bot$ and $\mathsf{Y}f$ in b by $\tilde{\bot}$ and $\tilde{\mathsf{Y}}\{f\}$, respectively. In $\mathcal{M}_\kappa$ we have $[\bot] = \tilde{\bot} = \bot$ and $[\mathsf{Y}f] = \tilde{\mathsf{Y}}\{[f]\} = \mathsf{Y}[f]$. This allows to prove $\mathcal{M}_\kappa \models [a] = a$ for all terms a by structural induction.

For all E-free MT programs b, $[b]$ is an E-free MT_{def} program. Define $b \stackrel{1}{\to} c$ as in Section 3.3 and 3.4. We have:

$$\begin{array}{llll}
\bot & \stackrel{1}{\to} & \bot & \text{in MT} \\
\mathsf{Y}f & \stackrel{1}{\to} & f(\mathsf{Y}f) & \text{in MT} \\
\tilde{\bot} & \stackrel{1}{\to} & \tilde{\bot} & \text{in } \text{MT}_{\text{def}} \\
\tilde{\mathsf{Y}}\{f\} & \stackrel{1}{\to} & f(\tilde{\mathsf{Y}}\{f\}) & \text{in } \text{MT}_{\text{def}} \\
[\bot] & \stackrel{1}{\to} & [\bot] & \text{in } \text{MT}_{\text{def}} \\
[\mathsf{Y}f] & \stackrel{1}{\to} & [f(\mathsf{Y}f)] & \text{in } \text{MT}_{\text{def}}
\end{array}$$

In general, if $b \stackrel{1}{\to} c$ in MT then $[b] \stackrel{1}{\to} [c]$ in MT_{def} by structural induction in b and c.

Let a be an MT program and assume $\mathcal{M}_\kappa \models a \neq \bot$. Now $\mathcal{M}_\kappa \models [a] \neq \bot$.

Recall that for each a, $a \stackrel{1}{\to} b$ holds for at most one b (up to naming of bound variables). Let $a_1, a_2, \ldots$ be the unique longest finite or infinite sequence such that $a \stackrel{1}{\to} a_1 \stackrel{1}{\to} a_2 \stackrel{1}{\to} \cdots$ in MT. By Theorem A.5.3, the sequence $[a] \stackrel{1}{\to} [a_1] \stackrel{1}{\to} [a_2] \stackrel{1}{\to} \cdots$ is finite and ends with a term in root normal form (i.e. is T or an abstraction). Hence, $a \stackrel{1}{\to} a_1 \stackrel{1}{\to} a_2 \stackrel{1}{\to} \cdots$ has the same property, so $a \in \mathcal{N}_t \cup \mathcal{N}_f$.

For programs that may contain E we have:

Theorem A.5.5. *$\mathcal{M}_\omega$ is computationally adequate for* $\mathrm{MT}_{\mathrm{def}}$ *programs and for MT programs.*

Proof of A.5.5 The proof is similar to that of A.5.4. Define $a \stackrel{3}{\to} d \Leftrightarrow \exists b, c\colon a \stackrel{1}{\to} b \wedge b \stackrel{1}{\to} c \wedge c \stackrel{1}{\to} d$ and let $a \stackrel{*}{\to} b$ be the transitive closure of $a \stackrel{1}{\to} b$.

Recall the definition of $\mathsf{E}_{\mathrm{syntactic}}$ from Section 3.9. The definition is recursive and thus implicitly uses Y. Now define

$$\tilde{\mathsf{E}} \equiv \tilde{\mathsf{Y}}\{\lambda f, a.\, a\mathsf{C}_1 \parallel \cdots \parallel a\mathsf{C}_7 \parallel a(\lambda x.\, fx) \parallel f(\lambda x.\, f(\lambda y.\, a(xy)))\}$$

We have $\mathsf{E}_{\mathrm{syntactic}} = \tilde{\mathsf{E}}$ and

$$\tilde{\mathsf{E}}a \stackrel{3}{\to} a\mathsf{C}_1 \parallel \cdots \parallel a\mathsf{C}_7 \parallel a(\lambda x.\, \tilde{\mathsf{E}}x) \parallel \tilde{\mathsf{E}}(\lambda x.\, \tilde{\mathsf{E}}(\lambda y.\, a(xy)))$$

For all terms b of MT we define the E-less transform $[b]$ to be the term which results when replacing all occurences of $\mathsf{E}a$ by $\tilde{\mathsf{E}}a$. In $\mathcal{M}_\omega$ we have $[\mathsf{E}a] = \tilde{\mathsf{E}}[a] = \mathsf{E}_{\mathrm{syntactic}}[a] = \mathsf{E}[a]$. This allows to prove $\mathcal{M}_\omega \models [a] = a$ by structural induction.

If $b \stackrel{1}{\to} c$ in MT then $[b] \stackrel{1}{\to} [c]$ or $[b] \stackrel{3}{\to} [c]$ in MT and, in any case, $[b] \stackrel{*}{\to} [c]$.

The theorem follows from $\mathcal{M}_\omega \models a \neq \bot \Rightarrow a \in \mathcal{N}_t \cup \mathcal{N}_f$ which we now prove. Assume $\mathcal{M}_\omega \models a \neq \bot$. Let $a \stackrel{1}{\to} a_1 \stackrel{1}{\to} a_2 \stackrel{1}{\to} \cdots$ be the unique reduction sequence for a. Now $[a] \stackrel{*}{\to} [a_1] \stackrel{*}{\to} [a_2] \stackrel{*}{\to} \cdots$ is finite by A.5.4, so $a \in \mathcal{N}_t \cup \mathcal{N}_f$.

The case $\kappa > \omega$ is open:

Open Question A.5.6. *Is $\mathcal{M}_\kappa$ computationally adequate for* $\mathrm{MT}_{\mathrm{def}}$ *programs and for* MT *programs for $\kappa > \omega$?*

A.6. Soundness

Recall from Section 3.6 that $a =_\kappa b$ is shorthand for $\mathcal{M}_\kappa \models a = b$.

Theorem A.6.1 (Soundness of $\mathcal{M}_\kappa$ and $\mathcal{M}_\omega$).

(a) $a =_\kappa b \Rightarrow a =_{\mathrm{obs}} b$ *for E-free* MT *programs a and b.*
(b) $a =_\omega b \Rightarrow a =_{\mathrm{obs}} b$ *for all* MT *programs a and b.*
(c) $a =_\kappa b \Rightarrow a =_{\mathrm{obs}} b$ *for E-free* $\mathrm{MT}_{\mathrm{def}}$ *programs a and b.*
(d) $a =_\omega b \Rightarrow a =_{\mathrm{obs}} b$ *for all* $\mathrm{MT}_{\mathrm{def}}$ *programs a and b.*
(e) $a =_\kappa b \Rightarrow a =_{\mathrm{obs}} b$ *for all* MT_0 *programs a and b.*

Note that observational equality $a =_{\mathrm{obs}} b$ of MT, $\mathrm{MT}_{\mathrm{def}}$, and MT_0 is true if $ca \sim cb$ for all MT, $\mathrm{MT}_{\mathrm{def}}$, and MT_0 programs c, respectively, so the notions of observational equality are slightly different. Also note that MT_0 does not have E in its syntax, so all MT_0 programs are born E-free.

Proof of A.6.1 Soundness follows trivially from computational adequacy. We only prove (a). Assume $a =_\kappa b$. Assume c is an MT program. We have $ca =_\kappa cb$ so $ca =_\kappa \mathsf{T} \Leftrightarrow cb =_\kappa \mathsf{T}$ and, by Theorem A.5.4, $ca \in \mathcal{N}_t \Leftrightarrow cb \in \mathcal{N}_t$. Likewise, $ca \in \mathcal{N}_f \Leftrightarrow cb \in \mathcal{N}_f$ and $ca \in \mathcal{N}_\bot \Leftrightarrow cb \in \mathcal{N}_\bot$. Thus, $ca \sim cb$ for all MT programs c which, by definition of $=_{\mathrm{obs}}$, gives $a =_{\mathrm{obs}} b$.

Above, we use computational adequacy to prove soundness, and Open Question A.5.6 may be restated thus:

Open Question A.6.2.

(a) $a =_\kappa b \Rightarrow a =_{\mathrm{obs}} b$ *for* MT *programs a and b and $\kappa > \omega$?*

(b) $a =_\kappa b \Rightarrow a =_{\mathrm{obs}} b$ *for* $\mathrm{MT}_{\mathrm{def}}$ *programs a and b and $\kappa > \omega$?*

A.7. Full abstraction

Definition A.7.1. *A model $\mathcal{M}$ is* fully abstract *for MT/*$\mathrm{MT}_{\mathrm{def}}$/$\mathrm{MT}_0$ *if* $a =_{\mathrm{obs}} b \Leftrightarrow \mathcal{M} \models a = b$ *for all MT/*$\mathrm{MT}_{\mathrm{def}}$/$\mathrm{MT}_0$ *programs a and b.*

We now state and prove that $\mathcal{M}_\omega$ is fully abstract for MT:

Theorem A.7.2 (Full Abstraction of $\mathcal{M}_\omega$). $a =_{\mathrm{obs}} b \Leftrightarrow a =_\omega b$ *for MT programs a and b.*

Proof. ($\Leftarrow$) follows from Theorem A.6.1. ($\Rightarrow$) Assume $a =_{\mathrm{obs}} b$. Assume $p \in \mathbf{P}_\omega$. From $a =_{\mathrm{obs}} b$ we have $\mathcal{T}_{\langle\{p\},t\rangle} a \in \mathcal{N}_t \Leftrightarrow \mathcal{T}_{\langle\{p\},t\rangle} b \in \mathcal{N}_t$. Hence, by Theorem A.5.4, $\mathcal{T}_{\langle\{p\},t\rangle} a =_\omega \mathsf{T} \Leftrightarrow \mathcal{T}_{\langle\{p\},t\rangle} b =_\omega \mathsf{T}$. Thus, by Corollary A.3.2, $(\downarrow\langle\{p\},t\rangle)\underline{a} = \mathsf{T} \Leftrightarrow (\downarrow\langle\{p\},t\rangle)\underline{b} = \mathsf{T}$ so $p \in \underline{a} \Leftrightarrow p \in \underline{b}$ for all $p \in \mathbf{P}_\omega$. Hence, $\underline{a} = \underline{b}$ and $a =_\omega b$. $\square$

Theorem A.7.2 also holds for $\mathrm{MT}_{\mathrm{def}}$, i.e. $\mathcal{M}_\omega$ is also fully abstract for $\mathrm{MT}_{\mathrm{def}}$.

MT_0 lacks parallel or and Theorem A.7.2 does not hold for MT_0, i.e. $\mathcal{M}_\omega$ is not fully abstract for MT_0. As a counterexample, take

$$\begin{array}{rcl} a & = & \lambda x.\,\mathsf{if}[\,x\mathsf{T}\bot\ddot{\wedge}x\bot\mathsf{T}\ddot{\wedge}\ddot{\neg}x\mathsf{FF}\,,\,\mathsf{T}\,,\,\bot\,] \\ b & = & \lambda x.\,\bot \end{array}$$

The map a above is a parallel or tester, i.e. $ax = \mathsf{T}$ if xuv is the parallel or of u and v. We have $a =_{\mathrm{obs}} b$ in MT_0.

A.8. Negative results

We now prove that $\mathcal{M}_\kappa$ is not fully abstract for MT for $\kappa > \omega$, κ regular:

Theorem A.8.1. *If $\kappa > \omega$, κ regular, then there exist MT programs a and b for which $a =_{\mathrm{obs}} b$ and $a \neq_\kappa b$.*

Proof. Take $a = \mathsf{E}_{\mathrm{semantic}} = \lambda x.\,\mathsf{E}x$. Take $b = \mathsf{E}_{\mathrm{syntactic}}$ so that $b = \lambda x.\,(x\mathsf{C}_1 \parallel \cdots \parallel x\mathsf{C}_8 \parallel b\lambda u.\,b\lambda v.\,x(uv))$, c.f. Section 3.4 and 3.9.

We first prove $a =_{\mathrm{obs}} b$. According to Theorem A.7.2 is is enough to prove $a =_\omega b$. Furthermore, a and b are both characteristic maps, so it is enough to prove $ap =_\omega \mathsf{T} \Leftrightarrow bp =_\omega \mathsf{T}$ for all $p \in \mathcal{M}_\omega$. Now $ap =_\omega \mathsf{T}$ iff $px =_\omega \mathsf{T}$ for some $x \in \mathcal{M}_\omega$ and $bp =_\omega \mathsf{T}$ iff $px =_\omega \mathsf{T}$ for some MT program x. If $px =_\omega \mathsf{T}$ for some $x \in \mathcal{M}_\omega$ then $pc =_\omega \mathsf{T}$ for some compact $c \in \mathcal{M}_\omega$ so $p\mathcal{T}_c =_\omega \mathsf{T}$ proving $bp =_\omega \mathsf{T}$. Hence, $ap =_\omega \mathsf{T} \Rightarrow bp =_\omega \mathsf{T}$. If $bp =_\omega \mathsf{T}$ then $px =_\omega \mathsf{T}$ for some MT program x so $px =_\omega \mathsf{T}$ for some $x \in \mathcal{M}_\omega$ proving $ap =_\omega \mathsf{T}$. Hence, $bp =_\omega \mathsf{T} \Rightarrow ap =_\omega \mathsf{T}$ which ends the proof of $a =_\omega b$.

We then prove $a \neq_\kappa b$. Let $t_0 \equiv t$ and $t_{n+1} \equiv \langle\emptyset, t_n\rangle$ for $n \in \mathbf{N}$. We have $t_i \supset t_j \Leftrightarrow i = j$. Now let $g \in \mathbf{N} \rightarrow \mathbf{N}$ be non-computable. Let $Q \equiv \{\langle\{t_i\}, t_{g(i)}\rangle \mid i \in \mathbf{N}\}$, $q = \downarrow Q$, and $p = \downarrow\langle Q, t\rangle$. We have $p, q \in \mathcal{M}_\kappa$ and $pq =_\kappa \mathsf{T}$

so $ap =_\kappa \mathsf{T}$. Furthermore, $px =_\kappa \mathsf{T}$ for no program x since g is non-computable, so $bp \neq_\kappa \mathsf{T}$ proving $a \neq_\kappa b$. □

Theorem A.8.1 is not too surprising since E quantifies over $\mathcal{M}_\kappa$ whereas the computable approximation b in the proof essentially quantifies over $\{\downarrow p \mid p \in \mathbf{P}_\omega\}$. We may however strengthen the theorem above as follows:

Theorem A.8.2. *If $\kappa > \omega$, κ regular, then there exist E-free MT programs a and b for which $a =_{\mathrm{obs}} b$ and $a \neq_\kappa b$.*

The proof of Theorem A.8.2 spans the rest of this section.

Let $I' = \downarrow\{\langle\{p\}, p\rangle \mid p \in \mathbf{P}_\omega\}$, i.e. let I' be the smallest element of $\mathcal{M}_\kappa$ for which $I'(\downarrow p) = \downarrow p$ for all $p \in \mathbf{P}_\omega$. I' is compact but $I' \notin \mathbf{C}_\omega$. As we shall see in a moment, there exists an MT-term b which denotes I'.

To prove the lemma, we take $a = \lambda x.x$ and we take b to be a term which denotes I'. Now $a =_{\mathrm{obs}} b$ is true and $a =_\kappa b$ is false.

The rest of the proof is a definition of a b which denotes I'. The definition is long and technical.

Sections A.1–A.3 define $\mathcal{T}_p$ in ZFC. We now reflect that definition in MT.

Recall $(x{::}y)\mathsf{T} = x$ and $(x{::}y)\mathsf{F} = y$. Let $(x_1, \ldots, x_n)$ be shorthand for $x_1{::}\cdots{::}x_n{::}\mathsf{T}$. We shall refer to $(x_1, \ldots, x_n)$ as a *list* and use lists to represent finite sets. We now port the constructs of Section A.2 from ZFC to MT:

$$\begin{array}{lcl}
\sum_{x\in y} \mathcal{A} & \equiv & \sum' y(\lambda x.\mathcal{A}) \\
\sum' ya & \equiv & \mathsf{if}[\, y\,,\, \mathsf{F}\,,\, a(y\mathsf{T}) \parallel \sum'(y\mathsf{F})a\,] \\
\prod_{x\in y} \mathcal{A} & \equiv & \prod' y(\lambda x.\mathcal{A}) \\
\prod' ya & \equiv & \mathsf{if}[\, y\,,\, \mathsf{T}\,,\, a(y\mathsf{T})\ \&\ \prod'(y\mathsf{F})a\,]
\end{array}$$

Above, $\sum$ ($\prod$) expresses existential (universal) quantification. We also need a strict version of universal quantification:

$$\begin{array}{lcl}
\bigwedge_{x\in y} \mathcal{A} & \equiv & \bigwedge' y(\lambda x.\mathcal{A}) \\
\bigwedge' ya & \equiv & \mathsf{if}[\, y\,,\, \mathsf{T}\,,\, a(y\mathsf{T})\ddot{\wedge} \bigwedge'(y\mathsf{F})a\,]
\end{array}$$

We now proceed to port the definitions of $\mathbf{P}_\omega$ and $\mathbf{C}_\omega$ from ZFC to MT. We represent the elements $\mathbf{P}_\omega$ thus:

$$\begin{array}{lcl}
t & \equiv & \mathsf{T} \\
f & \equiv & \mathsf{T}{::}\mathsf{T} \\
\langle c, p\rangle & \equiv & \mathsf{T}{::}c{::}p
\end{array}$$

Recall that $x{::}y$ is right associative so that $\mathsf{T}{::}c{::}p$ is shorthand for $\mathsf{T}{::}(c{::}p)$. We have $\langle c, p\rangle\mathsf{FT} = c$ and $\langle c, p\rangle\mathsf{FF} = p$.

Elements of $\mathbf{C}_\omega$ are finite sets of elements of $\mathbf{P}_\omega$, so we represent them by lists. As an example, $(\langle(t), t\rangle, \langle(f), f\rangle)$ represents the element of $\mathbf{C}_\omega$ whose downward closure is the interpretation of $\lambda x.\,\mathsf{if}[\, x\,,\, \mathsf{T}\,,\, \lambda y.\perp]$.

A list like (t, f) does not represent an element of $\mathbf{C}_\omega$ since t and f are incoherent. We now define the coherence relations $\frown_0$ ($\frown_1$) on $\mathbf{P}_\omega$ ($\mathbf{C}_\omega$):

$$\begin{array}{lcl} p \frown_0 q & \equiv & \mathsf{if}[\,p\,,\,\mathsf{if}[\,q\,,\,\mathsf{T}\,,\,\mathsf{F}\,]\,,\,\mathsf{if}[\,q\,,\,\mathsf{F}\,, \\ & & \quad \mathsf{if}[\,p\mathsf{F}\,,\,\mathsf{T}\,,\,\mathsf{if}[\,q\mathsf{F}\,,\,\mathsf{T}\,, \\ & & \qquad p\mathsf{FT} \frown_1 q\mathsf{FT} \ddot{\Rightarrow} p\mathsf{FF} \frown_0 q\mathsf{FF}]]]] \\ c \frown_1 d & \equiv & \bigwedge_{p \in c} \bigwedge_{q \in d} p \frown_0 q \end{array}$$

The definitions above allow to define characteristic maps $\chi_{\mathbf{P}_\omega}$, $\chi_{\mathbf{C}_\omega}$, and $\chi_{\mathbf{C}_\omega^{<\omega}}$ which test for membership of $\mathbf{P}_\omega$, $\mathbf{C}_\omega$, and $\mathbf{C}_\omega^{<\omega}$, respectively:

$$\begin{array}{lcl} \chi_{\mathbf{P}_\omega} p & \equiv & \mathsf{if}[\,p\mathsf{F}\,,\,\mathsf{T}\,,\,\chi_{\mathbf{C}_\omega}(p\mathsf{FT}) \ddot{\wedge} \chi_{\mathbf{P}_\omega}(p\mathsf{FF})\,] \\ \chi_{\mathbf{C}_\omega} c & \equiv & c \frown_1 c \ddot{\wedge} \bigwedge_{p \in c} \chi_{\mathbf{P}_\omega} p \\ \chi_{\mathbf{C}_\omega^{<\omega}} \bar{c} & \equiv & \mathsf{if}[\,\bar{c}\,,\,\mathsf{T}\,,\,\chi_{\mathbf{C}_\omega}(\bar{c}\mathsf{T}) \ddot{\wedge} \chi_{\mathbf{C}_\omega^{<\omega}}(\bar{c}\mathsf{F})\,] \end{array}$$

We now port the definitions in Section A.3 from ZFC to MT. The definitions of $\mathcal{T}_t$, $\mathcal{T}_f$, and $\mathcal{T}_{\langle c,p \rangle}$ in Section A.3 define $\mathcal{T}_p$ for all $p \in \mathbf{P}_\omega$. Below, $\mathcal{T}_0 p$ is the MT translation of the ZFC construct $\mathcal{T}_p$:

$$\begin{array}{lcl} \mathcal{T}_0 p & \equiv & \mathsf{if}[\,p\,,\,\mathsf{T}\,, \\ & & \quad \mathsf{if}[\,p\mathsf{F}\,,\,\lambda x.\,\bot_{\mathrm{Curry}}\,, \\ & & \qquad \lambda x.\,\mathsf{if}[\,\chi_1(p\mathsf{FT})x\,,\,\mathcal{T}_0(p\mathsf{FF})\,,\,\bot_{\mathrm{Curry}}\,]]] \end{array}$$

The definitions of χ_p, χ_c, $\chi_{\bar{c}}$, and χ_s of Section A.3 translate into the following:

$$\begin{array}{lcl} \chi_0 p x & \equiv & \mathsf{if}[\,p\,,\,\mathsf{if}[\,x\,,\,\mathsf{T}\,,\,\bot_{\mathrm{Curry}}\,]\,, \\ & & \quad \mathsf{if}[\,p\mathsf{F}\,,\,\mathsf{if}[\,x\,,\,\bot_{\mathrm{Curry}}\,,\,\mathsf{T}\,]\,, \\ & & \qquad \chi_0(p\mathsf{FF})(x\mathcal{T}_1(p\mathsf{FT}))]] \\ \chi_1 c x & \equiv & \bigwedge_{p \in c} \chi_0 p x \\ \chi_2 \bar{c}\bar{x} & \equiv & \mathsf{if}[\,\bar{c}\,,\,\mathsf{T}\,,\,\bar{c}\mathsf{T}(\bar{x}\mathsf{T})\ \&\ \chi_2(\bar{c}\mathsf{F})(\bar{x}\mathsf{F})\,] \\ \chi_3 s \bar{x} & \equiv & \sum_{\bar{c} \in s} \chi_2 \bar{c}\bar{x} \end{array}$$

The union of two sets represented by lists is a classical:

$$c \cup d \quad \equiv \quad \mathsf{if}[\,c\,,\,d\,,\,c\mathsf{T}{::}(c\mathsf{F} \cup d)\,]$$

The discriminator constructs $\delta_{pp'}$, $\delta_{cc'}$, $\delta_{\bar{c}\bar{c}'}$, and $\delta_{ss'}$ of Section A.3 translate into the following:

$$\begin{array}{lcl} \delta_0 p p' x & \equiv & \mathsf{if}[\,p\,,\,\mathsf{if}[\,x\,,\,\mathsf{T}\,,\,\mathsf{F}\,]\,, \\ & & \quad \mathsf{if}[\,p\mathsf{F}\,,\,\mathsf{if}[\,x\,,\,\mathsf{F}\,,\,\mathsf{T}\,]\,, \\ & & \qquad \delta_0(p\mathsf{FF})(p'\mathsf{FF})(x(\mathcal{T}_1(p\mathsf{FT} \cup p'\mathsf{FT})))]] \\ \delta_1 c c' x & \equiv & \prod_{p \in c} \sum_{p' \in c'} \delta_0 p p' x \\ \delta_2 \bar{c}\bar{c}'\bar{x} & \equiv & \mathsf{if}[\,\bar{c}\mathsf{T} \frown_1 \bar{c}'\mathsf{T}\,,\,\delta_2(\bar{c}\mathsf{F})(\bar{c}'\mathsf{F})(\bar{x}\mathsf{F})\,,\,\delta_1(\bar{c}\mathsf{T})(\bar{c}'\mathsf{T})(\bar{x}\mathsf{T})\,] \\ \delta_3 s s' \bar{x} & \equiv & \sum_{\bar{c} \in s} \prod_{\bar{c}' \in s'} \delta_2 \bar{c}\bar{c}'\bar{x} \end{array}$$

The empty set and the singleton set is straightforward:

$$\begin{array}{lcl} \emptyset & \equiv & \mathsf{T} \\ \{x\} & \equiv & x{::}\mathsf{T} \end{array}$$

The ZFC construct $\mathsf{def}(n,c)$ of Section A.3 translates into the MT construct $\mathsf{def}\bar{x}c$ below where we represent the natural number n in the ZFC construct by a list $\bar{x}$ of length n in the MT construct.

$$\begin{array}{lcl} \mathsf{def}\bar{x}c & \equiv & \mathsf{if}[\,c\,,\,\emptyset\,,\,\mathsf{def}'\bar{x}(c\mathsf{T})\mathsf{T}\cup\mathsf{def}\bar{x}(c\mathsf{F})\,] \\ \mathsf{def}'\bar{x}c\bar{c} & \equiv & \mathsf{if}[\,\bar{x}\,,\,\{\bar{c}\}\,,\,\mathsf{if}[\,p\mathsf{F}\,,\,\emptyset\,,\,\mathsf{def}'(\bar{x}\mathsf{F})(p\mathsf{FF})(p\mathsf{FT}{::}\bar{c})\,]\,] \end{array}$$

$\mathsf{def}(n,c)$ is a set of tuples and $\mathsf{def}\bar{x}c$ is a list of lists. If $(p_1,\ldots,p_n)$ is an element of $\mathsf{def}(n,c)$ then $(p_n,\ldots,p_1)$ is an element of $\mathsf{def}\bar{x}c$. Note the list reversal.

Note that the parameter $\bar{c}$ of def' *accumulates* a list in reverse order. Use of such accumulating parameters is a standard trick in functional programming.

We now proceed:

$$\begin{array}{lcl} \mathsf{true}\bar{x}c & \equiv & \mathsf{if}[\,c\,,\,\emptyset\,,\,\mathsf{true}'\bar{x}(c\mathsf{T})\mathsf{T}\cup\mathsf{true}\bar{x}(c\mathsf{F})\,] \\ \mathsf{true}'\bar{x}c\bar{c} & \equiv & \mathsf{if}[\,\bar{x}\,,\,\mathsf{if}[\,p\,,\,\{\bar{c}\}\,,\,\emptyset\,]\,,\,\mathsf{if}[\,p\mathsf{F}\,,\,\emptyset\,,\,\mathsf{true}'(\bar{x}\mathsf{F})(p\mathsf{FF})(p\mathsf{FT}{::}\bar{c})\,]\,] \\ \mathsf{false}\bar{x}c & \equiv & \mathsf{if}[\,c\,,\,\emptyset\,,\,\mathsf{false}'\bar{x}(c\mathsf{T})\mathsf{T}\cup\mathsf{false}\bar{x}(c\mathsf{F})\,] \\ \mathsf{false}'\bar{x}c\bar{c} & \equiv & \mathsf{if}[\,\bar{x}\,,\,\mathsf{if}[\,p\,,\,\emptyset\,,\,\{\bar{c}\}\,]\,,\,\mathsf{if}[\,p\mathsf{F}\,,\,\emptyset\,,\,\mathsf{false}'(\bar{x}\mathsf{F})(p\mathsf{FF})(p\mathsf{FT}{::}\bar{c})\,]\,] \end{array}$$

We now define $\mathcal{T}_1c$ which corresponds to $\mathcal{T}_c$ in Section A.3. We do so using an accumulating parameter $\bar{x}$ which accumulates $(x_n,\ldots,x_1)$ where $x_1,\ldots,x_n$ are the bound variables in the definition of $\mathcal{T}_c$ in Section A.3. The definition reads:

$$\begin{array}{lcl} \mathcal{T}_1c & \equiv & \mathcal{T}_1'c\mathsf{T} \\ \mathcal{T}_1'c\bar{x} & \equiv & \mathsf{if}[\,\chi_3(\mathsf{def}\bar{x}c)\bar{x}\,,\,\perp_{\mathrm{Curry}}\,, \\ & & \quad \mathsf{if}[\,\delta_3(\mathsf{true}\bar{x}c)(\mathsf{false}\bar{x}c)\bar{x}\,,\,\chi_3(\mathsf{true}\bar{x}c)\bar{x}\,, \\ & & \quad\quad \chi_3(\mathsf{false}\bar{x}c)\bar{x}\,!\,\lambda x.\,\mathcal{T}_1'c(x{::}\bar{x})]] \end{array}$$

This completes the port of Sections A.1–A.3 from ZFC to MT. We now define constructs with the following properties:

$$\begin{array}{lcl} \mathsf{apply}x(y_n,\ldots,y_1) & = & xy_1\cdots y_n \\ (c_n,\ldots,c_1)\mapsto p & = & \langle c_1,\cdots\langle c_n,p\rangle\cdots\rangle \end{array}$$

Note the list reversal. The definitions read:

$$\begin{array}{lcl} \mathsf{apply}x\bar{y} & \equiv & \mathsf{if}[\,\bar{y}\,,\,x\,,\,\mathsf{apply}x(\bar{y}\mathsf{F})(\bar{y}\mathsf{T})\,] \\ \bar{c}\mapsto p & \equiv & \mathsf{if}[\,\bar{c}\,,\,p\,,\,\bar{c}\mathsf{F}\mapsto(\mathsf{T}{::}\bar{c}\mathsf{T}{::}p)\,] \end{array}$$

Finally, we may define a term b which denotes I' where I' is the smallest element of $\mathcal{M}_\kappa$ for which $I'(\downarrow p)=\downarrow p$ for all $p\in\mathbf{P}_\omega$. The definition uses an accumulating parameter $\bar{y}$:

$$\begin{array}{lcl} bx & \equiv & b'x\mathsf{T} \\ b'x\bar{y} & \equiv & \mathsf{if}[\,\mathsf{apply}x\bar{y}\,,\,\mathsf{E}\bar{c}\colon\chi_{\mathbf{C}_\omega^{<\omega}}\bar{c}\ddot{\wedge}\chi_0(\bar{c}\mapsto t)x\ddot{\wedge}\chi_2\bar{c}\bar{y}\,, \\ & & \quad \mathsf{E}\bar{c}\colon\chi_{\mathbf{C}_\omega^{<\omega}}\bar{c}\ddot{\wedge}\chi_0(\bar{c}\mapsto f)x\ddot{\wedge}\chi_2\bar{c}\bar{y}\,!\,\lambda y.\,b'x(y{::}\bar{y})] \end{array}$$

As an example, if $bxy_1\cdots y_{n-1}\not\in\{\mathsf{T},\perp\}$ and $xy_1\cdots x_n=\mathsf{T}$ then

$$\begin{array}{lcl} bxy_1\cdots y_n & = & b'x\mathsf{T}y_1\cdots y_n \\ & = & b'x(y_n,\ldots,y_1) \\ & = & \mathsf{E}\bar{c}\colon\chi_{\mathbf{C}_\omega^{<\omega}}\bar{c}\ddot{\wedge}\chi_0(\bar{c}\mapsto t)x\ddot{\wedge}\chi_2\bar{c}(y_n,\ldots,y_1) \end{array}$$

Thus, in the situation above, $bxy_1\cdots y_n$ returns T iff there exists a $\bar{c}\in\mathbf{C}_\omega^{<\omega}$ such that $\downarrow(\bar{c}\mapsto t)\preceq_{\mathcal{M}} x$ and $\downarrow\bar{c}\preceq_{\mathcal{M}}(y_1,\ldots,y_n)$.

B. Summary of MT

B.1. Syntax

$$
\begin{array}{lcl}
\mathcal{V} & ::= & x \mid y \mid z \mid \cdots \\
\mathcal{T} & ::= & \mathcal{V} \mid \lambda\mathcal{V}.\mathcal{T} \mid \mathcal{T}\mathcal{T} \mid \mathsf{T} \mid \mathsf{if}[\mathcal{T},\mathcal{T},\mathcal{T}] \mid \bot \mid \mathsf{Y} \mid \mathcal{T}\|\mathcal{T} \mid \mathsf{E}\mathcal{T} \mid \varepsilon\mathcal{T} \\
\mathcal{W} & ::= & \mathcal{T} = \mathcal{T}
\end{array}
$$

B.2. Definitions

$\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$ denote (possibly open) terms. $a, b, \ldots z, \theta$ denote variables.

Elementary definitions

$$
\begin{array}{rll}
\bot \equiv & (\lambda x.\, xx)(\lambda x.\, xx) & \text{(Only in MT}_{\text{def}}\text{)} \\
\mathsf{Y} \equiv & \lambda f.\, (\lambda x.\, f(xx))(\lambda x.\, f(xx)) & \text{(Only in MT}_{\text{def}}\text{)} \\
\lambda xy.\, \mathcal{A} \equiv & \lambda x.\, \lambda y.\, \mathcal{A} & \\
\mathsf{F} \equiv & \lambda x.\, \mathsf{T} & \\
\mathbf{1} \equiv & \lambda xy.xy & \\
x \downarrow y \equiv & \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \bot], \mathsf{if}[y, \bot, \lambda z.(xz) \downarrow (yz)]] & \\
x \preceq y \equiv & x = x \downarrow y & \\
\approx\! x \equiv & \mathsf{if}[\, x \,,\, \mathsf{T} \,,\, \mathsf{F}\,] & \\
x \circ y \equiv & \lambda z.x(yz) & \\
\chi \equiv & \lambda x.\lambda z.\mathsf{if}[xz, \mathsf{T}, \bot] & \\
x \rightarrow y \equiv & \mathsf{if}[x, y, \mathsf{F}] = \mathsf{if}[x, \mathsf{T}, \mathsf{F}] & \\
!x \equiv & \mathsf{if}[x, \mathsf{T}, \mathsf{T}] & \\
\ddot{\neg} x \equiv & \mathsf{if}[x, \mathsf{F}, \mathsf{T}] &
\end{array}
$$

Quantification

$$
\begin{array}{rl}
\ddot{\exists} p \equiv & \approx\!(p(\varepsilon p)) \\
\ddot{\exists} x\colon \mathcal{A} \equiv & \ddot{\exists} \lambda x.\, \mathcal{A} \\
\ddot{\forall} x\colon \mathcal{A} \equiv & \ddot{\neg}\ddot{\exists} x\colon \ddot{\neg}\mathcal{A} \\
\varepsilon x\colon \mathcal{A} \equiv & \varepsilon \lambda x.\, \mathcal{A} \\
\ddot{\forall} p \equiv & \ddot{\forall} x\colon px \\
\mathsf{E}x.\, \mathcal{A} \equiv & \mathsf{E}\lambda x.\, \mathcal{A}
\end{array}
$$

The definition of ψ

$$
\begin{array}{rl}
\psi \equiv & \sqcup s \\
s \equiv & \mathsf{Y} S \\
S \equiv & \lambda f.\, \bar{S} f(\sqcup f) \\
\bar{S} \equiv & \lambda f \theta a.\, \mathsf{if}[\, a \,,\, P \,,\, \mathsf{if}[\, a\mathsf{T} \,,\, Q(f(a\mathsf{F})) \,,\, Rf\theta(a\mathsf{T})(a\mathsf{F})\,]\,] \\
P \equiv & \lambda y.\, \mathsf{if}[\, y \,,\, \mathsf{T} \,,\, \bot\,] \\
Q \equiv & \lambda v.\, \mathsf{D} v \,!\, \lambda y.\, \ddot{\forall} z.\, v(y(z \,/\, v)) \\
R \equiv & \lambda f \theta bc.\, \theta c \,!\, R_1 f \theta bc \,!\, R_0 f \theta bc \\
R_1 \equiv & \lambda f \theta bc.\, \ddot{\forall} z.\, \mathsf{D}(f(b(cz \,/\, \theta))) \\
R_0 \equiv & \lambda f \theta bcy.\, \mathsf{E}z.\, (\theta z \,!\, f(b(cz \,/\, \theta))y) \\
\sqcup \equiv & \lambda fy.\, \mathsf{E}x.\, fxy \\
x \,!\, y \equiv & \mathsf{if}[\, x \,,\, y \,,\, \bot\,] \\
\mathsf{D} \equiv & \lambda x.\, \mathsf{if}[\, x \,,\, \mathsf{T} \,,\, \mathsf{T}\,] \\
f \,/\, g \equiv & \mathsf{if}[\, f \,,\, \mathsf{T} \,,\, \lambda x.\, gx \,!\, (fx \,/\, g)\,]
\end{array}
$$

B.3. Axioms and inference rules

Elementary axioms and inference rules

Trans	$\mathcal{A}=\mathcal{B};\mathcal{A}=\mathcal{C}\vdash\mathcal{B}=\mathcal{C}$
Sub	$\mathcal{B}=\mathcal{C}\vdash\mathcal{A}\mathcal{B}=\mathcal{A}\mathcal{C}$
Gen	$\mathcal{A}=\mathcal{B}\vdash\lambda x.\mathcal{A}=\lambda x.\mathcal{B}$
A1	$\mathsf{T}\mathcal{B}=\mathsf{T}$
A2 (β)	$(\lambda x.\mathcal{A})\mathcal{B}=\mathcal{C}$ if $\mathcal{C}\overset{\alpha}{=}\langle\mathcal{A}\mid x:=\mathcal{B}\rangle$
A3	$\bot\mathcal{B}=\bot$
I1	$\mathsf{if}[\mathsf{T},\mathcal{B},\mathcal{C}]=\mathcal{B}$
I2	$\mathsf{if}[\lambda x.\mathcal{A},\mathcal{B},\mathcal{C}]=\mathcal{C}$
I3	$\mathsf{if}[\bot,\mathcal{B},\mathcal{C}]=\bot$
QND	$\mathcal{A}\mathsf{T}=\mathcal{B}\mathsf{T};\mathcal{A}(\mathbf{1}\mathcal{C})=\mathcal{B}(\mathbf{1}\mathcal{C});\mathcal{A}\bot=\mathcal{B}\bot\vdash\mathcal{A}\mathcal{C}=\mathcal{B}\mathcal{C}$
P1	$\mathsf{T}\parallel\mathcal{B}=\mathsf{T}$
P2	$\mathcal{A}\parallel\mathsf{T}=\mathsf{T}$
P3	$\lambda x.\mathcal{A}\parallel\lambda y.\mathcal{B}=\lambda z.\mathsf{T}$
Y	$\mathsf{Y}\mathcal{A}=\mathcal{A}(\mathsf{Y}\mathcal{A})$ (Not needed in $\mathrm{MT}_{\mathrm{def}}$)

Monotonicity and minimality

Mono	$\mathcal{B}\preceq\mathcal{C}\vdash\mathcal{A}\mathcal{B}\preceq\mathcal{A}\mathcal{C}$
Min	$\mathcal{A}\mathcal{B}\preceq\mathcal{B}\vdash\mathsf{Y}\mathcal{A}\preceq\mathcal{B}$

Extensionality

Ext	if x and y are not free in $\mathcal{A}$ and $\mathcal{B}$ then $\approx(\mathcal{A}x)=\approx(\mathcal{B}x);\mathcal{A}xy{=}\mathcal{A}\mathcal{C};\mathcal{B}xy{=}\mathcal{B}\mathcal{C}\vdash\mathcal{A}x=\mathcal{B}x$

Axioms on E

ET	$\mathsf{E}\mathsf{T}=\mathsf{T}$
EB	$\mathsf{E}\bot=\bot$
EX	$\mathsf{E}x=\mathsf{E}(\chi x)$
EC	$\mathsf{E}(x\circ y)\to\mathsf{E}x$

Quantification axioms

Elim	$(\ddot{\forall}x\colon px)\wedge\psi y\to py$
Ackermann	$\varepsilon x\colon px=\varepsilon x\colon(\psi x\wedge px)$
StrictE	$\psi(\varepsilon x\colon px)=\ddot{\forall}x\colon!(px)$
StrictA	$!(\ddot{\forall}x\colon px)=\ddot{\forall}x\colon!(px)$

C. Index

Greek letters

Arrows

Equal signs/Equivalences

Order relations

Double dot constructs

Superscript set constructs

Other nulary constructs

Other unary constructs

Other binary constructs

[1] P. Aczel. *Non-Well-Founded Sets*, volume 14 of *Lecture Notes*. CSLI, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA, 1988.

[2] C. Berline. From computation to foundations via functions and application : the lambda-calculus and its webbed models. *Theoretical Computer Science*, 249:81–161, 2000.

[3] C. Berline and K. Grue. A κ-denotational semantics for Map Theory in ZFC+SI. *Theoretical Computer Science*, 179(1–2):137–202, June 1997.

[4] A. Church. A set of postulates for the foundations of logic I. *Ann. Math.*, 1933.

[5] A. Church. A set of postulates for the foundations of logic II. *Ann. Math.*, 1934.

[6] R.C. Flagg. κ-continuous lattices and comprehension principles for Frege structures. *Annals of Pure and Applied Logic*, 36:1–16, 1987.

[7] R.C. Flagg and J. Myhill. A type-free system extending ZFC. *Annals of Pure and Applied Logic*, 43:79–97, 1989.

[8] K. Grue. Map theory. *Theoretical Computer Science*, 102(1):1–133, July 1992.

[9] K. Grue. Map theory with classical maps. Technical Report 02/21, DIKU, Universitetsparken 1, DK-2100 Copenhagen, Denmark, 2002. http://www.diku.dk/˜grue/papers/classical/classical.html.

[10] K. Grue. Logiweb. In Fairouz Kamareddine, editor, *Mathematical Knowledge Management Symposium 2003*, volume 93 of *Electronic Notes in Theoretical Computer Science*, pages 70–101. Elsevier, 2004.

[11] Rainer Kerth. Définissabilité dans les domaines réflexifs. *Comptes rendus de l'Acadmie des sciences. Série 1, Mathématique*, 318(7):685–688, 1994.

[12] Rainer Kerth. Isomorphisme et équivalence équationelle entre modéles du lambda-calcul. Ph.D thesis, Université Paris 7, 1995.

[13] G. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5(3):223–255, December 1977.

[14] G. Plotkin. Set-theoretical and other elementary models of the lambda-calculus (in a collection of contributions in honour of Corrado Bohm on the occasion of his 70th birthday). *Theoretical Computer Science*, 121(1-2):159–192, 1993.

[15] Thierry Vallée. "Map theory" et antifondation. *Electr. Notes Theor. Comput. Sci.*, 79:1–260, 2003. (Ph.D. Thesis, Université Paris 7, 2001).

[16] Thierry Vallée. Map theory: From well-foundation to antifoundation. In *Proceedings of the 6th Workshop on domains (WD6)*, volume 73 of *Electron. Notes Theor. Comput. Sci.*, pages 217–245, Amsterdam, The Netherlands, The Netherlands, October 2004. Elsevier Science Publishers B. V.