λ-calculus as a foundation of mathematics

Klaus Grue*

August 28, 1997

Abstract

Church introduced λ-calculus in the beginning of the thirties as a foundation of mathematics and map theory from around 1992 fulfilled that primary aim.

The present paper presents a new version of map theory whose axioms are simpler and better motivated than those of the original version from 1992. The paper focuses on the semantics of map theory and explains this semantics on basis of κ-Scott domains.

The new version sheds some light on the difference between Russells and Burali-Fortis paradoxes, and also sheds some light on why it is consistent to allow non-well-founded sets in a ZF-style system.

*DIKU, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark, E-mail grue@diku.dk

Contents

1 Introduction

As mentioned in the abstract, Church introduced λ-calculus [4, 5, 6] in the beginning of the thirties as a foundation of mathematics and map theory from around 1992 fulfilled that primary aim.

In the meanwhile, λ-terms have shown very useful for expressing semantics in computer science, but there has been no natural choice of a theory for reasoning about these λ-terms.

In the lack of such a natural choice, computer science has turned to syntactic methods in which one reasons about the structure and conversion of λ-terms rather than their meaning.

Mathematics, on the contrary, has had set theory which allows to reason about sets rather than syntax. Set theory offers the luxury of referential transparency, in which every term has a meaning and every term implicitly denotes that meaning.

Map theory resembles set theory in that it assigns meaning to λ-terms and treats λ-terms in a referentially transparent fashion. Map theory also resembles set theory when comparing metamathematical power: For every consistent set theory Z there is a consistent map theory M more powerful than Z and vice versa. The deepest difference between set and map theory shows up in the treatment of infinite looping: Russell's sentence $\{x \mid x \notin x\} \in \{x \mid x \notin x\}$ is just a term that takes infinitely long time to compute, but set theory deals with this sentence by forbidding it rather than taking its value serious. Map theory, on the contrary, assigns meaning to the corresponding term $(\lambda x.\neg(xx))(\lambda x.\neg(xx))$.

Map theory is not a marriage of convenience between λ-calculus and set theory [7]. Rather, map theory is a theory based entirely on λ-calculus in which, among other, set membership, logical connectives, and quantification over all sets are definable concepts and in which all axioms and inference rules of ZFC are provable without resort to any syntactical considerations.

Map theory has the potential to serve as a foundation of both mathematics (due to its power that is equivalent to that of set theory) and of computer science (due to its treatment of λ-terms). Set and map theory are both inherently difficult to learn due to their level of abstraction, but set theory has the advantage of a century of pedagogical engineering that has made it easier to approach. On the contrary, the version of map theory from 1992 appears as a somewhat random collection of axioms that accidentally have the power of set theory and accidentally describe λ-terms.

Since 1992 it has turned out, however, that map theory is a natural choice of a theory of λ-calculus. More specifically, it has turned out that every sufficiently large so-called κ-Scott domain $\mathcal{D}$ for which $\mathcal{D} \cong [\mathcal{D} \rightarrow \mathcal{D}] \oplus_\perp \mathbf{1}$ where $\mathbf{1}$ is a one-element set contains a model for map theory [2]. When Dana Scott invented what is now known as Scott domains he was fully aware that the notion could be generalised to κ-Scott domains, but did not publish this finding as he saw no application of them at that time. Hence, in some sense, a λ-based foundation of mathematics has been around for a long time without anybody recognising it.

The present paper presents a new version of map theory which will be referred to as *MTC* (Map Theory with Classical maps) as opposed to the version from 1992 which will be referred to as *MTW* (Map Theory with Well-founded maps). The axioms of MTC are simpler and better motivated than those of MTW. The step from MTW to MTC is intended as a step in the direction of a theory that is easier to learn and teach. The paper focuses on the semantics of MTC and explains this semantics on basis of κ-Scott domains.

1.1 Differences with Churchs approach

It is a pity that Church did not find a theory like map theory right away since that could have saved a lot of work in computer science. There are, however, three good reasons why that did not happen, and these reasons are stated in the next three sections.

1.2 Inclusion of non-functions

The first reason is that Churchs theory may be seen as a theory about functions only, and as such is a theory about only one concept. Classical logic is built around the distinction between truth and falsehood, i.e. the semantic distinction between two concepts. The semantic nature of classical logic and set theory stems from this distinction. λ-calculus, on the contrary, deals with functions only, and in a theory with only one concept it is impossible to make a semantic distinction. For that reason, λ-calculus can merely deal with the provability or non-provability of the equivalence of λ-terms which is syntactic of nature.

In map theory, this problem is solved by insisting that the universe of map theory must contain at least one non-function. Just one non-function is enough to make a distinction, namely a distinction between functions and non-functions. In map theory, the minimalistic approach has been taken to include only one non-function.

Having both functions and non-functions in map theory allows to represent truth and falsehood, and the convention has been chosen to let functions represent falsehood and let non-functions represent truth. This convention has been chosen very carefully on basis of what makes definitions inside map theory easiest to read, but this issue will not be treated here.

Since there is only one non-function in map theory, it is convenient to introduce a name for that non-function, and since the non-function represents truth, the name T has been chosen.

In addition to functions and non-functions, the universe of map theory contains an element which is neither a function, nor a non-function. That element is denoted $\perp$ and represents infinite looping. This element violates the axiom of Tertium Non Datur, which in this context says that every object is either a function of a non-function. Nevertheless map theory is still classical of nature because it has another axiom called Quartum Non Datur which says that any map is either a function or T or $\perp$, there is no fourth possibility.

The inclusion of the non-function T in map theory is a trivial step, but it is a step that is very important for the semantic nature of the theory.

1.3 Set abstraction versus λ-abstraction

Set theory has set abstraction $\{x \mid p(x)\}$ and λ-calculus has λ-abstraction $\lambda x.p(x)$. It is tempting to identify the two kinds of abstraction and try to represent the class $\{x \mid p(x)\}$ by $\lambda x.p(x)$, i.e. to represent classes by their characteristic functions. This approach, however, has not succeeded, and set abstraction and λ-abstraction seem to be two completely different kinds of abstraction.

In map theory, a function g does not represent the class $\{x \mid g(x) = \mathsf{T}\}$. Rather, g represents the class $\{g(x) \mid x \in S\}$ where S is a fixed class of maps. In MTW, S is the class W of well-founded maps, and in MTC, S is the class C of classical maps. $\{g(x) \mid x \in S\}$ contains at least one element, so to allow to represent the empty set, the non-function T is taken to represent that set.

With this encoding, all sets of ZFC can be represented by well-founded maps in MTW and by classical maps in MTC. And, opposite, all well-founded maps in MTW and all classical maps in MTC represent sets. Classes may also be represented, but they are represented by maps that are not well-founded/classical.

In conclusion, the non-identification of set abstraction and λ-abstraction has been an important point in turning λ-calculus into a foundation.

1.4 Selection of well-behaved maps

The third problem in turning λ-calculus into a foundation of mathematics was to find a class S of maps that was sufficiently well-behaved to represent the sets of ZFC. In MTW, the class W of well-founded maps was chosen and in MTC, the class C of classical maps was chosen.

Insisting that $\{g(x) \mid x \in S\}$ should be a set of ZFC for all $g \in S$ puts many restrictions on S, and insisting that all sets of ZFC should be representable by an element of S puts strong requirements on the size and richness of S. Nevertheless, these restrictions by no means determine S uniquely, and finding a natural S is no trivial task.

1.5 Relation between MTW and MTC

One advantage of MTC over MTW is that eleven complicated axioms and inference rules that describe well-foundedness in MTW has been replaced by a single definition of classicality in MTC.

Another advantage is that MTC contains some inference rules that were missing in MTW. These are rules Y, M and E in Appendix B.5. In particular, rule Y says that the fixed point operator generates a minimal fixed point. It is interesting that the theorem of transfinite induction is provable in MTC from the minimality of fixed points combined with the recursive definition of classicality used in MTC. The details are worked out in [12].

A third advantage is the distinction between discontinuous and continuous occurrences of variables introduced in Section 3.1 which allows to treat equations as terms. The syntax of MTW specifies the syntax of terms and well-formed formulas. MTC is simpler in that it does not distinguish between terms and well-formed formulas. This may turn out to be convenient in computer assisted proof systems for MTC because it allows to represent both theorems and inference rules as terms. The notion of discontinuous occurrences permits the inference rule $a = \mathsf{T} \vdash a$ which is also valid in the systems of Feferman [8] and Flagg and Myhill [10]. The notion of discontinuous occurrences also permits the opposite rule $a \vdash a = T$. The distinction between discontinuous and continuous occurrences of variables seems to be new way to deal with the anomalies of equality.

MTW and MTC may be compared both on a syntactic and a semantic basis. A syntactic comparison is made in [12] with the following result:

If the classicality predicate of MTC is used to simulate the well-foundedness predicate in MTW, then all axioms and inference rules of MTW except Axiom Well-2 in [11] are provable in MTC. Axiom Well-2 in MTW is disprovable in MTC, which is just a consequence of the slight difference between well-foundedness and classicality. In [11], Axiom Well-2 is only used to prove Lemma C-K and Lemma C-P, both of which are provable in MTC. Hence, all theorems of MTW proved in [11] are also provable in MTC.

Among other, [11] proves all axioms and inference rules of ZFC in MTW. Combining these observations we have that all axioms and inference rules of ZFC are provable in MTC. This shows that MTC is adequate as a foundation of mathematics (provided ZFC is considered adequate).

MTW and MTC may also be compared semantically. This paper introduces the semantics of MTC by means of a κ-Scott domain $\mathcal{D} = (D, \leq)$ where D is the universe of all maps and $\leq$ is a partial order on all maps. The κ-Scott domain used has the property that it models both MTW and MTC, so it makes sense to compare MTW and MTC in this particular model. A class A of maps will be said to be "coherent" if any two elements of A has an upper bound in $\mathcal{D}$. The correspondence between the classes W and C of well-founded and classical maps, respectively, may now be formulated as follows:

- Any well-founded map is classical.
- Any non-empty, coherent class of well-founded maps has a greatest lower bound, and that greatest lower bound is classical.
- Any classical map is the lower bound of a coherent class of well-founded maps.

Hence, the class C of classical maps can be seen as the closure of W under greatest lower bounds of non-empty, coherent sets. An important difference between W and C is:

- For all well-founded maps g except T there exists a well-founded map h such that $h < g$.

- For all classical maps g there exists a minimal classical map h such that $h \leq g$ (h is a minimal classical map if $h' \leq h \Rightarrow h' = h$ for all classical maps h').

1.6 The structure and contents of the paper

Section 2 describes the semantics of MTC based on the κ-denotational framework developed in [2]. Section 3 gives a rather quick tour through the syntax, axioms and inference rules of MTC. Section 4 concludes by remarks on Russells and Burali-Fortis paradoxes and non-well-founded sets. Appendix A outlines a model of MTC based on the model of MTW in [2]. Merely the definition of the model is stated. The satisfaction of the axioms and inference rules remains to be proved. Appendix B summarises MTC.

A more detailed description of the individual axioms and inference rules and explanations of how they are used may be found in [12]. Note that the system in [12] contains an inconsistent axiom as pointed out by Chantal Berline. [12] plus errata may be obtained from //www.diku.dk/~grue.

2 The semantics of MTC

2.1 Maps over finite sets

In the following, words in italics and mathematical concepts in boxes occur in the index. Figure 1 shows a *map* over the set $I = \{1, 2, 3\}$.

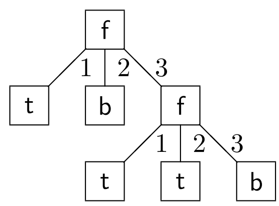

Figure 1: A map over $\{1, 2, 3\}$

In general, a map over I is a tree where each node is labelled by $\boxed{\mathsf{t}}$, $\boxed{\mathsf{f}}$ or $\boxed{\mathsf{b}}$, where each edge is labelled by an element of I, where each node labelled f has one downward edge for each element of I, and where each node labelled t or b has no downward edges. Maps may be infinitely deep. As an example, Figure 2 shows a map over $\{1, 2\}$.

If I is a set, if $x \in I$ and if g is a map over I, then we define g *applied to* x, denoted $\boxed{g'x}$, to be the subtree of g attached to the edge labelled x that extends downwards from the root of g. As an example, if g is the map in Figure 1 then Figure 3, 4 and 5 shows $g'1$, $g'2$ and $g'3$, respectively.

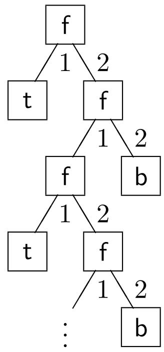

Figure 2: A map over $\{1, 2\}$

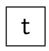

Figure 3: The map T

Application is left associative so that e.g. $g'3'1$ means $(g'3)'1$. If g is the map in Figure 1 then $g'3'1$ is the map in Figure 3. If h is the map in Figure 2 then $h'2'1 = h$.

The trees in Figure 3 and 4 will be denoted $\boxed{\mathsf{T}}$ and $\boxed{\bot}$, respectively. No edges extend downwards from the roots of T and $\bot$, so the definition of $g'x$ does not make sense for $g = \mathsf{T}$ and $g = \bot$. To make $g'x$ defined for all $x \in I$ and all maps g over I, we more or less arbitrarily define:

$$\begin{aligned} \mathsf{T}'x &= \mathsf{T} \\ \bot'x &= \bot \end{aligned}$$

2.2 Modelling of maps

Let t, f and b be three distinct objects. For all maps g over I we define $\boxed{r(g)}$ to be the label of the root of g. Hence, $r(\mathsf{T}) = \mathsf{t}$, $r(\bot) = \mathsf{b}$ and $r(g) = \mathsf{f}$ for all maps g over I except T and $\bot$.

Let $\boxed{I^{<\omega}}$ denote the set of finite lists $\langle x_1, \ldots, x_n \rangle$ of elements of I. For all maps g over I and all $\overline{x} = \langle x_1, \ldots, x_n \rangle \in I^{<\omega}$ let $g[\overline{x}]$ denote $g'x_1' \cdots' x_n$. As an example, $h[\langle 2, 1, 2, 1, 2, 1 \rangle] = h$ where h is the map in Figure 2. If $\overline{x}$ is the empty tuple $\langle\, \rangle$, then $g[\overline{x}]$ denotes g itself.

If g is a map over I and if $x \in I^{<\omega}$, then $r(g[\overline{x}])$ will be referred to as the label *indexed* by $\overline{x}$. As an example, if g is the map in Figure 1, then the labels indexed by $\langle 1 \rangle$, $\langle 3 \rangle$ and $\langle 3, 3 \rangle$ are t, f and b, respectively. The label indexed by

Figure 4: The map $\perp$

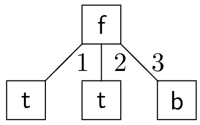

Figure 5: A map over $\{1, 2, 3\}$

$\langle 3, 3, 2\rangle$ is also b as shown by the following:

$$\begin{array}{rcl} g[\langle 3,3,2\rangle] & = & g'3'3'2 \\ & = & (g'3'3)'2 \\ & = & \perp'2 \\ & = & \perp \end{array}$$

Hence, computation of $r(g[\langle 3, 3, 2\rangle])$ depends on the convention that $\perp' x = \perp$.

In general, if $r(g[\langle x_1, \cdots, x_m\rangle]) = \mathsf{b}$ then $r(g[\langle x_1, \cdots, x_m, y_1, \cdots, y_n\rangle]) = \mathsf{b}$. Now for all $\overline{x}, \overline{y} \in I^{<\omega}$ let $\boxed{\overline{x} \cdot \overline{y}}$ denote the concatenation of the tuples x and y. We have

$$\begin{array}{lcl} r(g[\overline{x}]) = \mathsf{b} & \Rightarrow & r(g[\overline{x} \cdot \overline{y}]) = \mathsf{b} \\ r(g[\overline{x}]) = \mathsf{t} & \Rightarrow & r(g[\overline{x} \cdot \overline{y}]) = \mathsf{t} \end{array}$$

These two statements may be combined into one:

$$r(g[\overline{x}]) \neq \mathsf{f} \Rightarrow r(g[\overline{x} \cdot \overline{y}]) = r(g[\overline{x}])$$

Now let $\boxed{L} = \{\mathsf{t}, \mathsf{f}, \mathsf{b}\}$. For all functions u let $\boxed{\operatorname{dom} u}$ and $\boxed{\operatorname{rng} u}$ denote the domain and range of u, respectively. For all sets G and H let $\boxed{G \to H}$ denote the set of functions u for which $\operatorname{dom} u = G$ and $\operatorname{rng} u \subseteq H$.

If g is a map over I and if $\overline{x} \in I^{<\omega}$, then $r(g[\overline{x}]) \in L$. Two maps u and v over I are considered equal if $r(g[\overline{x}]) = r(h[\overline{x}])$ for all $\overline{x} \in I^{<\omega}$. Hence, a map g may be modelled by the function $\hat{g} \in I^{<\omega} \to L$ for which $\hat{g}(\overline{x}) = r(g[\overline{x}])$ for all $x \in I^{<\omega}$. From now on, maps are modelled this way which motivates to define the set M_I of maps over I by

$$\boxed{M_I} = \{g \in I^{<\omega} \to L \mid \forall \overline{x}, \overline{y} \in I^{<\omega} : (g(\overline{x}) \neq \mathsf{f} \Rightarrow g(\overline{x} \cdot \overline{y}) = g(\overline{x}))$$

2.3 Partially ordered sets

A *p.o* $\mathcal{D}$ is a *partially ordered set* $(D, \leq)$. As an example, $\boxed{\mathcal{L}} = (L, \leq_L)$ is a p.o where

$$\boxed{p \leq_L q} \Leftrightarrow p = \mathsf{b} \vee p = q$$

For all non-empty sets I, $\boxed{\mathcal{M}_I} = (M_I, \leq)$ is a p.o where

$$\boxed{g \leq h} \Leftrightarrow \forall x \in I^{<\omega} : g(x) \leq_L h(x)$$

For all p.o's $\mathcal{D} = (D, \leq)$ define the p.o $\boxed{\mathcal{D}^{<\omega}}$ by $\mathcal{D}^{<\omega} = (D^{<\omega}, \boxed{\leq^*})$ where

$$\langle x_1, \ldots, x_m \rangle \leq^* \langle y_1, \ldots, y_n \rangle \Leftrightarrow m = n \wedge x_1 \leq y_1 \wedge \cdots \wedge x_m \leq y_m$$

For all p.o's $\mathcal{D} = (D, \leq)$ and $\mathcal{E} = (E, \leq)$ let $x \in \mathcal{D}$, $A \subseteq \mathcal{D}$ and $g \in \mathcal{D} \to \mathcal{E}$ be shorthand for $x \in D$, $A \subseteq D$ and $g \in D \to E$, respectively.

For all non-empty sets I define $\boxed{\bot_I}, \boxed{\mathsf{T}_I}, \boxed{\mathsf{F}_I} \in \mathcal{M}_I$ by

$$\begin{array}{l} \bot_I(\langle x_1, \ldots, x_n \rangle) = \mathsf{b} \\ \mathsf{T}_I(\langle x_1, \ldots, x_n \rangle) = \mathsf{t} \\ \mathsf{F}_I(\langle \rangle) = \mathsf{f} \\ \mathsf{F}_I(\langle u, x_1, \ldots, x_n \rangle) = \mathsf{t} \end{array}$$

for all $n \geq 0$ and all $u, x_1, \ldots, x_n \in I$. $\bot_I$ is the unique bottom element of $\mathcal{M}_I$. T_I and F_I are two among many maximal elements.

2.4 κ-continuity

For all sets A and κ, A is said to be *κ-small* if A has cardinality strictly less than κ. From now on let κ be an infinite set.

An element x of a p.o $\mathcal{D} = (D, \leq)$ is said to be an *upper bound* of $A \subseteq \mathcal{D}$ if $\forall y{\in}A : y \leq x$. A subset $H \subseteq \mathcal{D}$ is said to be a *κ-chain* if all κ-small subsets of H have an upper bound in H. In particular, the empty set must have an upper bound in H so any κ-chain is non-empty. A p.o $\mathcal{D} = (D, \leq)$ is a *κ-cpo* if every κ-chain $H \subseteq \mathcal{D}$ has a supremum $\boxed{\sup H}$ in $\mathcal{D}$. As an example, for all non-empty sets I, $\mathcal{M}_I$ and $(\mathcal{M}_I)^{<\omega}$ are κ-cpo's. $\mathcal{M}_I$ is an example of a κ-cpo with a bottom element and $(\mathcal{M}_I)^{<\omega}$ is an example of one without.

For all κ-cpo's $\mathcal{D}$ and $\mathcal{E}$, a function $g \in \mathcal{D} \to \mathcal{E}$ is said to be *κ-continuous* if

$$g(\sup H) = \sup \{g(x) \mid x \in H\}$$

for all κ-chains $H \subseteq \mathcal{D}$. Let $\boxed{[\mathcal{D} \to \mathcal{E}]_\kappa}$ denote the κ-cpo of κ-continuous $g \in \mathcal{D} \to \mathcal{E}$, ordered by pointwise ordering. For all $\mathcal{D} = (D, \leq)$ define

$$\begin{array}{lcl} \boxed{M^\kappa_{\mathcal{D}}} & = & \{g{\in}M_D \mid g \in [\mathcal{D}^{<\omega} \to \mathcal{L}]_\kappa\} \\ \boxed{\mathcal{M}^\kappa_{\mathcal{D}}} & = & (M^\kappa_{\mathcal{D}}, \leq) \end{array}$$

We shall refer to elements of $\mathcal{M}^\kappa_{\mathcal{D}}$ as *κ-continuous maps over $\mathcal{D}$*.

2.5 Maps over R and maps over maps over R

At this point a few examples may show useful. Let $\mathbf{R}$ be the set of real numbers, let $\mathbf{Z} \subseteq \mathbf{R}$ be the set of integers, and let $\mathcal{G} = (G, \leq) = \mathcal{M}_{\mathbf{R}}$ be the p.o of maps over $\mathbf{R}$. We have that $\mathbf{Z}$ is $\mathbf{R}$-small since $\mathbf{Z}$ has cardinality strictly less than $\mathbf{R}$.

For all $A \subseteq \mathbf{R}$ define the characteristic map $\chi_A \in \mathcal{G}$ by

$$\begin{array}{lcl} \chi_A(\langle\,\rangle) & = & \mathsf{f} \\ \chi_A(\langle u, x_1, \ldots, x_n\rangle) & = & \begin{cases} \mathsf{t} & \text{if } u \in A \\ \mathsf{b} & \text{otherwise} \end{cases} \end{array}$$

As an example of a non-trivial $\mathbf{R}$-chain in $\mathcal{G}^{<\omega}$ we have

$$H = \{\langle \chi_A \rangle \mid A \subseteq \mathbf{R} \wedge A \text{ is } \mathbf{R}\text{-small}\}$$

The supremum $\sup H$ of this chain is $\langle \chi_{\mathbf{R}} \rangle$.

Now consider the set M_G of maps over $G = M_{\mathbf{R}}$, i.e. of maps over maps over $\mathbf{R}$. For all $B \subseteq \mathbf{R}$ define $\forall_B \in M_G$ by

$$\begin{array}{lcl} \forall_B(\langle\,\rangle) & = & \mathsf{f} \\ \forall_B(\langle u, x_1, \ldots, x_n\rangle) & = & \begin{cases} \mathsf{b} & \text{if } \exists v{\in}B : u(\langle v \rangle) = \mathsf{b} \\ \mathsf{t} & \text{if } \forall v{\in}B : u(\langle v \rangle) = \mathsf{t} \\ \mathsf{F}_G(x_1, \ldots, x_n) & \text{otherwise} \end{cases} \end{array}$$

$\forall_B$ satisfies

$$\forall_B{}'u = \begin{cases} \bot_G & \text{if } \exists v \in B : u'v = \bot_{\mathbf{R}} \\ \mathsf{T}_G & \text{if } \forall v \in B : u'v = \mathsf{T}_{\mathbf{R}} \\ \mathsf{F}_G & \text{otherwise} \end{cases}$$

If we take T, F and $\bot$ to represent truth, falsehood and undefined, respectively, then $\forall_B$ represents a universal quantifier that quantifies over B. The quantifier is strict in the sense that $\forall_B g$ is undefined if gx is undefined for some $x \in B$.

We have $\forall_{\mathbf{R}}(\sup H) = \mathsf{t}$ and $\sup\{\forall_{\mathbf{R}}(x) \mid x \in H\} = \mathsf{b}$ which shows that $\forall_{\mathbf{R}}$ is $\mathbf{R}$-discontinuous. On the contrary it is straightforward to prove that $\forall_{\mathbf{Z}}$ is $\mathbf{R}$-continuous (the proof is a somewhat lengthy proof by cases, but the point in the proof is that for all $\mathbf{R}$-chains H' there exists a function $h'' \in H'$ such that $\forall n{\in}\mathbf{Z}{:}\, h''(n) = h'(n)$ where $h' = \sup H'$). In general, $\forall_B$ is $\mathbf{R}$-continuous if and only if B is $\mathbf{R}$-small. Even more generally, quantification over a set B is κ-continuous if and only if B is κ-small.

2.6 κ-premodels

We have now seen maps over $\{1, 2, 3\}$, maps over $\mathbf{R}$ and $\mathbf{R}$-continuous maps over maps over $\mathbf{R}$. The maps of MTC are κ-continuous maps over maps of MTC. In other words, the domain $\mathcal{D}$ of all maps of MTC satisfies $\mathcal{D} \cong \mathcal{M}^{\kappa}_{\mathcal{D}}$.

If $\mathcal{D}$ is a *κ-Scott domain* [2], if σ is a *strongly inaccessible ordinal* [3], if κ is a *regular cardinal* [3] greater than σ and if $\mathcal{D} \cong \mathcal{M}^{\kappa}_{\mathcal{D}}$ then it follows from [2] that $\mathcal{D}$ can be expanded into a model of MTW (to see that one needs to prove $\mathcal{D} \cong \mathcal{M}^{\kappa}_{\mathcal{D}} \Rightarrow \mathcal{D} \cong [\mathcal{D} \rightarrow \mathcal{D}]_{\kappa} \oplus_{\perp} \{\mathsf{T}_D\}$; then let $A \in \mathcal{D} \rightarrow [\mathcal{D} \rightarrow \mathcal{D}]_{\kappa} \oplus_{\perp} \{\mathsf{T}_D\}$ be an order isomorphism, let λ be the inverse of A and apply Theorem 7.2.1 in [2] to $(\mathcal{D}, A, \lambda)$).

A model of MTC needs to satisfy more than this, partly because of the classical maps, partly because MTC contains inference rules that were missing in MTW.

From now on assume that there exist strongly inaccessible ordinals, assume that $\boxed{\sigma}$ is the least strongly inaccessible ordinal and assume that $\boxed{\kappa}$ is a regular cardinal greater than σ.

A *κ-premodel* P of MTC is a structure $(\mathcal{D}, \mathsf{a}, C, q)$ which satisfies the five properties below plus one more property which is stated in Section 2.7.

- $\mathcal{D} = (D, \leq)$ is a κ-Scott domain (and, in particular, a κ-cpo)
- $\mathsf{a} \in [\mathcal{D} \rightarrow \mathcal{M}^{\kappa}_{\mathcal{D}}]_{\kappa}$ is an isomorphism
- $C \subseteq \mathcal{D}$ is a κ-small set of so-called κ-compact elements (c.f. [2] and Appendix A.1).
- q is a choice function over $\mathcal{D}$, i.e. $q(A) \in A$ for $A \subseteq \mathcal{D}$, $A \neq \emptyset$
- $q(\emptyset) \in C$

κ-premodels are introduced here to present the intuitions behind MTC. The construction of a κ-premodel is outlined later. The detailed development and the verification of axioms and inference rules of MTC remains to be done. Now define

$$
\begin{aligned}
&\boxed{\bar{\mathsf{a}}} \text{ is the inverse of } \mathsf{a} \\
&\boxed{\mathsf{T}} = \bar{\mathsf{a}}(\mathsf{T}_D) \\
&\boxed{\perp} = \bar{\mathsf{a}}(\perp_D) \\
&\boxed{\mathsf{F}} = \bar{\mathsf{a}}(\mathsf{F}_D)
\end{aligned}
$$

For all $g, x \in \mathcal{D}$ let $\boxed{g'x}$ be the unique element of $\mathcal{D}$ for which

$$\mathsf{a}(g'x)(\langle y_1, \ldots, y_n \rangle) = \mathsf{a}(g)(\langle x, y_1, \ldots, y_n \rangle)$$

This defines the notion of applying a map g to an argument x which was first mentioned in Section 2.1. This concludes a circle: informal considerations about application of maps led to a representation of maps which was refined into a model of maps which allows to define application.

2.7 Classical maps

For all $S \subseteq D$ and $x, y \in D$ define $\boxed{x =_S y} \Leftrightarrow \forall z \in S^{<\omega}\colon \mathsf{a}(x)(z) = \mathsf{a}(y)(z)$. For all $x, y \in D^{<\omega}$ define $\boxed{x =^*_S y}$ by

$$\langle x_1, \cdots, x_m \rangle =^*_S y \langle y_1, \cdots, y_n \rangle \Leftrightarrow m = n \wedge x_1 = y_1 \wedge \cdots \wedge x_m = y_m$$

Let $\boxed{\mathcal{P}_\sigma(A)}$ denote the set of σ-small subsets of A. Let C' be the least subset of D which satisfies

$$\begin{array}{lcl} g \in C' & \Leftrightarrow & \forall x \in C : g'x \in C' \wedge \\ & & \exists V \in \mathcal{P}_\sigma(C') \forall x, y \in C^{<\omega}\colon (x =^*_V y \Rightarrow \mathsf{a}(g)(x) = \mathsf{a}(g)(y)) \end{array}$$

The existence of such a C' is easy to verify. For all $x \in \mathcal{D}$ and $A \subseteq \mathcal{D}$ let $\boxed{\uparrow x} = \{y \in \mathcal{D} \mid x \leq y\}$ and $\uparrow A = \bigcup \{\uparrow x \mid x \in A\}$. A κ-premodel P of MTC is a structure $(\mathcal{D}, \mathsf{a}, C, q)$ which satisfies the five properties in Section 2.6 plus the one below:

- $C' = \uparrow C$

3 Presentation of MTC

3.1 Syntax

The syntax $\mathcal{V}$ of variables and $\mathcal{T}$ of terms of MTC reads:

$$\begin{array}{lcl} \mathcal{V} & ::= & x_1 \mid x_2 \mid \cdots \\ \mathcal{T} & ::= & \mathcal{V} \mid \mathsf{T} \mid \mathcal{T}\mathcal{T} \mid \lambda\mathcal{V}.\mathcal{T} \mid \mathsf{P} \mid \varepsilon \mid \bar{\exists} \mid \epsilon(\mathcal{T}) \mid \mathcal{T} = \mathcal{T} \end{array}$$

The construct $\mathcal{T}\mathcal{T}$ has higher priority than $\lambda\mathcal{V}.\mathcal{T}$, which in turn has higher priority than $\mathcal{T} = \mathcal{T}$, so that e.g. $\lambda x_1.x_1x_1 = x_2$ means $(\lambda x_1.(x_1x_1)) = x_2$.

An occurrence of a variable v in a term t is said to be *discontinuous* if v occurs free in a subterm t' of t which has one of the forms $\epsilon(t'')$ or $t'' = t'''$. Occurrences that are not discontinuous are said to be *continuous*. As an example, the second occurrence of x_1 in $\mathsf{P}x_1x_2(x_1 = \lambda x_1.x_1)$ is discontinuous whereas the other occurrences of variables are continuous.

The following purely syntactical restriction is put on terms of MTC: Discontinuous occurrences of variables are not allowed to be bound. As an example, $\lambda x_1.(x_1 = \mathsf{T})$ is not a well-formed term.

3.2 Proofs

A proof in MTC is a sequence of terms in which each term is either an instances of an axiom scheme of follows from previous terms in the sequence by an inference rule. All axioms and inference rules of MTC are listed in Appendix B.5.

The interpretation of a proof is that it proves the last term in the proof, i.e. it proves that the last term in the proof equals T for all values of free variables.

3.3 Truth, equality and application

The term T in Section 3.1 denotes the value T defined in Section 2.6. The term $a = b$ equals T when a equals b and $a = b$ equals F otherwise. The value of $a = b$ may depend on the values of free variables in a and b. As an example, $x_1 = \mathsf{T}$ equals T when x_1 equals T and equals F otherwise. The inference rules of truth and equality read:

T	Transitivity	$a = b; a = c \vdash b = c.$
SA	Substitutivity	$a = c; b = d \vdash ab = cd.$
$S\lambda$	Substitutivity	$a = b \vdash \lambda x.a = \lambda x.b.$
$S\epsilon$	Substitutivity	$a = b \vdash \epsilon(a) = \epsilon(b).$
$=\vdash$	Equality	$a = \mathsf{T} \vdash a$
$\vdash=$	Equality	$a \vdash a = \mathsf{T}$

In inference rules, a, b, c, d, g and h denote arbitrary terms whereas x, y and z denote arbitrary, distinct variables.

The inference rule $a = \mathsf{T} \vdash a$ is inspired by [8, 10]. The opposite rule $a \vdash a = \mathsf{T}$ is made possible by the notion of discontinuous occurrences of variables which seems to be a new way of dealing with the discontinuity of equality.

For all terms a and b, $\boxed{ab}$ denotes a applied to b, i.e. ab denotes $a'b$. Since T applied to anything equals T by convention, we have the axiom

$A\mathsf{T}$ Application $\mathsf{T}a = \mathsf{T}.$

As an example of a proof we have

1	$A\mathsf{T}$	$\mathsf{T}a = \mathsf{T}$
2	$A\mathsf{T}$	$\mathsf{T}a = \mathsf{T}$
3	$1, 2, T$	$\mathsf{T} = \mathsf{T}$
4	$3, =\vdash$	T

3.4 Abstraction

For all variables x and terms $\mathcal{A}$ of ZFC let $x \mapsto \mathcal{A}$ be shorthand for $\{(x, \mathcal{A}) \mid x \in \mathcal{D}\}$, i.e. let $x \mapsto \mathcal{A}$ be the unique function g for which $\operatorname{dom} g = \mathcal{D}$ and $g(x) = \mathcal{A}$. For all $g \in [\mathcal{D} \to \mathcal{D}]_\kappa$ let $\lambda(g)$ be the unique element of $\mathcal{D}$ for which

$$\begin{array}{lcl} \lambda(g)(\langle\,\rangle) & = & \mathsf{f} \\ \lambda(g)(\langle u, x_1, \ldots, x_n\rangle) & = & g(u)(\langle x_1, \ldots, x_n\rangle) \end{array}$$

Finally define

$$\lambda x.\mathcal{A} = \lambda(x \mapsto \mathcal{A})$$

whenever $(x \mapsto \mathcal{A}) \in [\mathcal{D} \to \mathcal{D}]_\kappa$, i.e. whenever $x \mapsto \mathcal{A}$ is a κ-continuous function from maps to maps.

If x is a variable and $\mathcal{A}$ is a term of MTC, then $(x \mapsto \mathcal{A}) \in \mathcal{D} \to \mathcal{D}$. If, furthermore, there are no discontinuous free occurrences of x in $\mathcal{A}$ then $x \mapsto \mathcal{A}$ is κ-continuous so that $(x \mapsto \mathcal{A}) \in [\mathcal{D} \to \mathcal{D}]_\kappa$ and $(\lambda x.\mathcal{A}) \in \mathcal{D}$. This gives an interpretation of $\lambda x.\mathcal{A}$ whenever $\lambda x.\mathcal{A}$ is a well-formed term. We may now formulate two more axioms, namely

$A\lambda$	Application	$(\lambda x.a)b = \langle a \mid x{:=}b\rangle$ if b is free for x in a.
R	Renaming	$\lambda x.\langle a \mid y{:=}x\rangle = \lambda y.\langle a \mid x{:=}y\rangle$. if x is free for y in a and vice versa.

Here $\boxed{\langle a \mid x{:=}b\rangle}$ is the term that arises when replacing all free x in a by b. See [14] for a definition of *free for*. Axiom R allows renaming of bound variables and Axiom $A\lambda$ expresses that two terms are equal if they are β-equivalent. (Note, however, that terms may be equal in map theory without being β-equivalent. This holds even for terms that contain only variables, abstraction and application. The notion of equality in map theory is the semantic notion introduced in Section 2.2 and this semantic notion is not fully captured by β-equivalence).

Having abstraction and application allows to define the fixed point operator Y and the bottom element $\perp$:

$$\begin{array}{rcl} \boxed{\mathsf{Y}} & = & \lambda f.(\lambda x.f(xx))\lambda x.f(xx) \\ \boxed{\perp} & = & \mathsf{Y}\lambda x.x \end{array}$$

Having the bottom element we may state one more axiom and one more inference rule:

$A\perp$	Application	$\perp a = \perp$.
QND	Quartum Non Datur	$a\mathsf{T} = b\mathsf{T}; a\perp = b\perp; a\lambda y.xy = b\lambda y.xy \vdash ax = bx$.

Axiom $A\perp$ expresses that $\perp$ applied to anything yields $\perp$. Inference rule QND expresses that the root of any map is t, f or b, there is no fourth possibility. QND together with the map P described in Section 3.5 allows to develop classical propositional calculus (c.f. [11, 2, 12]). In the present paper, only the intended meaning of axioms and inference rules will be stated. The details of why they express the intended meaning and how they are used is stated in [12].

3.5 Selection

The map P is best described by the following three axioms:

PT	Selection	$\mathsf{P}ab\mathsf{T} = a$.
$\mathsf{P}\lambda$	Selection	$\mathsf{P}ab\lambda x.c = b$
$\mathsf{P}\perp$	Selection	$\mathsf{P}ab\perp = \perp$.

The map P allows to define many auxiliary concepts such as logical connectives (c.f. Appendix B.4) and to develop classical propositional calculus on basis of QND.

For $a, b \in \mathcal{D} = (D, \leq)$, the constructs introduced so far allow to define $a \preceq b$ such that $a \preceq b$ equals T if $a \leq b$ and equals F otherwise. The definition of $a \preceq b$ is stated in Appendix B.4 and reads

$$a{\preceq}b \doteq (a = a{\downarrow}b)$$

where $a{\downarrow}b$ is defined on basis of P in Appendix B.4. The definition of $\preceq$ uses $=$ which is a discontinuous construct in the sense that free occurrences of variables in $a = b$ are discontinuous occurrences. $\preceq$ inherits the discontinuity from $=$ so that free occurrences of variables in $a \preceq b$ are discontinuous, and no λ is allowed to bind a variable occurrence that is free in a subterm of form $a \preceq b$.

We may now state three further inference rules and one more axiom:

Y	Minimality	$ga \preceq a \vdash \mathsf{Y}g \preceq a$.
M	Monotonicity	$b \preceq c \vdash ab \preceq ac$.
E	Extensionality	$\wr gxy = \wr hxy; gxyz = gab; hxyz = hab \vdash gxy = hxy$ if x, y and z are not free in g and h
$!=$	Equality	$!(a = b)$

Rule Y states that $\mathsf{Y}g$ is minimal among all fixed points of g (this does not hold in all κ-premodels, but it holds in the particular one outlined in Appendix A). Rule M states that all maps a are monotonic in the sense that $b \leq c$ implies $ab \leq ac$. Rule E states that two maps g and h are equal if $\mathsf{a}(g)(\overline{x}) = \mathsf{a}(h)(\overline{x})$ for all $\overline{x} \in \mathcal{D}^{<\omega}$ (see [12] for details on the interpretation and applications of rule E). Axiom $!=$ states that $a = b$ is either true or false.

3.6 Simple existential quantification

The quantifier $\bar{\exists}$ is a particularly primitive quantifier. $\bar{\exists}g = \mathsf{T}$ if $gx = \mathsf{T}$ for some $x \in \mathcal{D}$ and $\bar{\exists}g = \bot$ otherwise. Contrary to $\exists$, which is defined in Section B.4, $\bar{\exists}g$ cannot be false. $\epsilon(g)$ is an $x \in \mathcal{D}$ such that $gx = \mathsf{T}$ if such an x exists. More formally, define $E \in \mathcal{D} \rightarrow \mathcal{D}$ by

$$E(g) = \begin{cases} \mathsf{T} & \text{if } \exists x \in \mathcal{D}\colon gx = \mathsf{T} \\ \bot & \text{otherwise} \end{cases}$$

Then define

$$\begin{array}{rcl} \bar{\exists} & = & \lambda(E) \\ \epsilon(g) & = & q(\{x \in \mathcal{D} \mid gx = \mathsf{T}\}) \end{array}$$

Here we use the choice function q from the premodel $(\mathcal{D}, \mathsf{a}, C, q)$. The requirement $q(\emptyset) \in C \subseteq \mathcal{D}$ ensures that $\epsilon(g)$ is a map even when $gx \neq \mathsf{T}$ for all maps x. The following axioms describe $\bar{\exists}$:

$\rightarrow\bar{\exists}$	Existence	$ab \rightarrow \bar{\exists}a$
$\bar{\exists}\rightarrow$	Existence	$\bar{\exists}a \rightarrow a\epsilon(a)$
$?\bar{\exists}$	Existence	$\bar{\exists}a = ?\bar{\exists}a$

Axiom $\rightarrow\bar{\exists}$ expresses that if ab is true for some particular b, then $\bar{\exists}a$ is true. Axiom $\bar{\exists}\rightarrow$ expresses that if ax is true for some x then, in particular, it is true for $x = \epsilon(a)$. Axiom $?\bar{\exists}$ expresses that $\bar{\exists}a$ equals either T or $\bot$. See Appendix B.4 for definitions of $a \rightarrow b$ and $?a$.

3.7 Hilberts ε-operator

Define

$$\begin{array}{lcl} Q(g) & = & \left\{ \begin{array}{ll} \bot & \text{if } \exists x \in C \colon gx = \bot \\ q(\{x \in C \mid gx = \mathsf{T}\}) & \text{otherwise} \end{array} \right. \\ \varepsilon & = & \lambda(Q) \end{array}$$

The map ε is Hilberts ε-operator [13]. The map ε allows to define many auxiliary concepts such as the universal and existential quantifiers (c.f. Appendix B.4) and to develop first order predicate calculus. Furthermore, C is rich enough to allow all sets of ZFC to be modelled by classical maps which allows to define set membership and prove all axioms and inference rules of ZFC inside MTC. The details may be found in [11, 12].

The constructs introduced so far allows to define the map ℓ (the definition is stated in Appendix B.4). Now let

$$C' = \{x \in \mathcal{D} \mid \ell x = \mathsf{T}\}$$

(this C' is actually the same C' as the one defined in Section 2.7). The axioms that describe ε read:

$Q1$	Quantification	$\ell a \wedge \forall b \rightarrow ba$
$Q2$	Quantification	$\varepsilon x \colon a = \varepsilon x \colon \ell x \wedge a$
$Q3$	Quantification	$\ell(\varepsilon x \colon a) = \forall x \colon !a$
$Q4$	Quantification	$!\forall x \colon a = \forall x \colon !a.$

Axiom Q1 states $C' \subseteq C$ and Axiom Q2 states $C \subseteq C'$ so that $C = C'$ as in Section 2.7. Axiom Q2 furthermore expresses Ackermanns axiom ([9], p.244). The definition of ε entails that $\varepsilon g \in C$ if $\forall x {\in} C \colon gx \neq \bot$ and $\varepsilon g = \bot$ if $\exists x {\in} C \colon gx = \bot$. The former is expressed directly by Axiom Q3 and the latter is expressed indirectly by Axiom Q4.

4 Conclusion

The most immediate translation of Russells paradoxical sentence $\{x \mid x \notin x\} \in \{x \mid x \notin x\}$ into MTC is the term $(\lambda x.\neg(xx))(\lambda x.\neg(xx))$ whose value is $\bot$. Hence, Russells paradox is essentially avoided by having a third truth value $\bot$ which is the value of terms that make a computer loop indefinitely. An important point in MTC is that this third truth value can be introduced without loosing the classical, semantic flavour of the theory and without resort to intuitionistic logic.

The above translation of Russells paradox to MTC translates set abstraction to λ-abstraction, which gives insight to how Russells paradox is avoided. As noted in Section 1.3, set abstraction is not the same as λ-abstraction, so one may also look at how Russells paradox is avoided when modelling set abstraction as in [11]. This turns out to be trivial, however, since that modelling ensures that all sets are well-founded so that all sets x satisfy $x \notin x$. Furthermore, that modelling ensures that classes that contain all sets are not sets themselves.

A map x in MTC is classical if $\ell x = \mathsf{T}$ where

$$\ell \doteq \lambda f.f \left\{ \begin{array}{l} \mathsf{T} \\ (\forall x{:}\, \ell(fx)) \wedge \bar{\exists} S{:}\, \ell S \wedge \forall x \forall y{:}\, x \sim^2_S y \Rightarrow fx \sim fy \end{array} \right.$$

The precise structure and meaning of this definition is not important here. The important observation is that the definition of ℓ is recursive in that ℓ occurs on the right hand side of $\doteq$ (recursive definitions are shorthand for definitions that use the fixed point operator Y explicitly).

The two occurrences of ℓ in the definition combined with the minimality of fixed points implies that classical maps are well-founded in two, distinct ways. The first occurrence of ℓ makes classical maps well-founded in the sense that for all classical maps $g, x_1, x_2, \ldots$ there exists an n such that $gx_1 \cdots x_n = \mathsf{T}$. This kind of well-foundedness corresponds to the well-foundedness of sets expressed by the axiom of restriction in ZFC (no infinitely descending $\in$-chains).

The second occurrence of ℓ makes classical maps well-founded in a much more subtle sense. The closest analogue in ZFC to this well-foundedness is the limitation of size present in ZFC. However, maps contain more structure than sets and the second kind of well-foundedness is more a limitation of complexity than just a simple limitation of size. In any case, it is this second kind of well-foundedness that avoids Burali-Fortis paradox.

The first kind of well-foundedness does not avoid paradoxes. If the first kind of well-foundedness is abandoned, if the second kind is kept, and if the representation of sets used in [11] is used, then the representable sets become those of Aczel's AFA set theory [1] (the one in which the equation $X = \{X\}$ has exactly one solution). Hence, it is consistent to allow infinite descending $\in$-chains because the paradoxes are avoided by the second kind of well-foundedness.

As noted by Aczel, all known theories about non-well-foundedness share the peculiarity that they start by constructing a well-founded universe and then proceed to the non-well-founded. It is an interesting topic for further work to try to formulate a version of map theory which does not share this peculiarity. Such a theory could be a theory about non-well-founded, classical maps, i.e. maps that satisfy the second kind of well-foundedness without satisfying the first.

Acknowledgement

My thanks are due to the referee from who's report I copied a paragraph to the abstract.

A Outline of a model of MTC

A.1 The κ-denotational semantics

The following exposition follows [2] except that in the following exposition, the empty set is κ-small and κ-cpos need not have a bottom element.

A p.o $\mathcal{D}$ is a partially ordered set, $(D, \leq)$. We use $x \in \mathcal{D}$ and $x \subseteq \mathcal{D}$ as shorthand for $x \in D$ and $x \subseteq D$, respectively. $\boxed{\downarrow u}$ means $\{v \mid v \leq u\}$. $A \subseteq \mathcal{D}$ is *κ-chain* if every κ-small $B \subseteq A$ is bounded by an element of A. $\mathcal{D}$ is a *κ-cpo* if every κ-chain A has a sup (i.e. least upper bound). $\mathcal{D}$ is a *κ-ccpo* if moreover every bounded A has a sup (hence, like in [2], every κ-ccpo has a bottom). A *κ-compact* element of a κ-cpo $\mathcal{D}$ is an element $u \in \mathcal{D}$ such that, for all κ-chains A, $u \leq \sup(A)$ implies $u \leq v$ for some $v \in A$. $\boxed{\mathcal{D}_c}$ is the set of κ-compact elements and $\boxed{\downarrow_c u} = \downarrow u \cap \mathcal{D}_c$. A κ-ccpo $\mathcal{D}$ is a *κ-Scott domain* iff for all $u \in \mathcal{D}$, $u = \sup \downarrow_c u$. A *prime* element of a κ-ccpo $\mathcal{D}$ is an element u such that, for all bounded A, $u \leq \sup A$ implies $u \leq v$ for some $v \in A$. $\boxed{\mathcal{D}_p}$ is the set of prime elements of $\mathcal{D}$, and $\boxed{\downarrow_p u} = \downarrow u \cap \mathcal{D}_p$. A κ-Scott domain is *κ-prime algebraic* if for all $u \in \mathcal{D}$, $u = \sup \downarrow_p u$.

A.2 Outline of a model

A model of MTC is outlined in the following. Merely the definition of the model will be stated. The satisfaction of the axioms and inference rules remains to be proved.

The model construction has many similarities with the construction in Section 8 of [2]. The construction below differs from that in [2] in the following ways: (1) The domain of the model is constructed as coherent, complete, initial segments of κ-compact elements rather than coherent, initial segments of κ-prime elements and the PCS of κ-prime elements is not constructed at all below. (2) The model construction is based on two fixed point constructions rather than one. The first fixed point generates the σ-compact maps of MTW which is a set large enough to contain the well-founded maps of MTW. Then the well-founded maps are turned into classical maps by discarding information, and then a second fixed point construction is used to generate the κ-compact maps of MTC. Finally, the κ-premodel of MTC is constructed from coherent, initial segments of κ-compact elements.

To simplify the exposition, assume that κ is strongly inaccessible, i.e. assume σ is the least strongly inaccessible ordinal and assume κ is a strongly inaccessible ordinal greater than σ. The assumption that κ is inaccessible rather than just regular is merely a luxury.

For all relations $\preceq$ define $\boxed{\preceq^*}$ by

$$\langle x_1, \ldots, x_m \rangle \preceq^* \langle y_1, \ldots, y_n \rangle \Leftrightarrow m = n \wedge x_1 \preceq y_1 \wedge \cdots \wedge x_m \preceq y_m$$

An *applicative structure* $\mathcal{D}$ is a pair (D, a) where D is a set and $a \in D \to M_D$. For applicative structures $\mathcal{D} = (D, a)$ define $\boxed{\leq_{\mathcal{D}}}$ and $\boxed{M_{\mathcal{D}}}$ by

$$x \leq_{\mathcal{D}} y \Leftrightarrow a(x) \leq a(y)$$
$$M_{\mathcal{D}} = \{g \in M_D \mid \forall x, y \in D^{<\omega} \colon (x \leq^*_{\mathcal{D}} y \Rightarrow a(g)(x) \leq_L a(g)(y))\}$$

An applicative structure $\mathcal{D} = (D, a)$ is said to be *monotonic* if $\perp_D \in \operatorname{rng} a \subseteq M_{\mathcal{D}}$. For all monotonic applicative structures $\mathcal{D} = (D, a)$ define

$$\begin{array}{ll} & D_{\mathcal{D}} = D \cup M_{\mathcal{D}} \\ a^+_{\mathcal{D}} \in D_{\mathcal{D}} \to M_{\mathcal{D}} & a^+_{\mathcal{D}}(x) = \begin{cases} a(x) & \text{if } x \in D \\ x & \text{otherwise} \end{cases} \\ & g \preceq_{\mathcal{D}} h \Leftrightarrow a^+_{\mathcal{D}}(g) \leq a^+_{\mathcal{D}}(h) \\ a_{\mathcal{D}} \in D_{\mathcal{D}} \to M_{D_{\mathcal{D}}} & a_{\mathcal{D}}(g)(x) = \sup \{a^+_{\mathcal{D}}(g)(y) \mid y \in D^{<\omega} \wedge y \preceq^*_{\mathcal{D}} x\} \\ & \mathcal{D}^+ = (D_{\mathcal{D}}, a_{\mathcal{D}}) \end{array}$$

To ensure the soundness of the definition, we make the arbitrary convention that $\sup A = \mathsf{b}$ when $A \subseteq L$ has no supremum.

For applicative structures $\mathcal{D} = (D, a)$ and $\mathcal{D}' = (D', a')$ define $\boxed{\mathcal{D} \sqsubseteq \mathcal{D}'} \Leftrightarrow D \subseteq D' \wedge \forall g{\in}D \forall x{\in}D^{<\omega}\colon a(g)(x) = a'(g)(x)$. We have $\mathcal{D} \sqsubseteq \mathcal{D}^+$. Now define $\boxed{\mathcal{D}^\alpha}$ for all ordinals α by transfinite induction as follows:

$$\begin{array}{lcll} \mathcal{D}^0 & = & \mathcal{D} & \\ \mathcal{D}^{\alpha+1} & = & (\mathcal{D}^\alpha)^+ & \\ \mathcal{D}^\delta & = & \sqcup_{\alpha\in\delta} \mathcal{D}^\alpha & \text{for limit ordinals } \delta \end{array}$$

Above, $\sqcup_{\alpha\in\delta}$ has the obvious interpretaion, i.e. if $\mathcal{D}^\alpha = (D_\alpha, a_\alpha)$ then $D_\delta = \cup_{\alpha\in\delta} D_\alpha$ and $a_\delta(g)(x) = a_\alpha(g)(x)$ whenever $g{\in}D_\alpha$ and $x{\in}D_\alpha^{<\omega}$.

Let $\boxed{\perp}$ be an arbitrary element of the universe of ZFC which is not a function, let $\boxed{\mathcal{D}_1} = (D_1, a_1)$ where $D_1 = \{\perp\}$ and where $a_1 \in D_1 \to (D_1^{<\omega} \to L)$ is defined by $a_1(g)(x) = \mathsf{b}$. Furthermore, let $\boxed{\mathcal{D}_2} = (D_2, a_2) = \mathcal{D}_1^\sigma$. D_2 is essentially the set of σ-compact elements of MTW.

Let $\boxed{S^\infty}$ denote the set of infinite lists of elements of S. For all $x = (x_1, \ldots)$ and $n \in \omega$ let $\boxed{(x|n)}$ denote the tuple $\langle x_1, \cdots, x_n\rangle$.

Define $\boxed{G^\circ} = \{g{\in}\mathcal{D}_2 \mid \forall x{\in}D_2^\infty \exists n{\in}\omega\colon a_2(g)(x|n) = \mathsf{t}\}$, and let $\boxed{\Phi}$ be the least subset of $\mathcal{D}_2$ for which $G^{\circ\circ} \subseteq \Phi$ whenever $G \subseteq \Phi$ is σ-small. This Φ is essentially the set of well-founded maps introduced in [2, 11].

Now let $\boxed{\mathcal{D}_3} = (D_3, a_3)$ where $D_3 = \Phi \cup \{\perp\}$ and where $a_3 \in D_3 \to (D_3^{<\omega} \to L)$ is defined by $a_3(g)(x) = a_2(g)(x)$. $\mathcal{D}_3$ is Φ from which some information is discarded, namely the information about the applicative behaviour of well-founded maps when applied to non-well-founded maps. This makes $\mathcal{D}_3$ useful as a model of the classical maps.

Now let $\boxed{\mathcal{D}_4} = (D_4, a_4) = \mathcal{D}_3^\kappa$. D_4 is essentially the set of κ-compact elements of MTC. Transfinite induction up to κ gives the κ-compact elements because κ is assumed inaccessible. More care would be needed in the definition of $\mathcal{D}^+$ and $\mathcal{D}^\alpha$ if κ was not assumed inaccessible.

$A \subseteq D_4$ is an initial segment if A is non-empty and $x \leq_{D_4} y \wedge y \in A \Rightarrow x \in A$. A is coherent if any two elements of A have an upper bound in D_4. A is complete if any subset of A that has an upper bound in $\mathcal{D}$ also has an upper bound in A. Let D_5 be the set of coherent, complete, initial segments and define the relation $\leq_5$ on D_5 by $x \leq_5 y \Leftrightarrow x \subseteq y$. Define $a_5 \in D_5 \to (D_5^{<\omega} \to L)$ by $a_5(g)(\langle x_1, \ldots, x_n \rangle) = \sup \{a_4(h)(\langle y_1, \ldots, y_n \rangle) \mid h \in g \wedge y_1 \in x_1 \wedge \cdots \wedge y_n \in x_n\}$. Define $C_5 = \{\downarrow x \mid x \in D_3 \setminus \{\perp\}\}$ and let $q \in \mathcal{P}(D_5) \to D_5$ satisfy $q(A) \in A$ when A is non-empty and $q(A) \in C_5$ when A is empty.

$P = ((D_5, \leq_5), a_5, C_5, q_5)$ is claimed to be a κ-premodel which satisfies all axioms and inference rules of MTC.

B Summary of MTC

B.1 Syntax

$$\begin{array}{lll} \mathcal{V} & ::= & x_1 \mid x_2 \mid \cdots \\ \mathcal{T} & ::= & \mathcal{V} \mid \mathsf{T} \mid \mathcal{T}\mathcal{T} \mid \lambda \mathcal{V}.\mathcal{T} \mid \mathsf{if}(\mathcal{T}, \mathcal{T}, \mathcal{T}) \mid \varepsilon \mid \bar{\bar{\exists}} \mid \epsilon(\mathcal{T}) \mid \mathcal{T} = \mathcal{T} \end{array}$$

B.2 Priority

The priority is as follows. Functional application fx has highest priority and appears at the top of the table. Operators on the same line have the same priority.

$$\begin{array}{l} fx \\ x, y \\ x \downarrow y \quad \downarrow x \\ x \sim y \quad x{\in}y \quad x{\in\in}y \quad x \sim^2_S y \\ \neg x \quad \wr x \quad !x \quad ?x \\ x \wedge y \quad x \mathrel{\tilde{\wedge}} y \\ x \vee y \quad x \mathrel{\tilde{\vee}} y \\ x \Rightarrow y \quad x \mathrel{\tilde{\Rightarrow}} y \quad x \Leftrightarrow y \\ x \left\{ \begin{array}{l} y \\ z \end{array} \right. \\ \lambda x.y \\ x = y \quad x \preceq y \\ x \to y \end{array}$$

B.3 Associativity

fx, $x \wedge y$, $x \mathrel{\tilde{\wedge}} y$, $x \vee y$ and $x \mathrel{\tilde{\vee}} y$ are left associative so that e.g. fxy means $(fx)y$.

$x \sim y$, $x{\in}y$, $x{\in\in}y$, $x \sim^2_S y$, $x \Rightarrow y$, $x \mathrel{\tilde{\Rightarrow}} y$, $x \Leftrightarrow y$, $x = y$, $x \preceq y$, and $x \to y$ are "and"-associative so that e.g. $x \sim y \sim z$ means $(x \sim y) \wedge (y \sim z)$ and $x \Rightarrow y \Rightarrow z$ means $(x \Rightarrow y) \wedge (y \Rightarrow z)$.

B.4 Definitions used in axioms

$\mathsf{F} \qquad \doteq \lambda x.\mathsf{T}$

$$x\left\{\begin{array}{l} a \\ b \end{array}\right. \doteq \mathsf{P}abx$$

$$\mathsf{Y} \doteq \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

$$\bot \doteq \mathsf{Y}\lambda x.x$$

$$\neg x \doteq x\left\{\begin{array}{l} \mathsf{F} \\ \mathsf{T} \end{array}\right.$$

$$\wr x \doteq x\left\{\begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array}\right.$$

$$!x \doteq x\left\{\begin{array}{l} \mathsf{T} \\ \mathsf{T} \end{array}\right.$$

$$?x \doteq x\left\{\begin{array}{l} \mathsf{T} \\ \bot \end{array}\right.$$

$$x \mathrel{\tilde{\wedge}} y \doteq x\left\{\begin{array}{l} y \\ \mathsf{F} \end{array}\right.$$

$$x \mathrel{\tilde{\Rightarrow}} y \doteq x\left\{\begin{array}{l} y \\ \mathsf{T} \end{array}\right.$$

$$x \wedge y \doteq x\left\{\begin{array}{l} y\left\{\begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array}\right. \\ y\left\{\begin{array}{l} \mathsf{F} \\ \mathsf{F} \end{array}\right. \end{array}\right.$$

$$x \Rightarrow y \doteq x\left\{\begin{array}{l} y\left\{\begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array}\right. \\ y\left\{\begin{array}{l} \mathsf{T} \\ \mathsf{T} \end{array}\right. \end{array}\right.$$

$$x \Leftrightarrow y \doteq (x \Rightarrow y) \wedge (y \Rightarrow x)$$

$$x \rightarrow y \doteq x \mathrel{\tilde{\wedge}} y = x \mathrel{\tilde{\wedge}} \mathsf{T}$$

$$\exists \doteq \lambda f.\wr f(\varepsilon f)$$

$$\exists x\colon a \doteq \exists \lambda x.a$$

$$\forall x\colon a \doteq \neg\forall x\colon \neg a$$

$$\forall \doteq \lambda f.\forall x\colon fx$$

$$f \sim g \doteq f\left\{\begin{array}{l} g\left\{\begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array}\right. \\ g\left\{\begin{array}{l} \mathsf{F} \\ \forall\lambda x.fx \sim gx \end{array}\right. \end{array}\right.$$

$$x \in y \doteq y\left\{\begin{array}{l} \mathsf{F} \\ \exists z\colon xz \sim y \end{array}\right.$$

$$x \in\in y \doteq \exists z\colon x \in z \wedge z \in y$$

$$\forall x{\in\in}a\colon b \doteq (\lambda y.\forall x\colon x \in\in y \mathrel{\tilde{\Rightarrow}} b)a$$

$$
\begin{array}{ll}
f \sim^2_S g & \doteq f \left\{ \begin{array}{l} g \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array} \right. \\ g \left\{ \begin{array}{l} \mathsf{F} \\ \forall x {\in\in} S \colon fx \sim^2_S gx \end{array} \right. \end{array} \right. \\
\ell & \doteq \lambda f.f \left\{ \begin{array}{l} \mathsf{T} \\ (\forall x \colon \ell(fx)) \wedge \bar{\exists} S \colon \ell S \wedge \forall x \forall y \colon x \sim^2_S y \Rightarrow fx \sim fy \end{array} \right. \\
x \downarrow y & \doteq x \left\{ \begin{array}{l} y \left\{ \begin{array}{l} \mathsf{T} \\ \bot \end{array} \right. \\ y \left\{ \begin{array}{l} \bot \\ \lambda z.xz \downarrow yz \end{array} \right. \end{array} \right. \\
x \preceq y & \doteq x = x \downarrow y
\end{array}
$$

B.5 Axioms and inference rules

Below, a, b, c, d, f and g denote arbitrary terms whereas x, y and z denote arbitrary, distinct variables.

T	Transitivity	$a = b; a = c \vdash b = c.$
SA	Substitutivity	$a = c; b = d \vdash ab = cd.$
$S\lambda$	Substitutivity	$a = b \vdash \lambda x.a = \lambda x.b.$
$S\epsilon$	Substitutivity	$a = b \vdash \epsilon(a) = \epsilon(b).$
$\mathsf{P}\mathsf{T}$	Selection	$\mathsf{P}ab\mathsf{T} = a.$
$\mathsf{P}\lambda$	Selection	$\mathsf{P}ab\lambda x.c = b$
$\mathsf{P}\bot$	Selection	$\mathsf{P}ab\bot = \bot.$
$A\mathsf{T}$	Application	$\mathsf{T}a = \mathsf{T}.$
$A\lambda$	Application	$(\lambda x.a)b = \langle a \mid x{:=}b \rangle$ if b is free for x in a.
$A\bot$	Application	$\bot a = \bot.$
R	Renaming	$\lambda x.\langle a \mid y{:=}x \rangle = \lambda y.\langle a \mid x{:=}y \rangle.$ if x is free for y in a and vice versa.
QND	Quartum Non Datur	$a\mathsf{T} = b\mathsf{T}; a\bot = b\bot; a\lambda y.xy = b\lambda y.xy$ $\vdash ax = bx.$
$Q1$	Quantification	$\ell a \wedge \forall b \rightarrow ba$
$Q2$	Quantification	$\varepsilon x\colon a = \varepsilon x\colon \ell x \wedge a$
$Q3$	Quantification	$\ell(\varepsilon x\colon a) = \forall x\colon !a$
$Q4$	Quantification	$!\forall x\colon a = \forall x\colon !a.$
Y	Minimality	$ga \preceq a \vdash \mathsf{Y}g \preceq a.$
M	Monotonicity	$b \preceq c \vdash ab \preceq ac.$
E	Extensionality	$\wr gxy = \wr hxy; gxyz = gab; hxyz = hab$ $\vdash gxy = hxy$ if x, y and z are not free in g and h
$\rightarrow\bar{\exists}$	Existence	$ab \rightarrow \bar{\exists} a$
$\bar{\exists}\rightarrow$	Existence	$\bar{\exists} a \rightarrow a\epsilon(a)$
$?\bar{\exists}$	Existence	$\bar{\exists} a = ?\bar{\exists} a$
$=\vdash$	Equality	$a = \mathsf{T} \vdash a$
$\vdash=$	Equality	$a \vdash a = \mathsf{T}$
$!=$	Equality	$!(a = b)$

Index

References

[1] P. Aczel. Frege structures and the notions of proposition, truth and set. In J. Barwise, H. J. Keisler, and K. Kunen, editors, *The Kleene Symposiom*, pages 31–59, Amsterdam, 1980. North-Holland.

[2] 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.

[3] C. C. Chang and K. J. Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1973.

[4] A. Church. A set of postulates for the foundations of logic I. *Ann. Math.*, 1933.

[5] A. Church. A set of postulates for the foundations of logic II. *Ann. Math.*, 1934.

[6] A. Church. *The Calculi of Lambda Conversion*. Princeton University Press, 1941.

[7] S. Feferman. Recursion theory and set theory, a marriage of convenience. In J. E. Fenstad, R. O. Gandy, and G. E. Sacks, editors, *Generalized Recursion Theory II, Proceedings of the 1977 Oslo Symposium*, pages 55–98. North-Holland, 1978.

[8] S. Feferman. Toward useful type-free theories.I. *Journal of Symbolic Logic*, 49:75–111, 1984.

[9] U. Felgner. Choice functions on sets and classes. In *Sets and Classes: On the works by Paul Bernays*, pages 217–255. North-Holland, 1976.

[10] R.C. Flagg and J. Myhill. A type-free system extending ZFC. *Annals of Pure and Applied Logic*, 43:79–97, 1989.

[11] K. Grue. Map theory. *Theoretical Computer Science*, 102(1):1–133, July 1992.

[12] K. Grue. Stable map theory. DIKU Report 96/10, Department of Computer Science, University of Copenhagen, DIKU, Universitetsparken 1, DK–2100 Copenhagen, Denmark, April 1996. Available from //www.diku.dk/~grue.

[13] D. Hilbert and P. Bernays. *Grundlagen der Mathematic*, volume 2. Springer-Verlag, 1939.

[14] E. Mendelson. *Introduction to Mathematical Logic*. Wadsworth and Brooks, 3. edition, 1987.

[15] D. Scott. Models for various type-free λ-calculi. In *Studies in Logic and The Foundation of Mathematics*, volume 74, pages 157–187. Proceedings of the IVth Int. Congress for Logic, Methodology and Philosophy of Science, 1973.