A synthetic axiomatization of Map Theory

Chantal Berline, Klaus Grue

A synthetic axiomatization of Map Theory[☆]

Chantal Berline[a,1], Klaus Grue[b,2,*]

[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 substantially simplified axiomatization of Map Theory and proves the consistency of this axiomatization (called MT) 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 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. The first axiomatization MT_0 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 MT 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.

In the paper we define the notions of canonical and non-canonical κ- and $\kappa\sigma$-expansions and prove that if σ is the smallest strongly inaccessible ordinal then canonical $\kappa\sigma$-expansions are models of MT (which proves the consistency). Furthermore, in Appendix A, we prove that canonical ω-expansions are fully abstract models of the computational part of Map Theory.

Keywords: lambda calculus, foundation of mathematics, map theory, kappa-Scott semantics, Hilbert's epsilon operator

[☆] This document is a collaborative effort.

[*] Corresponding author

Email addresses: `berline@pps.univ-paris-diderot.fr` (Chantal Berline), `grue@di.ku.dk` (Klaus Grue)

[1] Present address: 85 rue Dutot, F-75015 Paris, France

[2] Present address: Edlund A/S, Bjerregårds Sidevej 4 , DK-2500 Valby, Denmark

Contents

1. Introduction

1.1. What Map Theory is

Intuitively, Map Theory is the theory of a universe $\mathcal{M}$ which is a Big ordered model of untyped lambda calculus. The universe $\mathcal{M}$ is big enough to contain a model of ZFC. The elements of $\mathcal{M}$ are called maps. Applying any map to any map yields a map, and Map Theory supports unrestricted use of lambda abstraction. Application is monotonic in the order of $\mathcal{M}$.

Closed lambda terms are, of course, interpreted by maps, but this is also the case for sets, classes, set- and class-constructors, logical connectives and quantifiers (c.f. Section 3 of [9]). Any wellformed formula of ZFC is a term of Map Theory (through definitions in Map Theory of $\in$, $\neg$, $\Rightarrow$ and $\forall$, cf. Example 4.5.1).

Map theory interprets ZFC as follows: A closed, wellformed formula $\mathcal{A}$ of ZFC is a closed term of Map Theory. If $\mathcal{A}$ is a theorem of ZFC then $\mathcal{A} = \mathsf{T}$ is a theorem of Map Theory where T is a special map which represents truth. If the negation of $\mathcal{A}$ is a theorem of ZFC then $\mathcal{A} = \mathsf{F}$ where F is another map. Russell's paradoxical sentence $\mathcal{R}$ is not a wellformed formula of ZFC but is easy to express in Map Theory; it satisfies $\mathcal{R} = \perp$ where $\perp$ is the minimal element of $\mathcal{M}$ and that does not give rise to any inconsistency (cf. Section 2.4). Computationally speaking, $\mathcal{R}$ evaluates to "does not terminate" in the λ-calculus sense (cf. Section 2.6).

Map Theory has several axiomatizations like the axiomatization MT considered in the present paper and the original axiomatization MT_0 in [9]. This is like Set Theory which has e.g. the axiomatizations ZFC and NBG. MT is defined in Section 3.2 and Section 4. Appendix D contains a summary of MT.

As shown in [4], some big Scott-like models can be enriched to be suitable universes of Map Theory.

Syntactically, Map Theory comprises a computer programming language plus Hilbert's epsilon operator. All theorems of ZFC set theory including the axiom of foundation are provable in all the various axiomatizations of Map Theory, and if one omits Hilbert's epsilon operator from Map Theory then one is left with a computer programming language (cf. Section 2.5).

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. A core benefit of Map Theory is that it allows to mix recursive programs and quantifiers freely, as exemplified in Section 2.3 and Example 4.5.2. All this was largely developed in [9, Part 1].

1.2. Map Theory and its axiomatizations

The first axiomatization of Map Theory [9], which we call MT_0 in this paper, had complex axioms and a complex model. [4] provided a simpler model. The present paper provides a simpler and more synthetic axiomatization which we call MT and which is summarized in Appendix D, and proves the consistency of the enhanced system starting from the *canonical* models of MT_0 built in [4]. On a quite solid basis we conjecture (Conjecture 2.2.3 and Appendix B) that MT is more powerful than MT_0. We also introduce (in Section 3.4) a natural and minor variant $\mathrm{MT}_{\mathrm{def}}$ of MT, and derive its consistency from that of MT.

When speaking of "Map Theory" in this paper we always refer either to the generic intuition or to properties shared by all the axiomatizations we have proposed for that underlying intuition.

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 refer to the 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 (i.e. language construct) ε, whose semantics is that of Hilbert's choice operator acting over a universe Φ of "wellfounded maps". The ε construct is axiomatized through the "quantification axioms" (four equations).

Apart from ε, MT and MT_0 have in common a few elementary constructs (λ-abstraction, application, T, $\perp$ and if) and related axioms and inference rules (the Elem group, cf. Section 4.1) which take care of the computational part of Map Theory. These constructs simultaneously bear set theoretical and/or logical meanings [9]. Some "sugar" (the construct Y and parallel or and the associated Elem' group of rules) has also been added to MT, but this is inessential.

1.3. How MT enhances MT_0

We now explain why we felt a need for designing MT, and what is the key difference between MT and MT_0. While MT to some extend obsoletes MT_0 (cf. Section 14.4), MT_0 is still important here since the consistency proof of MT builds on that of MT_0.

Apart from ε and the elementary constructs, MT_0 has only one construct ϕ, which is in spirit the characteristic function of Φ. As a set of rules (where "rules" means "axioms and inference rules"), and with the terminology above, we have

$$\mathrm{MT}_0 = \mathsf{Elem} + \mathsf{Quant}[\phi] + \mathsf{WF}[\phi]$$

cf. Section 5.1. $\mathsf{Quant}[\phi]$ axiomatizes the notion of quantification over Φ and $\mathsf{WF}[\phi]$ contains ten axioms, each axiomatizing one specific closure property of Φ, plus one inference rule of transfinite induction (cf. Section 5.1). MT_0 has the power to embody ZFC because ϕ satisfies $\mathsf{WF}[\phi]$.

Having ten axioms, even if some of them are not intuitive, 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, [4], also stated here in Section 7.8). GCP was one of the founding intuitions behind Map Theory (cf. [9]), it was satisfied in our models of MT_0, and our desire was to reflect it at the syntactic level.

With the present MT not only do we solve this problem (whence "synthetic") but we also eliminate ϕ and $\mathsf{WF}[\phi]$, replacing them by ... *nothing* (whence "simpler"). Moreover, the new system is stronger (Provided Conjecture 2.2.3 is true). *Nothing* should be taken with three grains of salt as explained in the following.

The Definability Theorem (Theorem 10.1), which is the most difficult result of the present paper, tells us that if we take Φ to be the smallest universe satisfying GCP, then its characteristic function ϕ happens to be definable from other MT constructs as a term ψ (defined in Section 4.7).

The first grain of salt is that we replace $\mathsf{Quant}[\phi]$ by $\mathsf{Quant}[\psi]$. In other words, when we eliminate ϕ, we replace it by ψ.

The second grain of salt is that for defining ψ we need to add a construct E ("pure existence") and its related axioms (the Exist group, cf. Section 4.4). 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 syntactic level 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. In fact, besides finding the order, the cost is the addition of three inference rules which express monotonicity (Mono), minimality of the fixed point operator Y (Min) and extensionality (Ext), c.f. Section 4.2 and 4.3 and the table in Section 5.5.

Thus, we can summarize MT by

$$\mathrm{MT} = \mathsf{Elem} + \mathsf{Elem}' + \mathsf{Mono} + \mathsf{Min} + \mathsf{Ext} + \mathsf{Exist} + \mathsf{Quant}[\psi]$$

cf. Section 5.1.

Finally, $\mathrm{MT}_{\mathrm{def}}$ is just the "economical" version of MT where all the occurrences of Y and $\bot$ are replaced by $\mathsf{Y}_{\mathrm{Curry}} \equiv \lambda f.\, (\lambda x.\, f(xx))(\lambda x.\, f(xx))$ and $\bot_{\mathrm{Curry}} \equiv \mathsf{Y}_{\mathrm{Curry}} \lambda x.\, x = (\lambda x.\, xx)(\lambda x.\, xx)$. In this paper, $\equiv$ is used for definitions.

1.4. The consistency of MT

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 noticing that a first "synthetic" version of MT, called MT_c, was present in [10], 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 come back to MT_c in Section 2.2.

We prove the consistency of MT in ZFC+SI where ZFC+SI is ZFC extended with the assumption that there exists an inaccessible ordinal (where *inaccessible* means *strongly inaccessible*, c.f. Section 6.1).

We prove the consistency by showing that some of the models of MT_0 built in [4] can be expanded to model MT also. More specifically, the "canonical" models of MT_0 are also models of MT, provided they are constructed using the *first* inaccessible ordinal σ_o (Theorem 2.2.1/Theorem 13.1).

The most difficult part of the consistency proof for MT is the Definability Theorem (Theorem 10.1) which states that $\phi = \psi$. The proof can be found in Sections 10–12 and uses that σ_o is the smallest inaccessible.

Furthermore, MT has some new inference rules (Mono, Min and Ext) whose satisfaction requires canonicity. They are treated in Section 9.

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 c.f. Section 5.4.

The natural and minor variant $\mathrm{MT}_{\mathrm{def}}$ of MT mentioned in Sections 1.2 and 1.3 and defined in Section 3.4 (the one where $\mathsf{Y}_{\mathrm{Curry}}$ and $\perp_{\mathrm{Curry}}$ replace Y and $\perp$) has even less models than MT and is a bit more difficult to prove consistent. This is one reason why we chose MT as the main subject of the present paper and present the consistency of $\mathrm{MT}_{\mathrm{def}}$ as a corollary of the consistency of MT (c.f. Section 13).

1.5. Relation to the consistency proof for MT_0

The present paper reuses a substantial amount of material from the consistency proof for MT_0 in [4]. In the present paper we repeat definitions and theorems from [4] that we need, but we do not repeat proofs. The intention is to keep the size of the present paper down and at the same time make the present paper readable without having [4] available. MT can be seen as obsoleting MT_0, but the present paper cannot be seen as obsoleting [4] since some theorems needed in the present paper are proved in [4].

Furthermore, as stated at the end of Section 5.4, it is infeasible to prove the consistency of $\mathsf{Quant}[\psi]$ directly due to the complexity of the definition of ψ (cf. Section 4.9). Instead, we reuse the consistency of $\mathsf{Quant}[\phi]$ from [4] and prove $\psi = \phi$ in Section 10–12. Since ϕ only lives in MT_0 and not in MT, this is another point in favor of keeping MT_0 around.

Thus, we compare MT and MT_0 throughout the present paper so that we may reuse results from [4] and explain which new theorems are needed.

$\mathsf{Quant}[\psi]$, $\mathsf{Quant}[\phi]$ and the other sets of rules of MT and MT_0 are discussed in Section 5.5.

1.6. Relation of Map Theory to other systems

MT_0 was the first system fulfilling Church's original aim at the origin of the creation of (untyped) λ-calculus [5, 6]. 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 (since classes, classes of classes, operators, constructors, etc. also quite directly live in Map Theory), all this was developed in [4, 9] and remains true for MT.

For a comparison of Map Theory with other foundational+computational systems see [4, 9] and also Section 2.2 below. For a version of MT_0 with anti-foundation axioms à la Aczel [1], see [16, 17].

1.7. Computational properties of the canonical models

As a bonus, Appendix A explores the computational properties of the simplest (i.e. the canonical) models of the equational theory MT, w.r.t. the computational rules which are behind it (see Sections 5.2–5.3 for an introduction to canonical and non-canonical models and premodels). Among others, Appendix A addresses the adequacy, soundness and full abstraction of canonical models. In particular we will prove that the "smallest" canonical premodel of MT (case $\kappa = \omega$) is fully abstract w.r.t. the computational rules.

These supplementary results are deferred to an appendix because they are independent of the consistency proof and are quite technical.

1.8. The structure of the paper

To ease navigation, the paper ends with an index (Appendix E). The table of contents of course also supports navigation in addition to exposing the structure of the paper.

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 describes the consistency proof, the models in use and compares MT to MT_0.
Sections 6–13: The consistency proof. See Section 5.6 for an overview.
Sections 14–15: Conclusion and acknowledgements.
Appendix A explores computational properties of canonical models.
Appendix B compares the strength of MT and MT_0.
Appendix C ties up a loose end.
Appendix D summarizes the rules (i.e. axioms and inference rules) of MT.
Appendix E contains the index.

2. Preview of MT

2.1. Map Theory is an equational theory

MT is a Hilbert style axiomatic system which comprises syntactic definitions of terms and wellformed 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 refer to such formulas as *equations*. In MT one cannot (Theorem 2.2.1) prove $\mathsf{T} = \mathsf{F}$.

2.2. Relation to ZFC

Set membership of ZFC is definable as a term E of MT such that $\mathrm{E}xy = \mathsf{T}$ iff the set represented by x belongs to the set represented by y (c.f. Example 4.5.1). We use the infix notation $x \ddot{\in} y$ for $\mathrm{E}xy$. Also definable in MT are universal quantification $\ddot{\forall}$, negation $\ddot{\neg}$, implication $\ddot{\Rightarrow}$, the empty set $\ddot{\emptyset}$ and so on.

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

We now make the statements above more precise. Let σ_o be the smallest inaccessible. Let κ be a regular cardinal greater than σ_o. Let $\mathcal{M}_{\kappa\sigma_o}$ be the *canonical $\kappa\sigma_o$-expansion* built inside ZFC+SI in Section 8 (cf. Definition 8.6.2). The present paper proves the following main theorem:

Theorem 2.2.1 (Consistency Theorem). *$\mathcal{M}_{\kappa\sigma_o}$ satisfies MT.*

Stated another way, the $\kappa\sigma_o$-expansion $\mathcal{M}_{\kappa\sigma_o}$ is a model of MT. Since T trivially differs from F in all $\kappa\sigma$-expansions, the statement trivially implies the consistency of MT. We prove the Consistency Theorem in Sections 6–13 and conclude the proof in Section 13 where we restate the theorem as Theorem 13.1.

Now let $\neg$SI be the assumption that there exist no inaccessible ordinals and let V_{σ_o} be the usual model of ZFC+$\neg$SI in ZFC+SI. For arbitrary, closed formulas $\mathcal{A}$ of ZFC we have:

Theorem 2.2.2. *V_{σ_o} satisfies $\mathcal{A}$ iff $\mathcal{M}_{\kappa\sigma_o}$ satisfies $\ddot{\mathcal{A}} = \mathsf{T}$.*

Theorem 2.2.2 follows easily from [4, Appendix A.4] and the fact that $\mathcal{M}_{\kappa\sigma_o}$ builds on top of the model built in [4]. 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 which are common to MT and MT_0, and Theorem 2.2.2 carries over from MT_0 to MT without changing the definition of $\ddot{\mathcal{A}}$.

Conjecture 2.2.3. *If $\mathcal{A}$ is provable in ZFC+$\neg$SI then $\ddot{\mathcal{A}} = \mathsf{T}$ is provable in MT.*

Conjecture 2.2.3 is supported by the following:

Theorem 2.2.4 ([9]). *If $\mathcal{A}$ is provable in ZFC then $\ddot{\mathcal{A}} = \mathsf{T}$ is provable in* MT_0.

Theorem 2.2.5 ([10]). *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 [10].*

MT_c resembles MT, but all attempts to prove MT_c consistent have failed. A proof of $(\neg\mathrm{SI}) = \mathsf{T}$ in MT_c should be easy. To prove Conjecture 2.2.3 one has to prove $(\neg\mathrm{SI}) = \mathsf{T}$ in MT and to translate the proof of Theorem 2.2.5 to MT. This remains to be done.

It is not really intended that $(\neg\mathrm{SI}) = \mathsf{T}$ should be provable in MT; it is rather a side effect. The original MT_0 was designed to be "as flexible as ZFC", and is in particular consistent with $\mathrm{SI} = \mathsf{T}$ as well as $(\neg\mathrm{SI}) = \mathsf{T}$. As mentioned in Section 1.3, the MT_0 system has a constant ϕ and a group $\mathsf{WF}[\phi]$ of rules. MT replaces the characteristic function ϕ of Φ by ψ. That makes MT more rigid since ψ corresponds to the minimal possible Φ. This should make $(\neg\mathrm{SI}) = \mathsf{T}$ provable since the minimal Φ is analogous to the minimal V_{σ_o} in ZFC+SI. The proof of $(\neg\mathrm{SI}) = \mathsf{T}$ in MT remains to be worked out.

2.3. 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 that $x \ddot{\cup} y$, $\dot{\bigcup} x$, $\dot{\{} x \dot{\}}$ and $\dot{\{} \mathcal{A}[x] \mid x \ddot{\in} \mathcal{B} \dot{\}}$ are the binary union, unary union, unit set and replacement set operators of ZFC, respectively, translated into MT. One may define the successor ordinal $\mathrm{succ}(x)$ thus in MT:

$$\mathrm{succ}(x) \equiv x \ddot{\cup} \dot{\{} x \dot{\}}$$

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

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

Recall from Section 1.3 that we use $\equiv$ for definitional equality. 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 (cf. Section 3.2).

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

Now let $\ddot{\exists} x.\, \mathcal{A}$ and $\varepsilon x.\, \mathcal{A}$ be defined as in Section 4.5 and let $x \ddot{\in} y$ and $x \ddot{\wedge} y$ be defined as in Example 4.5.1. Under reasonable conditions, $\varepsilon x.\, \mathcal{A}$ chooses a wellfounded x such that $\mathcal{A}$ is true. 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:

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

As another example, in MT we may use Hilbert's choice operator ε recursively to define a well-ordering of any set. Let a be a map which represents the set to be well-ordered (for the representation of sets by maps see Example 4.5.1). Then define:

$$
\begin{array}{lcl}
f(\alpha) & \equiv & \varepsilon x.\, x \mathrel{\ddot{\in}} g(\alpha) \\
g(\alpha) & \equiv & a \mathbin{\ddot{\setminus}} \dot{\{} f(\gamma) \mathrel{\dot{|}} \gamma \ddot{\in} \alpha \dot{\}} \\
x \prec y & \equiv & \ddot{\exists} \alpha.\, x \mathrel{\ddot{\in}} a \ddot{\setminus} g(\alpha) \mathbin{\ddot{\wedge}} y \mathrel{\ddot{\in}} g(\alpha)
\end{array}
$$

Above, $\prec$ is a well-ordering of the set represented by a. Note that succ, ρ, Ord, $\prec$ and so on can themselves be taken to be terms of MT since we could define e.g. $\mathrm{Ord} \equiv \lambda x.\, \ddot{\exists} y.\, x \ddot{\in} \rho(y)$.

2.4. Russell's paradox

In naive set theory, define $S \equiv \{x | x \notin x\}$ and $R \equiv S \in S$. We have $x \in S \Leftrightarrow x \notin x$ and $R \Leftrightarrow S \in S \Leftrightarrow S \notin S \Leftrightarrow \neg R$ 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 restricting abstraction $\{x \mid p(x)\}$ (and thereby banning S), but that is not an option in MT which allows unlimited use of abstraction and recursion. As an example, one may define a variant $\mathcal{R}$ of Russell's paradoxical statement as follows 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". In particular, $\ddot{\neg} \perp = \perp$. The definition $\mathcal{R} \equiv \ddot{\neg} \mathcal{R}$ is shorthand for $\mathcal{R} \equiv \mathsf{Y} \ddot{\neg}$ and it is indeed provable in MT that $\mathcal{R} \equiv \mathsf{Y} \ddot{\neg} = \perp$.

One question remains: Map Theory allows to model ZFC and classes, so one may ask what happens to $\{x \mid x \notin x\}$ in Map Theory. We return to that in Example 4.5.1.

2.5. 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 ur-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 (cf. Section 3.5). 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 it 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 (cf. `http://lox.la/`).

Sections 3.5–3.9 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.

2.6. Computation of Russell's paradox

Russell's paradoxical statement $\mathcal{R} \equiv \ddot{\neg}\mathcal{R}$ does not contain ε so one may ask a computer to compute it. If doing so, the computer will loop indefinitely. Thus, according to the computer, $\mathcal{R} = \bot$ (even if the computer never says so).

If one asks the computer to compute $\ddot{\neg}\mathcal{R}$ it also loops indefinitely. Thus, $\ddot{\neg}\mathcal{R} = \bot$ according to the computer. Hence, $\mathcal{R} = \ddot{\neg}\mathcal{R}$ as expected.

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.

The syntax of MT is stated in Section 3.2 and the rules (i.e. axioms and inference rules) in Section 4. Appendix D provides a summary of MT.

3.2. Syntax

The syntax of variables $\langle\text{var}\rangle$, terms $\langle\text{term}\rangle$ and wellformed formulas $\langle\text{wff}\rangle$ of MT reads:

$$
\begin{array}{lcl}
\langle\text{var}\rangle & ::= & x \mid y \mid z \mid \cdots \\
\langle\text{term}\rangle & ::= & \langle\text{var}\rangle \\
 & \mid & \lambda\langle\text{var}\rangle.\,\langle\text{term}\rangle \\
 & \mid & \langle\text{term}\rangle\langle\text{term}\rangle \\
 & \mid & \mathsf{T} \\
 & \mid & \mathsf{if}[\langle\text{term}\rangle, \langle\text{term}\rangle, \langle\text{term}\rangle] \\
 & \mid & \bot \\
 & \mid & \mathsf{Y}\langle\text{term}\rangle \\
 & \mid & \langle\text{term}\rangle \| \langle\text{term}\rangle \\
 & \mid & \mathsf{E}\langle\text{term}\rangle \\
 & \mid & \varepsilon\langle\text{term}\rangle \\
\langle\text{wff}\rangle & ::= & \langle\text{term}\rangle = \langle\text{term}\rangle
\end{array}
$$

Or, terser:

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

Recall from Section 1 that we use *construct* as shorthand for *language construct* and from Section 1.3 that we use $\equiv$ for definitional equality. The intuition behind the constructs above is as follows:

$\lambda x.\,\mathcal{A}$ denotes lambda abstraction.

juxtaposition denotes functional application. As an example, fx denotes f applied to x.

T denotes truth. Falsehood F is not included in the syntax; we define it by $\mathsf{F} \equiv \lambda x.\,\mathsf{T}$. Later, we also use T to denote the empty set, the empty tuple and the natural number 0.

if denotes selection; we have $\mathsf{if}[\mathsf{T}, b, c] = b$ and $\mathsf{if}[\lambda x.\mathcal{A}, b, c] = c$. Later, we also use selection to define a pairing construct $b{::}c \equiv \lambda x.\,\mathsf{if}[\,x\,,\,b\,,\,c\,]$.

$\bot$ 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; $a \parallel b$ equals T if a or b or both equal T. 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 (Theorem 3.8.2). We use full abstraction to explain equality in Section 3.8.

E denotes pure existence; we have $\mathsf{E}a = \mathsf{T}$ iff $ax = \mathsf{T}$ for some x.

ε denotes Hilbert's choice operator; under reasonable conditions, εa is a *wellfounded* x such that $ax = \mathsf{T}$. Wellfoundedness is explained in Section 3.10

$=$ denotes equality. Equality is described in Section 3.8. As a preview, terms which are $\alpha\beta$-equivalent are equal. The opposite does not hold.

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 e.g. $\lambda xy.\,xyy$ means $\lambda x.\,\lambda y.\,((xy)y)$. The term $a\lambda b.\,cd$ means $a(\lambda b.\,(cd))$ since abstractions extend as far as possible to the right but cannot extend to the left. Binary operators like $x \parallel y$ have priority between application and abstraction so $\lambda x.\,xx \parallel xx$ means $\lambda x.\,((xx) \parallel (xx))$. Occasionally, formally superfluous parentheses are added for the sake of readability.

3.3. Expansions and models

In Section 5.2 we introduce the notions of κ- and $\kappa\sigma$-*expansions*.

κ-expansions are mathematical structures defined for all regular ordinals $\kappa \geq \omega$ and they model all constructs of MT except ε. In contrast, $\kappa\sigma$-expansions are defined for all regular $\kappa > \sigma$ where σ is inaccessible, and $\kappa\sigma$-expansions model all constructs of MT. Apart from that, κ- and $\kappa\sigma$-expansions are identical.

All κ- and $\kappa\sigma$-expansions satisfy some axioms and inference rules of MT and some $\kappa\sigma$-expansions satisfy all of MT. We refer to $\kappa\sigma$-expansions which model all of MT as $\kappa\sigma$-*models*.

Let $\mathcal{M}_\kappa$ and $\mathcal{M}_{\kappa\sigma}$ be the *canonical* κ- *and* $\kappa\sigma$-*expansion*, respectively, as introduced in Section 5.3. For each regular $\kappa \geq \omega$ there are many κ-expansions but $\mathcal{M}_\kappa$ is the only canonical one, and likewise for $\kappa\sigma$-expansions.

As already stated in the Consistency Theorem (Theorem 2.2.1/Theorem 13.1), $\mathcal{M}_{\kappa\sigma_o}$ models MT if σ_o is the first inaccessible and $\kappa > \sigma_o$ is regular.

3.4. $\mathrm{MT}_{\mathrm{def}}$

Define

$$\begin{array}{lcl} \mathsf{Y}_{\mathrm{Curry}} & \equiv & \lambda f.\, (\lambda x.\, f(xx))(\lambda x.\, f(xx)) \\ \bot_{\mathrm{Curry}} & \equiv & \mathsf{Y}_{\mathrm{Curry}} \lambda x.\, x \qquad = \quad (\lambda x.\, xx)(\lambda x.\, xx) \end{array}$$

In canonical κ-expansions we will prove (Theorem 9.5.3) that

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

Then, by the Min rule stated in Section 4.2, we trivially have

$$\bot = \bot_{\mathrm{Curry}}$$

Thus, without loss of power and consistency, one might omit $\bot$ and Y from the syntax and use $\bot_{\mathrm{Curry}}$ and $\mathsf{Y}_{\mathrm{Curry}}$ instead. Doing so, however, would reduce the number of possible models of MT.

We include $\bot$ and Y in the syntax. We prove $\mathsf{Y} f = \mathsf{Y}_{\mathrm{Curry}} f$ as a separate theorem (Theorem 9.5.3) which is only guaranteed in canonical expansions. Inclusion of $\bot$ and Y also simplifies the consistency proof since modelling of $\mathsf{Y} f$ and proving $\mathsf{Y} f = \mathsf{Y}_{\mathrm{Curry}} f$ can be treated separately.

We use $\mathrm{MT}_{\mathrm{def}}$ to denote the version of MT where we omit $\bot$ and Y from the syntax.

3.5. Basic computation

The constructs $\lambda x.\, \mathcal{A}$, $\mathcal{A}\mathcal{B}$, T and $\mathsf{if}[\mathcal{A}, \mathcal{B}, \mathcal{C}]$ 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 this section, $\mathcal{A}$ and $\mathcal{B}$ 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 [11], one can define the programming language by the smallest relation $\stackrel{1}{\rightarrow}$ which satisfies:

$$
\begin{array}{lcl}
\mathsf{T}b & \stackrel{1}{\rightarrow} & \mathsf{T} \\
(\lambda x.\, \mathcal{A})b & \stackrel{1}{\rightarrow} & \langle \mathcal{A} \mid x := b \rangle \\
\mathsf{if}[\mathsf{T}, b, c] & \stackrel{1}{\rightarrow} & b \\
\mathsf{if}[\lambda x.\, \mathcal{A}, b, c] & \stackrel{1}{\rightarrow} & c \\
a \stackrel{1}{\rightarrow} r & \Rightarrow & ab \stackrel{1}{\rightarrow} rb \\
a \stackrel{1}{\rightarrow} r & \Rightarrow & \mathsf{if}[a, b, c] \stackrel{1}{\rightarrow} \mathsf{if}[r, b, c]
\end{array}
$$

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

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

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 $a \stackrel{1}{\rightarrow} b$. 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 κ-expansions, even those which do not model all of MT (cf. Theorem 7.5.2). That holds for the definition of $a \stackrel{1}{\rightarrow} b$ given above as well as for the extensions given in the following.

We 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,

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

reduces to

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

which cannot be reduced further. In particular, the term above does not reduce to $\lambda y.\, (\lambda x.\, xxx)(\lambda x.\, xxx)$. We refer to terms of form T and $\lambda x.\, \mathcal{A}$ as *true* and *function* normal terms, respectively.

3.6. Further computation

One may extend the programming language by the constructs $\bot$, $\mathsf{Y}\mathcal{A}$, $\mathcal{A} \parallel \mathcal{B}$ and $\mathsf{E}\mathcal{A}$. One cannot extend the programming language by $\varepsilon\mathcal{A}$ because ε cannot be seen as computable.

In this section, 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 (case $\mathrm{MT}_{\mathrm{def}}$), they need no reduction rules. If they are included in the syntax (case MT), their reduction rules read:

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

The construct $a \parallel b$ can be computed as follows. 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 as follows. 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 [11] supports neither $\mathsf{E}a$ nor $a \parallel b$.

The construct $\mathsf{E}a$ is needed for defining ψ (c.f. Section 4.7) 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:

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

Note the swapping of arguments in the third rule above. The swapping makes reduction alternate between reduction of a and b. As an example, $(\lambda x.\, xx)(\lambda x.\, x) \parallel \mathsf{T} \stackrel{1}{\to} \mathsf{T} \parallel (\lambda x.\, x)(\lambda x.\, x) \stackrel{1}{\to} \mathsf{T}$.

Giving 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 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:

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

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 give no reduction rules for εa since, as mentioned, it is not computable.

3.7. Programs

We refer to closed, ε-free MT terms as *MT programs*. Likewise, we 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. $\bot$, and $\bot$ is counted among the programs.

3.8. Equality

Wellformed formulas of MT have form $\mathcal{A} = \mathcal{B}$ where $\mathcal{A}$ and $\mathcal{B}$ are terms. We now present some intuition concerning 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 $\mathcal{N}_\bot$ be the set of the remaining MT programs. We now define *root equivalence* $a \sim b$ and *observational equality* $a =_{\mathrm{obs}} b$.

Definition 3.8.1. *For MT programs a and b define:*

Root equivalence	$a \sim b$	*iff*	$(a \in \mathcal{N}_t \Leftrightarrow b \in \mathcal{N}_t) \wedge (a \in \mathcal{N}_f \Leftrightarrow b \in \mathcal{N}_f)$
Obs. equality	$a =_{\mathrm{obs}} b$	*iff*	*$ca \sim cb$ for all MT programs c.*

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

Recall that the canonical κ-expansion $\mathcal{M}_\kappa$ models all constructs of MT except ε for all regular $\kappa \geq \omega$. Now let $a =_\kappa b$ denote $\mathcal{M}_\kappa \models a = b$. We have:

Theorem 3.8.2 (Full Abstraction of $\mathcal{M}_\omega$). *$a =_{\mathrm{obs}} b \Leftrightarrow a =_\omega b$ for all MT programs a and b.*

See Theorem A.7.2 for a proof and Appendix A for related positive and negative results. Full abstraction may help understanding MT except ε.

We have $a \sim b \Leftrightarrow (a \in \mathcal{N}_t \Leftrightarrow b \in \mathcal{N}_t) \wedge (a \in \mathcal{N}_\perp \Leftrightarrow b \in \mathcal{N}_\perp)$ since each of a and b belongs to exactly one of $\mathcal{N}_t$, $\mathcal{N}_f$ and $\mathcal{N}_\perp$. Now for all $a, b \in \mathcal{M}_\kappa$ 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 is the following purely semantic observation:

Fact 3.8.3. *$a =^\kappa_{\mathrm{obs}} b \Leftrightarrow a =_\kappa b$ for all $a, b \in \mathcal{M}_\kappa$, $\kappa \geq \omega$, κ regular.*

The fact follows trivially from the definition of $\mathcal{M}_\kappa$ (cf. Section 8.7).

3.9. Semantic 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 directly from Theorem 3.8.2 (=Theorem A.7.2), Theorem 9.1.2 (using $\kappa = \omega$) and Theorem A.5.5, and provides another intuitive description of equality. We also have:

Fact 3.9.1. *Let $a, b \in \mathcal{M}_\kappa$, $\kappa \geq \omega$ regular. The following are equivalent:*

$ca \sim_\kappa cb$	*for all $c \in \mathcal{M}_\kappa$*
$ay_1 \cdots y_n \sim_\kappa by_1 \cdots y_n$	*for all $n \geq 0$ and all $y_1, \ldots, y_n \in \mathcal{M}_\kappa$.*

Fact 3.9.1 follows from Fact 3.8.3 and Theorem 9.1.2. The ZFC equivalent of Fact 3.9.1 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 refer to Fact 3.9.1 as *semantic extensionality*; we express it axiomatically in Section 4.3.

3.10. Wellfoundedness

We have now described all constructs of MT except ε. To describe ε we first have to introduce the notion of wellfoundedness.

To explain wellfoundedness we resort, as in [4], to any κ-expansion $\mathcal{M}$ (cf. Section 3.3) where κ is regular and greater than at least one inaccessible ordinal. We refer to elements of $\mathcal{M}$ as maps.

For each inaccessible $\sigma < \kappa$ there is a set Φ of maps as defined in Definition 7.8.2. At the present stage there is no need to know what Φ is precisely except that given κ there is one for each inaccessible $\sigma < \kappa$. We refer to elements of Φ as *wellfounded* maps.

As before, let V_σ be the usual (wellfounded) model of ZFC inside ZFC+SI in which $\mathcal{M}$ itself is built. There exists [4, Appendix A.4] a surjective function

$Z\colon\Phi\to V_\sigma$ which allows to represent all sets of V_σ by elements of Φ. Example 4.5.1 defines Z and elaborates on that.

The semantic definition of wellfoundedness given in Definition 7.8.2 is robust in that it is pertinent for a large class of structures, it is close to the semantic intuitions behind Map Theory and it is independent of its diverse possible axiomatisations. Therefore, it is the definition we retain in this paper, as we did in [4] when treating MT_0.

3.11. Provable wellfoundedness

Suppose now that σ is the *first* inaccessible. Then, by the Definability Theorem (Theorem 10.1), we have $\Phi = \{x{\in}\mathcal{M} \mid \psi x = \mathsf{T}\}$ where ψ is the term defined in Section 4.7.

Starting from ψ could hence give us an alternative definition of wellfoundedness, but only pertinent for MT. The interest of the second definition is that it comes with the proof theoretic notion of being provably wellfounded in MT, which we describe now.

By definition, a closed term a is provably wellfounded in MT if $\psi a = \mathsf{T}$ is provable in MT. Likewise a is provably wellfounded in MT_0 if $\phi a = \mathsf{T}$ is provable in MT_0 where ϕ is a construct of MT_0 intended to be the characteristic function of Φ.

In Section 4.8 we give examples illustrating that usual data structures are provably wellfounded in MT (they were also provably wellfounded in MT_0, but with very different proofs).

Provable wellfoundedness is the relevant tool for developing proofs inside MT and for interpreting ZFC in MT. But for the purpose of this paper, which is to prove the consistency of MT, wellfoundedness as defined in Definition 7.8.2 is the most relevant and enlightening.

Now let ψ_{Curry} be defined exactly like ψ except that all occurrences of Y and $\bot$ are replaced by $\mathsf{Y}_{\mathrm{Curry}}$ and $\bot_{\mathrm{Curry}}$, respectively. A closed term a is provably wellfounded in $\mathrm{MT}_{\mathrm{def}}$ if $\psi_{\mathrm{Curry}} a = \mathsf{T}$ is provable in $\mathrm{MT}_{\mathrm{def}}$. In canonical models we have $\mathsf{Y} = \mathsf{Y}_{\mathrm{Curry}}$ and $\bot = \bot_{\mathrm{Curry}}$. Thus, in canonical models, we have $\psi = \psi_{\mathrm{Curry}}$ and $\Phi = \{x{\in}\mathcal{M} \mid \psi_{\mathrm{Curry}} x = \mathsf{T}\}$.

3.12. Hilbert's choice operator

To explain ε we resort, like in Section 3.10, to a κ-expansion $\mathcal{M}$ where κ is regular and greater than at least one inaccessible ordinal.

We say that $a \in \mathcal{M}$ is *total*, written $\mathrm{Total}(a)$, if $ax \neq \bot$ for all $x \in \Phi$.

We use $\underline{\varepsilon}$ to denote the intended interpretation of Hilbert's choice operator. More specifically, $\underline{\varepsilon}$ is a function of type $\mathcal{M}\to\mathcal{M}$ which has the following properties for all $a \in \mathcal{M}$:

$$\begin{array}{lcll}
\underline{\varepsilon}(a) & = & \bot & \text{if } \neg\mathrm{Total}(a) \\
\underline{\varepsilon}(a) & \in & \Phi & \text{if } \mathrm{Total}(a) \\
a(\underline{\varepsilon}(a)) & = & \mathsf{T} & \text{if } \mathrm{Total}(a) \wedge \exists x{\in}\Phi\colon ax = \mathsf{T} \\
\underline{\varepsilon}(a) & = & \underline{\varepsilon} b & \text{if } \mathrm{Total}(a) \wedge \mathrm{Total}(b) \wedge \forall x{\in}\Phi\colon (ax = \mathsf{T} \Leftrightarrow bx = \mathsf{T})
\end{array}$$

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

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

3.13. The need for inaccessibility

We repeatedly assume that σ is the first inaccessible and that κ is greater than σ. That may give rise to the questions: Why inaccessible? Why first? Why greater?

Since we can interpret ZFC in Map Theory it should be no surprise that to prove the consistency of Map Theory we need something strong enough to prove the consistency of ZFC. In [9] there are some results which use relativization and the assumption that ZFC is consistent instead of assuming the existence of an inaccessible. But those results and their proofs are cumbersome and not very general. That hints at why we assume the existence of an inaccessible.

Then the Definability Theorem (Theorem 10.1) proves $\Phi = \{x{\in}\mathcal{M} \mid \psi x = \mathsf{T}\}$ for the first inaccessible σ where ψ is the term defined in Section 4.7 and Φ is the "universe of wellfounded maps" that we introduced informally in Section 1.2, and whose formal definition (Definition 7.8.2) depends on σ. The term ψ defined in Section 4.7 is the simplest one we have found so far which allows to formulate a version of Map Theory strong enough to develop ZFC in it.

That term ψ happens to be the characteristic function of the Φ associated to the smallest inaccessible. One could imagine the use of another ψ which was the characteristic function of another Φ, but in the present paper we use the ψ of Section 4.7 and that forces us to use the first inaccessible.

Fixing ψ is also the point which makes MT less flexible than MT_0 in terms of compatibility with extensions of ZFC. For each consistent extension of ZFC there is an associated consistent variant of MT_0 [9, Theorem 15.5.1] which can prove all the theorems of the given extension. To get something similar for MT one would have to find a new ψ (if any) for each extension of ZFC.

Finally, we assume that the regular cardinal κ satisfies $\kappa > \sigma$. The consistency proof presented in the present paper is based on so-called κ-Scott semantics and κ-continuity (where κ-Scott semantics and κ-continuity is usual Scott semantics and continuity, respectively, for $\kappa = \omega$). In the κ-Scott approach we can model ε iff $\underline{\varepsilon}$ as defined in Section 3.12 is κ-continuous, and that happens to require that Φ is the upwards closure of a set of cardinality less than κ (Theorem 7.7.3) which is true only if $\kappa > \sigma$.

To summarize the above, we use the inaccessibility of σ in many places (e.g. for defining Φ, for modelling ε and for proving the Definability Theorem). We only use that σ is the *first* inaccessible in the proof of Lemma 12.4.2 which constitutes part of the proof of the Definability Theorem.

3.14. Pure existence revisited

Let $\kappa \geq \omega$ be regular and let $\mathcal{M}$ be any κ-expansion. Pure existence E is designed to satisfy in $\mathcal{M}$ that $\mathsf{E}a = \mathsf{T}$ if $ax = \mathsf{T}$ for some x and $\mathsf{E}a = \bot$ otherwise (cf. Section 4.4). So, $\mathsf{E}a = \mathsf{T}$ in $\mathcal{M}$ iff $ax = \mathsf{T}$ for some $x \in \mathcal{M}$ while the reduction rule for $\mathsf{E}a$ given in Section 3.6 gives that $\mathsf{E}a = \mathsf{T}$ iff $ax = \mathsf{T}$ for some program x. We now compare these two notions of existential quantification. Define pure and computational existence as follows:

$$\begin{array}{lcl} \mathsf{E}_{\mathrm{pure}} & \equiv & \lambda a.\, \mathsf{E}a \\ \mathsf{E}_{\mathrm{comp}} & \equiv & \lambda a.\, [\, a\mathsf{C}_1 \parallel \cdots \parallel a\mathsf{C}_8 \parallel \mathsf{E}_{\mathrm{comp}}\, \lambda u.\, \mathsf{E}_{\mathrm{comp}}\, \lambda v.\, a(uv)] \end{array}$$

We have

$$\begin{array}{lclll} \mathsf{E}_{\mathrm{pure}}\, a & = & \mathsf{T} & \text{iff} & ax = \mathsf{T} \text{ for some map } x \\ \mathsf{E}_{\mathrm{comp}}\, a & = & \mathsf{T} & \text{iff} & ax = \mathsf{T} \text{ for some program } x \end{array}$$

The canonical ω-expansion $\mathcal{M}_\omega$ happens to be a simple and very pertinent model for 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 $\mathsf{E}_{\mathrm{pure}} = \mathsf{E}_{\mathrm{comp}}$ (cf. Lemma A.4.1). Now, this equation can be proved to be false in $\mathcal{M}_\kappa, \kappa > \omega$ (cf. Theorem A.8.1 and its proof), and more generally should be false in all the models of MT built from κ-premodels ($\kappa > \omega$), for a similar reason (these models are in a sense "too big").

The E of MT is the pure one. Indeed, the computational intuition behind E that we provided at the end of Section 3.6 is valid in $\mathcal{M}_\omega$ but does not hold in full MT.

4. Rules (i.e. axioms and inference rules)

MT has six groups of *rules* (where *rules* means *axioms and inference rules*):

Elem	Elementary rules common to MT and MT_0	Section 4.1
Elem′	Further elementary rules	Section 4.1
Mono/Min	Monotonicity and Minimality	Section 4.2
Ext	Extensionality	Section 4.3
Exist	The axioms on E	Section 4.4
Quant[ψ]	Quantification axioms	Section 4.5

The syntax of MT was stated in Section 3.2 and Appendix D summarizes MT.

4.1. Elementary axioms and inference rules

Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ be (possibly open) terms and let x and y be variables. Let $\mathbf{1} \equiv \lambda xy.\, xy$, i.e. let $\mathbf{1}$ be the term that Church happened to use for the number 1. The two first sets of rules (i.e. axioms and inference rules) of MT read:

Trans	$\mathcal{A} = \mathcal{B}; \mathcal{A} = \mathcal{C} \vdash \mathcal{B} = \mathcal{C}$
Sub	$\mathcal{A} = \mathcal{B}; \mathcal{C} = \mathcal{D} \vdash \mathcal{AC} = \mathcal{BD}$
Gen	$\mathcal{A} = \mathcal{B} \vdash \lambda x.\, \mathcal{A} = \lambda x.\, \mathcal{B}$
A1	$\mathsf{T}\mathcal{B} = \mathsf{T}$
A2 (β)	$(\lambda x.\, \mathcal{A})\mathcal{B} = \langle \mathcal{A} \mid x := \mathcal{B} \rangle$ if $\mathcal{B}$ is free for x in $\mathcal{A}$
A3	$\bot\mathcal{B} = \bot$
Rename (α)	$\lambda x.\, \langle \mathcal{A} \mid y := x \rangle = \lambda y.\, \langle \mathcal{A} \mid x := y \rangle$ if x is free for y in $\mathcal{A}$ and vice versa
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	$\langle \mathcal{A} \mid x := \mathsf{T} \rangle = \langle \mathcal{B} \mid x := \mathsf{T} \rangle$; $\langle \mathcal{A} \mid x := \mathbf{1}x \rangle = \langle \mathcal{B} \mid x := \mathbf{1}x \rangle$; $\langle \mathcal{A} \mid x := \bot \rangle = \langle \mathcal{B} \mid x := \bot \rangle \vdash$ $\mathcal{A} = \mathcal{B}$

The Elem group of rules

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	$\mathsf{Y}\mathcal{A} = \mathcal{A}(\mathsf{Y}\mathcal{A})$

The Elem′ group of rules

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

Monotonicity was part of the founding intuitions behind Map Theory [4], even if it was not reflected in the first axiomatization MT_0 of Map Theory.

Expressing this intuition at the syntactic level can of course only be done using a syntactic order $\preceq$ which has to be defined first:

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

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. That $\preceq$ is an order is forced by Rule Ext introduced in Section 4.3; this is explained in Example 4.3.2.

The rules of Monotonicity and Minimality read:

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

The Mono and Min rules

Mono and Min force the constant Y to behave, at the syntactic level, as a fixed point operator which is minimal w.r.t. the syntactic order $\preceq$.

As illustrated by the following example, the principle of induction follows from minimality.

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 refer to elements of $\mathcal{M}$ as maps.

We 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 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 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:

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

As an example, $0''$ denotes one among many maps which represents 'two'.

The semantics of $\bar{\chi}$ in the model $\mathcal{M}$ is that $\bar{\chi}x = \mathsf{T}$ if x is a natural number map and $\bar{\chi}x = \perp$ otherwise. For that reason we refer to $\bar{\chi}$ as the *characteristic map* of the class of natural number maps (cf. Definition 4.6.1). We say that a map m is a *covariant representation* of $\{x{\in}\mathcal{M} \mid mx = \mathsf{T}\}$ so $\bar{\chi}$ covariantly represents the set of natural number maps (Example 4.5.1 introduces a *contravariant* representation). The covariant representation is the one used by Church and others for representing sets.

For all 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 $\mathcal{E} \equiv (\lambda x.\, x \stackrel{\omega}{=} x) = \lambda x.\, \mathsf{if}[x, \mathsf{if}[x, \mathsf{T}, \mathsf{F}], \mathsf{if}[x, \mathsf{F}, x\mathsf{T} \stackrel{\omega}{=} x\mathsf{T}]] = \lambda x.\, \mathsf{if}[x, \mathsf{T}, x\mathsf{T} \stackrel{\omega}{=} x\mathsf{T}]$ where the latter equality requires QND. Hence, we can prove $\varpi\mathcal{E} = \mathcal{E}$ so $\varpi\mathcal{E} \preceq \mathcal{E}$ (cf. Example 4.3.2). Hence, we can prove $\bar{\chi} \preceq \mathcal{E}$ by Min.

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

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

From a program correctness point of view we have now done the following: We have defined an inductive data type (the natural numbers) and we have represented it by its characteristic map $\bar{\chi}$. Then we have written a program $\lambda xy.\, (x \stackrel{\omega}{=} y)$ which can compare two natural numbers for equality. And finally we have proved $\bar{\chi} \preceq \mathcal{E}$ which expresses that every natural number equals itself.

While this is a very simple example and even though we do not write out detailed proofs, 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 $\lambda xy.\, x \stackrel{\omega}{=} y$ is a program; one can compile it and run it on arguments x and y using the system described in [11].

In other logical frameworks than MT, given a recursive program like $\lambda xy.\, x \stackrel{\omega}{=} y$, proofs of theorems like $(x \stackrel{\omega}{=} x) = \mathsf{T}$ for all natural numbers x usually requires some sort of Peano induction. In MT, induction is expressed by Min.

In the example above, we applied Min to the characteristic function $\bar{\chi}$ of natural number maps to get something equivalent to Peano induction (cf. [10, Section 7.13]). One can do the same for arbitrary inductive data types and even for Φ: applying Min to the characteristic map ψ defined in Section 4.7 yields an induction scheme which resembles but is stronger than transfinite induction (cf. [10, Section 9.13]).

4.3. Rule Ext

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

Ext	If x and y are not free in $\mathcal{A}$ and $\mathcal{B}$ then $\approx(\mathcal{A}x) = \approx(\mathcal{B}x); \mathcal{A}xy{=}\mathcal{A}\mathcal{C}; \mathcal{B}xy{=}\mathcal{B}\mathcal{C} \vdash \mathcal{A}x = \mathcal{B}x$

The Ext rule

Note that if the premises of Ext hold, if $c = \lambda xy.\, \mathcal{C}$ and if $y_1, y_2, \ldots$ are not free in $\mathcal{A}$ and $\mathcal{B}$, then we have e.g.

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

More generally, we have $\approx(\mathcal{A}xy_1 \cdots y_n) = \approx(\mathcal{A}xy_1 \cdots y_n)$. Now, canonical κ-expansions $\mathcal{M}_\kappa$ have a semantic extensionality property which says that if $a, b \in \mathcal{M}_\kappa$ and if

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

for all natural numbers n and all $y_1, \ldots, y_n \in \mathcal{M}_\kappa$ then $a = b$ (cf. Fact 3.9.1). Rule Ext is a syntactical approximation of this property 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 as defined in Section 3.9 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 $a = b$. The set theory formula $\mathcal{P} \Leftrightarrow \mathcal{Q}$ corresponds to $\approx\mathcal{P} = \approx\mathcal{Q}$ in MT, and $y \in a$ corresponds to $ay_1 \cdots y_n$.

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 [10]).

When developing ZFC in MT, Ext plays a marginal but essential role [10]. 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.

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 $F_2 =_{\text{obs}} F_3$. 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. We conjecture that $F_2 = F_3$ is not provable in MT_0 (there is no reason why it should be).

Proving $F_2 = F_3$ directly (i.e. without establishing a collection of convenience lemmas first) is a tricky exercise. To get started, define $F_1 \equiv \lambda x.\, F_1$, $\mathcal{A} \equiv \lambda x.\, \mathsf{if}[\, x\, ,\, F_1\, ,\, F_1\,]$, $\mathcal{B} \equiv \lambda x.\, \mathsf{if}[\, x\, ,\, F_2\, ,\, \lambda y.\, F_2\,]$ and $\mathcal{C} \equiv \mathsf{if}[\, x\, ,\, \mathsf{F}\, ,\, \mathsf{T}\,]$. Then, using QND, prove the premises of Ext and conclude $\mathcal{A}x = \mathcal{B}x$. Then $\mathcal{A}\mathsf{T} = \mathcal{B}\mathsf{T}$ gives $F_1 = F_2$. Proceed by proving $F_1 = F_3$.

Section 14.1 mentions yet another use of Ext.

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 the model. Its axiomatization is a syntactical approximation of this. Now define:

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

We have $(g \circ h)z = g(hz)$ so $(g \circ h)$ is the functional composition of g and h. The equation $x \to y$ expresses "if $x = \mathsf{T}$ then $y = \mathsf{T}$". Finally, χg is the *characteristic map* (cf. Definition 4.6.1) for which $\chi g x = \mathsf{T}$ iff $gx = \mathsf{T}$ and $\chi g x = \bot$ otherwise.

The axioms on E read:

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

The Exist group of rules

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

4.5. Quantification axioms (i.e. axioms on ε)

Define:

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

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 the term ψ defined in Section 4.7. Let $\mathcal{M}$ be as in Section 3.12. From the properties of ε stated in Section 3.12 and for all maps $p \in \mathcal{M}$ we have

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

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

ElimAll	$(\ddot{\forall}x.\,\mathcal{A}) \wedge \psi\mathcal{B} \rightarrow (\lambda x.\,\mathcal{A})\mathcal{B}$
Ackermann	$\varepsilon x.\,\mathcal{A} = \varepsilon x.\,(\psi x \wedge \mathcal{A})$
StrictEpsilon	$\psi(\varepsilon x.\,\mathcal{A}) = \ddot{\forall}x.\,!(\mathcal{A})$
StrictAll	$!(\ddot{\forall}x.\,\mathcal{A}) = \ddot{\forall}x.\,!(\mathcal{A})$

The Quant[ψ] group of rules

The quantification axioms are axioms on ε, but in some of them ε only appears implicitly.

ElimAll says that if $p(x)$ is true for all wellfounded x and if $\mathcal{B}$ is wellfounded then $p(\mathcal{B})$ is true.

Ackermann (Ackermann's axiom) says that $\varepsilon x.p(x)$ only depends on the truth value of $p(x)$ for wellfounded x. In other words, $\varepsilon x.p(x)$ does not care about $p(x)$ for non-wellfounded x. Furthermore, if x is wellfounded and $p(x)$ is neither T nor $\bot$, then ε considers $p(x)$ false and does not care about the exact value of $p(x)$.

StrictEpsilon says that ε is strict (cf. Section 3.12) in the sense that that $\varepsilon x.p(x)$ is $\bot$ if $p(x)$ is $\bot$ for one or more wellfounded x. Likewise, StrictAll says that $\ddot{\forall}$ is strict.

Example 4.5.1. According to the Strong Induction Property (SIP, c.f. Section 7.8), elements of Φ are wellfounded w.r.t. Φ (see Example 4.2.1 for the definition of wellfoundedness with respect to a set). This allows to introduce a representation of sets of ZFC by elements of Φ which we shall refer to as the *contravariant* representation (see Example 4.2.1 for the covariant representation). We define the set $Z[x]$ contravariantly represented by $x \in \Phi$ thus:

$$\begin{array}{lcl} Z[\mathsf{T}] & \equiv & \emptyset \\ Z[x] & \equiv & \{\ Z[xz] \ \mid\ z \in \Phi\ \} \quad \text{if } x \neq \mathsf{T} \end{array}$$

For the usual model V_σ of ZFC in ZFC+SI and canonical $\kappa\sigma$-expansions $\mathcal{M}_{\kappa\sigma}$ of MT we have $V_\sigma = \{Z[x] \mid x \in \Phi\}$ (cf. [4, Appendix A.4]) so all sets of ZFC are representable by wellfounded maps $x \in \Phi$. Now define:

$$\begin{array}{lcl} x \ddot{\Rightarrow} y & \equiv & \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \mathsf{F}], \mathsf{if}[y, \mathsf{T}, \mathsf{T}]] \\ x \ddot{\wedge} y & \equiv & \ddot{\neg}(x \ddot{\Rightarrow} \ddot{\neg} y) \\ x \ddot{=} y & \equiv & \mathsf{if}[x, \mathsf{if}[y, \mathsf{T}, \mathsf{F}], \mathsf{if}[y, \mathsf{F}, (\ddot{\forall} u \ddot{\exists} v.\, xu \ddot{=} yv) \ddot{\wedge} (\ddot{\forall} v \ddot{\exists} u.\, xu \ddot{=} yv)]] \\ x \ddot{\in} y & \equiv & \mathsf{if}[y, \mathsf{F}, \ddot{\exists} z.\, x \ddot{=} yz] \end{array}$$

For all $x, y \in \Phi$ we have $(x \ddot{\in} y) = \mathsf{T}$ iff $Z[x] \in Z[y]$ and $(x \ddot{=} y) = \mathsf{T}$ iff $Z[x] = Z[y]$. The definition 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. By Theorem 2.2.2 all closed theorems of ZFC are satisfied by the canonical model $\mathcal{M}_{\kappa\sigma}$ of MT (actually, they are satisfied by all $\kappa\sigma$-expansions, σ inaccessible, $\kappa > \sigma$). As a conjecture (Conjecture 2.2.3), closed theorems of ZFC+$\neg$SI are provable in MT.

The map $I \equiv \lambda x.\, x$ happens not to be wellfounded. But if it were we would have $Z[I] = \{Z[Ix] \mid x \in \Phi\} = \{Z[x] \mid x \in \Phi\}$ so I is a reasonable representation for the class of all sets. In general, no wellfounded map represents the class of all sets.

For all wellfounded x we have $x \ddot{\notin} x$ so Russell's paradoxical $\{x \mid x \notin x\}$ is the class of all sets, and we could represent it by I. So Russell's paradoxical set is in Map Theory, but is not wellfounded in the sense of Map Theory. Likewise, Burali-Forti's "set" of all ordinals is in Map Theory, but is not wellfounded in the sense of Map Theory.

The covariant representation mentioned in Example 4.2.1 where a map m represents $\{x \mid mx = \mathsf{T}\}$ was the one used by Church and others for representing classes. That representation seems to be entirely unsuited for representing ZFC sets. In contrast, the contravariant representation introduced in the present example works well for developing ZFC. When working with MT, one typically has to use both co- and contravariant representations.

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

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

equals T. The proof involves a proof of $\ddot{\forall} y.\, x + y \stackrel{\omega}{=} y + x$ by induction on x (or, more precisely, a proof of $\bar{\chi} \preceq \lambda x.\, \ddot{\forall} y.\, x + y \stackrel{\omega}{=} y + x$ 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. The ability to mix recursive programs and quantifiers freely is a core benefit of MT.

4.6. Preliminaries for the definition of ψ

We conclude the presentation of the axioms by defining ψ. Like in Section 3.12 let $\mathcal{M}$ be any $\kappa\sigma$-expansion. We first define some auxiliary concepts.

Definition 4.6.1. for all $a \in \mathcal{M}$ define:
(a) a is a *characteristic map* if $a \in \mathcal{M} \setminus \{\mathsf{T}, \bot\}$ and $ax \in \{\mathsf{T}, \bot\}$ for all $x \in \mathcal{M}$.
(b) $\mathsf{D}[a] = \{x \in \mathcal{M} \mid ax = \mathsf{T}\}$
(c) a is the *characteristic map of S* if a is a characteristic map and $S = \mathsf{D}[a]$.

In Example 4.2.1 we referred to $\bar{\chi}$ as "the characteristic map of the set of natural number maps", which is coherent with the definition above.

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

The map $\sqcup$ trivially satisfies

Fact 4.6.3. $\mathsf{D}[\sqcup f] = \cup_{x \in \mathcal{M}} \mathsf{D}[fx]$.

Furthermore, $x : y$ is "y guarded by x" in the sense that if $x = \mathsf{T}$ then $x : y = y$ and if $x \neq \mathsf{T}$ then $x : y = \bot$. We make $x : y$ right associative so that $x : y : z$ means $x : (y : z)$. Thus, $x : y : z$ is z guarded by both x and y. Since $x : y$ is an infix operator we have that $xu : yv$ means $(xu) : (yv)$.

One may think of f/g as a projection in the sense that $(f/g)/g = f/g \preceq f$ holds in $\mathcal{M}$ (cf. Lemma 10.5.3). The f/g construct equals $\Downarrow_G f$ of [4]. Since f/g is an infix operator we have that fx/gy means $(fx)/(gy)$.

4.7. The definition of ψ

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, P, Q and R will satisfy:

$$\begin{array}{llcl} & \psi & \equiv & \sqcup s \\ \text{so} & \mathsf{D}[\psi] & = & \cup_{a \in \mathcal{M}} \mathsf{D}[sa] \\ \text{furthermore} & sa & \in & \{P, Q(s(a\mathsf{F})), Rs\psi(a\mathsf{T})(a\mathsf{F}), \bot\} \\ \text{and} & \mathsf{D}[\psi] & = & \mathsf{D}[P] \cup \left(\cup_{c \in \mathcal{M}} \mathsf{D}[Q(sc)]\right) \cup \left(\cup_{b,c \in \mathcal{M}} \mathsf{D}[Rs\psi bc]\right) \end{array}$$

For all $a, b, c \in \mathcal{M}$ we will have that ψ, sa, P, $Q(sc)$ and $Rs\psi bc$ are characteristic maps or $\bot$. For all $a \in \mathcal{M}$, $\mathsf{D}[sa]$ will be *essentially σ-small* in the sense that there exists a set $A \subseteq \mathcal{M}$ of cardinality less than σ such that $\mathsf{D}[sa] = \{w \in \mathcal{M} \mid \exists u {\in} A \colon u \preceq w\}$. See Sections 10–12 for proofs.

Now, the definition of ψ and the auxiliary terms reads:

Definition 4.7.1.

(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) $P \equiv \lambda y.\, \mathsf{if}[\, y\,,\, \mathsf{T}\,,\, \bot\,]$
(f) $Q \equiv \lambda c.\, !c : \lambda y.\, \ddot{\forall} z.\, c(y(z/c))$
(g) $R \equiv \lambda f \theta b c.\, \theta c : R_1 f \theta b c : R_0 f \theta b c$
(h) $R_1 \equiv \lambda f \theta b c.\, \ddot{\forall} z.\, !(f(b(cz/\theta)))$
(i) $R_0 \equiv \lambda f \theta b c y.\, \mathsf{E} z.\, (\theta z : f(b(cz/\theta))y)$

Like in Section 3.11 let ψ_{Curry} be defined exactly like ψ except that all occurrences of Y and $\bot$ are replaced by $\mathsf{Y}_{\text{Curry}}$ and $\bot_{\text{Curry}}$, respectively. In MT_{def}, ψ_{Curry} takes the place of ψ.

4.8. Some properties of ψ

Note that $s = \mathsf{Y}S = S(\mathsf{Y}S) = Ss = \bar{S}s(\sqcup s) = \bar{S}s\psi$. Hence, in Definition 4.7.1(d-i) above one may think of f and θ as s and ψ, respectively.

Now define $b{::}c \equiv \lambda z.\, \mathsf{if}[\, z\,,\, b\,,\, c\,]$. For $b, c \in \mathcal{M}$ we have $(b{::}c)\mathsf{T} = b$ and $(b{::}c)\mathsf{F} = c$. Thus,

$$\begin{array}{lllll} s\mathsf{T} & = & \bar{S}s\psi\mathsf{T} & = & P \\ s(\mathsf{T}{::}c) & = & \bar{S}s\psi(\mathsf{T}{::}c) & = & Q(sc) \\ s(b{::}c) & = & \bar{S}s\psi(b{::}c) & = & Rs\psi bc \quad \text{if } b \notin \{\mathsf{T}, \bot\}. \end{array}$$

Accordingly, for $b, c, y \in \mathcal{M}$ we have

$$\begin{array}{lll} \text{If } Py = \mathsf{T} & \text{then } \psi y = \mathsf{T} & \\ \text{If } Q(sc)y = \mathsf{T} & \text{then } \psi y = \mathsf{T} & \\ \text{If } Rs\psi bcy = \mathsf{T} & \text{then } \psi y = \mathsf{T} & (b \notin \{\mathsf{T}, \bot\}) \end{array}$$

Thus, P, Q and R represent three ways to prove that y is wellfounded in the sense of MT.

Example 4.8.1. From $P\mathsf{T} = \mathsf{T}$ we have $\psi\mathsf{T} = \mathsf{T}$ so T is wellfounded. This may be seen as the base case. Actually, P just has two purposes: it forces T to be wellfounded and it initiates the recursive population of the universe of wellfounded maps.

From $Q(s\mathsf{T})(\lambda u.\, \mathsf{T}) = \ddot{\forall} z.\, s\mathsf{T}((\lambda u.\, \mathsf{T})(z/s\mathsf{T})) = \ddot{\forall} z.\, s\mathsf{T}\mathsf{T} = \ddot{\forall} z.\, \mathsf{T} = \mathsf{T}$ we have that $\lambda z.\, \mathsf{T}$ is wellfounded.

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.

4.9. Discussion of the definition of ψ

The definition of ψ replaces the $\mathsf{WF}[\phi]$ group of MT_0 (ten axioms and one inference rule). The definition of ψ in MT covers the following axioms of 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, the axiom of restriction and the axiom of foundation.

The ability of MT to model ZFC stems from several sources. First, the quantification axioms (cf. Section 4.5) reference ψ in a way which forces MT quantifiers to quantify over the universe $\mathsf{D}[\psi] = \{x{\in}\mathcal{M} \mid \psi x = \mathsf{T}\}$. Second, as shown in Example 4.8.1, recursive use of $s = \mathsf{Y}S$ populates $\mathsf{D}[\psi]$, putting a lower bound on the size of the universe. Third, the minimality of Y permits a kind of transfinite induction over $\mathsf{D}[\psi]$, 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 $\mathsf{T} = \ddot{\forall} x, y.\, (y \ddot{\in} \mathcal{P}(x) \ddot{\Leftrightarrow} \ddot{\forall} z.\, (z \ddot{\in} y \ddot{\Rightarrow} z \ddot{\in} x))$ and $\mathsf{T} = \ddot{\forall} x.\, \psi(\mathcal{P}(x))$ from which the power set axiom is easy to prove. Proving $\mathsf{T} = \ddot{\forall} x.\, \psi(\mathcal{P}(x))$ makes use of the second point above by using the fact 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 on x and thereby uses the fact that the universe is so small that all sets have powersets.

Note that the definition of ψ (Definition 4.7.1) is somewhat complicated: E appears explicitly in (i) and implicitly in (a) through the definition of $\sqcup$. Y appears explicitly in (b) and implicitly in (f), (h) and (i) through the recursive definition of z/c. Finally, ε appears implicitly in (f) and (h) through the definition of $\ddot{\forall}$.

As stated in Section 5.4, the complexity of the definition of ψ makes it infeasible to model $\mathsf{Quant}[\psi]$ directly. So we instead reuse some results from [4] and combine them with the investigation of ψ in Sections 10–12.

That the definition of ψ is somewhat complicated should not be too surprising, given that the power of ZFC is hidden in it.

5. Introduction to the consistency proof

We now give some more information on expansions and models for Map Theory.

Recall from Section 1.2 that the axiomatization MT of Map Theory is the main topic of the present paper and that MT to some extend obsoletes the previous axiomatization MT_0 [4, 9]. Also recall that the consistency proof of MT of the present paper draws heavily on the consistency proof of MT_0 in [4] so that we cannot ignore MT_0.

Finally recall that $\mathrm{MT}_{\mathrm{def}}$ is the minor variant of MT in which we replace Y by $\mathsf{Y}_{\mathrm{Curry}} \equiv \lambda f.\, (\lambda x.\, f(xx))(\lambda x.\, f(xx))$ and $\bot$ by $\bot_{\mathrm{Curry}} \equiv \mathsf{Y}_{\mathrm{Curry}} \lambda x.\, x =$

$(\lambda x.\,xx)(\lambda x.\,xx)$ (cf. Section 3.4). In Section 13 we prove the consistency of $\mathrm{MT_{def}}$ as a corollary of the consistency of MT.

5.1. Axioms and inference rules

The rules (i.e. axioms and inference rules) of MT and MT_0 fall in the following groups:

Elem The elementary rules of MT except Y and P1–3 (Section 4.1). These are also rules of MT_0.

Elem′ The rules Y and P1–3 of MT (Section 4.1).

Mono The rule of monotonicity of MT (Section 4.2)

Min The rule of minimality of MT (Section 4.2)

Ext The rule of extensionality of MT (Section 4.3)

Exist The axioms on E of MT (Section 4.4)

Quant[a] The quantification axioms of MT (Section 4.5) in which ψ is replaced by a. The Quant[ψ] and Quant[ϕ] rules are rules of MT and MT_0, respectively.

WF[ϕ] Ten axioms, each axiomatizing one specific closure property of Φ, plus one inference rule of transfinite induction (the ten axioms are presented as three "wellfoundedness" and seven "construction" axioms in [9]).

We have:

$$\begin{array}{lcl} \mathrm{MT} & = & \mathsf{Elem} + \mathsf{Elem}' + \mathsf{Mono} + \mathsf{Min} + \mathsf{Ext} + \mathsf{Exist} + \mathsf{Quant}[\psi] \\ \mathrm{MT}_0 & = & \mathsf{Elem} + \mathsf{Quant}[\phi] + \mathsf{WF}[\phi] \end{array}$$

Modulo an inessential change of the definition of $\mathcal{A} \to \mathcal{B}$, Quant[$\phi$] already appears in [4, Appendix C]. The four axioms of Quant[ϕ] are equivalent to the original set of 5+1 axioms where the five ones were stated in [9] and the sixth one, as pointed out by Thierry Vallée, was used but not stated in [9].

5.2. Domains, premodels and expansions

We introduce here informally the notions of *$\kappa\sigma$-expansions* (σ inaccessible, $\kappa > \sigma$, κ regular) and *κ-expansions* ($\kappa \geq \omega$, κ regular), among which live the *canonical* expansions. Certain canonical $\kappa\sigma$-expansions will be proved to be the models of MT we are looking for. In contrast, one main result of [4] is that all $\kappa\sigma$-expansions satisfy MT_0 (cf. Fact 5.4.1(a,b)). The notion of $\kappa\sigma$-expansions is built in the following stages:

Underlying set	$\mathcal{M}^0$	
κ-Scott domain	$\mathcal{M}^1 = (\mathcal{M}^0, \leq)$	Section 6.3
reflexive κ-domain	$\mathcal{M}^2 = (\mathcal{M}^1, A, \lambda)$	Section 7.1
κ-premodel	$\mathcal{M}^3$: An $\mathcal{M}^2$ which satisfies	Definition 7.4.4
κ-expansion	$\mathcal{M}^4 = (\mathcal{M}^3, \mathsf{T}, \bot, \mathsf{if}, \mathsf{Y}, \|, \mathsf{E})$	Section 7.5
$\kappa\sigma$-expansion	$\mathcal{M}^5 = (\mathcal{M}^4, \varepsilon, \phi)$	Section 7.5

The notions of κ-Scott domains, reflexive κ-domains, κ-premodels and κ-expansions are defined for all regular $\kappa \geq \omega$. For each κ-premodel $\mathcal{M}^3$ there is exactly one κ-expansion $\mathcal{M}^4 = (\mathcal{M}^3, \mathsf{T}, \bot, \mathsf{if}, \mathsf{Y}, \|, \mathsf{E})$; we shall refer to that uniquely defined $\mathcal{M}^4$ as *the κ-expansion of* $\mathcal{M}^3$.

The notion of a $\kappa\sigma$-expansion is defined for all inaccessible σ and all regular $\kappa > \sigma$. For each κ-premodel $\mathcal{M}^3$ and given $\sigma < \kappa$ there is exactly one $\kappa\sigma$-expansion (modulo the choice of the choice function underlying ε); as before we refer to it as *the $\kappa\sigma$-expansion of* $\mathcal{M}^3$.

From now on, $x \in \mathcal{M}^1 = (\mathcal{M}^0, \leq)$ means $x \in \mathcal{M}^0$ and likewise for $\mathcal{M}^2$ to $\mathcal{M}^5$. Furthermore, we drop the superscripts of $\mathcal{M}$ and let $\mathcal{M}$ denote any one of $\mathcal{M}^0$ to $\mathcal{M}^5$ depending on context.

5.3. The canonical expansions $\mathcal{M}_\kappa$ and $\mathcal{M}_{\kappa\sigma}$

In Section 8 we construct a *canonical κ-premodel* for each regular cardinal $\kappa \geq \omega$. For each regular $\kappa \geq \omega$ there are many κ-premodels but only one canonical one. We shall refer to the κ-expansion of the canonical κ-premodel as *the canonical κ-expansion* $\mathcal{M}_\kappa$ and likewise for the canonical $\kappa\sigma$-expansion $\mathcal{M}_\kappa$
$\mathcal{M}_{\kappa\sigma}$ (c.f. Definition 8.6.2). $\mathcal{M}_{\kappa\sigma}$

We use the word *canonical* for a number of reasons. First, that is the word we used in [4] so "canonical" is convenient when referring to "the canonical models of [4]". Second, for each choice of κ and σ there is only one of them (except for the choice of choice function used to model ε, cf. Fact 8.3.3). Third, the κ-premodel we call "canonical" is the one which "feels right", just like ω is the model of the natural numbers which "feels right". Besides, it is the simplest premodel one can produce within κ-Scott semantics (cf. the introduction of Section 8), and probably the only one suited to MT (in contrast to MT_0).Finally, the word "canonical" formally does not really mean anything and thus does not need justification (as opposed to more suggestive words like "minimal").

5.4. Satisfaction of axioms and inference rules

A $\kappa\sigma$-expansion interprets application A, abstraction λ and the constructs T, $\bot$, if, Y, $\|$, E, ε and ϕ. The construct ϕ is not needed for modelling MT and the constructs Y, $\|$ and E are not needed for modelling MT_0.

A κ-expansion does not define ε and ϕ and thus cannot satisfy MT or MT_0. In particular, κ-expansions cannot satisfy $\mathsf{Quant}[\psi]$, $\mathsf{Quant}[\phi]$ and $\mathsf{WF}[\phi]$ but can satisfy the other groups of rules.

We shall use $\mathcal{M} \models S$ to denote that $\mathcal{M}$ satisfies the rule or group of rules S. Now let κ be regular. We have:

Fact 5.4.1.
(a) $\mathcal{M} \models \mathsf{Elem} + \mathsf{Elem}' + \mathsf{Exist}$ if $\mathcal{M}$ is any κ-expansion where $\kappa \geq \omega$ (c.f. Theorem 7.5.2).
(b) $\mathcal{M} \models \mathsf{Quant}[\phi] + \mathsf{WF}[\phi]$ if $\mathcal{M}$ is any $\kappa\sigma$-expansion where $\kappa > \sigma$ and σ is inaccessible (c.f. [4] or Theorem 7.9.2)
(c) $\mathcal{M}_\kappa \models \mathsf{Mono} + \mathsf{Min} + \mathsf{Ext}$ if $\kappa \geq \omega$ (c.f. Section 7.6).
(d) $\mathcal{M}_{\kappa\sigma_o} \models \mathsf{Quant}[\psi]$ if $\kappa > \sigma_o$ and σ_o is the first inaccessible (c.f. Theorem 7.9.2 and the Definability Theorem (Theorem 10.1)).

In particular, $\mathcal{M}_{\kappa\sigma_o} \models \mathrm{MT}$ if σ_o is the first inaccessible and $\kappa > \sigma_o$, c.f. the Consistency Theorem (Theorem 2.2.1) which we restate and prove as Theorem 13.1. In contrast, $\mathcal{M}_{\kappa\sigma} \models \mathrm{MT}_0$ for any inaccessible σ and $\kappa > \sigma$.

The proof of (a) is easy, the proof of (b) is less easy (c.f. Section 9) and, as already mentioned, the proof of (d) is by far the most difficult.

Proving $\mathcal{M}_{\kappa\sigma_o} \models \mathsf{Quant}[\psi]$ directly is infeasible because the definition of ψ is complicated (cf. Section 4.9). Instead, we reuse $\mathcal{M}_{\kappa\sigma_o} \models \mathsf{Quant}[\phi]$ from [4], prove $\phi = \psi$ in Sections 10–12 and conclude $\mathcal{M}_{\kappa\sigma_o} \models \mathsf{Quant}[\psi]$ from that. Reusing $\mathcal{M}_{\kappa\sigma_o} \models \mathsf{Quant}[\phi]$ and several other theorems from [4] substantially simplifies and shortens the present paper.

5.5. Subjective and objective difficulty of axioms and inference rules

We now move on to consider the "difficulty" of the rules (i.e. axioms and inference rules) of Map Theory. "Difficulty" is a multi-dimensional and subjective notion. When looking at the rules it is natural to ask oneself the following questions:

- *Naturality.* Are the rules intuitive or "natural" in some sense, i.e. is there a natural or simple or motivated intuition behind?
- *Strength.* Do we need $\kappa > \sigma$ for an inaccessible σ for modelling them or is $\kappa \geq \omega$ enough?
- *Conceptual difficulty.* Do we need to introduce original and/or high level tools for modelling them?
- *Technical difficulty.* Do we need difficult computations for modelling them?

The Elem and Elem' rules are natural (if one is used to λ-calculus) and can be modelled at no cost (i.e. in any κ-premodel, $\kappa \geq \omega$).

The Exist rules are at first glance purely technical, but 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 rules of MT are natural in some sense, since they were designed from semantic and computational intuitions (cf. [9]), but this naturality may 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 but fortunately works for canonical ones (cf. Sections 7.6 and 9.3).

The Ext rule requires "familiarization" in the sense that it is unintelligible in itself and requires some explanations like those given in Section 4.3. But the intuition behind it is easy (if g and h behave the same when applied to arbitrary lists of arguments, then they are equal). Satisfying Ext is both conceptually and technically not so easy. Again, Ext reduces the class of possible models, but is satisfied in canonical ones.

Concerning the Quant[a] rules 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 a bit harder for MT (because they refer to the defined ψ which replaces the WF[ϕ] rules) and technically much harder (cf. Sections 10–12).

The WF[ϕ] rules belong to MT_0 and are treated in [4]. Some of them are difficult to satisfy and very difficult to explain.

	Elem Elem′	Mono Min	Ext	Exist	Quant[ψ]	Quant[ϕ]	WF[ϕ]
Naturality	Easy	Easy	f	Easy	q	Easy	D
Strength	$\kappa{\geq}\omega$	$\kappa{\geq}\omega$	$\kappa{\geq}\omega$	$\kappa{\geq}\omega$	$\kappa{>}\sigma$	$\kappa{>}\sigma$	$\kappa{>}\sigma$
Conceptual difficulty	Easy	c	c	Easy	D	d	d
Technical difficulty	Easy	c	c	Easy	D	d	d

c Less easy. Requires canonicity and some less easy developments
d Difficult
D Very difficult
f Requires familiarization
q Easy in themselves but the definition of ψ is complicated

5.6. Overview of the consistency proof

Section 6 presents κ-Scott semantics. Section 7 defines the notions of expansions and related structures and treats the satisfaction of Elem, Elem′, Exist and Quant[ϕ]. Section 7 also gives some initial results concerning the satisfaction of Mono, Min and Ext.

Section 8 recalls the construction of canonical models from [4] which allows Section 9 to finish the treatment of Mono, Min and Ext.

Sections 10–12 prove $\psi = \phi$ where Section 11 proves $\psi \preceq_{\mathcal{M}} \phi$, Section 12 proves $\phi \preceq_{\mathcal{M}} \psi$ and Section 10 presents material needed in both Section 11 and 12. Section 13 restates the Consistency Theorem and finilizes its proof.

6. The κ-Scott semantics

As promised in Section 5.6 we now introduce κ-Scott semantics. In particular, we define the notion of κ-Scott domains (c.f. Section 5.2) and related concepts. The treatment is similar to that of [4] but is repeated here for the sake of self-containedness (c.f. Section 5.6).

Models of Map Theory are, in particular, models of λ-calculus (i.e. pure untyped λ-calculus) since Map Theory extends λ-calculus.

As is well-known, models of λ-calculus are exactly the reflexive objects of the Cartesian closed categories (ccc) with enough points (see e.g. [2]). The purpose of this section is to *describe* the ccc we use for modelling Map Theory, while the reflexive objects of the ccc will be introduced in Section 7.1.

Scott built the first non-syntactic model of λ-calculus within the ccc of complete lattices (as objects) with continuous functions (as morphisms), and came quickly to the more abundant ccc of Scott domains and continuous functions, usually called Scott semantics for short.

Scott semantics itself is too weak for modelling powerful foundational extensions of λ-calculus but, as explained in Section 6.2, it is very easy (as already Scott was aware) to develop, for each regular cardinal κ, a *κ-Scott semantics*, which has the required ability (for κ large enough). Usual Scott semantics (case $\kappa = \omega$) is sufficient for dealing with the computational aspects of Map Theory (c.f. Appendix A).

Sections 6.1–6.5 recall the basics of κ-Scott semantics, $\kappa \geq \omega$, mentioning why it is enough and convenient to consider only regular κ. Section 6.6 introduces a new notion of *κ-step functions*, which happens to be a very convenient tool (e.g. when modelling epsilon and in Section 11).

6.1. Notation

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

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. $G^{<\omega}$ $\langle\,\rangle$

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

Let $f{:}\,G{\rightarrow}H$ denote that f is a total function from G to H. $f{:}\,G{\rightarrow}H$

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. $\uparrow S$ $\downarrow S$

We say that a set G is *κ-small* if G has cardinality strictly 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. κ-small $\mathcal{P}(G)$ $\mathcal{P}_{<\kappa}(G)$

As usual, the *cofinality* $\mathrm{cf}(\alpha)$ of an ordinal α is the smallest ordinal β such that there is a $g{:}\,\beta{\rightarrow}\alpha$ for which $\alpha = \bigcup_{\gamma\in\beta} g(\gamma)$. The cofinality $\mathrm{cf}(\alpha)$ is always a cardinal. An ordinal α is a *regular cardinal* if $\mathrm{cf}(\alpha) = \alpha \geq \omega$. An ordinal σ is *inaccessible* (i.e. *strongly inaccessible*) if σ is regular, $\sigma > \omega$ and $\mathcal{P}(\gamma)$ is σ-small for all $\gamma < \sigma$. cofinality $\mathrm{cf}(\alpha)$ regular inaccessible

Note that there are many regular cardinals since e.g. all infinite successor cardinals are regular. In contrast, the existence of an inaccessible ordinal is independent of ZFC.

A key consequence of regularity is that κ-small unions of κ-small sets are κ-small for regular κ (and of course likewise for inaccessible σ).

6.2. 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. The pertinent κ-Scott notions merely depend on the cofinality of κ. Thus, as a convenience and without loss of generality, we only consider regular κ. As κ grows (κ regular) 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 [4]. For the reader familiar with Scott domain theory, passing from Scott to κ-Scott is straightforward and just amounts (provided κ is regular) to changing everywhere "finite" by "κ-small". We recall some key definitions and results in the following.

κ-Scott semantics was first used around 1987-89 in [7, 8] and was used independently in [4], 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.

From now on κ is regular.

6.3. κ-Scott domains

p.o. — κ-directed — κ-Scott domain

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, if 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 a κ-Scott domain 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 $\mathcal{M}_{\kappa\sigma}$ will not be a full power set, but will still be a set of sets, ordered by inclusion.

compact — prime — κ-algebraic — prime-algebraic

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 $v \in S$. The p.o. $\mathcal{D}$ 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. In κ-Scott domains, prime elements are κ-compact. Another key property, which is a straightforward generalization of the ω-case, is that (existing) sups of κ-small sets of κ-compact elements are themselves κ-compact. A κ-Scott domain is *prime-algebraic* if each element of $\mathcal{D}$ is the sup of the primes below it.

$\mathcal{D}_c$

Definition 6.3.1. *$\mathcal{D}_c$ is the set of compact elements of the κ-Scott domain $\mathcal{D}$.*

Both $(\mathcal{P}(D), \subseteq)$ and the domain $\mathcal{D}_{\kappa\sigma}$ underlying $\mathcal{M}_{\kappa\sigma}$ 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. The compact elements of $\mathcal{D}_{\kappa\sigma}$ are downward closures of adequate κ-small subsets of D, while primes are downward closures of singletons.

6.4. 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.

6.5. κ-open sets

$G \subseteq \mathcal{D}$ is κ-open if $G = \uparrow K$ for some set $K \subseteq \mathcal{D}_c$. 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 straightforward to check but crucial to note that the intersection of a κ-small family of κ-open sets is still κ-open.

The set $G \subseteq \mathcal{D}$ 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 $G = \uparrow V$ for some κ-small $V \subseteq \mathcal{D}_c$.

6.6. κ-step functions

We now introduce a notion of *κ-step functions*; such functions are particularly easy to prove to be κ-continuous and they are natural and convenient tools for our purposes. In particular, the interpretation of ε recalled from [4] in Section 7.4 is a κ-step function, and several families of κ-step functions will be used in Sections 10–12.

Definition 6.6.1. *For all $g\colon \mathcal{D} \rightarrow \mathcal{D}$ the domain* $\mathrm{Dom}[g]$ *is defined by* $\mathrm{Dom}[g] \equiv \{x \in \mathcal{D} \mid g(x) \neq \bot\}$.

Definition 6.6.2. $g\colon \mathcal{D} \rightarrow \mathcal{D}$ is a κ-step function if:
(a) $\mathrm{Dom}[g]$ is κ-open.
(b) $x \preceq_{\mathcal{M}} y \Rightarrow g(x) = \bot \vee g(x) = g(y)$.

Lemma 6.6.3. *Every κ-step function is κ-continuous.*

Proof of 6.6.3 Monotonicity is obvious. Now let $S \subseteq \mathcal{D}$ be κ-directed. We shall prove $g(\sup(S)) = \sup(\{g(x) \mid x \in S\})$. Because of Definition 6.6.2(b) this is equivalent to proving $\sup(S) \in \mathrm{Dom}[g] \Leftrightarrow \exists x {\in} S\colon x \in \mathrm{Dom}[g]$ which is obvious since $\mathrm{Dom}[g]$ is κ-open.

6.7. Conclusion

As promised in Section 5.6 we have now introduced κ-Scott semantics. Furthermore, we have introduced the notion of κ-Scott domains. As mentioned in Section 5.2, the construction of an adequate κ-Scott domain constitutes the first step in constructing a model of MT. We have also proved the small Lemma 6.6.3, but it is the presentation of κ-Scott semantics which was the main purpose of Section 6.

7. Premodels and expansions.

Having κ-Scott domains from Section 6 and following the plan laid out in Section 5.6, we now proceed with defining the rest of the concepts listed in Section 5.2 leading up to the definition of $\kappa\sigma$-expansions (Definition 7.9.1) which we eventually use to model MT.

In this section $\kappa \geq \omega$ is regular.

7.1. Reflexive κ-domains as models of pure λ-calculus

A *Reflexive κ-domain* (i.e. a reflexive object of the κ-Scott semantics) is a triple $(\mathcal{D}, A, \lambda)$ where $\mathcal{D}$ is a κ-Scott domain and $A\colon \mathcal{D} \to_\kappa [\mathcal{D} \to_\kappa \mathcal{D}]$ and $\lambda\colon [\mathcal{D} \to_\kappa \mathcal{D}] \to_\kappa \mathcal{D}$ 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, A2 and Rename when we use A and λ to interpret the pure λ-terms, in the standard way (see e.g. [2]).

Most of the time $A(u)(v)$ will be abbreviated as uv which we make left-associative so that uvw means $(uv)w$. 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 $\mathcal{D}$: for any such f there is an element $v \in \mathcal{D}$ such that $f(u_1, \ldots, u_n) = vu_1 \cdots u_n$ for all $u_1, \ldots, u_n \in \mathcal{D}$. In the case $n = 1$ we can take $v = \lambda(f)$.

7.2. 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

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

$\mathcal{Y}_{\mathrm{Tarski}}$

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

Lemma 7.2.1. *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 7.2.1 Easy and classical.

Lemma 7.2.2.

(a) If $\kappa = \omega$ then $\mathcal{Y}_{\mathrm{Tarski}} \in [\mathcal{D} \to_\kappa \mathcal{D}] \to_\kappa \mathcal{D}$ is total.

(b) If f has a fixed point then $(f^\alpha)_{\alpha \leq \kappa}$ and $\mathcal{Y}_{\mathrm{Tarski}}(f)$ are defined.

(c) If there are A, λ making $(\mathcal{D}, A, \lambda)$ a reflexive κ-domain then $\mathcal{Y}_{\text{Tarski}}$ is total and κ-continuous.

Proof of 7.2.2

(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 $\mathsf{Y}_{\text{Curry}}\lambda(f)$ is a fixed point where $\mathsf{Y}_{\text{Curry}} \equiv \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))$ exists in all models of λ-calculus and thus exists in all reflexive κ-domains. Continuity can be proved by a rather standard proof (which can be found e.g. in [16]).

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

Definition 7.2.3. $\mathsf{Y}_{\text{Tarski}} \equiv \lambda(\mathcal{Y}_{\text{Tarski}} \circ A)$ $\mathsf{Y}_{\text{Tarski}}$

Above, $\circ$ is composition.

Corollary 7.2.4.

(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 7.2.4 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 point operator.

Hence, $\mathsf{Y}_{\text{Tarski}}$ would 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 (c.f. Theorem 7.6.3 which is proved as Corollary 9.3.2):

7.3. The domain equation Eq_κ

Let $\bot'$ and T' be arbitrary, distinct constants which are not functions. Given a κ-Scott domain $\mathcal{D}'$ which does not contain T' and $\bot'$ we denote by $\mathcal{D}' \oplus_{\bot'} \{\mathsf{T}'\}$ the κ-Scott domain obtained by adding to $\mathcal{D}'$ the element T' which we decide to be incomparable to all the elements of $\mathcal{D}'$, and the bottom element $\bot'$ which we decide to be below T' and all the elements of $\mathcal{D}'$.

Definition 7.3.1. *Eq_κ is the domain equation $\mathcal{D} \simeq [\mathcal{D} {\to_\kappa} \mathcal{D}] \oplus_{\bot'} \{\mathsf{T}'\}$.*

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 κ-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 canonical solution, which will be rebuilt in Section 8.

7.4. κ-Premodels

Given a solution $\mathcal{D}$ of Eq_κ and an order isomorphism λ from $[\mathcal{D} \to_\kappa \mathcal{D}] \oplus_{\perp'} \{\mathsf{T}'\}$ to $\mathcal{D}$, define:

Definition 7.4.1.

$$\begin{array}{lcl} \mathsf{T} & \equiv & \lambda(\mathsf{T}') \\ \perp & \equiv & \lambda(\perp') \\ \mathcal{F} & \equiv & \{\lambda(f) \mid f \in [\mathcal{D} \to_\kappa \mathcal{D}]\} \end{array}$$

We clearly have:

Fact 7.4.2.
(a) $\perp$ is the bottom element of $\mathcal{D}$.
(b) $\mathcal{F} = \mathcal{D} \setminus \{\perp, \mathsf{T}\}$.
(c) $\mathcal{F}$ and $\{\mathsf{T}\}$ are disjoint, open subsets of $\mathcal{D}$.
(d) $\lambda x. \perp$ (i.e. $\lambda(x \mapsto \perp)$) is the bottom element of $\mathcal{F}$ because $\mathcal{F}$ is the isomorphic image of $[\mathcal{D} \to_\kappa \mathcal{D}]$ under λ.

Now let A be a morphism from $\mathcal{D}$ to $[\mathcal{D} \to_\kappa \mathcal{D}]$. Since uv abbreviates $A(u)(v)$ we have that $\mathsf{T}u$ abbreviates $A(\mathsf{T})(u)$ and likewise for $\perp u$ and $\mathbf{1}u$. The following theorem is easy to prove and the details can be found in [4, Section 3.1]:

Theorem 7.4.3. *If $\mathcal{D}$ is a solution of Eq_κ and if λ is an order isomorphism from $[\mathcal{D} \to_\kappa \mathcal{D}] \oplus_{\perp'} \{\mathsf{T}'\}$ to $\mathcal{D}$ then there exists an A such that $(\mathcal{D}, A, \lambda)$ is a reflexive κ-domain satisfying:*
(a) $\mathsf{T}u = \mathsf{T}$ and $\perp u = \perp$ for all $u \in \mathcal{D}$.
(b) $\mathcal{F} = \{u \in \mathcal{D} \mid u = \mathbf{1}u\} = \mathcal{D} \setminus \{\perp, \mathsf{T}\}$.
(c) $\mathcal{F}$ and $\{\mathsf{T}\}$ are disjoint κ-open subsets of $\mathcal{D}$.

Note that in (b) above, only the last equation (which repeats Fact 7.4.2(b)) uses the assumption that $\mathcal{D}$ is a solution to Eq_κ; the first equation is classic.

The interpretation of any term of form $\lambda x. \mathcal{A}$ is in $\mathcal{F}$. For all $u, v \in \mathcal{F}$ we have $u \preceq_{\mathcal{M}} v$ iff $ux \preceq_{\mathcal{M}} vx$ for all x in $\mathcal{D}$.

Conversely, any reflexive κ-domain $(\mathcal{D}, A, \lambda)$ satisfying (a), (b) and (c) of the above theorem can easily be turned into a solution of Eq_κ.

Definition 7.4.4. A *κ-premodel* $\mathcal{M}$ is a reflexive κ-domain $(\mathcal{D}, A, \lambda)$ for which $\mathcal{D}$ satisfies Eq_κ and which satisfies the three conditions of Theorem 7.4.3.

7.5. κ-expansions

We now define the notion of a κ-expansion. Later, in Definition 8.6.2, and as promised in Section 5.3, we define the *canonical* κ-expansion $\mathcal{M}_\kappa$ to be the κ-expansion of the *canonical* κ-premodel.

Definition 7.5.1. *The κ-expansion of a κ-premodel $\mathcal{M} = (\mathcal{D}, A, \lambda)$ is the tuple $(\mathcal{M}, \mathsf{T}, \perp, \mathsf{if}, \mathsf{Y}, \|, \mathsf{E})$ where T and $\perp$ are defined in Definition 7.4.1 and if, Y, $\|$ and E are defined below.*

Define $\mathsf{If}(u,v,w) = v$ if $u = \mathsf{T}$, w if $u \in \mathcal{F}$ and $\bot$ if $u = \bot$. Now If is clearly κ-continuous. Let if be the unique element of $\mathcal{M}$ such that $\mathsf{if}uvw = \mathsf{If}(u,v,w)$ for all $u,v,w \in \mathcal{M}$.

Let Y be $\mathsf{Y}_{\mathrm{Tarski}}$ as defined in Section 7.2.

Define $\mathsf{Paror}(u,v) = \mathsf{T}$ if u or v is T, $\lambda x.\mathsf{T}$ if $u,v \in \mathcal{F}$ and $\bot$ otherwise. Define $\mathsf{Ex}(u) = \mathsf{T}$ if $uv = \mathsf{T}$ for some $v \in \mathcal{M}$ and $\bot$ otherwise. Now Paror and Ex are clearly κ-continuous. Let $\|$ and E be the unique elements of $\mathcal{M}$ such that $\| uv = \mathsf{Paror}(u,v)$ and $\mathsf{E}u = \mathsf{Ex}(u)$ for all $u,v \in \mathcal{M}$.

Note that interpretation of T, $\bot$, if, $\|$ and E can only be done as above and that choosing $\mathsf{Y}_{\mathrm{Tarski}}$ for Y is the most natural (and possibly unique) way to interpret Y.

Recall Elem and Elem' from Section 4.1 and Exist from Section 4.4. We have:

Theorem 7.5.2. $\mathcal{M} \models \mathsf{Elem}+\mathsf{Elem}'+\mathsf{Exist}$ *if* $\mathcal{M}$ *is a* κ*-expansion where* $\kappa \geq \omega$.

Proof. The κ-premodel underlying the κ-expansion $\mathcal{M}$ satisfies rules Trans, Sub, Gen, A2 and Rename since it is a reflexive object of a Cartesian closed category. The κ-premodel satisfies A1 and A3 and rule QND because κ-premodels by definition satisfy the conditions of Theorem 7.4.3. Axioms I1, I2 and I3 follow from the definition of if. Axiom Y follows from Corollary 7.2.4(a). Axioms P1, P2 and P3 follow from the definition of $\|$; and the four axioms on E follow from the definition of E. □

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

We are not yet in a position to prove the monotonicity and minimality rules Mono and Min (cf. Section 4.2) and the extensionality rule Ext (cf. Section 4.3), but we have the following:

Theorem 7.6.1. *If* $\kappa \geq \omega$ *and if* $\mathcal{M}$ *is a* κ*-expansion, then* $\mathcal{M}$ *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 7.2.4. □

In Section 9, we prove Mono and Min in the canonical κ-expansion $\mathcal{M}_\kappa$ ($\kappa \geq \omega$) by proving that $\preceq_{\mathcal{M}}$ and $\preceq$ coincide in such models. Thus, no inaccessible is needed, but canonicity is crucial. Modelling of Mono, Min and Ext in Section 9 proceeds thus:

Theorem 7.6.2 (Section 9.2). $\mathcal{M}_\kappa \models \mathsf{Ext}$ *if* $\kappa \geq \omega$.

Theorem 7.6.3 (Section 9.3). $\mathcal{M}_\kappa$ *satisfies that the model order* $\preceq_{\mathcal{M}}$ *coincides with the syntactic order* $\preceq$ *if* $\kappa \geq \omega$.

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

Corollary 7.6.4. $\mathcal{M}_\kappa \models \mathsf{Mono} + \mathsf{Min}$ *if* $\kappa \geq \omega$.

Theorem 7.6.5 (Section 9.5). *$\mathcal{M}_\kappa$ satisfies $\mathsf{Y}_{\text{Tarski}} = \mathsf{Y}_{\text{Curry}}$ if $\kappa \geq \omega$.*

Corollary 7.6.6. *$\mathcal{M}_\kappa$ satisfies $\mathsf{Y}f = \mathsf{Y}_{\text{Curry}}f$ and $\bot = \bot_{\text{Curry}}$ if $\kappa \geq \omega$.*

Corollary 7.6.7. *$\mathcal{M}_\kappa$ satisfies Mono and Min of MT_{def} if $\kappa \geq \omega$.*

7.7. Quantifier axioms

We now turn to the quantifier axioms (cf. Section 4.5). The $\mathsf{Quant}[\phi]$ axioms of MT_0 were easy to model (the difficulty was carried by some of the ϕ-axioms). For MT, the complexity of the term ψ, whose definition involves ε and Y, makes $\mathsf{Quant}[\psi]$ very difficult to model. Our trick will be to use that $\mathsf{Quant}[\phi]$ holds and to prove (in Sections 10–12) that ψ and ϕ coincide in all $\kappa\sigma$-expansions, provided σ is the first inaccessible and provided ε and ϕ are defined as in Definition 7.9.1.

Recall $\mathsf{D}[w] \equiv \{u{\in}\mathcal{M} \mid wu = \mathsf{T}\}$ (cf. Definition 4.6.1) and define:

Definition 7.7.1. *For all $U \subseteq \mathcal{M}$ and $w \in \mathcal{M}$ we let:*
(a) $wU \equiv \{wu \mid u \in U\}$
(b) $\chi_U\colon \mathcal{M} \to \mathcal{M}$ is defined by $\chi_U(x) = \mathsf{T}$ if $x \in U$ and $\chi_U(x) = \bot$ otherwise.

Remark 7.7.2.
(a) $\mathsf{D}[w]$ is a κ-open set for all $w \in \mathcal{M}$
(b) χ_U is κ-continuous iff U is κ-open

Theorem 7.7.3 ([4]). *Let $\mathcal{M}$ be a κ-expansion ($\kappa \geq \omega$), and let $\Phi \subseteq \mathcal{M}$ be such that $\Phi = \uparrow\Psi$ for some κ-small set Ψ such that $\mathsf{T} \in \Psi$ and $\bot \notin \Psi$. Then there is an $\varepsilon \in \mathcal{M}$ such that, when the syntactical ε is interpreted by this ε, $\mathcal{M}$ satisfies $\mathsf{Quant}[\chi_\Phi]$.*

Proof. We first recall the proof in [4, Section 4.1]: let ξ be a choice function on Φ, i.e. a function $\xi\colon \mathcal{P}(\Phi) \to \Phi$ such that $\xi(V) \in V$ for all non-empty $V \subseteq \Phi$. Let $e\colon \mathcal{M} \to \Phi \cup \{\bot\}$ be defined by: $e(u) = \bot$ if $\bot \in u\Phi$, $e(u) = \mathsf{T}$ if $u\Phi \subseteq \mathcal{F}$ and $e(u) = \xi(\{x{\in}\Phi \mid ux = \mathsf{T}\})$ otherwise. Then e is a κ-step function: It is indeed clear that $u \preceq_{\mathcal{M}} v \Rightarrow e(u) = \bot \vee e(u) = e(v)$. It remains to prove that $\text{Dom}[e] \equiv \{x{\in}\mathcal{M} \mid e(x) \neq \bot\}$ is κ-open. Now, $\text{Dom}[e] = \{u{\in}\mathcal{M} \mid \Phi \subseteq \text{Dom}[u]\} = \{u{\in}\mathcal{M} \mid \Psi \subseteq \text{Dom}[u]\} = \cap_{x\in\Psi}\{u{\in}\mathcal{M} \mid ux \neq \bot\}$. Thus, $\text{Dom}[e]$ is the intersection of a κ-small family of κ-open sets, and hence is κ-open. Thus, e is a κ-step function and, hence, κ-continuous. Now $\varepsilon \equiv \lambda(e)$ has the required properties by [4, Theorem 4.3.1]. □

7.8. The definition of Φ

We suppose now that $\sigma < \kappa$ is inaccessible. We define *σ-small* sets and *essentially σ-small* sets as it was done for κ (cf. Section 6.1 and 6.5), 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 7.8.1. *[4] For any $U, V, H \subseteq \mathcal{M}$ where H is open define:*

(a) $\mathcal{O}_{<\sigma}(H)$ *and* $\mathcal{O}_{<\kappa}(H)$ *are the sets of all essentially* σ- *and* κ*-small open subsets of* H*, respectively*
(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) $\mathcal{F}_\sigma(H) \equiv \{\mathsf{T}\} \cup \bigcup\{G^\circ{\to}G \mid G \in \mathcal{O}_{<\sigma}(H)\}$

In the present paper we define Φ thus:

Definition 7.8.2. $\Phi \subseteq \mathcal{M}$ *is the smallest set such that* Φ

$$\mathsf{T} \in \Phi \quad \text{and} \quad G \in \mathcal{O}_{<\sigma}(\Phi) \Rightarrow G^\circ{\to}G \subseteq \Phi$$

The elements of Φ *are, by definition, the wellfounded maps.*

The Φ defined above equals the Φ defined in [4] (c.f. Lemma 10.4.2). Furthermore, Φ satisfies $\Phi = \mathcal{F}_\sigma(\Phi)$ and is the smallest solution to this equation. Also, Φ satisfies the Generic Closure Property (GCP) of [4] which says $\Phi = \cup\{G^\circ{\to}\Phi \mid G \in \mathcal{O}_{<\sigma}(\Phi)\}$ (c.f. [4, Theorem 7.1.1]).

Another important property is $\Phi \subseteq \Phi^\circ$ which is called the Strong Induction Property (SIP) in [4] and which is stated here as Lemma 10.4.3(f). Furthermore, $\Phi \in \mathcal{O}_{<\kappa}(\mathcal{M})$ according to Lemma 10.4.6 or [4]. In fact it is proved in [4] that Φ has essential cardinality exactly σ in the sense that Φ is not essentially σ-small and, furthermore, $\Phi = {\uparrow}\Psi$ where $\Psi \subseteq \mathcal{M}_c$ is defined in [4] and where Ψ has cardinality σ.

7.9. $\kappa\sigma$*-Expansions*

Definition 7.9.1. *Given a* κ*-expansion* $\mathcal{M}$ *define the* $\kappa\sigma$*-expansion* $(\mathcal{M}, \varepsilon, \phi)$ *of* $\mathcal{M}$ *as follows:* ε *is defined as in the proof of Theorem 7.7.3 and* $\phi = \lambda(\chi_\Phi)$*.*

Later, in Definition 8.6.2, and as promised in Section 5.3, we define the *canonical* $\kappa\sigma$-expansion $\mathcal{M}_{\kappa\sigma}$ to be the $\kappa\sigma$-expansion of the *canonical* κ-premodel.

Theorem 7.9.2 ([4]). $\mathcal{M} \models \mathsf{Quant}[\phi] + \mathsf{WF}[\phi]$ *if* $\mathcal{M}$ *is a* $\kappa\sigma$*-expansion where* $\kappa > \sigma$ *and* σ *is inaccessible.*

In particular, we have $\mathcal{M} \models \mathrm{MT}_0$ proving the consistency of MT_0. We now return to models of MT.

To model the quantification axioms of MT it is enough to show that, if σ is the first inaccessible ordinal, then $\psi = \phi$. The proof of this result, called the "Definability Theorem" (Theorem 10.1) occupies Section 10–12 and is, by far, the most difficult proof in the present paper. The proof of the Definability Theorem is split into two parts, called the Upper Bound Theorem (UBT) and the Lower Bound Theorem (LBT).

UBT says $\psi \preceq_{\mathcal{M}} \phi$. It puts an upper bound on ψ and is proved in Section 11. The proof uses the existence of an inaccessible σ (actually, the mere definitions of Φ and ϕ need 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 $\phi \preceq_{\mathcal{M}} \psi$. It puts a lower bound on ψ and is proved in Section 12. 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).

We interpret $\psi \preceq_{\mathcal{M}} \phi$ as an upper bound of ψ rather than e.g. a lower bound on ϕ since ϕ was given already in [4] whereas ψ is the quantity being investigated in the present paper.

We can now outline how $\mathsf{Quant}[\psi]$ is going to be modelled:

Theorem 7.9.3. *(Outline) If σ_o is the first inaccessible and $\kappa > \sigma_o$, then:*

(a) Any $\kappa\sigma_o$-expansion satisfies $\mathsf{Quant}[\psi]$.

(b) The canonical $\kappa\sigma_o$-expansion $\mathcal{M}_{\kappa\sigma_o}$ satisfies $\mathsf{Quant}[\psi_{\mathrm{Curry}}]$.

Proof of 7.9.3 (Outline) Let $\mathcal{M}$ be a $\kappa\sigma_o$-expansion. From Theorem 7.9.2 we have $\mathcal{M} \models \mathsf{Quant}[\phi]$. From the Definability Theorem (Theorem 10.1) we have $\phi = \psi$. Thus $\mathcal{M} \models \mathsf{Quant}[\psi]$. Then (b) follows from Corollary 7.6.6.

As already noticed in the introduction (Section 1.4), the minor variant $\mathrm{MT}_{\mathrm{def}}$ of MT which is presented in Section 3.4 is a priori more difficult to model than MT. Fortunately, $\mathcal{M}_{\kappa\sigma_o}$ models $\mathsf{Quant}[\psi_{\mathrm{Curry}}]$ as noted above and, more generally, models all of $\mathrm{MT}_{\mathrm{def}}$.

7.10. Conclusion

We have now defined the concepts listed in Section 5.2. We have also proved some theorems like Theorem 7.5.2 which says that all κ-expansions satisfy the Elem, Elem' and Exist groups of axioms and inference rules (c.f. Section 5.1). We have also recalled from [4] that $\kappa\sigma$-expansions satisfy $\mathsf{Quant}[\phi]$ (and in fact all of MT_0). In Section 8 we prove that there exist κ- and $\kappa\sigma$-expansions. We use all that in the proof of the Consistency Theorem in Section 13. However, the main purpose of Section 7 was to define the notion of $\kappa\sigma$-expansions.

8. Building the canonical κ-premodel

As promised in Section 5.6 we now construct the canonical κ-premodel (c.f. Section 5.3). The treatment is similar to that of [4, Section 8] but is repeated here for the sake of self-containedness (c.f. Section 1.5).

Constructing the canonical κ-premodel has two purposes. First, the construction proves that κ-premodels and, hence, κ- and $\kappa\sigma$-expansions exist. Second, some axioms and inference rules of MT and $\mathrm{MT}_{\mathrm{def}}$ do not hold in all $\kappa\sigma$-expansions but do hold in canonical ones. Canonicity is needed for the Definability Theorem (Theorem 10.1) and for satisfying Mono, Min and Ext. For $\mathrm{MT}_{\mathrm{def}}$, canonicity is furthermore needed for satisfying $\mathsf{Quant}[\psi_{\mathrm{Curry}}]$.

In the following, $\kappa \geq \omega$ can be any regular cardinal and no inaccessible σ is needed.

A classical method for building domains or solving recursive domain equations in Scott's semantics (or its variants) is to look for webbed domains whose web satisfies a "derived" but more feasible equation (cf. e.g. [3]). That is what we will do here for building the canonical κ-premodel. We will indeed replace the domain equation Eq_κ of Definition 7.3.1 by the equation Eq'_κ of Definition 8.3.1. We will note (Fact 8.3.3) that the simplest solution of Eq'_κ is in fact its unique solution. This is of no use in the consistency proof, but can be seen as a further argument in favor of the word "canonical".

Here the web is a *preordered coherent space (pcs)* $\mathbf{P} = (P, \leq, \frown)$ where $\leq$ and $\frown$ are a preorder and a reflexive, symmetric relation on P, respectively, and where we refer to $\frown$ as a *coherence relation*. The terminology of "webbed model" was introduced in [3] and preordered coherent spaces (pcs's) are defined in Section 8.1.

The canonical κ-premodel $((\mathcal{M}, \preceq_\mathcal{M}), A, \lambda)$ and the web $(P, \leq, \frown)$ from which it is built satisfy that $(\mathcal{M}_p \setminus \{\perp\}, \preceq_\mathcal{M})$ is isomorphic to $(P, \leq)/(\leq \cap \geq)$ where $\mathcal{M}_p$ is the set of prime elements of the κ-premodel. Furthermore, $a \frown b$ iff the corresponding elements of $\mathcal{M}_p$ have an upper bound in $\mathcal{M}$.

The notion of pcs's generalizes the notion of preordered sets as well as Girard's definition of coherence spaces, both of which are well known to be relevant for building mathematical models of λ-calculus.

8.1. Preordered coherent spaces (pcs's)

A *pcs-structure* (or *structure* for short) is a tuple $\mathbf{D} = \langle D, \leq, \frown \rangle$ for which $\leq$ and $\frown$ are binary relations on D.

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

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

The compatibility requirement above may be motivated thus: if x and y have an upper bound (i.e. $\exists z {\in} D \colon x \leq z \wedge y \leq z$) then they are coherent (i.e. $x \frown y$). The opposite is not true: even if $x \frown y$ then x and y need not have an upper bound. However, $x \frown y$ denotes that x and y are *intended* to have an upper bound. Recall that pcs's are used for constructing κ-Scott domains. The coherence relation $x \frown y$ is used to record at an early stage of a construction that x and y are going to have an upper bound at a later stage of the construction.

If $x \leq x'$ and $y \leq y'$ and if z is an upper bound of x' and y' then z is also an upper bound of x and y. Compatibility expresses the reasonable requirement that if $x \leq x'$ and $y \leq y'$ and if x' and y' are intended to have an upper bound then x and y are also intended to have an upper bound.

From now on, $\mathbf{D} = \langle D, \leq, \frown \rangle$ and $\mathbf{D}' = \langle D', \leq', \frown' \rangle$ denote structures. We say that $\mathbf{D}$ is a *substructure* of $\mathbf{D}'$, written $\mathbf{D} \sqsubseteq \mathbf{D}'$, if the following hold:

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

A set S of structures is a *chain* if $\forall \mathbf{D}, \mathbf{D}' {\in} S {:}\, \mathbf{D} \sqsubseteq \mathbf{D}' \vee \mathbf{D}' \sqsubseteq \mathbf{D}$. Now for all structures $\mathbf{D} = \langle D, \leq, \frown \rangle$, all $u, v \subseteq D$ and all $p \in D$ define

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

Intuitively, $\mathrm{Coh}_{\mathbf{D}} u$ states that the set u is coherent, i.e. it is intended to have an upper bound. We have that u is coherent iff ${\downarrow_{\mathbf{D}}} u$ is coherent. Thus $\mathcal{I}(\mathbf{D})$ also denotes the set of coherent, initial segments of $\mathbf{D}$.

Fact 8.1.1. *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 Sections 8.2–8.3 is to define a pcs $\mathbf{P}$ such that $(\mathcal{I}(\mathbf{P}), \subseteq)$ satisfies Eq_κ.

8.2. Pcs generators

Fact 8.2.1. *Let* $\mathrm{U}(t) \equiv \langle \{t\}, \{t\} \times \{t\}, \{t\} \times \{t\} \rangle$*. Now* $\mathrm{U}(t)$ *is a pcs for all objects* t *(of ZFC).*

Fact 8.2.2. *Let* $\mathbf{D}_f \equiv \langle D \cup \{f\}, \leq'', \frown'' \rangle$ *where* $x \leq'' y \Leftrightarrow x = f \vee x \leq y$ *and* $x \frown'' y \Leftrightarrow x = f \vee y = f \vee x \frown y$*. If* $\mathbf{D}$ *is a pcs and* $f \notin D$ *then* $\mathbf{D}_f$ *is a pcs.*

Fact 8.2.3. *For all chains* S *of pcs's* $\mathbf{D} = \langle D_{\mathbf{D}}, \leq_{\mathbf{D}}, \frown_{\mathbf{D}} \rangle$ *let* $\cup S \equiv \langle \cup_{\mathbf{D} \in S} D_{\mathbf{D}}, \cup_{\mathbf{D} \in S} \leq_{\mathbf{D}}, \cup_{\mathbf{D} \in S} \frown_{\mathbf{D}} \rangle$*. If* S *is a chain of pcs's then* $\cup S$ *is a pcs.*

Fact 8.2.4. *Let* $\mathbf{D} \oplus \mathbf{D}' \equiv \langle D \cup D', \leq \cup \leq', \frown \cup \frown' \rangle$*. 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 8.2.5. *Let* $\mathbf{D} {\to} \mathbf{D}' \equiv \langle D \times D', \leq'', \frown'' \rangle$ *where* $(x, x') \leq'' (y, y') \Leftrightarrow y \leq x \wedge x' \leq y'$*, and* $(x, x') \frown'' (y, y') \Leftrightarrow x \not\frown y \vee x' \frown' y'$*. If* $\mathbf{D}$ *and* $\mathbf{D}'$ *are pcs's then* $\mathbf{D} {\to} \mathbf{D}'$ *is a pcs.*

Fact 8.2.6. *Let* $\mathcal{P}^{\mathrm{coh}}_{<\kappa}(\mathbf{D}) \equiv \langle E, \leq^*_{\mathbf{D}}, \frown^*_{\mathbf{D}} \rangle$ *where* $E \equiv \{a {\in} \mathcal{P}_{<\kappa}(D) \mid \mathrm{Coh}_{\mathbf{D}} a\}$*. If* $\mathbf{D}$ *is a pcs and* κ *is a cardinal, then* $\mathcal{P}^{\mathrm{coh}}_{<\kappa}(\mathbf{D})$ *is a pcs.*

8.3. The web of the canonical κ*-premodel*

Recall that κ is a regular cardinal. Let t and f be distinct non-pairs (e.g. $t = \emptyset$ and $f = \{\emptyset\}$). For all structures $\mathbf{D}$, define

Definition 8.3.1.
(a) $\mathbf{H}(\mathbf{D}) \equiv (\mathcal{P}^{\mathrm{coh}}_{<\kappa}(\mathbf{D}) {\to} \mathbf{D})_f \oplus \mathrm{U}(t)$
(b) Eq'_κ is the equation $\mathbf{H}(\mathbf{D}) = \mathbf{D}$

Fact 8.3.2. *If a pcs* $\mathbf{D}$ *satisfies* Eq'_κ *then* $(\mathcal{I}(\mathbf{D}), \subseteq)$ *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 $\mathbf{P}_\beta = \langle P_\beta, \leq_\beta, \frown_\beta \rangle$ is a pcs, that $\{\mathbf{P}_\beta \mid \beta \in \delta\}$ is a chain of pcs's and that:

Fact 8.3.3. *The pcs* $\mathbf{P}$ *is the* $\sqsubseteq$*-minimal (and in fact unique) solution of* Eq'_κ.

We define the rank $rk(p)$ of $p \in \mathbf{P}$ as the smallest ordinal α for which $p \in P_\alpha$. Recall that $P_0 = \emptyset$ and note that $P_1 = \{t, f\}$.

8.4. Some properties of the web

From now on $\downarrow$ means $\downarrow_{\mathbf{P}}$. Define $\mathbf{C} \equiv \mathcal{P}^{\mathrm{coh}}_{<\kappa}(\mathbf{P})$. For all $p \in \mathbf{P}$ and $\bar{c} = \langle c_1, \ldots, c_n \rangle \in \mathbf{C}^{<\omega}$ let $\ell(\bar{c})$ denote n (i.e. the length of $\bar{c}$) and define

$$\langle \bar{c}, p \rangle \equiv \langle c_1, \langle c_2, \langle \cdots \langle c_n, p \rangle \cdots \rangle \rangle \rangle$$

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

Lemma 8.4.1 ([4]). *For each* $p \in \mathbf{P}$ *there is a unique decomposition of* p *as* $p = \langle \bar{c}, t \rangle$ *or* $p = \langle \bar{c}, f \rangle$ *where* $\bar{c} \in \mathbf{C}^{<\omega}$.

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

Remark 8.4.2.
$\langle c, p \rangle \leq r \in \mathbf{P}$ *implies* $r = \langle e, q \rangle$ *for some* e, q.
$\langle c, p \rangle \leq \langle e, q \rangle$ *iff* $e \subseteq {\downarrow} c$ *and* $p \leq q$.

8.5. 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 8.5.1.
$\mathcal{M}_p \equiv \{{\downarrow} p \mid p \in \mathbf{P}\}$ *is the set of prime maps of* $\mathcal{M}$.
$\mathcal{M}_c \equiv \{{\downarrow} a \mid a \in \mathbf{C}\}$ *is the set of compact maps of* $\mathcal{M}$.
In $\mathcal{M}$*, sups are unions and infs are intersections.*

The definition of $\mathcal{M}_c$ above is compatible with the one in Section 6.3.

8.6. The canonical κ-premodel

Recall that T' and $\bot'$ are arbitrary, distinct constants which are not functions so $\mathsf{T}', \bot' \notin [\mathcal{M}{\to_\kappa}\mathcal{M}]$. Now define λ, T, $\bot$ and A by

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

We have:

Fact 8.6.1.
(a) $\lambda : \Big([\mathcal{M}{\to_\kappa}\mathcal{M}] \oplus_{\bot'} \{\mathsf{T}'\}\Big) \to \mathcal{M}$ is an order isomorphism.
(b) $\mathcal{M}$ is a solution to the domain equation Eq_κ.
(c) $(\mathcal{M}, A, \lambda)$ is a κ-premodel.

We are now able to define the canonical κ-premodel and thus also able to define the canonical κ- and $\kappa\sigma$-expansions as promised in Section 5.3.

Definition 8.6.2.
(a) The canonical κ-premodel is the triple $(\mathcal{M}, A, \lambda)$ with $\mathcal{M}$ defined as in Section 8.5 and λ and A defined as above.
(b) The canonical κ-expansion $\mathcal{M}_\kappa$ is the κ-expansion (cf. Definition 7.5.1) of the canonical κ-premodel.
(c) The canonical $\kappa\sigma$-expansion $\mathcal{M}_{\kappa\sigma}$ is the $\kappa\sigma$-expansion (cf. Definition 7.9.1) of the canonical κ-premodel.

Note that we have $\mathsf{T} = \{t\}$, $\bot = \emptyset$ and $\mathcal{F} = \mathcal{M} \setminus \{\mathsf{T}, \bot\}$ with $\mathcal{F}$ defined as in Definition 7.4.1. We have:

Fact 8.6.3.
(a) $u \in \mathcal{F}$ iff $u \in \mathcal{M}$ and $f \in u$.
(b) $\{f\}$ is the minimal element of $\mathcal{F}$ and models $\lambda x.\bot$.
(c) $\bot, \mathsf{T}, \{f\} \in \mathcal{M}_c$.

8.7. Tying up a loose end

As promised in Section 3.8, we are now able to prove the non-trivial direction of Fact 3.8.3. Recall the definitions of $a \sim_\kappa b$, $a =^\kappa_{\text{obs}} b$ and $a =_\kappa b$ from Section 3.8. Note that if $\forall c{\in}\mathcal{M}_\kappa{:}\, ca \sim_\kappa cb$ 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 3.8.3.

8.8. Conclusion

We have now constructed the canonical κ-premodel and the canonical κ-expansion $\mathcal{M}_\kappa$ and the canonical $\kappa\sigma$-expansion $\mathcal{M}_{\kappa\sigma}$. Thus, as promised in Section 5.6, we have finished the definition of the concepts introduced in Section 5.2–5.3 and are thus prepared to develop the consistency proof in Sections 9–13.

9. Canonical premodels satisfy Mono, Min and Ext

Recall the Mono and Min rules from Section 4.2 and the Ext rules from Section 4.3.

In this section we only suppose $\kappa \geq \omega$, and prove that $\mathcal{M}_\kappa$, the canonical κ-expansion, satisfies Mono, Min and Ext, that its model order $\subseteq$ coincides with the syntactical order $\preceq$ and that we could eliminate the constant Y in favor of Curry's paradoxical combinator. It is essential that $\mathcal{M}_\kappa$ is canonical since we constantly use Lemma 8.4.1.

From now on we use the same notation for terms and for their interpretations in $\mathcal{M}_\kappa$. Monotonicity of application w.r.t. $\subseteq$ will be used constantly, most often without mention.

9.1. A characterization of the order of $\mathcal{M}_\kappa$ via application

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

Definition 9.1.1. *Let* $\mathsf{F} \equiv \lambda x.\, \mathsf{T}$ *and* $r \equiv \lambda u.\, \mathsf{if}[u, \mathsf{T}, \mathsf{F}]$.

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

Theorem 9.1.2. *For all $u, v \in \mathcal{M}_\kappa$ the following are equivalent:*

(i) $u \subseteq v$

(ii) *For all* $\bar{w} \in \mathcal{M}_\kappa^{<\omega}$ *we have* $r(u\bar{w}) \subseteq r(v\bar{w})$

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

(ii) $\Rightarrow$ (i). The proof is by contradiction. Choose a p in $\mathbf{P}$ of minimal length for which there exist u, v satisfying (ii) such that $p \in u$ and $p \notin v$.

From $r(u) \subseteq r(v)$ we have $t \in u \Rightarrow t \in v$ and $f \in u \Rightarrow f \in v$ so $p \neq t$ and $p \neq f$. Thus, p has form $\langle c, q\rangle$.

From $p = \langle c, q\rangle \in u$ we have $r(u) = r(v) = \mathsf{F}$ so $u, v \in \mathcal{F}$. Hence, using the definition of A (cf. Section 8.6) we have $q \in u(\downarrow c) \Leftrightarrow \exists c' \subseteq \downarrow c \colon \langle c', q\rangle \in u \Leftrightarrow \langle c, q\rangle \in u \Leftrightarrow p \in u$. Likewise, $q \in v(\downarrow c) \Leftrightarrow p \in v$.

From $\ell(q) < \ell(p)$ and the minimality of $\ell(p)$ we have $q \in u' \Rightarrow q \in v'$ for all u', v' satisfying (ii). Thus, $p \in u \Leftrightarrow q \in u(\downarrow c) \Rightarrow q \in v(\downarrow c) \Leftrightarrow p \in v$ yielding a contradiction. □

Corollary 9.1.3. *For all $u, v \in \mathcal{M}_\kappa$ 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 $\mathcal{M}_\kappa$ was a premodel ($\mathcal{M}_\kappa$ is not required to be canonical for the corollary). □

9.2. Ext

Theorem 9.2.1. $\mathcal{M}_\kappa \models \mathsf{Ext}$

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) $\mathcal{M}_\kappa \models \forall w \forall v\colon (\mathcal{A}wv = \mathcal{A}\mathcal{C}[w/x, v/y] \wedge \mathcal{B}wv = \mathcal{B}\mathcal{C}[w/x, v/y])$. The task is to prove that $\mathcal{M}_\kappa \models \forall w\colon (\mathcal{A}w = \mathcal{B}w)$ under the hypothesis that $\mathcal{M}_\kappa \models \forall w\colon (r(\mathcal{A}w) = r(\mathcal{B}w))$. Now, the hypothesis on $\mathcal{A}$ and $\mathcal{B}$ obviously imply that, given $w \in \mathcal{M}_\kappa$, the elements $\mathcal{A}w$ and $\mathcal{B}w$ satisfy point (ii) of Theorem 9.1.2; by (i) we hence have $\mathcal{A}w \subseteq \mathcal{B}w$. Similarly, $\mathcal{B}w \subseteq \mathcal{A}w$ so $\mathcal{A}w = \mathcal{B}w$. □

9.3. λ-definability of the order of $\mathcal{M}_\kappa$

Theorem 9.3.1. $\mathcal{M}_\kappa \models u \downarrow v = u \cap v$ *for all* $u, v \in \mathcal{M}_\kappa$.

Proof. The proof is by contradiction. Choose a p in $\mathbf{P}$ of minimal length for which there exist u, v such that $\neg(p \in u \downarrow v \Leftrightarrow p \in u \cap v)$.

Since $t \in u \downarrow v \Leftrightarrow u = v = \mathsf{T} \Leftrightarrow t \in u \cap v$ we have $p \neq t$. Likewise, since $f \in u \downarrow v \Leftrightarrow u, v \in \mathcal{F} \Leftrightarrow f \in u \cap v$ we have $p \neq f$. Thus, p has form $\langle c, q \rangle$.

If $p = \langle c, q \rangle \in u \downarrow v$ then $u, v \in \mathcal{F}$. If $p = \langle c, q \rangle \in u \cap v$ then also $u, v \in \mathcal{F}$. Thus, in any case, $u, v \in \mathcal{F}$. Hence, using the definition of A (cf. Section 8.6) we have $q \in u(\downarrow c) \Leftrightarrow \exists c' \subseteq \downarrow c\colon \langle c', q \rangle \in u \Leftrightarrow \langle c, q \rangle \in u \Leftrightarrow p \in u$. Likewise, $q \in v(\downarrow c) \Leftrightarrow p \in v$ and $q \in (u \downarrow v)(\downarrow c) \Leftrightarrow p \in u \downarrow v$.

From $\ell(q) < \ell(p)$ and the minimality of $\ell(p)$ we have $q \in u' \downarrow v' \Leftrightarrow q \in u' \cap v'$ for all u', v'. Thus, $p \in u \downarrow v \Leftrightarrow q \in (u \downarrow v)(\downarrow c) \Leftrightarrow q \in u(\downarrow c) \downarrow v(\downarrow c) \Leftrightarrow q \in u(\downarrow c) \cap v(\downarrow c) \Leftrightarrow q \in u(\downarrow c) \wedge q \in v(\downarrow c) \Leftrightarrow p \in u \wedge p \in v \Leftrightarrow p \in u \cap v$ yielding a contradiction. □

Corollary 9.3.2. $(\mathcal{M}_\kappa \models u \preceq v) \Leftrightarrow u \subseteq v$ *for all* $u, v \in \mathcal{M}_\kappa$.

Corollary 9.3.3. *In $\mathcal{M}_\kappa$ the binary κ-continuous function inf is definable by a λ-term (using* if, $\bot$ *and* T*), and hence the model order $\subseteq$ is equationally definable.*

Remark 9.3.4. *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 [15], was rediscovered independently by Kerth [12], who proved that it also holds in Berry's and Girard's stable semantics, and Ehrhard's strongly stable semantics [13] (with different formulas).*

9.4. Mono *and* Min

Theorem 9.4.1. $\mathcal{M}_\kappa \models \mathsf{Mono} + \mathsf{Min}$.

Proof of 9.4.1 Follows from Corollary 9.3.2 and from the fact that application is monotonic w.r.t. the model order, and that $\mathsf{Y} = \mathsf{Y}_{\text{Tarski}}$ acts as a minimal fixed point w.r.t. the model order.

9.5. Definability of the fixed point operator

Now we show that $\mathcal{M}_\kappa$ 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 9.5.1. *For all $u \in \mathcal{M}_\kappa$ and ordinals α let $u_\alpha \equiv \downarrow(u \cap \mathbf{P}_\alpha) \in \mathcal{M}_\kappa$.*

Lemma 9.5.2. *For all $u, v \in \mathcal{M}_\kappa$ 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* $\beta \geq 0$.

In particular, $u_1 v = u_1 v_0$.

Proof. (i) Obvious.

(ii) Obvious when $u \in \{\mathsf{T}, \bot\}$. Now assume $c \in \mathcal{F}$. We have $\cup_{\beta<\delta}(u_\beta v) \subseteq u_\delta v$ by monotonicity. Now assume $q \in u_\delta v$. Choose $c \subseteq v$ such that $\langle c, q\rangle \in u_\delta \equiv \downarrow(u \cap \mathbf{P}_\delta)$. Choose $p = \langle e, q'\rangle \in u \cap \mathbf{P}_\delta$ such that $\langle c, q\rangle \leq p$. Choose $\beta < \delta$ such that $p \in \mathbf{P}_\beta$. We have $q \leq q'$ and $e \subseteq \downarrow c$ (cf. Remark 8.4.2). Now $q \leq q' \in (\downarrow p)(\downarrow e) \subseteq u_\beta v$ so $q \in u_\beta v$.

(iii) Obvious when $u \in \{\mathsf{T}, \bot\}$. Now assume $c \in \mathcal{F}$. We have $u_{\beta+1} v_\beta \subseteq u_{\beta+1} v$ by monotonicity. Now assume $q \in u_{\beta+1} v$. Choose $c \subseteq v$ such that $\langle c, q\rangle \in u_{\beta+1} \equiv \downarrow(u \cap \mathbf{P}_{\beta+1})$. Choose $p = \langle e, q'\rangle \in u \cap \mathbf{P}_{\beta+1}$ such that $\langle c, q\rangle \leq p$. We have $q \leq q'$ and $e \subseteq \downarrow c$ (cf. Remark 8.4.2). Furthermore, $p \in \mathbf{P}_{\beta+1}$ implies $e \subseteq \mathbf{P}_\beta$. Now $q \leq q' \in (\downarrow p)(\downarrow e) \subseteq u_{\beta+1} v_\beta$ so $q \in u_{\beta+1} v_\beta$. □

Theorem 9.5.3. $\mathcal{M}_\kappa \models \mathsf{Y}_{\mathrm{Curry}} = \mathsf{Y}$.

Proof. Recall that $\mathcal{M}_\kappa$ interprets Y as $\mathsf{Y}_{\mathrm{Tarski}}$. Since $\mathsf{Y}_{\mathrm{Tarski}}$ acts as the least fixed point operator on $\mathcal{M}_\kappa$ it is enough to prove that, for all $u \in \mathcal{M}_\kappa$, 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 comes by Lemma 9.5.2(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 9.5.2(iii) and the last inclusion by induction hypothesis. □

Remark 9.5.4. *Most usual models of untyped λ-calculus 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 sup of an increasing sequence u_α, $\alpha \in \kappa$ (usually $\kappa = \omega$) satisfying all the properties listed in Lemma 9.5.2 except $u_1 v = u_1 v_0$. This last equation is really the key point here. The equation $u_1 v = u_1 v_0$ holds e.g. for Scott's first model $\mathcal{D}_\infty$ and fails for Park's variant of $\mathcal{D}_\infty$ which does not satisfy* Min.

10. Concepts for proving the Definability Theorem

Recall the definition of Φ (Definition 7.8.2), of ϕ as the characteristic map of Φ, of ψ (Definition 4.7.1) and of $\mathsf{D}[g]$ (Definition 4.6.1). In particular, $\Phi = \mathsf{D}[\phi]$.

The aim of Section 10–12 is to prove:

Theorem 10.1 (Definability Theorem). *If $\sigma < \kappa$ is the smallest inaccessible ordinal then any $\kappa\sigma$-expansion satisfies $\psi = \phi$.*

Section 11 proves UBT ($\psi \preceq_{\mathcal{M}} \phi$, Theorem 11.3.1) for any inaccessible σ using $\mathsf{Y} = \mathsf{Y}_{\text{Tarski}}$. Section 12 proves LBT ($\phi \preceq_{\mathcal{M}} \psi$, Theorem 12.4.3) for the first inaccessible σ. The proof of LBT uses UBT (in Lemma 12.4.1) and the minimality of σ (in Lemma 12.4.2).

Section 10 provides preliminary material and results which will be used in Section 11 and 12. Sections 10.2–10.6 present and reorganize concepts and results which were either explicit or implicit in [4] (including its appendices). Section 10.7 applies 10.5–10.6 to the "components" Q and R of ψ and is hence new material.

The notation is essentially that of [4] except that the notation g/h introduced here replaces $\Downarrow_H g$ where $H = \mathsf{D}[h]$, and $G^\bullet$ replaces $G^{\circ\delta}$.

In the following, $\sigma < \kappa$ is inaccessible and κ is still understood to be regular. We work in a $\kappa\sigma$-expansion $\mathcal{M}$. We refer to elements of $\mathcal{M}$ as *maps*. Unless otherwise noted, variables range over $\mathcal{M}$.

10.1. Necessity of assumptions

The proof of UBT uses the minimality of Y and the proof of LBT uses the minimality of σ, the minimality of Y and UBT. The last two dependencies may be seen as a convenience whereas the two other dependencies are essential. We elaborate on this in the following.

Recall that UBT says $\psi \preceq_{\mathcal{M}} \phi$ where $\psi \equiv \sqcup s$ and $s \equiv \mathsf{Y}S$. The proof of UBT uses that s is the minimal fixed point of S. To see that this is needed, it is enough to show that S has a non-minimal fixed point for which UBT fails:

Lemma 10.1.1. *Let $\sigma < \kappa$ be inaccessible and let $\mathcal{M}$ be any $\kappa\sigma$-expansion. There exists an $s' \in \mathcal{M}$ such that $Ss' = s'$ and $\mathsf{D}[\psi'] = \mathcal{M}$ where $\psi' = \sqcup s'$.*

For the proof see Appendix C where Lemma 10.1.1 is restated as Lemma C.1.

For monotonicity reasons, if LBT is true for $\mathcal{M}$ when interpreting Y by the minimal fixed point operator, then it is obviously also true for any other fixed point operator of $\mathcal{M}$. In other words, the satisfaction of LBT does not require $\mathsf{Y} = \mathsf{Y}_{\text{Tarski}}$ and we can conjecture that there exists a proof of LBT not using it; since the proof of UBT needs minimality (c.f. Lemma C.1), getting rid of minimality for LBT would also mean getting rid of UBT.

Finally, LBT does indeed depend on σ being the *first* inaccessible: According to LBT we have $\mathsf{D}[\psi] = \Phi$ when σ is minimal, and since different choices of σ give rise to different Φ we cannot have $\mathsf{D}[\psi] = \Phi$ for non-minimal σ.

10.2. Duals, boundaries, closure and functions

We now state some definitions, many of which are repetitions of earlier definitions.

Definition 10.2.1. Let $G, H \subseteq \mathcal{M}$ and $g, h \in \mathcal{M}$.

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

We refer to G° as the *dual* of G. The set $G^\bullet$ is the set of minimal elements of G°.

Definition (a) above repeats Definition 7.8.1(c). Definition (b) makes explicit how to understand $\emptyset^\circ$. Definition (d) makes a definition in Section 6.1 explicit. Definition (e) repeats Definition 7.8.1(b).

Fact 10.2.2.
(a) $G \subseteq H \Rightarrow H^\circ \subseteq G^\circ$
(b) $G' \subseteq G \wedge H \subseteq H' \Rightarrow G{\to}H \subseteq G'{\to}H'$
(c) $G \neq \emptyset \Rightarrow G^\circ = G{\to}G^\circ$
(d) $G \subseteq H \subseteq H^{\circ\circ} \Rightarrow G^+ \equiv G^\circ{\to}G \subseteq H^\circ{\to}H^{\circ\circ} = H^{\circ\circ}$

Note that G° is anti-monotonic in G and that $G{\to}H$ is monotonic in H but anti-monotonic in G. That allows to combine G° and $G{\to}H$ into monotonic operators $G^{\circ\circ}$ and $G^+ \equiv G^\circ{\to}G$:

Fact 10.2.3.
(a) $G \subseteq H \Rightarrow G^{\circ\circ} \subseteq H^{\circ\circ}$
(b) $G \subseteq H \Rightarrow G^+ \subseteq H^+$

For all $G \subseteq \mathcal{M}$ recall from Sections 6.1, 6.5 and 7.8 that G is *essentially σ-small* if there exists a σ-small V such that $V \subseteq G \subseteq \uparrow\! V$. If G is open then G is essentially σ-small iff $G = \uparrow\! V$ for some σ-small V.

Let $\mathcal{O}(G)$ denote the set of open subsets of G. Recall from Definition 7.8.1(a) that $\mathcal{O}_{<\sigma}(G)$ denotes the set of essentially σ-small open subsets of G. We define $\mathcal{O}_{<\kappa}(G)$ likewise. We use $\mathcal{O}(G)$, $\mathcal{O}_{<\sigma}(G)$ and $\mathcal{O}_{<\kappa}(G)$ only for G open (and mostly for $G = \Phi$ and $G = \mathcal{M}$). From Remark 7.7.2 we have:

Fact 10.2.4.
(a) If $g \in \mathcal{M}$ then $\mathsf{D}[g] \in \mathcal{O}(\mathcal{M})$.
(b) If $G \in \mathcal{O}(\mathcal{M})$ then $G = \mathsf{D}[g]$ for some $g \in \mathcal{M}$.
(c) $\mathcal{O}(\mathcal{M}) = \{\mathsf{D}[g] \mid g \in \mathcal{M}\}$.

We use $\mathsf{D}[g]$ only when $g \in \mathcal{M}$. Thus, whenever we assume $G = \mathsf{D}[g]$ we implicitly assume $g \in \mathcal{M}$.

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

Theorem 10.2.5.
(a) If $G \in \mathcal{O}(\mathcal{M})$ then $G^\bullet$ is a set of incompatible elements, $G^\circ = \uparrow\!(G^\bullet)$, and if $G \neq \emptyset$ then $G^\bullet$ is infinite.

(b) If $G \in \mathcal{O}_{<\kappa}(\mathcal{M})$ then $G^\circ \in \mathcal{O}(\mathcal{M})$ and $G^\bullet \in \mathcal{M}_c$.
(c) If $G \in \mathcal{O}_{<\sigma}(\mathcal{M})$ then $G^\circ \in \mathcal{O}_{<\sigma}(\mathcal{M})$ and $G^\bullet \in \mathcal{P}_{<\sigma}(\mathcal{M}_c)$.

Proof of 10.2.5 For $G \neq \emptyset$ this is [4, Theorem 6.1.11] adapted to the notation of the present paper. For $G = \emptyset$ the theorem follows trivially from the definition of $\emptyset^\circ$ (Definition 10.2.1(b)).

Lemma 10.2.6 (Closure properties of $\mathcal{O}_{<\sigma}(\mathcal{M})$ and $\mathcal{O}_{<\sigma}(\Phi)$).
(a) If $G \in \mathcal{O}_{<\sigma}(\mathcal{M})$ then $G^\circ \in \mathcal{O}_{<\sigma}(\mathcal{M})$.
(b) $\mathcal{O}_{<\sigma}(\Phi)$ and $\mathcal{O}_{<\sigma}(\mathcal{M})$ are closed under σ-small unions.
(c) If $G \in \mathcal{O}_{<\sigma}(\Phi)$ then $G^+ \in \mathcal{O}_{<\sigma}(\Phi)$.
(d) If $G \in \mathcal{O}_{<\sigma}(\Phi)$ then $G^{\circ\circ} \in \mathcal{O}_{<\sigma}(\Phi)$.

Proof of 10.2.6
(a) Is part of Theorem 10.2.5.
(b) Follows from the regularity of σ, i.e. the fact that a σ-small union of σ-small sets is σ-small.
(c) Follows directly from the definition of Φ (Definition 7.8.2) (and implicitly uses that σ is inaccessible).
(d) Will be re-stated and proved as Lemma 10.4.4; we do not yet have the material to prove it, but we include it here for the sake of completeness. Note that $\mathcal{O}_{<\sigma}(\Phi)$ is closed under $^{\circ\circ}$ but not under $^\circ$.

10.3. Elementary observations

We now list some facts which we shall use without reference in the rest of the paper. Some of the facts have been used before.

Fact 10.3.1.
(a) $\uparrow\{\bot\} = \mathcal{M}$
(b) $\uparrow\{\mathsf{T}\} = \{\mathsf{T}\}$
(c) $\uparrow\mathcal{F} = \mathcal{F}$

Fact 10.3.2.
(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 10.3.3.
(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 10.3.4.
(a) $!x = \mathsf{T} \Leftrightarrow x \neq \bot$

Fact 10.3.5.
(a) $f \preceq_{\mathcal{M}} g \Rightarrow \mathsf{D}[f] \subseteq \mathsf{D}[g]$
(b) $\mathsf{D}[\sqcup f] = \cup_{x \in \mathcal{M}} \mathsf{D}[fx]$

10.4. On the definition of Φ

There are many ways to build Φ. The definition chosen in the present paper was stated as Definition 7.8.2. The one chosen in [4] was to build a set Ψ of maps as the union of an increasing sequence $(\Psi_\alpha)_{\alpha\in\sigma}$ of σ-small sets and then take $\Phi = \uparrow\Psi$. However, as shown in [4] and below, Φ could as well be inductively defined as the limits of certain increasing sequences $(\Phi_\alpha)_{\alpha\in\sigma}$ and $(\mathcal{H}_\alpha)_{\alpha\in\sigma}$ of essentially σ-small open sets. Using these two sequences will be pertinent for proving UBT and LBT.

Note that [4, Theorem 7.1.1] states that there exists a Φ with certain properties which is enough for the development in [4]. Then the proof of [4, Theorem 7.1.1] constructs a concrete Φ which is the one we refer to here as "the Φ defined in [4]".

We now define $(\Phi_\alpha)_{\alpha\in\sigma}$ and $(\mathcal{H}_\alpha)_{\alpha\in\sigma}$ and then move straight to Lemma 10.4.2 which is important because it allows to use all theorems about Φ in [4] in the present paper.

Definition 10.4.1. For all $\alpha \leq \sigma$ define Φ_α and $\mathcal{H}_\alpha$ thus:

(a) $\Phi_0 \equiv \{\mathsf{T}\}$
(b) $\Phi_{\alpha+1} \equiv \Phi_\alpha^+$
(c) $\Phi_\delta \equiv \cup_{\beta\in\delta}\Phi_\beta$ for limit ordinals δ.
(d) $\mathcal{H}_0 \equiv \{\mathsf{T}\}$
(e) $\mathcal{H}_{\alpha+1} \equiv \mathcal{H}_\alpha^{\circ\circ}$
(f) $\mathcal{H}_\delta \equiv \cup_{\beta\in\delta}\mathcal{H}_\beta$ for limit ordinals δ.

Lemma 10.4.2.

(a) $\Phi = \Phi_\sigma$.
(b) The Φ defined in the present paper equals the Φ defined in [4].

Proof of 10.4.2

(a) By transfinite induction using Fact 10.2.3 and Lemma 10.2.6 we have $\Phi_\alpha \subseteq \Phi_\beta$ for all $\alpha \leq \beta \leq \sigma$ and $\Phi_\alpha \in \mathcal{O}_{<\sigma}(\Phi)$ for all $\alpha < \sigma$ (for reference, these two easy results are stated again below as Lemma 10.4.3(a) and 10.4.5, respectively). Thus $\Phi_\alpha \subseteq \Phi$ by transfinite induction using the definition of Φ (Definition 7.8.2). In particular $\Phi_\sigma \subseteq \Phi$. Now assume $G \in \mathcal{O}_{<\sigma}(\Phi_\sigma)$. Then $G \subseteq \Phi_\alpha$ for some $\alpha \in \sigma$ since σ is regular so $G^+ \subseteq \Phi_{\alpha+1} \subseteq \Phi_\sigma$. Thus, $G \in \mathcal{O}_{<\sigma}(\Phi_\sigma) \Rightarrow G^+ \subseteq \Phi_\sigma$ so $\Phi \subseteq \Phi_\sigma$ by the minimality of Φ (c.f. Definition 7.8.2).

(b) Let Φ' denote the Φ defined in the proof of [4, Theorem 7.1.1] and let Φ'_α denote the Φ_α defined in the proof of [4, Lemma A.1.1]. We now prove $\Phi' = \Phi'_\sigma = \Phi_\sigma = \Phi$.

Proof of $\Phi' = \Phi'_\sigma$. As stated without proof in the proof of [4, Lemma A.1.1] we have $\Phi' = \Phi'_\sigma$; it is an easy consequence of [4, Lemma 7.1.2].

Proof of $\Phi'_\sigma = \Phi_\sigma$. By the definition of Φ'_0 we have $\Phi'_0 = \{\mathsf{T}\} = \Phi_0$. Furthermore, according to the proof of [4, Lemma A.1.1] we have $\Phi'_\alpha = \cup_{\beta\in\alpha}(\Phi'^{\circ}_\alpha \rightarrow \Phi'_\alpha)$ if $0 < \alpha < \sigma$, so $\Phi'_\alpha = \Phi_\alpha$ for all $\alpha \leq \sigma$ by transfinite induction. In particular, $\Phi'_\sigma = \Phi_\sigma$.

Finally, $\Phi_\sigma = \Phi$ by (a) which finishes the proof.

Lemma 10.4.3.
(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) $\forall G \in \mathcal{O}_{<\sigma}(\Phi)\, \exists \alpha \in \sigma\colon G \subseteq \Phi_\alpha \subseteq \mathcal{H}_\alpha$
(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 10.4.3
(a,b) By transfinite induction using Fact 10.2.3.
(c) We have $\mathcal{H}_\alpha \subseteq \mathcal{H}_{\alpha+1} = \mathcal{H}_\alpha{}^{\circ\circ}$ by (b) and Definition 10.4.1(e). Hence, $\Phi_\alpha \subseteq \mathcal{H}_\alpha$ by transfinite induction using Fact 10.2.2(d).
(d) For $\Phi = \Phi_\sigma$ see Lemma 10.4.2(a). For $\mathcal{H}_\sigma \subseteq \Phi$ see [4, Theorem A.2.2] and its proof. Finally, $\Phi_\sigma \subseteq \mathcal{H}_\sigma$ is given by (c).
(e) Let V be σ-small and such that $G = \uparrow V$. For each $g \in V$ let $\rho(g)$ be the smallest ordinal for which $g \in \Phi_{\rho(g)}$. Take $\alpha = \cup_{g \in V} \rho(g)$.
(f) [4, Theorem 7.1.1].
(g) Follows trivially from (a-d,f) and Fact 10.2.2(a).

Lemma 10.4.4. *$G^{\circ\circ} \in \mathcal{O}_{<\sigma}(\Phi)$ for all $G \in \mathcal{O}_{<\sigma}(\Phi)$.*

Proof of 10.4.4 This is the announced re-statement of Lemma 10.2.6(d). We have $G^{\circ\circ} \in \mathcal{O}_{<\sigma}(\mathcal{M})$ by Lemma 10.2.6(a). It remains to prove that $G^{\circ\circ} \subseteq \Phi$. Using Lemma 10.4.3(e) take α such that $G \subseteq \mathcal{H}_\alpha$. Then $G^{\circ\circ} \subseteq \mathcal{H}_\alpha^{\circ\circ} \equiv \mathcal{H}_{\alpha+1}$ by Fact 10.2.3(a). Thus $G^{\circ\circ} \subseteq \Phi$ by Lemma 10.4.3(g).

Lemma 10.4.5. *$\Phi_\alpha \in \mathcal{O}_{<\sigma}(\Phi)$ and $\mathcal{H}_\alpha \in \mathcal{O}_{<\sigma}(\Phi)$ for all $\alpha \in \sigma$.*

Proof of 10.4.5 By transfinite induction using Lemma 10.2.6 and 10.4.4.

Lemma 10.4.6. *$\Phi \in \mathcal{O}_{<\kappa}(\mathcal{M})$.*

Proof of 10.4.6 From Lemma 10.4.5, $\sigma < \kappa$ and the regularity of κ we have $\Phi = \Phi_\sigma \in \mathcal{O}_{<\kappa}(\mathcal{M})$.

10.5. Projections

Definition 10.5.1. Let $G, H \subseteq \mathcal{M}$ and $g, h \in \mathcal{M}$.
(a) $g/h \equiv \mathsf{if}[\, g\,,\, \mathsf{T}\,,\, \lambda x.\, hx : (gx/h)\,]$
(b) $G/h \equiv \{g/h \mid g \in G\}$.
(c) $gH \equiv \{gh \mid h \in H\}$.

Definition (a) repeats Definition 4.6.2(d).

Recall that since / is an infix operator we have that ab/cd means $(ab)/(cd)$. Likewise, gH/k means $(gH)/k$ which equals $\{(gh)/k \mid h \in H\}$.

As mentioned in Section 4.6, the g/h construct is a kind of "transitive restriction" of the function g to the domain $H = \mathsf{D}[h]$. But what makes the construct interesting here is that it is a projection in the sense that $(g/h)/h = g/h \preceq_{\mathcal{M}} g$ (cf. Lemma 10.5.3). More specifically, $g \mapsto g/h$ is a projection from H° onto $H^\bullet$:

Lemma 10.5.2. *If* $H = \mathsf{D}[h]$ *then* $H^\bullet = H^\circ/h$.

Proof of 10.5.2 Like was the case for Theorem 10.2.5, this is part of [4, Theorem 6.1.11] adapted to the notation of the present paper. For $H = \emptyset$ the lemma follows trivially from the definition of $\emptyset^\circ$ (Definition 10.2.1(b)).

Lemma 10.5.3. Let $H = \mathsf{D}[h]$. We have:

(a)	$g/h \preceq_{\mathcal{M}} g$.	
(b)	$g/h = \mathsf{T}$	iff $g = \mathsf{T}$.
(c)	$g/h = g'/h$	if $g \in H^\circ$ and $g \preceq_{\mathcal{M}} g'$.
(d)	$g/h \in K \Leftrightarrow g \in \uparrow K$	if $K \subseteq H^\bullet$.
(e)	$g/h = (g/h')/h$	if $\mathsf{D}[h] \subseteq \mathsf{D}[h']$.

Proof of 10.5.3

(a) In Example 4.3.1 we defined i such that $ig = \mathsf{if}[\,g\,,\,\mathsf{T}\,,\,\lambda x.\,i(gx)\,]$. From that we proved $ig = g$ using Ext. Now define $\ddot{\imath} \equiv \lambda hg.\,\mathsf{if}[\,g\,,\,\mathsf{T}\,,\,\lambda x.\,\ddot{\imath}h(gx)\,]$. Repeating the argument in Example 4.3.1 on $\ddot{\imath}h$ in place of i we get $\ddot{\imath}hg = g$.

We have $a : b \preceq_{\mathcal{M}} b$ when a is T, $\bot$, or a function, so $a : b \preceq_{\mathcal{M}} b$ by QND. Using $a : b \preceq_{\mathcal{M}} b$, $\ddot{\imath}hg = g$ and that recursive definitions are shorthand for definitions that use Y we have

$$\begin{array}{rcl} g/h & \equiv & (\mathsf{Y}\lambda fhg.\,\mathsf{if}[\,g\,,\,\mathsf{T}\,,\,\lambda x.\,hx : fh(gx)\,])hg \\ & \preceq_{\mathcal{M}} & (\mathsf{Y}\lambda fhg.\,\mathsf{if}[\,g\,,\,\mathsf{T}\,,\,\lambda x.\,fh(gx)\,])hg \quad \equiv \quad \ddot{\imath}hg = g \end{array}$$

Above, we have taken the liberty to consider h as the first and g as the second parameter of g/h. That is immaterial, but avoids some technicalities here.

(b) By the definition of g/h.

(c) From $g \in H^\circ$ and $g \preceq_{\mathcal{M}} g'$ we have $g' \in H^\circ$. Then by Theorem 10.2.5(a) and Lemma 10.5.2 we have $g/h \in H^\bullet$, $g'/h \in H^\bullet$ and g/h and g'/h are either equal or incompatible. But $g/h \preceq_{\mathcal{M}} g'/h$ by monotonicity, so g/h and g'/h are equal.

(d) First we note that $\uparrow K \subseteq H^\circ$ (Theorem 10.2.5) and that $\Rightarrow$ follows from (a). Suppose now that $g \in \uparrow K$ and take $g' \in K$ such that $g' \preceq_{\mathcal{M}} g$. By (c) we have $g/h = g'/h$. Furthermore, $g' \in H^\bullet$ implies $g'/h = g' \in K$. Hence, $g/h \in K$.

(e) If $\mathsf{D}[h] \subseteq \mathsf{D}[h']$ then $hy : z = hy : (h'y : z)$. The claim then follows from the definition of g/h.

We use $G \leq_c H$ to denote that G has the same or smaller cardinality than H.

Lemma 10.5.4. $\mathsf{D}[g]/h \leq_c \mathsf{D}[g']/h'$ *if* $\mathsf{D}[g] \subseteq \mathsf{D}[g']$ *and* $\mathsf{D}[h] \subseteq \mathsf{D}[h']$

Proof of 10.5.4 We prove that the function $k(x) = x/h$ is surjective from a subset of $\mathsf{D}[g']/h'$ onto $\mathsf{D}[g]/h$. Suppose $y \in \mathsf{D}[g]/h$. Let $x \in \mathsf{D}[g]$ satisfy $y = x/h$. Now $z \equiv x/h' \in \mathsf{D}[g]/h' \subseteq \mathsf{D}[g']/h'$ satisfies $k(z) = (x/h')/h = x/h = y$ by Lemma 10.5.3(e).

Lemma 10.5.5. Assume $G = \mathsf{D}[g] \subseteq \Phi$. We have:

(a) $\Phi/g \subseteq G^\bullet$
(b) $a\Phi/\phi \in \mathcal{P}_{<\sigma}(\Phi^\bullet)$ if $a \in \Phi$.
(c) $\Phi/g \in \mathcal{P}_{<\sigma}(\mathcal{M}_c)$ if $G \in \mathcal{O}_{<\sigma}(\Phi)$.

Proof of 10.5.5

(a) $G \subseteq \Phi$ gives $\Phi \subseteq \Phi^\circ \subseteq G^\circ$ by Lemma 10.4.3(f) and Fact 10.2.2(a). We conclude using Lemma 10.5.2.
(b) Choose $\alpha < \sigma$ such that $a \in \Phi_{\alpha+1} = \Phi_\alpha^\circ \to \Phi_\alpha$. Since $\Phi \subseteq \Phi_\alpha^\circ$ (cf. Lemma 10.4.3) we have $a\Phi \subseteq \Phi_\alpha \subseteq \Phi$. Thus, $a\Phi/\phi \subseteq \Phi/\phi \subseteq \Phi^\bullet$ by (a). From Lemma 10.4.5 we have $\Phi_\alpha \in \mathcal{O}_{<\sigma}(\Phi)$. Take $K \in \mathcal{P}_{<\sigma}(\Phi)$ such that $\Phi_\alpha = \uparrow K$. We have $a\Phi/\phi \subseteq \Phi_\alpha/\phi = K/\phi$ which is σ-small.
(c) By (a) and Theorem 10.2.5(c).

Lemma 10.5.6. *Let $g \in \mathcal{M}$. Assume $G \equiv \mathsf{D}[g] \in \mathcal{O}_{<\kappa}(\mathcal{M})$. Let $k\colon G^\bullet \to \mathcal{M}$. Then there exists an $h \in \mathcal{F}$ such that $hx = k(x/g)$ when $x \in G^\circ$ and $hx = \perp$ otherwise. Note that $hx = h(x/g)$ for all $x \in \mathcal{M}$.*

Proof of 10.5.6 Let $k'\colon \mathcal{M} \to \mathcal{M}$ satisfy $k'(x) = k(x/g)$ when $x \in G^\circ$ and $k'(x) = \perp$ otherwise. Then k' is a κ-step function. Suppose indeed $x \preceq_{\mathcal{M}} y$ and $k'(x) \neq \perp$; then $x \in G^\circ$, hence $x/g = y/g$ by Lemma 10.5.3. Hence, k' is κ-continuous (Lemma 6.6.3) and $h = \lambda(k')$ satisfies the first conclusion of the lemma.

Now if $x \in G^\circ$ then $h(x/g) = hx$ since $h(x/g) \equiv k((x/g)/g)$ and $(x/g)/g = x/g$; finally if $x \notin G^\circ$ then $h(x/g) = h(x) = \perp$.

10.6. Self-extensionality

We now recall the definition of *self-extensionality* plus some auxiliary concepts from [4, Appendix A.2]. First recall $r = \lambda u.\, \mathsf{if}[u, \mathsf{T}, \lambda x.\, \mathsf{T}]$ from Definition 9.1.1. Then recall the definition of $x =_G y$ from [4]:

Definition 10.6.1. $x =_G y$ *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 9.1.2. Now the definition of self-extensionality reads:

Definition 10.6.2. $G \subseteq \mathcal{M}$ 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" is borrowed from [4] and refers to the property $x =_G y \Rightarrow x =_\Phi y$ which happens to follow from (c) above and [4, Lemma A.2.1]. We shall neither use (c) nor $x =_G y \Rightarrow x =_\Phi y$ explicitly in the present paper.

Note that $G^{\circ\circ} = G^\circ \to G^{\circ\circ}$ for all G (cf. Fact 10.2.2(c)).

Lemma 10.6.3. *If $G \equiv \mathsf{D}[g]$ is self-extensional then $G^{\circ\circ}$ is self-extensional and $G^\bullet \subseteq G^{\circ\circ}/g$.*

Proof of 10.6.3 This is [4, Lemma A.2.4 and A.2.5].

Lemma 10.6.4. *If* $G \equiv \mathsf{D}[g]$ *is self-extensional then* $\Phi/g = G^\bullet$

Proof of 10.6.4 We have $\Phi/g \subseteq G^\bullet$ by Lemma 10.5.5(a). From Lemma 10.6.3 we have $G^{\circ\circ} \subseteq \Phi$ and $G^\bullet \subseteq G^{\circ\circ}/g \subseteq \Phi/g$.

Lemma 10.6.5. $\mathcal{H}_\alpha$ *is self-extensional for all* $\alpha \in \sigma$.

Proof of 10.6.5 By Lemma 10.4.5 and Lemma 10.4.3(b) we have that $\mathcal{H}_\alpha$ satisfies Definition 10.6.2 (a) and (b), respectively. By transfinite induction on α using [4, Theorem A.2.1] $\mathcal{H}_\alpha$ also satisfies Definition 10.6.2(c).

Lemma 10.6.6. *For all* $G \in \mathcal{O}_{<\sigma}(\Phi)$ *there is a self-extensional* H *such that* $G \subseteq H$.

Proof of 10.6.6 By Lemma 10.4.3(e) and Lemma 10.6.5

10.7. Properties of Q *and* R

The definition of ψ (Definition 4.7.1) includes definitions of the auxiliary maps P, Q and R. The lemma below states the properties of P, Q and R that we use for proving UBT and LBT.

Lemma 10.7.1. Let $g, a, b, c, \theta \in \mathcal{M}$.

(a)	$\mathsf{D}[Qg] = \Phi/g \rightarrow \mathsf{D}[g]$	if $Qg \neq \bot$
(b)	$\mathsf{D}[Qg] \supseteq \mathsf{D}[g]^+$	if $\mathsf{D}[g] \subseteq \Phi$ and $Qg \neq \bot$
(c)	$\mathsf{D}[Qg] = \mathsf{D}[g]^+$	if $\mathsf{D}[g]$ is self-extensional.
(d)	$\mathsf{D}[Ra\theta bc] = \cup_{z \in \mathsf{D}[\theta]} \mathsf{D}[a(b(cz/\theta))]$	if $Ra\theta bc \neq \bot$

Lemma 10.7.1(c) is used in Section 11 which proves UBT. Lemma 10.7.1(b) is used in Section 12 which proves LBT.

Note that the definition of Q gives $Qg \neq \bot \Leftrightarrow g \neq \bot$.

Proof of 10.7.1

(a) $Qg \neq \bot$ gives $!g = \mathsf{T}$. We have

$y \in \mathsf{D}[Qg]$	
$\Leftrightarrow Qgy = \mathsf{T}$	Definition of D
$\Leftrightarrow \ddot{\forall} z.\, g(y(z/g)) = \mathsf{T}$	$!g = \mathsf{T}$ and the definition of Q
$\Leftrightarrow \forall z {\in} \Phi \colon g(y(z/g)) = \mathsf{T}$	Properties of $\ddot{\forall}$
$\Leftrightarrow \forall z {\in} \Phi \colon y(z/g) \in \mathsf{D}[g]$	Definition of D
$\Leftrightarrow y \in \Phi/g \rightarrow \mathsf{D}[g]$	Definition of Φ/g and $\rightarrow$

(b) Follows from (a) and Lemma 10.5.5(a)

(c) From $\mathsf{D}[g]$ self-extensional we have $\mathsf{D}[g] \neq \emptyset$, so $g \neq \bot$ and $Qg \neq \bot$. Now (c) follows from (a) and Lemma 10.6.4.

(d) $Ra\theta bc \neq \bot$ and the definition of R gives $\theta c = \mathsf{T}$ and $R_1 a\theta bc = \mathsf{T}$. Now:

$y \in \mathsf{D}[Ra\theta bc]$	
$\Leftrightarrow Ra\theta bcy = \mathsf{T}$	Definition of D
$\Leftrightarrow R_0 a\theta bcy = \mathsf{T}$	$\theta c = \mathsf{T}$, $R_1 a\theta bc = \mathsf{T}$, and definition of R.
$\Leftrightarrow \big(\mathsf{E}z.\,(\theta z : a(b(cz/\theta))y)\big) = \mathsf{T}$	Definition of R_0
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon \big(\theta z : a(b(cz/\theta))y = \mathsf{T}\big)$	Properties of E
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon \big(\theta z = \mathsf{T} \wedge a(b(cz/\theta))y = \mathsf{T}\big)$	Properties of guards
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon \big(z \in \mathsf{D}[\theta] \wedge a(b(cz/\theta))y = \mathsf{T}\big)$	Definition of $\mathsf{D}[\theta]$
$\Leftrightarrow \exists z{\in}\mathsf{D}[\theta]\colon \big(a(b(cz/\theta))y = \mathsf{T}\big)$	Trivial
$\Leftrightarrow \exists z{\in}\mathsf{D}[\theta]\colon \big(y \in \mathsf{D}[a(b(cz/\theta))]\big)$	Definition of D
$\Leftrightarrow y \in \cup_{z\in\mathsf{D}[\theta]}\mathsf{D}[a(b(cz/\theta)))]$	Trivial

11. Proof of the Upper Bound Theorem (UBT)

Recall that UBT states that $\psi \preceq_{\mathcal{M}} \phi$ (c.f. Theorem 11.3.1). In this section we need that σ is inaccessible (but not necessarily minimal), that $\mathcal{M}$ is any $\kappa\sigma$-expansion where $\kappa > \sigma$ is regular and that Y acts as $\mathsf{Y}_{\mathrm{Tarski}}$. We will use repeatedly without mention the fact that application is monotonic w.r.t. $\preceq_{\mathcal{M}}$.

11.1. Restriction and step maps

We shall say that $g \in \mathcal{M}$ is a *step map* if $x \mapsto gx \in \mathcal{M}{\to}\mathcal{M}$ is a κ-step function in the sense of Definition 6.6.2. For convenience we drop κ in "step map" and "step chain" below. For all $g, h \in \mathcal{M}$ we shall say that g is a *restriction* of h if $\forall a{\in}\mathcal{M}\colon ga = \bot \vee ga = ha$. If h is a step map and g is a restriction of h then obviously g is also a step map. Now define

$$g \trianglelefteq h \Leftrightarrow \forall a, b{\in}\mathcal{M}\colon (a \preceq_{\mathcal{M}} b \wedge ga \neq \bot \Rightarrow ga = hb)$$

Fact 11.1.1.

(a) If h is a step map then $g \trianglelefteq h$ iff g is a restriction of h.
(b) $g \trianglelefteq g$ iff g is a step map.
(c) $g \trianglelefteq h \wedge h \trianglelefteq k \Rightarrow g \trianglelefteq k$.

For ordinals $\alpha \leq \kappa$ we say that $(g_\beta)_{\beta\in\alpha}$ is a *step chain* if $g_\gamma {\trianglelefteq} g_\beta$ for all $\gamma \leq \beta < \alpha$. In particular, $g_\beta \trianglelefteq g_\beta$ implies that the elements of a step chain are step maps.

Lemma 11.1.2. Suppose $(g_\beta)_{\beta\in\alpha}$ is a step chain and has a supremum g w.r.t. $\preceq_{\mathcal{M}}$. We have:

(a) $g_\beta \trianglelefteq g$ for all $\beta \in \alpha$.
(b) If $g_\beta \trianglelefteq h$ for all $\beta \in \alpha$ then $g \trianglelefteq h$.

Proof of 11.1.2

(a) Assume $a \preceq_{\mathcal{M}} b$ and $g_\beta a \neq \bot$. We shall prove $g_\beta a = gb$. Now $g_\beta a = g_\beta b$ since g_β is a step map. Furthermore, for all $\gamma \in \alpha$, $g_\gamma b = \bot \vee g_\gamma b = g_\beta b$ since $(g_\beta)_{\beta\in\alpha}$ is a step chain so $gb = g_\beta b$.

(b) Assume $a \preceq_{\mathcal{M}} b$ and $ga \neq \bot$. We shall prove $ga = hb$. Choose $\beta \in \alpha$ such that $g_\beta a \neq \bot$. Now $g_\gamma a = \bot \vee g_\gamma a = g_\beta a$ for all $\gamma \in \alpha$, so $g_\beta a = ga$. Furthermore, $g_\beta b = g_\beta a$ since g_β is a step map, so $g_\beta b = hb$ since $g_\beta \trianglelefteq h$.

Lemma 11.1.3. *$g \trianglelefteq h \wedge \theta \preceq_{\mathcal{M}} \phi \Rightarrow \bar{S}g\theta \trianglelefteq \bar{S}h\theta$.*

Proof of 11.1.3 Assume $a \preceq_{\mathcal{M}} b$ and $\bar{S}g\theta a \neq \bot$. We shall prove $\bar{S}g\theta a = \bar{S}h\theta b$. From $\bar{S}ga\theta \neq \bot$ and the definition of $\bar{S}$ we have $a \neq \bot$. If $a = \mathsf{T}$ then $\bar{S}k\theta a = Pa$ for all k so $\bar{S}g\theta a = Pa = \bar{S}h\theta a$. Now assume $a \in \mathcal{F}$.

From $\bar{S}g\theta a \neq \bot$ and $a \in \mathcal{F}$ we have $a\mathsf{T} \neq \bot$. We proceed by two cases: $a\mathsf{T} = \mathsf{T}$ and $a\mathsf{T} \in \mathcal{F}$.

Case 1. Assume $a\mathsf{T} = \mathsf{T}$. Now $\bar{S}k\theta a = Q(k(a\mathsf{F}))$ for all k. From $Q(g(a\mathsf{F})) = \bar{S}g\theta a \neq \bot$ we have $a\mathsf{F} \neq \bot$. Thus, from $g \trianglelefteq h$ we have $g(a\mathsf{F}) = h(b\mathsf{F})$ so $\bar{S}g\theta a = Q(g(a\mathsf{F})) = Q(h(b\mathsf{F})) = \bar{S}h\theta b$.

Case 2. Assume $a\mathsf{T} \in \mathcal{F}$. Now $\bar{S}k\theta a = Rk\theta(a\mathsf{T})(a\mathsf{F})$ for all k. From $Rg\theta(a\mathsf{T})(a\mathsf{F}) = \bar{S}g\theta a \neq \bot$ we have $\theta(a\mathsf{F}) = \mathsf{T}$ and $R_1 g\theta(a\mathsf{T})(a\mathsf{F}) = \mathsf{T}$. From the latter we have $g(a\mathsf{T}(a\mathsf{F}z/\theta)) \neq \bot$ for all $a \in \Phi$ and thus in particular for all $a \in \mathsf{D}[\theta]$ (since $\theta \preceq_{\mathcal{M}} \phi$). Thus, from $g \trianglelefteq h$ we have $g(a\mathsf{T}(a\mathsf{F}z/\theta)) = h(b\mathsf{T}(bFz/\theta))$, from which $\bar{S}g\theta a = \bar{S}h\theta b$ follows by the definitions of $\bar{S}$, R, R_1 and R_0.

Definition 11.1.4. For all ordinals $\alpha \leq \kappa$ and for all $\theta \in \mathcal{M}$ define $\bar{\theta}_\alpha$ by
(a) $\bar{\theta}_0 = \bot$
(b) $\bar{\theta}_{\alpha+1} = \bar{S}\bar{\theta}_\alpha\theta$
(c) $\bar{\theta}_\delta = \sup_{\alpha\in\delta}\bar{\theta}_\alpha$ for limit ordinals δ

By transfinite induction we have that $\bar{\theta}_\alpha \preceq_{\mathcal{M}} \mathsf{Y}\lambda f.\,\bar{S}f\theta$ and that all the sups exist (since $(\bar{\theta}_\alpha)_{\alpha\in\delta}$ is bounded).

Lemma 11.1.5. *If $\theta \preceq_{\mathcal{M}} \phi$ then $(\bar{\theta}_\alpha)_{\alpha\in\kappa}$ is a step chain.*

Proof of 11.1.5 From Lemma 11.1.3 we have $\bar{\theta}_\gamma \trianglelefteq \bar{\theta}_{\gamma'} \Rightarrow \bar{\theta}_{\gamma+1} \trianglelefteq \bar{\theta}_{\gamma'+1}$. We now prove that $(\theta_\alpha)_{\alpha\leq\beta+1}$ is a step chain by induction on $\beta \in \kappa$. The zero case follows from $\bot = \theta_0 \trianglelefteq \theta_1$. The successor case follows from $\bar{\theta}_\beta \trianglelefteq \bar{\theta}_{\beta+1} \Rightarrow \bar{\theta}_{\beta+1} \trianglelefteq \bar{\theta}_{\beta+2}$. For limit ordinals δ suppose $(\theta_\alpha)_{\alpha\leq\beta+1}$ is a step chain for all $\beta \in \delta$. Then $(\bar{\theta}_\alpha)_{\alpha<\delta}$ is a step chain. Then $\bar{\theta}_\alpha \trianglelefteq \bar{\theta}_\delta$ by Lemma 11.1.2(a). Then $\bar{\theta}_\alpha \trianglelefteq \bar{\theta}_{\alpha+1} \trianglelefteq \bar{\theta}_{\delta+1}$ so $\bar{\theta}_\delta \trianglelefteq \bar{\theta}_{\delta+1}$ by Lemma 11.1.2(b), so $(\theta_\alpha)_{\alpha\leq\delta+1}$ is a step chain.

11.2. Limited size

Lemma 11.2.1. $\mathsf{D}[Qg] \in \mathcal{O}_{<\sigma}(\Phi)$ *if* $\mathsf{D}[g] \in \mathcal{O}_{<\sigma}(\Phi)$

Proof of 11.2.1 Let $h = \lambda y.\,\mathsf{if}[\,gy\,,\,\mathsf{T}\,,\,\bot\,]$. Now $\mathsf{D}[h] = \mathsf{D}[g]$ and $\mathsf{D}[Qh] = \mathsf{D}[Qg]$. Furthermore, h is a characteristic map. Using Lemma 10.6.6, choose $K \in \mathcal{O}_{<\sigma}(\Phi)$ such that K is self-extensional and contains $\mathsf{D}[h]$ as a subset. Let k be the characteristic map of K. Since h and k are characteristic maps and $\mathsf{D}[h] \subseteq \mathsf{D}[k]$ we have $h \preceq_{\mathcal{M}} k$ so $Qh \preceq_{\mathcal{M}} Qk$ by monotonicity. Hence, $\mathsf{D}[Qh] \subseteq \mathsf{D}[Qk] = \mathsf{D}[k]^+ \in \mathcal{O}_{<\sigma}(\Phi)$ by Lemma 10.2.6(c) and 10.7.1(c). Thus,

$\mathsf{D}[Qg] = \mathsf{D}[Qh] \subseteq \Phi$. Furthermore, $\mathsf{D}[Qg]$ is open (c.f. Fact 10.2.4(a)) so it remains to prove that $\mathsf{D}[Qg]$ is essentially σ-small. This is trivial for $g = \bot$ so assume $g \neq \bot$. Now $Qg \neq \bot$.

Let $G \equiv \mathsf{D}[g]$. By Lemma 10.7.1(a) we have $\mathsf{D}[Qg] = \Phi/g{\to}G$. By hypothesis we have $G \in \mathcal{O}_{<\sigma}(\Phi)$, and by Lemma 10.5.5 we have $\Phi/g \subseteq G^{\bullet}$ and $\Phi/g \in \mathcal{P}_{<\sigma}(\mathcal{M}_c)$. Choose a σ-small $V \subseteq \Phi$ such that $G = {\uparrow}V$.

Let $h' \in \Phi/g{\to}V$. Now $h' \neq \bot$. If $h' \in \mathcal{F}$ and using Lemma 10.5.6 let $h'' \in \mathcal{F}$ be such that $h''x = h'(x/g)$ when $x \in {\uparrow}(\Phi/g)$ and $h''x = \bot$ otherwise. If $h' = \mathsf{T}$ let $h'' = \mathsf{T}$. Using Lemma 10.5.3(a,d) we have $h'' \preceq_{\mathcal{M}} h'$ and $h'' \in \Phi/g{\to}V$. So $\Phi/g{\to}V = {\uparrow}W$ where $W \equiv \{h'' \mid h' \in \Phi/g{\to}V\}$ is σ-small, and $\Phi/g{\to}G = \Phi/g{\to}{\uparrow}V = {\uparrow}(\Phi/g{\to}V) = {\uparrow}{\uparrow}W = {\uparrow}W$, which finishes the proof.

Lemma 11.2.2. Assume $f, a, b, c, v, \theta \in \mathcal{M}$, $\theta \preceq_{\mathcal{M}} \phi$ and $\forall x{\in}\mathcal{M}\colon \mathsf{D}[fx] \in \mathcal{O}_{<\sigma}(\Phi)$. We have:

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

Proof of 11.2.2

(a) Trivial.
(b) Follows from Lemma 11.2.1.
(c) If $Rf\theta bc = \bot$ then $\mathsf{D}[Rf\theta bc] = \emptyset \in \mathcal{O}_{<\sigma}(\Phi)$. Now assume $Rf\theta bc \neq \bot$. From the definition of R we have $\theta c = \mathsf{T}$ so $c \in \Phi$ since $\theta \preceq_{\mathcal{M}} \phi$. Hence, $c\Phi/\phi \subseteq \Phi^{\bullet}$ is σ-small by Lemma 10.5.5(b). Thus $b(c\Phi/\phi)$ and $K \equiv b(c\mathsf{D}[\theta]/\phi)$ are σ-small too. Hence, using Lemma 10.7.1(d), Lemma 10.2.6 and the hypothesis $\mathsf{D}[fx] \in \mathcal{O}_{<\sigma}(\Phi)$ we have $\mathsf{D}[Rf\theta bc] = \cup_{z\in\mathsf{D}[\theta]}\mathsf{D}[f(b(cz/\phi))] = \cup_{x\in K}\mathsf{D}[fx] \in \mathcal{O}_{<\sigma}(\Phi)$.
(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 (a-d).

In the following lemma, $(\bar{\theta}_\beta)_{\beta<\kappa}$ is the step chain produced from $\theta \in \mathcal{M}$ by Definition 11.1.4.

Lemma 11.2.3. *If $\theta \preceq_{\mathcal{M}} \phi$ and $\beta \leq \kappa$ then $\forall x{\in}\mathcal{M}\colon \mathsf{D}[\bar{\theta}_\beta x] \in \mathcal{O}_{<\sigma}(\Phi)$.*

Proof of 11.2.3 By induction on β. For $\beta = 0$ we have $\mathsf{D}[\bar{\theta}_\beta x] = \emptyset \in \mathcal{O}_{<\sigma}(\Phi)$. The successor case follows from Lemma 11.2.2(e). The limit case follows from Lemma 11.1.5.

11.3. Proof of UBT

In this section we prove UBT (i.e. $\psi \preceq_{\mathcal{M}} \phi$), and a refined form of it which sheds some light on the intuition behind the definitions of ψ and s (Definition 4.7.1).

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

Theorem 11.3.1 (Upper Bound Theorem/UBT). *$\psi \preceq_{\mathcal{M}} \phi$ holds in all $\kappa\sigma$-expansions where $\sigma < \kappa$ is any inaccessible ordinal.*

Proof of 11.3.1 Recall that $\psi \equiv \sqcup s$ where $s \equiv \mathsf{Y}S$ (Definition 4.7.1). From Lemma 11.2.3 and for all $\alpha \in \sigma$ and $x \in \mathcal{M}$, we have $\mathsf{D}[\bar{\phi}_\alpha x] \subseteq \Phi$ and so, using Fact 4.6.3, $\mathsf{D}[\sqcup\bar{\phi}_\alpha] = \cup_{x\in\mathcal{M}}\mathsf{D}[\bar{\phi}_\alpha x] \subseteq \Phi = \mathsf{D}[\phi]$. Since both $\sqcup\bar{\phi}_\alpha$ and ϕ are characteristic maps we have $\sqcup\bar{\phi}_\alpha \preceq_{\mathcal{M}} \phi$.

Now define s_α for all $\alpha \le \kappa$ by:

$$\begin{array}{lcll} s_0 & = & \bot & \\ s_{\alpha+1} & = & S s_\alpha & \\ s_\delta & = & \sup_{\alpha\in\delta} s_\alpha & \text{for limit ordinals } \delta \end{array}$$

By transfinite induction we have that $s_\alpha \preceq_{\mathcal{M}} \mathsf{Y}S$, that the sequence is increasing and that all the sups are defined (since $(s_\alpha)_{\alpha\in\delta}$ is bounded by $\mathsf{Y}S$). Furthermore, $s \equiv \mathsf{Y}S = s_\kappa$ since Y is $\mathsf{Y}_{\text{Tarski}}$ (c.f. Section 7.2).

We have $s_\alpha \preceq_{\mathcal{M}} \bar{\phi}_\alpha$ by transfinite induction: The zero and limit cases are trivial. We now assume $s_\alpha \preceq_{\mathcal{M}} \bar{\phi}_\alpha$ and prove $s_{\alpha+1} \preceq_{\mathcal{M}} \bar{\phi}_{\alpha+1}$. From $s_\alpha \preceq_{\mathcal{M}} \bar{\phi}_\alpha$ and monotonicity we have $\sqcup s_\alpha \preceq_{\mathcal{M}} \sqcup\bar{\phi}_\alpha$ so $\sqcup s_\alpha \preceq_{\mathcal{M}} \phi$. Hence, $s_{\alpha+1} \equiv S s_\alpha \equiv \bar{S} s_\alpha(\sqcup s_\alpha) \preceq_{\mathcal{M}} \bar{S}\bar{\phi}_\alpha\phi \equiv \bar{\phi}_{\alpha+1}$.

Now $\psi \equiv \sqcup s = \sqcup s_\kappa \preceq_{\mathcal{M}} \sqcup\bar{\phi}_\kappa \preceq_{\mathcal{M}} \phi$.

Theorem 11.3.2 below is a strengthening of UBT which we do not need but which captures some of the intuition behind the definitions of s and ψ.

Theorem 11.3.2 (Strong UBT). *For all $a \in \mathcal{M}$ we have* $\mathsf{D}[sa] \in \mathcal{O}_{<\sigma}(\Phi)$.

Proof of 11.3.2 We have $\psi \preceq_{\mathcal{M}} \phi$ by UBT so $\mathsf{D}[\bar{\psi}_\kappa a] \in \mathcal{O}_{<\sigma}(\Phi)$ by Lemma 11.2.3.

Define s_α like in the proof of UBT. We have $\sqcup s_\alpha \preceq_{\mathcal{M}} \sqcup s \equiv \psi$. We now prove $s_\alpha \preceq_{\mathcal{M}} \bar{\psi}_\alpha$ by transfinite induction on α. If $s_\alpha \preceq_{\mathcal{M}} \bar{\psi}_\alpha$ then $s_{\alpha+1} \equiv S s_\alpha \equiv \bar{S} s_\alpha(\sqcup s_\alpha) \preceq_{\mathcal{M}} \bar{S}\bar{\psi}_\alpha\psi \equiv \bar{\psi}_{\alpha+1}$. The zero and limit cases are trivial.

Now $\mathsf{D}[sa] = \mathsf{D}[s_\kappa a] \subseteq \mathsf{D}[\bar{\psi}_\kappa a] \in \mathcal{O}_{<\sigma}(\Phi)$.

12. Proof of the Lower Bound Theorem (LBT)

Recall that LBT states that $\phi \preceq_{\mathcal{M}} \psi$ (c.f. Theorem 12.4.3). As already mentioned, the proof of LBT uses UBT (in Lemma 12.2.3 and 12.4.1), the minimality of Y (in Lemma 12.1.2(b)) and that σ is the *first* inaccessible (in Lemma 12.4.2). The dependency on UBT and the minimality of Y should be seen as a convenience whereas the dependency on σ being the first inaccessible is essential, c.f. Section 10.1.

In the following, $\mathcal{M}$ can be any $\kappa\sigma$-expansion ($\kappa > \sigma$). We only require σ to be minimal when needed (in Lemma 12.4.2 and LBT itself).

12.1. Characteristic maps

Recall that we refer to elements of $\chi \equiv (\mathcal{M} \rightarrow \{\mathsf{T}, \bot\}) \cap \mathcal{F}$ as *characteristic maps*. For all $G \subseteq \mathcal{M}$ we have $\mathsf{D}[g] = G$ for at most one $g \in \chi$. We refer to that g, if any, as the *characteristic map of* G. As examples, ϕ and ψ are the characteristic maps of Φ and $\mathsf{D}[\psi]$, respectively. Define $\chi_\bot \equiv \chi \cup \{\bot\}$. Note that if $g \in \chi_\bot$ then $\mathsf{D}[g] = \mathrm{dom}[g] \equiv \{x \in \mathcal{M} \mid gx \neq \bot\}$. Also recall the following facts:

Fact 12.1.1. Let $g, h \in \mathcal{M}$. We have:
(a) $g \in \chi \Leftrightarrow g \preceq_{\mathcal{M}} \lambda x.\, \mathsf{T} \wedge g \neq \bot$
(b) $g \in \chi_\bot \Leftrightarrow g \preceq_{\mathcal{M}} \lambda x.\, \mathsf{T}$
(c) If $g \in \chi_\bot$ and $h \in \chi$ then $g \preceq_{\mathcal{M}} h \Leftrightarrow \mathsf{D}[g] \subseteq \mathsf{D}[h]$

Now recall the definition of s (Definition 4.7.1(b)). We have:

Lemma 12.1.2.
(a) $sa = \bar{S} s \psi a$
(b) $sa \preceq_{\mathcal{M}} \lambda x.\, \mathsf{T}$
(c) $sa \preceq_{\mathcal{M}} \psi$
(d) $sa \preceq_{\mathcal{M}} sb \Leftrightarrow \mathsf{D}[sa] \subseteq \mathsf{D}[sb]$ provided $sa \neq \lambda x.\, \bot$

Proof of 12.1.2
(a) By the definitions of S and ψ (Definition 4.7.1).
(b) Let $\mathsf{T}_1 \equiv \lambda x.\, \mathsf{T}$ and $\mathsf{T}_2 \equiv \lambda y.\, \mathsf{T}_1$. It is enough to prove
$$\forall a \in \mathcal{M} \colon S\mathsf{T}_2 a \preceq_{\mathcal{M}} \mathsf{T}_1 \qquad (1)$$
since if (1) holds then $S\mathsf{T}_2 = \lambda a.\, S\mathsf{T}_2 a \preceq_{\mathcal{M}} \lambda a.\, \mathsf{T}_1 \equiv \mathsf{T}_2$ so $\mathsf{Y}S \preceq_{\mathcal{M}} \mathsf{T}_2$ (since $\mathsf{Y} = \mathsf{Y}_{\mathrm{Tarski}}$). Hence, $sa = \mathsf{Y}Sa \preceq_{\mathcal{M}} \mathsf{T}_2 a = \mathsf{T}_1$.
It remains to prove (1). By Lemma 11.2.2 we have
$$S\mathsf{T}_2 a \in \{\bot, P, QT_1, R\mathsf{T}_2(\sqcup \mathsf{T}_2)(a\mathsf{T})(a\mathsf{F})\}$$
Since clearly $P \preceq_{\mathcal{M}} \mathsf{T}_1$, it only remains to check that the two last terms are smaller than T_1. From $Q\mathsf{T}_1 = !\mathsf{T}_1 : \lambda y.\, \ddot{\forall} z.\, \mathsf{T}_1(y(z/v)) = \lambda y.\, \ddot{\forall} z.\, \mathsf{T} = \lambda y.\, \mathsf{T} = \mathsf{T}_1$ we have $Q\mathsf{T}_1 \preceq_{\mathcal{M}} \mathsf{T}_1$. From $\mathsf{E}z.\, \mathcal{A} \preceq_{\mathcal{M}} \mathsf{T}$ for all terms $\mathcal{A}$ we have $R\mathsf{T}_2(\sqcup \mathsf{T}_2)bc = \cdots : \lambda y.\, \mathsf{E}z.\cdots \preceq_{\mathcal{M}} \mathsf{T}_1$.
(c) From $\psi = \sqcup s$ and Fact 4.6.3 we get $\mathsf{D}[sa] \subseteq \mathsf{D}[\psi]$. Thus $sa \preceq_{\mathcal{M}} \psi$ by Fact 12.1.1(b,c) and (b) of the present Lemma..
(d) If $sb \neq \bot$ then the lemma follows from (b) and Fact 12.1.1(c). The lemma is trivially true if $sb = \bot$. Actually, $sa \preceq_{\mathcal{M}} sb \Leftrightarrow \mathsf{D}[sa] \subseteq \mathsf{D}[sb]$ only fails for $(sa = \lambda x.\, \bot) \wedge (sb = \bot)$.

12.2. Analysis of s applied to pairs

We analyze here the shape of $\mathsf{D}[sa]$ when either $a = \mathsf{T}$ or a is a pair as defined below. UBT is used in the proof of Lemma 12.2.3(c) below.

Lemma 12.2.1.
(a) $s\mathsf{T} = P$
(b) $\mathsf{D}[s\mathsf{T}] = \{\mathsf{T}\}$

Proof of 12.2.1

(a) $s\mathsf{T} = \bar{S}s\psi\mathsf{T} = P$ by Lemma 12.1.2(a) and the definition of $\bar{S}$

(b) Follows from (a) and the definition of P.

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

Fact 12.2.2.

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

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

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

Lemma 12.2.3.

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

(b) $Q(sa) \neq \bot$ if $sa \neq \bot$

(c) $\mathsf{D}[s(\mathsf{T}{::}a)] \supseteq \mathsf{D}[sa]^{+}$ if $sa \neq \bot$ (Uses UBT)

Proof of 12.2.3

(a)

$$\begin{array}{lll} & s(\mathsf{T}{::}a) & \\ = & \bar{S}s\psi(\mathsf{T}{::}a) & \text{Lemma 12.1.2(a)} \\ = & Q(s((\mathsf{T}{::}a)\mathsf{F})) & \text{Definition of } \bar{S} \\ = & Q(sa) & \text{Fact 12.2.2(c)} \end{array}$$

(b) From $sa \neq \bot$ we have $!(sa) = \mathsf{T}$ so

$$\begin{array}{lll} & Q(sa) & \\ = & !(sa) : \lambda y.\cdots & \text{Definition of } Q \\ = & \mathsf{T} : \lambda y.\cdots & \text{From the assumption} \\ = & \lambda y.\cdots & \text{Definition of guards} \\ \neq & \bot & \text{Trivial} \end{array}$$

(c) We have $\mathsf{D}[sa] \subseteq \Phi$ by UBT (Theorem 11.3.1) and $Q(sa) \neq \bot$ by (b). Hence,

$$\begin{array}{lll} & \mathsf{D}[s(\mathsf{T}{::}a)] & \\ = & Q(sa) & \text{(a)} \\ \supseteq & \mathsf{D}[sa]^{+} & \text{Lemma 10.7.1(b)} \end{array}$$

Lemma 12.2.4. Assume $b \in \mathcal{F}$, $\psi c = \mathsf{T}$ and $\forall z{\in}\Phi\colon s(b(cz/\psi)) \neq \bot$

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

(b) $s(b{::}c) = \lambda y.\,\mathsf{E}z.\,(\psi z : s(b(cz/\psi))y)$

(c) $s(b{::}c) \neq \bot$

(d) $\mathsf{D}[s(b{::}c)] = \cup_{z\in\mathsf{D}[\psi]}\mathsf{D}[s(b(cz/\psi))]$

Proof of 12.2.4

(a)

$$\begin{array}{lll} & R_1 s\psi(b{::}c) & \\ = & \ddot{\forall} z.\,!(s((b{::}c)\mathsf{T}((b{::}c)\mathsf{F}z/\psi))) & \text{Definition of } R_1 \\ = & \ddot{\forall} z.\,!(s(b(cz/\psi))) & \text{Fact 12.2.2} \\ = & \mathsf{T} & \text{Third assumption} \end{array}$$

(b) $\quad s(b{::}c)$
$= \bar{S}s\psi(b{::}c)$ Lemma 12.1.2(a)
$= Rs\psi(b{::}c)$ Definition of $\bar{S}$
$= \psi c : R_1 s\psi(b{::}c) : R_0 s\psi(b{::}c)$ Definition of R and Fact 12.2.2
$= R_0 s\psi(b{::}c)$ $\psi c = \mathsf{T}$ and (a)
$= \lambda y.\, \mathsf{E}z.\, (\psi z : s(b(cz/\psi))y)$ Definition of R_0 and Fact 12.2.2

(c) Follows from (b)

(d) $\quad y \in \mathsf{D}[s(b{::}c)]$
$\Leftrightarrow s(b{::}c)y = \mathsf{T}$ Definition of D
$\Leftrightarrow \mathsf{E}z.\, (\psi z : s(b(cz/\psi))y) = \mathsf{T}$ (b)
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon (\psi z : s(b(cz/\psi))y) = \mathsf{T}$ Properties of E
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon \psi z = \mathsf{T} \wedge s(b(cz/\psi))y = \mathsf{T}$ Properties of guards
$\Leftrightarrow \exists z{\in}\mathcal{M}\colon z \in \mathsf{D}[\psi] \wedge y \in \mathsf{D}[s(b(cz/\psi))]$ Definition of D
$\Leftrightarrow y \in \cup_{z \in \mathsf{D}[\psi]} \mathsf{D}[s(b(cz/\psi))]$ Trivial

12.3. Further properties of projections

Recall that since / is an infix operator we have that ab/cd means $(ab)/(cd)$. Likewise, gH/k means $(gH)/k$ which equals $\{(gh)/k \mid h \in H\}$.

Lemma 12.3.1. *If $G = \mathsf{D}[g] \in \mathcal{O}_{<\kappa}(\mathcal{M})$ and $\emptyset \neq G \subseteq H \subseteq G^\circ$ then $\exists h{\in}G^+\colon G/g = hH/g$.*

Proof of 12.3.1 If $h \in G^+ \equiv G^\circ \to G$ then $hG^\circ \subseteq G$. From $G \subseteq H \subseteq G^\circ$ we have $hG/g \subseteq hH/g \subseteq hG^\circ/g \subseteq G/g$. It remains to find an $h \in G^+$ such that, furthermore, $G/g \subseteq hG/g$.

Let $k\colon G^\bullet \to G$ satisfy $k(x)/g = x$ for all $x \in G/g \subseteq G^\circ/g = G^\bullet$. For $x \notin G/g$ we merely require $k(x) \in G$ which is tenable since $G \neq \emptyset$.

Using Lemma 10.5.6 let $h \in \mathcal{F}$ satisfy $hx = k(x/g)$ when $x \in G^\circ$. Obviously, $h \in G^+$.

Assume $x \in G/g$. Let $y \in G$ satisfy $y/g = x$. By the definition of h and k we have $x = k(x)/g = k(y/g)/g = hy/g \in hG/g$; whence $G/g \subseteq hG/g$.

Lemma 12.3.2. *If $G = \mathsf{D}[g] \in \mathcal{O}_{<\kappa}(\mathcal{M})$, $G \subseteq G^\circ$ and $2 \leq_c G/g$ then $\mathcal{P}(G/g) \leq_c G^+/g$.*

Proof of 12.3.2 Let $a, b \in G$ satisfy $a/g \neq b/g$.

From Lemma 10.5.2 we have $G/g \subseteq G^\bullet$. For all $U \subseteq G/g$ define $k_U\colon G^\bullet \to \mathcal{M}$, $h_U \in \mathcal{F}$ and $i_U \in \mathcal{M}$ as follows using Lemma 10.5.6:

$$
\begin{array}{lcll}
k_U(x) & = & \left\{ \begin{array}{ll} a & \text{if } x \in U \\ b & \text{otherwise} \end{array} \right. & \text{for all } x \in G^\bullet \\
h_U x & = & k_U(x/g) & \text{for all } x \in G^\circ \\
i_U & = & h_U/g &
\end{array}
$$

Since $\{a, b\} \subseteq G$ we have $h_U \in G^+$ and $i_U \in G^+/g$ for all $U \subseteq G/g$.

Thus to prove $\mathcal{P}(G/g) \leq_c G^+/g$ there only remains to prove that $U \mapsto i_U$ is injective.

Now assume $U, V \subseteq G/g$ and $U \neq V$. Without loss of generality assume $U \setminus V \neq \emptyset$ and take $x \in G$ such that $x/g \in U \setminus V$. Thus $h_U x = a$ and $h_V x = b$.

Using the definition of / we have $i_U x = (h_U/g)x = gx : (h_U x/g) = \mathsf{T} : (a/g) = a/g$. Likewise, $i_V x = b/g$ so $i_U \neq i_V$ which ends the proof.

Let ϕ_α be the characteristic map for Φ_α. We have $\Phi_\alpha = \mathsf{D}[\phi_\alpha]$.

Lemma 12.3.3. *$\mathcal{P}(\Phi_\alpha/\phi_\alpha) \leq_c \Phi_{\alpha+1}/\phi_{\alpha+1}$ for all $\alpha \in \sigma$.*

Proof of 12.3.3 Φ_0/ϕ_0 is finite and Φ_1/ϕ_1 is infinite so the lemma holds for $\alpha = 0$. For all $\alpha > 0$ we have $\mathsf{T}, \mathsf{F} \in \Phi_\alpha$ and so $2 \leq_c \Phi_\alpha/\phi_\alpha$. Hence, $\mathcal{P}(\Phi_\alpha/\phi_\alpha) \leq_c \Phi_\alpha^+/\phi_\alpha = \Phi_{\alpha+1}/\phi_\alpha$ by Lemma 12.3.2. Furthermore, $\Phi_{\alpha+1}/\phi_\alpha \leq_c \Phi_{\alpha+1}/\phi_{\alpha+1}$ by Lemma 10.5.4 and Lemma 10.4.3(a).

Lemma 12.3.4. *$\alpha \leq_c \Phi_\alpha/\phi_\alpha$ for all $\alpha \in \sigma$.*

Proof of 12.3.4 By transfinite induction using Lemma 12.3.3 for the successor case and Lemma 10.5.4 for the limit case..

12.4. Proof of LBT

We use UBT twice in the proof of the following lemma.

Lemma 12.4.1. *Let α be a limit ordinal. For all $\gamma \in \alpha$ assume that $b_\gamma \in \mathcal{M}$ satisfies $\Phi_\gamma \subseteq \mathsf{D}[sb_\gamma]$. Suppose $a \in \mathcal{M}$ satisfies $\mathrm{cf}(\alpha) \leq_c \mathsf{D}[sa]/sa$. Then there exists a $b_\alpha \in \mathcal{M}$ such that $\Phi_\alpha \subseteq \mathsf{D}[sb_\alpha]$*

Proof of 12.4.1 The core idea is to take $b_\alpha \equiv c{::}d$ with c and d chosen as below, and to apply Lemma 12.2.4(d) to $\mathsf{D}[s(c{::}d)]$.

Let $g \equiv sa$, $G \equiv \mathsf{D}[g]$ and $H = \mathsf{D}[\psi]$.

From $sxy = \mathsf{T} \Rightarrow \psi y = \mathsf{T}$ we have $\mathsf{D}[sx] \subseteq \mathsf{D}[\psi]$. Thus, by UBT we have $\mathsf{D}[sx] \subseteq \mathsf{D}[\psi] \subseteq \Phi \subseteq \Phi^\circ \subseteq \mathsf{D}[sx]^\circ$ for all $x \in \mathcal{M}$.

From the hypotheses we have $\mathsf{D}[sa] \neq \emptyset$ and $\mathsf{D}[sb_\gamma] \neq \emptyset$, so $sa \neq \perp$ and $sb_\gamma \neq \perp$ for all $\gamma \in \alpha$.

Step 1: definition of d and properties. From $\mathsf{D}[sa] \subseteq \mathsf{D}[\psi] \subseteq \mathsf{D}[sa]^\circ$ we have $G \subseteq H \subseteq G^\circ$. Using Lemma 12.3.1 choose $d \in G^+$ such that $G/g = dH/g$. Using Lemma 12.2.3 (and hence once more UBT) we have $d \in G^+ \subseteq \mathsf{D}[s(\mathsf{T}{::}a)] \subseteq H$.

Step 2: definition of c and B and properties. Let $k' \in G/g \to \alpha$ be cofinal in α, and let $k \in G^\bullet \to \mathcal{M}$ be defined by $k(x) = b_{k'(x)}$ if $x \in G/g$ and $k(x) = \perp$ otherwise. By Lemma 10.5.6 there exists a $c \in \mathcal{F}$ such that $cx = k(x/g)$ when $x \in G^\circ$, $cx = \perp$ otherwise, and $cx = c(x/g)$ for all $x \in \mathcal{M}$. For such a c, using Lemma 10.5.3(e) we have $c(x/g) = c(x/\psi/g) = c(x/\psi)$. Finally, let $B \equiv c(G/g) = \mathrm{range}[k]$. We have $B \equiv c(G/g) = c(dH/g) = c(dH/\psi)$.

Step 3: computation of $\mathsf{D}[s(c{::}d)]$. We have $c \in \mathcal{F}$, $\psi d = \mathsf{T}$ and $s(c(dz/\psi)) = s(k(dz/\psi/g)) = sb_{k'(dz/g)} \neq \perp$ for all $z \in \Phi$, so by Lemma 12.2.4 we have $\mathsf{D}[s(c{::}d)] = \cup_{z \in H} \mathsf{D}[s(c(dz/\psi))] = \cup_{u \in B} \mathsf{D}[su]$.

Step 4: computation of Φ_α. We now prove $\Phi_\alpha = \cup_{\gamma\in\alpha}\Phi_\gamma = \cup_{x\in G/g}\Phi_{k'(x)} \subseteq \cup_{x\in G/g}\mathsf{D}[sb_{k'(x)}] = \cup_{u\in B}\mathsf{D}[su]$: the second equality uses that α is a limit ordinal, k' is cofinal in α and the sequence Φ_α is increasing; the inclusion uses the hypothesis and range$[k'] \subseteq \alpha$. Taking $b_\alpha = c{::}d$ we have $\Phi_\alpha \subseteq \cup_{u\in B}\mathsf{D}[su] = \mathsf{D}[s(c{::}d)] = \mathsf{D}[sb_\alpha]$ as required.

The following lemma is the one where we use that σ is not only inaccessible but is furthermore the smallest inaccessible.

Lemma 12.4.2. *Suppose $\sigma < \kappa$ is the smallest inaccessible ordinal. Let $\alpha \in \sigma$. For all $\gamma \in \alpha$ assume that $b_\gamma \in \mathcal{M}$ satisfies $\Phi_\gamma \subseteq \mathsf{D}[sb_\gamma]$. Then there exists a $b_\alpha \in \mathcal{M}$ such that $\Phi_\alpha \subseteq \mathsf{D}[sb_\alpha]$.*

Proof of 12.4.2 If $\alpha = 0$ take $b_\alpha = \mathsf{T}$. Then $\Phi_\alpha = \{\mathsf{T}\} = \mathsf{D}[sb_\alpha]$. If $\alpha = \beta + 1$ take $b_\alpha = \mathsf{T}{::}b_\beta$. Then $\Phi_\alpha = \Phi_\beta^+ \subseteq \mathsf{D}[sb_\beta]^+ \subseteq \mathsf{D}[sb_\alpha]$. Now assume that α is a limit ordinal.

Thanks to Lemma 12.4.1 we just have to find an $a \in \mathcal{M}$ such that $\mathrm{cf}(\alpha) \leq_c \mathsf{D}[sa]/sa$. Since $\alpha \in \sigma$ and since σ is the smallest inaccessible ordinal we have that α is not inaccessible so $\mathrm{cf}(\alpha) < \alpha \vee \exists\beta{\in}\alpha{:}\, \alpha \leq_c \mathcal{P}(\beta)$. We proceed by considering two cases: $\mathrm{cf}(\alpha) < \alpha$ and $\mathrm{cf}(\alpha) = \alpha$.

Case 1. Assume $\mathrm{cf}(\alpha) < \alpha$. Let $\beta \equiv \mathrm{cf}(\alpha)$ and $a = b_\beta$. From the hypothesis we have $\Phi_\beta \subseteq \mathsf{D}[sb_\beta]$, so $\Phi_\beta/\phi_\beta \leq_c \mathsf{D}[sb_\beta]/sb_\beta$ by Lemma 10.5.4. Furthermore, $\beta \leq_c \Phi_\beta/\phi_\beta$ by Lemma 12.3.4. Thus, $\mathrm{cf}(\alpha) \equiv \beta \leq_c \Phi_\beta/\phi_\beta \leq_c \mathsf{D}[sb_\beta]/sb_\beta = \mathsf{D}[sa]/sa$.

Case 2. Assume $\mathrm{cf}(\alpha) = \alpha$. Choose $\beta \in \alpha$ such that $\alpha \leq_c \mathcal{P}(\beta)$ and let $a = b_{\beta+1}$. Since α is a limit ordinal we have $\beta+1 < \alpha$. Thus, by the hypothesis, $\Phi_{\beta+1} \subseteq \mathsf{D}[sb_{\beta+1}]$. So $\mathrm{cf}(\alpha) = \alpha \leq_c \mathcal{P}(\beta) \leq_c \mathcal{P}(\Phi_\beta/\phi_\beta) \leq_c \Phi_{\beta+1}/\phi_{\beta+1} \leq_c \mathsf{D}[sb_{\beta+1}]/sb_{\beta+1} = \mathsf{D}[sa]/sa$ by Lemmas 10.5.4, 12.3.4 and 12.3.3.

Theorem 12.4.3 (Lower Bound Theorem/LBT). *$\phi \preceq_\mathcal{M} \psi$ holds in all $\kappa\sigma$-expansions provided $\sigma < \kappa$ is the first inaccessible ordinal.*

Proof of 12.4.3 From Lemma 12.4.2 we have $\Phi_\alpha \subseteq \mathsf{D}[sb_\alpha] \subseteq \mathsf{D}[\psi]$ for all $\alpha \in \sigma$ so $\Phi = \cup_{\alpha\in\sigma}\Phi_\alpha \subseteq \mathsf{D}[\psi]$. Thus $\phi \preceq_\mathcal{M} \psi$ since ϕ and ψ are the characteristic maps of Φ and $\mathsf{D}[\psi]$, respectively.

13. The consistency of MT

The main result of the present paper is that MT (as defined in Section 3.2 and Section 4) is consistent (also see Appendix D for a summary of MT). We formulate the main result thus:

Theorem 13.1 (Consistency of MT). *If σ is the first inaccessible ordinal and $\kappa > \sigma$ is regular then $\mathcal{M}_{\kappa\sigma} \models$ MT and $\mathcal{M}_{\kappa\sigma} \not\models \mathsf{T}{=}\mathsf{F}$.*

Proof of 13.1 From Theorem 7.5.2 we have $\mathcal{M}_{\kappa\sigma} \models \mathsf{Elem}+\mathsf{Elem}'+\mathsf{Exist}$. From Section 7.6 we have $\mathcal{M}_{\kappa\sigma} \models \mathsf{Mono} + \mathsf{Min} + \mathsf{Ext}$. From UBT (Theorem 11.3.1) and LBT (Theorem 12.4.3) we have the Definability Theorem (Theorem 10.1). From Theorem 7.9.2 and the Definability Theorem we have $\mathcal{M}_{\kappa\sigma} \models \mathsf{Quant}[\psi]$. Thus, $\mathcal{M}_{\kappa\sigma} \models$ MT. Finally, from Theorem 7.4.3 and Definition 7.4.4 we have $\mathcal{M}_{\kappa\sigma} \not\models \mathsf{T}{=}\mathsf{F}$.

Also the "economical" minor variant $\mathrm{MT}_{\mathrm{def}}$ of MT is consistent (both Section 3.4 and Appendix D mention how $\mathrm{MT}_{\mathrm{def}}$ differs from MT). We state that as a corollary:

Corollary 13.2 (Consistency of $\mathrm{MT}_{\mathrm{def}}$). *If σ is the first inaccessible ordinal and $\kappa > \sigma$ is regular then $\mathcal{M}_{\kappa\sigma} \models \mathrm{MT}_{\mathrm{def}}$.*

Proof of 13.2 Follows from Theorem 13.1 and $\mathsf{Y}_{\mathrm{Curry}} = \mathsf{Y}$ (Theorem 9.5.3).

14. Conclusion

We have now introduced the axiomatization MT of Map Theory and proved its consistency. To some extent, MT obsoletes the previous axiomatization [9, 4], which we call MT_0 in the present paper.

Furthermore, we have introduced the natural and minor variant $\mathrm{MT}_{\mathrm{def}}$ of MT and also proved the consistency of $\mathrm{MT}_{\mathrm{def}}$. This shows that it is a matter of taste whether or not Y and $\perp$ are included in the syntax.

What can be learned from Map Theory and its consistency proofs is that if we make Scott domains big enough, we can use them as universes for all of mathematics. Or, more precisely, if we make reflexive Scott domains big enough and use a suitable notion of continuity (κ-continuity), then we can interpret ZFC (including predicate calculus) in them via λ-calculus plus Hilbert's ε operator. Moreover, we can express this ability of these big reflexive Scott domains axiomatically (the equational theories MT, $\mathrm{MT}_{\mathrm{def}}$ and MT_0 being examples).

MT enhances MT_0 in three ways: First, it contains three new rules named Mono, Min and Ext. Second, it contains parallel or. Third, it contains a definition of wellfoundedness rather than axiomatizing wellfoundedness by a more or less random collection of rules. We elaborate on that in the following.

14.1. The Mono*,* Min *and* Ext *rules*

The Mono and Min rules express well known properties of Scott domains: all constructs are monotonic in the Scott order, and Tarski's fixed point operator generates minimal fixed points. Maybe somewhat surprising, Min replaces induction and transfinite induction (see Example 4.2.1 for induction on natural numbers).

Curiously, the Ext rule does not resemble anything the authors have ever come across, so it may be an entirely new axiom. The Ext rule corresponds to extensionality in set theory. Ext may be seen as a transtive version of Gen which says $\mathcal{A} = \mathcal{B} \vdash \lambda x.\, \mathcal{A} = \lambda x.\, \mathcal{B}$. Gen merely considers one level of lambdas. In contrast, Ext compares two maps by traversing their lambdas to an arbitrary depth. To some extent, one may think of Ext as structural induction on lambdas.

While the Ext rule seems entirely new, it does not seem to have any surprising consequences. The results in Example 4.3.1, 4.3.2 and 4.3.3 are representative examples of uses of Ext. As another example, and as a continuation of Example 4.2.1, one may show that the data type of natural numbers is a retraction: if we define the set of natural numbers (i.e. the first infinite ordinal number) by $\omega \equiv \lambda x. \mathsf{if}[\, x \,,\, \mathsf{T} \,,\, \lambda y. \omega(x\mathsf{T}) \,]$ then we can prove $\omega(\omega x) = \omega x$ using Ext.

14.2. Parallel or

The second enhancement of MT over MT_0 is that MT contains parallel or. As mentioned in Section 3.2, parallel or is neither needed for developing ZFC in MT nor convenient when programming. Nevertheless, it is nice to have parallel or since it allows to prove a full abstraction result (Theorem 3.8.2). That result in turn makes it easier to explain the notion of equality in Map Theory as is done in Section 3.8. Thus, full abstraction is a nice to have property when getting introduced to Map Theory in general and to its computational part in particular.

14.3. Definability of wellfoundedness

The third enhancement of MT over MT_0 is that it contains a definition of wellfoundedness rather than axiomatizing wellfoundedness by a more or less random collection of rules. That gives a number of advantages.

First, it is an interesting result in itself that one can define the characteristic function ψ of Φ in λ-calculus plus Hilbert's ε-operator. In ZFC one has to populate the universe by axioms like the power and union set axioms. One can do the same in Map Theory (as is done in MT_0). But one also has the choice just to postulate that λ-calculus plus Hilbert's ε-operator makes sense and then *define* a ψ which corresponds to the universe of ZFC.

Second, having a precise definition ψ of the notion of wellfoundedness inside Map Theory allows to investigate the notion inside the theory itself. In particular, applying the Min rule to the definition of ψ happens to produce a rule of transfinite induction [10] which in turn may be used for proving each and every axiom of ZFC.

Third, having a definition of wellfoundedness allows to build up a better intuition of what wellfoundedness means. In MT_0 one was forced to take a more or less random collection of wellfoundedness axioms for granted, but it remained unclear what wellfoundedness meant precisely. That was to some extent solved in [4] where the Generic Closure Property (GCP) and Strong Induction Property (SIP) provided a clearer picture of wellfoundedness. But MT takes that a step further by internalizing the notion of wellfoundedness in the axiomatization itself.

14.4. Does MT obsolete MT_0*?*

As mentioned, MT enhances MT_0 in three ways: First, it contains three new rules named Mono, Min and Ext. Second, it contains parallel or. Third, it contains a definition of wellfoundedness rather than axiomatizing wellfoundedness by a more or less random collection of rules.

The Mono, Min and Ext rules are uncontroversial. Mono and Min express well-known properties shared by all the non-syntactic models of untyped λ-calculus. It was not considered to include something expressing monotonicity and minimality in MT_0 because these concepts were not needed in MT_0 for developing ZFC, and MT_0 was the first demonstration of the fact that lambda calculus can be used as a foundation of mathematics. The Ext rule was not yet conceived at the time MT_0 was constructed, but it expresses some fundamental intuition behind Map Theory [9, Section 2.3], and as such is uncontroversial. Thus, even if one decided to go back to MT_0 one would probably keep Mono, Min and Ext and extend MT_0 by these rules.

The parallel or construct is also uncontroversial; and if one does not like it, one can just drop it. Having full abstraction can have a reassuring effect, but apart from that, parallel or can be in- or excluded according to taste. Thus, like was the case for Mono, Min and Ext, if one goes back to MT_0 then one may decide to keep parallel or and include it in MT_0.

Having a definition of wellfoundedness is more of a game changer. MT_0 leaves it open whether or not there exist inaccessible ordinals which are wellfounded in the sense of Map Theory, and MT_0 can consistently be extended to satisfy either. In contrast, MT is completely clear: inaccessible ordinals are non-wellfounded in the sense of MT. That is unimportant for the vast majority of mathematicians, but it is bad news for users of inaccessible ordinals, and could be a reason for them to prefer MT_0.

Note that MT does not say that inaccessible ordinals do not exist. It just says that they are not wellfounded. Actually, the class of all ordinals exists in MT and is in some sense the first inaccessible. But since it is not wellfounded, it is not in the range which quantifiers like ε and $\ddot{\forall}$ quantify over.

If one needs inaccessible ordinals which are wellfounded in the sense of Map Theory, then the easy solution is to go back to MT_0 and add an axiom saying that the needed ordinals exist. A more complicated but probably more viable solution would be to change the definition of ψ in MT to make the needed inaccessible ordinals wellfounded.

On the other hand one may also take the complete opposite point of view. If one sees undecidable propositions as a nuisance which should be kept to a minimum, it is nice that the question of existence of wellfounded inaccessible ordinals has a definite answer.

14.5. Further work

A key benefit of using lambda calculus as a foundation of mathematics is that it allows to use the same formalism for mathematics and computer programming. That could be particularly useful for proving mathematical results about computer programs, since theorems, proofs and programs could be expressed in the same framework. As an example, that could allow to treat numerical software in a setting where mathematical analysis is available.

To make use of that it would be convenient to have an implementation of the computational part of Map Theory. The implementation described in [11] is such an implementation. Further work could be to make that implementation more mature by enhancing such practicalities as its I/O capabilities, its responsivity to external interrupts, its garbage collection and so on.

Another obvious piece of further work would be to port the proof in [10] to MT. That proof is expressed in the axiomatization MT_c (cf. Section 1.4) whose consistency has never been proved. Porting the proof to MT plus proving the probably easy ¬SI would confirm Conjecture 2.2.3 which says that MT can interpret ZFC+¬SI.

Since a proof checker has already been implemented in the system described in [11], it would also be an obvious piece of further work to run the above mentioned ported proof through that proof checker. The proof in [10] has already been verified by other proof checkers, but it would be interesting to verify the ported proof in a proof checker which directly implements MT.

On a different note, one could try to add further rules to MT. As an example, one could imagine a rule saying that all maps are κ-continuous. Or, more precisely, κ-continuous for $\kappa = \sigma^+$ where σ^+ is the smallest regular ordinal greater than σ. That would express the continuity of maps but would also force κ to be σ^+. Expressing that maps are κ-continuous would be a step forward. Restricting κ to be σ^+ could be seen as a benefit or a drawback depending on taste.

15. Acknowledgement

The authors would like to thank the referees for their remarkable and unusually careful reviews which have enhanced the paper at all levels from the readability of individual formulations to the general presentation and structure of the paper.

A. Computational properties of canonical premodels

We now proceed to compare the observational, computational behavior of programs (i.e. closed, ε-free MT terms) with their semantics as defined by the canonical κ-expansions $\mathcal{M}_\kappa$ (cf. Definition 8.6.2).

Recall that $\mathcal{M}_\omega$ does not model all of MT. Modelling ε requires $\kappa > \sigma$ for an inaccessible σ, but modelling the other constructs just requires $\kappa \geq \omega$. Now assume $\kappa \geq \omega$.

Sections A.1 and A.2 introduce and define auxiliary constructs and terms needed for Section A.3. Section A.3 proves that all the compact (and prime) elements of $\mathcal{M}_\omega$, as well as some kinds of "analogues" in $\mathcal{M}_\kappa$, $\kappa > \sigma$, are definable using A, λ, T, if and parallel or (Corollary A.3.2). Section A.4 proves $\mathcal{M}_\omega \models \mathsf{E}_{\mathrm{pure}} = \mathsf{E}_{\mathrm{comp}}$ (as defined in Section 3.14). Section A.5 proves that $\mathcal{M}_\omega$ is computationally adequate for E-free MT and $\mathrm{MT}_{\mathrm{def}}$ programs, and leaves open whether this is true for $\kappa > \omega$. Section A.6 states soundness results and questions. Section A.7 proves that $\mathcal{M}_\omega$ is fully abstract for MT and $\mathrm{MT}_{\mathrm{def}}$. Section A.8 proves that this is false for $\mathcal{M}_\kappa$, $\kappa > \omega$.

Similar definability, adequacy and full abstraction results (case $\kappa = \omega$) were proved for diverse typed λ-calculi, starting from the paradigmatic paper of Plotkin on PCF [14]. The proofs, already non-trivial in the typed case, are here (untyped case) technically much more difficult.

For all MT, $\mathrm{MT_{def}}$ and $\mathrm{MT_0}$ 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 $\mathbf{C}_\omega = \mathcal{P}^{\mathrm{coh}}_{<\omega}(\mathbf{P}_\omega)$. Recall from Section 8.5 that if $p \in \mathbf{P}_\omega \subseteq \mathbf{P}$ then $\downarrow p \in \mathcal{M}_\kappa$ is a prime map and if $c \in \mathbf{C}_\omega \subseteq \mathbf{C}$ then $\downarrow c \in \mathcal{M}_\kappa$ is a compact map. For all $p \in \mathbf{P}_\omega$ 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:

$$\begin{array}{lcl} \underline{\mathcal{T}_p} & = & \downarrow p \\ \underline{\mathcal{T}_c} & = & \downarrow c \\ \underline{\chi_p} x & = & \left\{ \begin{array}{ll} \mathsf{T} & \text{if } \downarrow p \preceq_{\mathcal{M}} x \\ \bot & \text{otherwise} \end{array} \right. \\ \underline{\chi_c} x & = & \left\{ \begin{array}{ll} \mathsf{T} & \text{if } \downarrow c \preceq_{\mathcal{M}} x \\ \bot & \text{otherwise} \end{array} \right. \end{array}$$

To define the terms above, we also define a number of auxiliary concepts. For all $n \in \omega$ and for n-tuples $\bar{c} = \langle c_1, \ldots, c_n \rangle$ and $\bar{e} = \langle e_1, \ldots, e_n \rangle$ in $\mathbf{C}^n_\omega$ we define

$$\bar{c} \frown \bar{e} \Leftrightarrow c_1 \frown e_1 \wedge \cdots \wedge c_n \frown e_n$$

and

$$\downarrow \bar{c} = \langle \downarrow c_1, \ldots, \downarrow c_n \rangle$$

For $\bar{x} = \langle x_1, \ldots, x_n \rangle$ and $\langle \bar{y} = y_1, \ldots, y_n \rangle$ in $(\mathcal{M}_\kappa)^n$ we define

$$\bar{x} \preceq_{\mathcal{M}} \bar{y} \Leftrightarrow x_1 \preceq_{\mathcal{M}} y_1 \wedge \cdots \wedge x_n \preceq_{\mathcal{M}} y_n$$

For sets of n-tuples $u, v \in \mathcal{P}_{<\omega}(\mathbf{C}^n_\omega)$ we define

$$u \frown v \Leftrightarrow \exists \bar{c} {\in} u \exists \bar{e} {\in} v \colon \bar{c} \frown \bar{e}$$

For $\bar{x} = \langle x_1, \ldots, x_n \rangle$ let $\lambda \bar{x}.\, a$ and $a\bar{x}$ denote $\lambda x_1 \cdots x_n.\, a$ and $a x_1 \cdots x_n$, respectively.

For all $p, q \in \mathbf{P}_\omega$, $c, e \in \mathbf{C}_\omega$, $\bar{c}, \bar{e} \in \mathbf{C}^n_\omega$ and $u, v \in \mathcal{P}_{<\omega}(\mathbf{C}^n_\omega)$ for which $p \not\frown q$, $c \not\frown e$, $\bar{c} \not\frown \bar{e}$ and $u \not\frown v$, we are going to define $\mathrm{MT_{def}}$ programs δ_{pq}, δ_{ce}, $\delta_{\bar{c}\bar{e}}$ and δ_{uv} which satisfy:

$$\begin{array}{lcll} \underline{\delta_{pq}} x & = & \mathsf{T} & \text{if } \downarrow p \preceq_{\mathcal{M}} x \\ \underline{\delta_{pq}} x & = & \mathsf{F} & \text{if } \downarrow q \preceq_{\mathcal{M}} x \\ \underline{\delta_{ce}} x & = & \mathsf{T} & \text{if } \downarrow c \preceq_{\mathcal{M}} x \\ \underline{\delta_{ce}} x & = & \mathsf{F} & \text{if } \downarrow e \preceq_{\mathcal{M}} x \\ \underline{\delta_{\bar{c}\bar{e}}} \bar{x} & = & \mathsf{T} & \text{if } \downarrow \bar{c} \preceq_{\mathcal{M}} \bar{x} \\ \underline{\delta_{\bar{c}\bar{e}}} \bar{x} & = & \mathsf{F} & \text{if } \downarrow \bar{e} \preceq_{\mathcal{M}} \bar{x} \\ \underline{\delta_{uv}} \bar{x} & = & \mathsf{T} & \text{if } \exists \bar{c} {\in} u \colon \downarrow \bar{c} \preceq_{\mathcal{M}} \bar{x} \\ \underline{\delta_{uv}} \bar{x} & = & \mathsf{F} & \text{if } \exists \bar{e} {\in} v \colon \downarrow \bar{e} \preceq_{\mathcal{M}} \bar{x} \end{array}$$

Finally, for all $\bar{c} \in \mathbf{C}_\omega^n$ and $u \in \mathcal{P}_{<\omega}(\mathbf{C}_\omega^n)$ we are going to define $\mathrm{MT}_{\mathrm{def}}$ programs $\chi_{\bar{c}}$ and χ_u which satisfy:

$$
\begin{aligned}
\underline{\chi_{\bar{c}}}\bar{x} &= \begin{cases} \mathsf{T} & \text{if } \downarrow\!\bar{c} \preceq_{\mathcal{M}} \bar{x} \\ \bot & \text{otherwise} \end{cases} \\
\underline{\chi_u}\bar{x} &= \begin{cases} \mathsf{T} & \text{if } \exists \bar{c} \in u\colon \downarrow\!\bar{c} \preceq_{\mathcal{M}} \bar{x} \\ \bot & \text{otherwise} \end{cases}
\end{aligned}
$$

A.2. Parallel constructs

As a supplement to parallel or define parallel and:

$$x \;\&\; y = \ddot{\neg}(\ddot{\neg}x \parallel \ddot{\neg}y)$$

For finite sets $I = \{i_1, \ldots, i_n\}$ and $\mathrm{MT}_{\mathrm{def}}$ programs a_i, $i \in I$, define the $\mathrm{MT}_{\mathrm{def}}$ programs $\sum_{i \in I} a_i$ and $\prod_{i \in I} a_i$ by

$$
\begin{aligned}
\textstyle\sum_{i \in I} a_i &= \mathsf{F} \parallel a_{i_1} \parallel \cdots \parallel a_{i_n} \\
\textstyle\prod_{i \in I} a_i &= \mathsf{T} \;\&\; a_{i_1} \;\&\; \cdots \;\&\; a_{i_n}
\end{aligned}
$$

where the order $i_1, \ldots, i_n$ of elements of I is chosen in some arbitrary, fixed way.

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

For all $p, q \in \mathbf{P}_\omega$, $c, e \in \mathbf{C}_\omega$, $n \in \omega$, $\bar{c}, \bar{e} \in \mathbf{C}_\omega^n$ and $u, v \in \mathcal{P}_{<\omega}(\mathbf{C}_\omega^n)$ we define the following $\mathrm{MT}_{\mathrm{def}}$ programs by induction on the set rank of $p, q, c, e, \bar{c}, \bar{e}, u, v$:

$$
\begin{aligned}
\mathcal{T}_t &= \mathsf{T} \\
\mathcal{T}_f &= \lambda x.\, \bot_{\mathrm{Curry}} \\
\chi_t &= \lambda x.\, \mathsf{if}[\, x \,,\, \mathsf{T} \,,\, \bot_{\mathrm{Curry}} \,] \\
\chi_f &= \lambda x.\, \mathsf{if}[\, x \,,\, \bot_{\mathrm{Curry}} \,,\, \mathsf{T} \,] \\
\delta_{tp} &= \lambda x.\, \mathsf{if}[\, x \,,\, \mathsf{T} \,,\, \mathsf{F} \,] && \text{if } p \neq t \\
\delta_{pt} &= \lambda x.\, \mathsf{if}[\, x \,,\, \mathsf{F} \,,\, \mathsf{T} \,] && \text{if } p \neq t \\
\mathcal{T}_{\langle c,p \rangle} &= \lambda x.\, \mathsf{if}[\, \chi_c x \,,\, \mathcal{T}_p \,,\, \bot_{\mathrm{Curry}} \,] \\
\chi_{\langle c,p \rangle} &= \lambda x.\, \chi_p(x\mathcal{T}_c) \\
\chi_c &= \lambda x.\, \mathsf{if}[\, \textstyle\prod_{p \in c} \chi_p x \,,\, \mathsf{T} \,,\, \bot_{\mathrm{Curry}} \,] \\
\chi_{\bar{c}} &= \lambda \bar{x}.\, \chi_{c_1} x_1 \;\&\; \cdots \;\&\; \chi_{c_n} x_n \\
\chi_u &= \lambda \bar{x}.\, \textstyle\sum_{\bar{c} \in u} \chi_{\bar{c}} x_1 \cdots x_n \\
\delta_{c,e} &= \lambda x.\, \textstyle\prod_{p \in c} \sum_{q \in e} \delta_{pq} x && \text{if } c \not\supset e \\
\delta_{u,v} &= \lambda \bar{x}.\, \textstyle\sum_{\bar{c} \in u} \prod_{\bar{e} \in v} \delta_{\bar{c}\bar{e}} \bar{x} && \text{if } u \not\supset v
\end{aligned}
$$

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

$$\delta_{\langle c,p \rangle \langle e,q \rangle} = \lambda x.\, \delta_{pq}(x\mathcal{T}_{c \cup e})$$

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

For $\langle c_1, \cdots, c_n\rangle \not\supset \langle e_1, \ldots, e_n\rangle$ define

$$\delta_{\bar{c}\bar{e}} = \lambda \bar{x}.\, \delta_{c_i e_i} x_i$$

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

To define $\mathcal{T}_c$, recall the definition of $\langle \bar{c}, p\rangle$ from Section 8.4 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:

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

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 $p, q \in \mathbf{P}_\omega$, $c, e \in \mathbf{C}_\omega$, $\bar{c}, \bar{e} \in \mathbf{C}^n_\omega$ and $u, v \in \mathcal{P}_{<\omega}(\mathbf{C}^n_\omega)$ satisfy $p \not\supset q$, $c \not\supset e$, $\bar{c} \not\supset \bar{e}$ and $u \not\supset v$, respectively. Under these conditions, $\mathcal{T}_p$, $\mathcal{T}_c$, χ_p, χ_c, $\chi_{\bar{c}}$, χ_u, δ_{pq}, δ_{ce}, $\delta_{\bar{c}\bar{e}}$ and δ_{uv} have the properties stated in Section A.1.*

Proof. By induction on α we have that the theorem holds for all p, q, c, e, $\bar{c}$, $\bar{e}$, u and v of set rank less than α. ☐

Corollary A.3.2. *For all $p \in \mathbf{P}_\omega$ 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 $\underline{\mathcal{T}_c} = \downarrow c$.*

Let $\mathsf{C}_1, \ldots, \mathsf{C}_8$ be the combinators defined in Section 3.6 where C_1 and C_2 are the usual S and K combinators, respectively. We refer to terms built up from these combinators and functional application as MT combinator programs. We refer to the 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 $\bot$ 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 all $p \in \mathbf{P}_\omega$ 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 $\underline{\mathcal{T}'_c} = \downarrow c$.*

Of course Corollary A.3.2 and Corollary A.3.3 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 3.14:

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

Proof of A.4.1 Both E_{pure} and E_{comp} are characteristic functions. They satisfy

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

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. $\mathcal{M}_\omega \models \mathsf{E}a = \mathsf{E}_{\text{pure}}\, a = \mathsf{E}_{\text{comp}}\, a$.

A.5. Computational adequacy

Recall the notions of $\mathcal{N}_t$, $\mathcal{N}_f$ and $\mathcal{N}_\perp$ from Section 3.8.

Definition A.5.1. *$\mathcal{M}$ is* computationally adequate *for a set $\mathcal{T}$ of* MT_0*,* MT_{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*,* MT_{def} *and* MT*, respectively.*

As we shall see in a moment, $\mathcal{M}_\kappa$ is computationally adequate for MT_0 programs, for E-free MT_{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 $\mathcal{M} \models a = \perp$ (cf. Section 7.4). 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

$$a \in \mathcal{N}_t \quad \Leftarrow \quad \mathcal{M} \models a = \mathsf{T}$$
$$a \in \mathcal{N}_f \quad \Leftarrow \quad \mathcal{M} \models a = \lambda x.\, ax$$

follows trivially. The notion of computational adequacy of a model, as well as the notion of full abstraction, were introduced by Plotkin in [14] (for a paradigmatic simply typed lambda calculus called PCF). The definition of computational adequacy given above is equivalent to the one in [14] 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.

Theorem B.0.2 of [4] 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 of Theorem B.0.2 in [4] with the following two modifications. First, one has to include parallel or at the relevant places. Second, the proof of Lemma B.0.4 of [4], which is by structural induction, has one more case, namely the 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

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

which we prove in the following. For all terms g let $\tilde{\mathsf{Y}}g$ be the term

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

where x is chosen such that x is not free in g. Here, $\tilde{\mathsf{Y}}$ is a term function, i.e. a function from terms to terms, and $\tilde{\mathsf{Y}}g$ denotes application of the term function $\tilde{\mathsf{Y}}$ to the term g. In contrast, $\mathsf{Y}g$ denotes the term Y applied to the term g using the application operation of MT.

Since $\mathcal{M}_\kappa$ is canonical we have $\mathcal{M}_\kappa \models \perp = \perp_{\mathrm{Curry}}$ and $\mathcal{M}_\kappa \models \mathsf{Y}g = \tilde{\mathsf{Y}}g$.

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

For each E-free MT program b, $[b]$ is an E-free $\mathrm{MT}_{\mathrm{def}}$ program. Define $b \xrightarrow{1} c$ as in Section 3.5 and 3.6. We have:

$$
\begin{array}{llll}
\bot & \xrightarrow{1} & \bot & \text{in MT} \\
\mathsf{Y}g & \xrightarrow{1} & g(\mathsf{Y}g) & \text{in MT} \\
\bot_{\mathrm{Curry}} & \xrightarrow{1} & \bot_{\mathrm{Curry}} & \text{in } \mathrm{MT}_{\mathrm{def}} \\
\tilde{\mathsf{Y}}g & \xrightarrow{1} & g(\tilde{\mathsf{Y}}g) & \text{in } \mathrm{MT}_{\mathrm{def}} \\
[\bot] & \xrightarrow{1} & [\bot] & \text{in } \mathrm{MT}_{\mathrm{def}} \\
[\mathsf{Y}g] & \xrightarrow{1} & [g(\mathsf{Y}g)] & \text{in } \mathrm{MT}_{\mathrm{def}}
\end{array}
$$

In general, if $b \xrightarrow{1} c$ in MT then $[b] \xrightarrow{1} [c]$ in $\mathrm{MT}_{\mathrm{def}}$ by structural induction on 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, $a \xrightarrow{1} b$ holds for at most one b (up to renaming of bound variables). Let $a_1, a_2, \ldots$ be the unique longest finite or infinite sequence such that $a \xrightarrow{1} a_1 \xrightarrow{1} a_2 \xrightarrow{1} \cdots$ in MT. By Theorem A.5.3, the sequence $[a] \xrightarrow{1} [a_1] \xrightarrow{1} [a_2] \xrightarrow{1} \cdots$ is finite and ends with a term in root normal form (i.e. is T or an abstraction). Hence, $a \xrightarrow{1} a_1 \xrightarrow{1} a_2 \xrightarrow{1} \cdots$ has the same property, so $a \in \mathcal{N}_t \cup \mathcal{N}_f$ which was to be shown.

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 Theorem A.5.4. Define $a \xrightarrow{3} d \Leftrightarrow \exists b, c\colon a \xrightarrow{1} b \xrightarrow{1} c \xrightarrow{1} d$ and let $a \xrightarrow{*} b$ be the transitive closure of $a \xrightarrow{1} b$.

Recall the definition of $\mathsf{E}_{\mathrm{comp}}$ from Section 3.14. The definition is recursive and thus implicitly uses Y. Now define

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

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

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

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

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

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 $a \xrightarrow{1} a_1 \xrightarrow{1} a_2 \xrightarrow{1} \cdots$ be the unique reduction sequence for a. Now $[a] \xrightarrow{*} [a_1] \xrightarrow{*} [a_2] \xrightarrow{*} \cdots$ is finite by Theorem 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 3.8 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 all* 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 all* 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 $ca \in \mathcal{N}_\perp \Leftrightarrow cb \in \mathcal{N}_\perp$. 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) *Does $a =_\kappa b \Rightarrow a =_{\mathrm{obs}} b$ for all* MT *programs a and b and for $\kappa > \omega$?*
(b) *Does $a =_\kappa b \Rightarrow a =_{\mathrm{obs}} b$ for all* $\mathrm{MT}_{\mathrm{def}}$ *programs a and b and for $\kappa > \omega$?*

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 all 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

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

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 $a = \mathsf{E}_{\mathrm{pure}} = \lambda x.\, \mathsf{E}x$. Take $b = \mathsf{E}_{\mathrm{comp}}$ so that $b = \lambda x.\, (x\mathsf{C}_1 \parallel \cdots \parallel x\mathsf{C}_8 \parallel b\lambda u.\, b\lambda v.\, x(uv))$ (cf. Section 3.6 and 3.14).

We first prove $a =_{\mathrm{obs}} b$. According to Theorem A.7.2 it 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 every $n \in \mathbb{N}$. We have $t_i \supset\!\!\!\subset t_j \Leftrightarrow i = j$. Now let $g : \mathbb{N} \to \mathbb{N}$ be non-computable. Let $Q \equiv \{\langle \{t_i\}, t_{g(i)} \rangle \mid i \in \mathbb{N}\}$, $q = {\downarrow}Q$ 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, we cannot have $px =_\kappa \mathsf{T}$ for any program x since g is non-computable, so $bp \neq_\kappa \mathsf{T}$ proving $a \neq_\kappa b$. $\Box$

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 $p \in \mathbf{P}_\omega$. Now 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 about the definition of b which is long and technical.

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

Recall that $(x{::}y)\mathsf{T} = x$ and $(x{::}y)\mathsf{F} = y$. Let $(x_1, \ldots, x_n)$ be shorthand for $x_1{::}\cdots{::}x_n{::}\mathsf{T}$. We 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 $\mathsf{T}{::}c{::}p$ means $\mathsf{T}{::}(c{::}p)$. 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 $\lambda x.\, \mathsf{if}[\, x\,,\, \mathsf{T}\,,\, \lambda y.\, \bot\,]$.

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 $\asymp_0$ and $\asymp_1$ on $\mathbf{P}_\omega$ and $\mathbf{C}_\omega$, respectively:

$$\begin{array}{lcl}
p \asymp_0 q & \equiv & \mathsf{if}[\, p\,,\, \mathsf{if}[\, q\,,\, \mathsf{T}\,,\, \mathsf{F}\,]\,,\, \mathsf{if}[\, q\,,\, \mathsf{F}\,, \\
& & \quad \mathsf{if}[\, p\mathsf{F}\,,\, \mathsf{T}\,,\, \mathsf{if}[\, q\mathsf{F}\,,\, \mathsf{T}\,, \\
& & \qquad p\mathsf{FT} \asymp_1 q\mathsf{FT} \ddot{\Rightarrow} p\mathsf{FF} \asymp_0 q\mathsf{FF}]]]] \\
c \asymp_1 e & \equiv & \bigwedge_{p \in c} \bigwedge_{q \in e} p \asymp_0 q
\end{array}$$

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 in $\mathbf{P}_\omega$, $\mathbf{C}_\omega$ and $\mathbf{C}_\omega^{<\omega}$, respectively:

$$\begin{array}{lcl}
\chi_{\mathbf{P}_\omega} p & \equiv & \mathsf{if}[\, p\mathsf{F}\,,\, \mathsf{T}\,,\, \chi_{\mathbf{C}_\omega}(p\mathsf{FT}) \ddot{\wedge} \chi_{\mathbf{P}_\omega}(p\mathsf{FF})\,] \\
\chi_{\mathbf{C}_\omega} c & \equiv & c \asymp_1 c \ddot{\wedge} \bigwedge_{p \in c} \chi_{\mathbf{P}_\omega} p \\
\chi_{\mathbf{C}_\omega^{<\omega}} \bar{c} & \equiv & \mathsf{if}[\, \bar{c}\,,\, \mathsf{T}\,,\, \chi_{\mathbf{C}_\omega}(\bar{c}\mathsf{T}) \ddot{\wedge} \chi_{\mathbf{C}_\omega^{<\omega}}(\bar{c}\mathsf{F})\,]
\end{array}$$

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_u of Section A.3 translate into the following:

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

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

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

The discriminator constructs δ_{pq}, δ_{ce}, $\delta_{\bar{c}\bar{e}}$ and δ_{uv} of Section A.3 translate into the following:

$$
\begin{array}{lcl}
\delta_0 pqx & \equiv & \mathsf{if}[\,p\,,\,\mathsf{if}[\,x\,,\,\mathsf{T}\,,\,\mathsf{F}\,]\,, \\
& & \quad \mathsf{if}[\,p\mathsf{F}\,,\,\mathsf{if}[\,x\,,\,\mathsf{F}\,,\,\mathsf{T}\,]\,, \\
& & \qquad \delta_0(p\mathsf{FF})(q\mathsf{FF})(x(\mathcal{T}_1(p\mathsf{FT}\cup q\mathsf{FT})))]] \\
\delta_1 cex & \equiv & \prod_{p\in c}\sum_{q\in e}\delta_0 pqx \\
\delta_2 \bar{c}\bar{e}\bar{x} & \equiv & \mathsf{if}[\,\bar{c}\mathsf{T}\ \bigcirc_1\ \bar{e}\mathsf{T}\,,\,\delta_2(\bar{c}\mathsf{F})(\bar{e}\mathsf{F})(\bar{x}\mathsf{F})\,,\,\delta_1(\bar{c}\mathsf{T})(\bar{e}\mathsf{T})(\bar{x}\mathsf{T})\,] \\
\delta_3 uv\bar{x} & \equiv & \sum_{\bar{c}\in u}\prod_{\bar{e}\in v}\delta_2\bar{c}\bar{e}\bar{x}
\end{array}
$$

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 $\mathcal{T}_1 c$ 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:

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

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

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

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 $\bar{y}$:

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

As an example, if $b x y_1 \cdots y_{n-1} \notin \{\mathsf{T}, \bot\}$ and $x y_1 \cdots y_n = \mathsf{T}$ then

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

Thus, in the situation above, $b x y_1 \cdots y_n$ 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. Conjectures on the strength of MT versus MT$_0$

We now continue the discussion of the strength of MT and MT$_0$ initiated in Section 2.2. The prerequisites for reading the present appendix are included in Sections 1.1, 1.2, 2.1 and 2.2.

MT is very likely stronger than MT$_0$. Indeed, MT$_0$ can prove neither SI $= \mathsf{T}$ nor $(\neg\text{SI}) = \mathsf{T}$ since it can be consistently extended by either one. In contrast, $(\neg\text{SI}) = \mathsf{T}$ is conjectured to be provable in MT (Conjecture 2.2.3). Furthermore, MT can prove more pure lambda terms equivalent such as $F_2 = F_3$ (cf. Example 4.3.3) which we conjecture is not provable in MT$_0$. Furthermore, we conjecture the following:

Conjecture B.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 2.2.3 follows from Conjecture B.1 and Theorem 2.2.4. Conjecture B.1 is true if the ϕ-axioms of MT_0 are provable in MT with ψ replacing ϕ. Less support exists for Conjecture B.1 than for Conjecture 2.2.3.

C. On the necessity of minimality for proving UBT

We now restate Lemma 10.1.1 which states that the proof of UBT ($\psi \preceq_{\mathcal{M}} \phi$) needs that s is the minimal fixed point of S.

Lemma C.1. *Let $\sigma < \kappa$ be inaccessible and let $\mathcal{M}$ be any $\kappa\sigma$-expansion. There exists an $s' \in \mathcal{M}$ such that $Ss' = s'$ and $\mathsf{D}[\psi'] = \mathcal{M}$ where $\psi' = \sqcup s'$.*

Proof of C.1 To prove the lemma it is enough to find $s', u \in \mathcal{M}$ such that $Ss' = s'$ and $s'u = \lambda x.\,\mathsf{T}$, since then $\mathsf{D}[\psi'] = \cup_{a\in\mathcal{M}}\mathsf{D}[s'a] \supseteq \mathsf{D}[s'u] = \mathcal{M}$.

For any $C \subseteq \mathcal{M}$ let $\sup(C)$ denote the least upper bound of C (when such a one exists). The idea is to take $u \equiv \sup(B)$ for $B \equiv \{\lambda x_1 x_2 \cdots x_n.\,\bot \mid n \in \omega\} \equiv \{K_n\bot \mid n \in \omega\}$ where $K \equiv \lambda xy.\,x$, $K_0 \equiv \bot$ and $K_{n+1} \equiv KK_n$. And to define s' from u. Note that B has a sup because B is bounded (e.g. by any fixed point of K). We prove below that u is κ-compact and that we can produce an adequate s' from it. The proof has six steps, preceded by two more general lemmas.

Since $\mathcal{M}$ is a $\kappa\sigma$-expansion, it is in particular a κ-Scott domain and a κ-premodel. This in particular means that application is monotonic w.r.t. the κ-Scott order $\preceq_{\mathcal{M}}$ and that if $g, h \in \mathcal{F} \equiv \mathcal{M} \setminus \{\mathsf{T}, \bot\}$ and $gx \preceq_{\mathcal{M}} hx$ for all $x \in \mathcal{M}$ then $g \preceq_{\mathcal{M}} h$.

Lemma 1. Let $G \subseteq \mathcal{M}$. If $h = \sup(G) \in \mathcal{F}$ then $h\bot \equiv \sup(G\bot)$, where $G\bot \equiv \{g\bot \mid g \in G\}$.

Note that the hypothesis on h implies that G contains a non-$\bot$ element and does not contain T. The proof of the lemma is trivial if application commutes with all sups (which is for example true if $\mathcal{M}$ is canonical); the proof for the general case is a little tricky and will be given at the end.

Lemma 2. If $p \in \mathcal{M}$ is prime then $Kp = \lambda x.\,p$ is prime too.

Proof. Suppose $\lambda x.\,p \preceq_{\mathcal{M}} \sup(G)$ for some $G \subseteq \mathcal{M}$. We have $p = (\lambda x.\,p)\bot \preceq_{\mathcal{M}} \sup(G)\bot = \sup(G\bot)$ by monotonicity plus Lemma 1. Since p is prime we have $p \preceq_{\mathcal{M}} g\bot$ for some $g \in G$. Note that $g = \bot$ could occur only if $p = \bot$, in which case g could be replaced by any other element of G, so we can always take $g \in \mathcal{F}$. Now $p \preceq_{\mathcal{M}} gx$ for all $x \in \mathcal{M}$, whence $\lambda x.\,p \preceq_{\mathcal{M}} \lambda x.\,gx$ from which we have $\lambda x.\,p \preceq_{\mathcal{M}} g$ since $g = \lambda x.\,gx$ because $g \in \mathcal{F}$.

Step 1. Recall $B \equiv \{\lambda x_1 x_2 \cdots x_n.\,\bot \mid n \in \omega\} \equiv \{K_n\bot \mid n \in \omega\}$. We prove that $u \equiv \sup(B)$ is κ-compact and that $u \preceq_{\mathcal{M}} ux$ for all $x \in \mathcal{M}$: by Lemma 2, B is a countable set of primes, and hence a κ-small set of κ-compact elements (since $\kappa > \omega$). Hence its sup is κ-compact too. Now, $K_{n+1} \preceq_{\mathcal{M}} u$ implies

$K_n = \mathsf{K}_{n+1}x \preceq_{\mathcal{M}} ux$, so ux is an upper bound of B. Hence, $u \preceq_{\mathcal{M}} ux$ since u is the least one.

Step 2. Define $\tilde{s}$ by $\tilde{s}x = \lambda y.\mathsf{T}$ if $x \succeq_{\mathcal{M}} u$ and $\tilde{s}x = \bot$ otherwise. Such an $\tilde{s}$ exists because the corresponding function is κ-continuous (since u is κ-compact).

Step 3. Now $\theta \equiv \sqcup\tilde{s} = \lambda y.\mathsf{T}$. Proof: $\sqcup\tilde{s} \equiv \lambda y.\mathsf{E}x.\tilde{s}xy$ where $\mathsf{E}x.\tilde{s}xy = \mathsf{T}$, because $\tilde{s}xy = \mathsf{T}$ for $x = u$.

Step 4. $S\tilde{s}u = \lambda y.\mathsf{T}$. Proof: We have that $\lambda y.\mathsf{T}$ is maximal (because $\mathcal{M}$ is a premodel) so it is enough to prove $S\tilde{s}u \succeq_{\mathcal{M}} \lambda y.\mathsf{T}$. We have $S\tilde{s}u \equiv \bar{S}\tilde{s}(\sqcup\tilde{s})u \equiv \bar{S}\tilde{s}\theta u$. Using the definition of $\bar{S}$ and $ux \succeq_{\mathcal{M}} u$ we have $\bar{S}\tilde{s}\theta u \succeq_{\mathcal{M}} R\tilde{s}\theta uu$. From $\theta = \lambda y.\mathsf{T}$ we have $\theta u = \mathsf{T}$. From the definition of R_1 we have $R_1\tilde{s}\theta uu \equiv \big(\ddot{\forall}z.!(u(\cdots))\big) \succeq_{\mathcal{M}} \big(\ddot{\forall}z.!u\big) = \big(\ddot{\forall}z.\mathsf{T}\big) = \mathsf{T}$. From the definition of R_0 we have $R_0\tilde{s}\theta uu = \big(\lambda y.\mathsf{E}z.(\theta u{:}\,\tilde{s}(u(\cdots))y)\big) \succeq_{\mathcal{M}} \big(\lambda y.\mathsf{E}z.\tilde{s}uy\big) = \big(\lambda y.\mathsf{E}z.\mathsf{T}\big) = \lambda y.\mathsf{T}$. Thus, $R\tilde{s}\theta uu \equiv \big(\theta u : R_1\tilde{s}\theta uu : R_0\tilde{s}\theta uu\big) \succeq_{\mathcal{M}} \big(\mathsf{T} : \mathsf{T} : \lambda y.u\big) = \lambda y.\mathsf{T}$. In conclusion, $S\tilde{s}u = \bar{S}\tilde{s}\theta u \succeq_{\mathcal{M}} R\tilde{s}\theta uu \succeq_{\mathcal{M}} \lambda y.\mathsf{T}$ as required.

Step 5. $\tilde{s} \preceq_{\mathcal{M}} S\tilde{s}$. Proof: We have $\tilde{s}x = \lambda y.\mathsf{T} = S\tilde{s}u \preceq_{\mathcal{M}} S\tilde{s}x$ if $u \preceq_{\mathcal{M}} x$. Furthermore $\tilde{s}x = \bot \preceq_{\mathcal{M}} S\tilde{s}x$ if $u \not\preceq_{\mathcal{M}} x$, so $\tilde{s}x \preceq_{\mathcal{M}} S\tilde{s}x$ for all x. Hence, $\tilde{s} \preceq_{\mathcal{M}} S\tilde{s}$ as required.

Step 6. Define $s' \equiv f^{\kappa}(\tilde{s})$, where f is the κ-continuous function coded by S, so that $f(x) = Sx$ for all x. From $\tilde{s} \preceq_{\mathcal{M}} S\tilde{s}$ we have $f^0(\tilde{s}) \preceq_{\mathcal{M}} f^1(\tilde{s})$. Like in Lemma 7.2.1, $f^{\alpha}(\tilde{s}) \preceq_{\mathcal{M}} \lambda ay.\mathsf{T}$ is defined for all α and increasing in α, and $s' \equiv f^{\kappa}(\tilde{s})$ is a fixed point of S (though not minimal). From $\tilde{s} \preceq_{\mathcal{M}} s'$ we have $s'u = \lambda x.\mathsf{T}$ as required.

It only remains to prove Lemma 1:

Proof of Lemma 1. The proof only uses that $\mathcal{M}$ is a κ-premodel. From $h \in \mathcal{F}$ we have that G contains a non-$\bot$ element and does not contain T. Since G is bounded, $G\bot$ is also bounded. Hence, $a \equiv \sup(G\bot)$ exists. Moreover, $a \preceq_{\mathcal{M}} h\bot$, since $h\bot$ is an upper bound of $G\bot$ (by monotonicity). Now let $h' \in \mathcal{F}$ satisfy $h'\bot = a$ and $h'x = hx$ otherwise (the existence of h' follows from the fact that the corresponding function is easily seen to be κ-continuous). Now $g \preceq_{\mathcal{M}} h' \preceq_{\mathcal{M}} h$ because $gx \preceq_{\mathcal{M}} h'x \preceq_{\mathcal{M}} hx$ for all x, and $h, h' \in \mathcal{F}$ and $g = \bot$ or $g \in \mathcal{F}$. Hence, $h' = h$ (by minimality of h) and $h\bot = a$ as required.

D. Summary of MT

We now summarize MT as defined in Section 3.2 and Section 4. We also state how the minor variant $\mathrm{MT}_{\mathrm{def}}$ (cf. Section 3.4) differs from MT.

D.1. Syntax

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

For $\mathrm{MT}_{\mathrm{def}}$, $\bot$ and Y are omitted from the syntax.

D.2. Definitions

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

Elementary definitions

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

Quantifier definitions

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

The definition of ψ

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

D.3. Rules (i.e. axioms and inference rules)

Elementary rules (Elem)

Trans	$\mathcal{A} = \mathcal{B}; \mathcal{A} = \mathcal{C} \vdash \mathcal{B} = \mathcal{C}$
Sub	$\mathcal{A} = \mathcal{B}; \mathcal{C} = \mathcal{D} \vdash \mathcal{A}\mathcal{C} = \mathcal{B}\mathcal{D}$
Gen	$\mathcal{A} = \mathcal{B} \vdash \lambda x.\, \mathcal{A} = \lambda x.\, \mathcal{B}$
A1	$\mathsf{T}\mathcal{B} = \mathsf{T}$
A2 (β)	$(\lambda x.\, \mathcal{A})\mathcal{B} = \langle \mathcal{A} \mid x := \mathcal{B} \rangle$ if $\mathcal{B}$ is free for x in $\mathcal{A}$
A3	$\bot\mathcal{B} = \bot$
Rename (α)	$\lambda x.\, \langle \mathcal{A} \mid y := x \rangle = \lambda y.\, \langle \mathcal{A} \mid x := y \rangle$ if x is free for y in $\mathcal{A}$ and vice versa
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	$\langle \mathcal{A} \mid x := \mathsf{T} \rangle = \langle \mathcal{B} \mid x := \mathsf{T} \rangle;$ $\langle \mathcal{A} \mid x := \mathbf{1}x \rangle = \langle \mathcal{B} \mid x := \mathbf{1}x \rangle;$ $\langle \mathcal{A} \mid x := \bot \rangle = \langle \mathcal{B} \mid x := \bot \rangle \vdash$ $\mathcal{A} = \mathcal{B}$

Further elementary rules (Elem′)

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 (Mono) and minimality (Min)

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)

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 (Exist)

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

Quantification axioms ($\mathsf{Quant}[\psi]$)

ElimAll	$(\ddot{\forall} x.\, px) \wedge \psi y \rightarrow py$
Ackermann	$\varepsilon x.\, px = \varepsilon x.\, (\psi x \wedge px)$
StrictEpsilon	$\psi(\varepsilon x.\, px) = \ddot{\forall} x.\, !(px)$
StrictAll	$!(\ddot{\forall} x.\, px) = \ddot{\forall} x.\, !(px)$

E. Index

Latin letters

[1] P. Aczel. *Non-Well-Founded Sets*, volume 14 of *Lecture Notes*. CSLI, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA, 1988.

[2] H.P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and The Foundation of Mathematics*. North-Holland, 1984.

[3] C. Berline. From computation to foundations via functions and application : the lambda-calculus and its webbed models. *Theoretical Computer Science*, 249:81–161, 2000.

[4] 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.

[5] A. Church. A set of postulates for the foundations of logic I. *Ann. Math.*, 33:346–366, 1932.

[6] A. Church. A set of postulates for the foundations of logic II. *Ann. Math.*, 34:839–864, 1933.

[7] R.C. Flagg. κ-continuous lattices and comprehension principles for Frege structures. *Annals of Pure and Applied Logic*, 36:1–16, 1987.

[8] R.C. Flagg and J. Myhill. A type-free system extending ZFC. *Annals of Pure and Applied Logic*, 43:79–97, 1989.

[9] K. Grue. Map theory. *Theoretical Computer Science*, 102(1):1–133, July 1992.

[10] 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.

[11] 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.

[12] Rainer Kerth. Définissabilité dans les domaines réflexifs. *Comptes rendus de l'Acadmie des sciences. Série 1, Mathématique*, 318(7):685–688, 1994.

[13] Rainer Kerth. Isomorphisme et équivalence équationelle entre modéles du lambda-calcul. Ph.D thesis, Université Paris 7, 1995.

[14] G. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5(3):223–255, December 1977.

[15] 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.

[16] Thierry Vallée. "Map theory" et antifondation. *Electr. Notes Theor. Comput. Sci.*, 79:1–260, 2003. (Ph.D. Thesis, Université Paris 7, 2001).

[17] 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.