Stable map theory

Klaus Grue

April 18, 1996

Abstract

Map theory (MT) is a foundation of mathematics which is based on λ-calculus instead of logic and has at least the same expressive power as ZFC set theory. This paper presents "stable map theory" which is much easier to learn, teach and comprehend than "original map theory" from 1992. Pedagogical simplicity is important since MT is a candidate for a common foundation of classical mathematics and computer science. MT has the benefit that it allows a complete integration of classical mathematics and computer science. As a particular example, the free mixing of quantification and general recursion has many applications.

The long list of well-foundedness axioms in the original version (which corresponds to the list of proper axioms of ZFC) has been replaced by a single definition of "classicality" in the stable version. Furthermore, the stable version has been enhanced by axioms of stability, minimality of fixed points and a particular kind of extensionality. The stability axiom represents a change of semantics compared to original MT which is why the new version is called "stable". The axioms of minimality of fixed points is particularly important since it allows to prove the axiom of transfinite induction in the original version.

This paper presents stable MT and develops original MT in it. As a corollary of this development, also ZFC can be developed in stable MT. In addition, the paper presents an extension of the stable version which resembles NBG but is more general. The paper conjectures two specific structures to be models of stable MT.

The stable version sheds some light on why it is consistent to allow infinite descending chains of the membership relation in a ZF-style system (as Aczel did in his theory of non-well-founded sets). This is because the Burali-Forti paradox is avoided by "limitation" rather than "well-foundedness" as made explicit in the definition of "classical mathematical object".

Contents

1 Introduction

1.1 Summary for impatient readers

This section can be skipped without lack of continuity.

In MT, the syntax of variables v, terms t and well-formed formulas e reads:

$$
\begin{array}{lcl}
\mathsf{v} & ::= & x_1 \mid x_2 \mid x_3 \mid \cdots \\
\mathsf{t} & ::= & \mathsf{v} \mid \mathsf{tt} \mid \hat{\mathsf{v}}\mathsf{t}. \mid \mathsf{T} \mid \mathsf{P} \mid \varepsilon \\
\mathsf{e} & ::= & \mathsf{t} = \mathsf{t}
\end{array}
$$

The construct fx denotes the function f applied to x. $\hat{x}\mathcal{A}.$ denotes lambda abstraction, T represents truth, $\mathsf{P}xy$ is the pair of x and y and εf denotes an x such that $fx = \mathsf{T}$. Well-formed formulas $\mathcal{A} = \mathcal{B}$ are called equations. MT has a number of simple axioms (axioms with no antecedents) such as

Axiom 1.1.1 (PT) $\mathsf{P}ab\mathsf{T} = a$.

and a number of complex axioms (inference rules with at least one antecedent) such as

Axiom 1.1.2 (Sab) $a = c; b = d \vdash ab = cd$.

In MT, a proof is a sequence of equations such as

1	PT	$\mathsf{PTTT} = \mathsf{T}$
2	PT	$\mathsf{P}\varepsilon\mathsf{PT} = \varepsilon$
3	1,2,Sab	$(\mathsf{PTTT})(\mathsf{P}\varepsilon\mathsf{PT}) = \mathsf{T}\varepsilon$

In MT, it is possible to define constructs $\neg a$, $a \Rightarrow b$, $a \in b$ and $\forall_{\mathrm{set}} x\colon b$ such that the following holds:

Theorem 1.1.3 Let A be a closed, well-formed formula of ZFC and let A' be the term of MT obtained by replacing $\neg$, $\Rightarrow$, $\in$ and $\forall$ in ZFC by the corresponding constructs $\neg$, $\Rightarrow$, $\in$ and $\forall_{\mathrm{set}}$, respectively, in MT. Now, if A is provable in ZFC then $A' = \mathsf{T}$ is provable in MT.

As a corollary, if $\neg A$ is provable in ZFC then $(\neg A') = \mathsf{T}$ is provable in MT. In MT, $A' = \mathsf{T}$ and $(\neg A') = \mathsf{T}$ exclude each other which ensures that the above theorem is non-trivial.

1.2 Overview

Section 2 and 3 introduce the stable version of MT. Section 2 treats all computational constructs of stable MT, i.e. functional application (fx), lambda abstraction ($\hat{x}\mathcal{A}.$), and the constructs T and P. Section 3 treats quantification, i.e. the choice construct ε and derived concepts like universal and existential quantification and the notion of "classicality". Section 2 and 3 aim at readers who have seen both a lambda abstraction and a quantifier before. For an introduction that aims at first year university students, see [7]. Section 4 introduces various extensions of the stable version. The extensions will be referred to as non-monotonic, unstable and pure MT, respectively. The extensions resemble NBG set theory but are more general.

Appendix A verifies that all axioms and theorems (except one) in [6] of the original version of MT are provable in the stable version and, in particular, that all theorems of ZFC set theory are provable in the stable version.

Formally, the consistency of the stable version is an open question. Appendix B, however, defines two structures which are both conjectured to be models of the stable version. The two structures are obtained by minor modifications from the models of the original version in [3] and [6]. The tedious verification that the models satisfy all the axioms remains. The second of the structures in Appendix D is furthermore conjectured to be a model of non-monotonic MT. The consistency of the remaining extensions is completely open.

Appendix C gives an example of use of the concepts introduced in Section 4 by defining the notion of well-foundedness from original MT inside pure MT.

Appendix D provides a summary of syntax, priority rules, definitions, and axioms of stable MT. All words that appear in italics and all formulas that appear in a box in the text can be found in the index, which appears at the end right before the list of references.

1.3 Comparison of the original and stable versions

The stable version differs from the original one both by including the axiom of stability stated later and by replacing the notion of "well-foundedness" by "classicality". The notion of classicality has the benefit that it is definable where well-foundedness in the original version had to be expressed by a long list of complicated axioms.

Models of both versions are partially ordered sets, ordered by a relation $u \preceq v$ which is defined later. Elements of the ordered sets are called maps. A set C of maps is called a *chain* if

$$\forall x, y \in C : x \preceq y \vee y \preceq x$$

Informally, any chain of well-founded maps in the original version has a greatest lower bound in the stable version, and the classical maps are exactly the greatest lower bounds of such chains. In particular, all well-founded maps are classical. In other words, the set of classical maps can be found by a closure operation on the collection of well-founded maps. (In models of the original version that contain unstable, well-founded maps, "well-founded" has to be replaced by "stable and well-founded" above).

The basic constructs of the two versions relate as follows:

Original	Stable
$f(x)$	fx
$\lambda x.f$	$\hat{x}f.$
T	T
P	P
ε	ε
ϕ	ℓ

Only the correspondence between ϕ and ℓ is incomplete. The map ϕ is the well-foundedness predicate in the original version whereas ℓ is the classicality predicate in the stable version. Furthermore, ϕ is a basic concept in the original version whereas ℓ is a defined concept in the stable version.

2 Computational constructs

2.1 The syntax of map theory

Well-formed formulas will be referred to as *equations*. The syntax of variables v, terms t and equations e of MT reads:

$$\begin{array}{lll} \mathsf{v} & ::= & x_1 \mid x_2 \mid x_3 \mid \cdots \\ \mathsf{t} & ::= & \mathsf{v} \mid \mathsf{tt} \mid \hat{\mathsf{v}}\mathsf{t}. \mid \mathsf{T} \mid \mathsf{P} \mid \varepsilon \\ \mathsf{e} & ::= & \mathsf{t} = \mathsf{t} \end{array}$$

Hence, functional application fx, abstraction $\hat{x}a.$ and the maps T, P and ε are the fundamental constructs of MT whereas all other constructs are defined.

Section 2 deals with all of stable MT except ε.

2.2 Functions

MT is mainly a theory about functions. In MT, the word "map" is used almost as a synonym for the word "function".

In MT, a function can be *applied* to an argument. The result of applying a function f to an argument x is denoted

$$fx$$

Other notations used in the literature for fx read $f(x)$, $(f\ x)$ [12, 6] and $f`x$ [14]. Functional application is left associative so that fxy means $(fx)y$.

2.3 Abstraction

If f is defined by

$$f = \hat{x}x + 4.$$

then f is the function that satisfies

$$fx = x + 4$$

In general, for arbitrary terms a (that may contain x free), the term $\hat{x}a.$ denotes the function f that satisfies $fx = a$ for all x. (The integer 4 and the addition operator $+$ used in the example above are defined concepts in MT. The definitions will not be stated in the present paper. For definitions, see [7]). Other notations used in the literature for $\hat{x}a.$ read $\lambda x.a$ [4, 2] and $x \mapsto a$.

An *abstraction* construct $\hat{x}a.$ starts with a variable with a hat and ends with a dot. As an example, $\hat{x}ab.$ denotes the function f that satisfies $fx = ab$ whereas $\hat{x}a.b$ denotes the function $\hat{x}a.$ applied to b.

For arbitrary terms a and b and arbitrary variables x, let $\langle a \mid x{:=}b \rangle$ denote the result of replacing all free occurrences of x in a by b. One of the axioms of MT reads

Axiom 2.3.1 ($\hat{x}b$) *$\hat{x}a.b = \langle a \mid x{:=}b \rangle$ if b is free for x in a.*

For a definition of "b is free for x in a" consult [14].

In MT, the meaning of a term is unaffected by renaming of bound variables:

Axiom 2.3.2 (R) *$\hat{x}\langle a \mid y{:=}x \rangle. = \hat{y}\langle a \mid x{:=}y \rangle.$ if x is free for y in a and vice versa.*

2.4 Equations

In this paper, inference rules will be called *complex axioms* and axioms without antecedents will be called *simple axioms*. The benefit is that this allows to refer to the two kinds collectively simply as *axioms*. As examples, Axiom $\hat{x}b$ and R above are simple and Axiom T, Sab and S$\hat{x}$ below are complex.

All well-formed formulas of MT have the form $a = b$ where a and b are terms. Hence, equations are the only kind of formulas that can be proved in MT. The following three axioms describe equality:

Axiom 2.4.1 (T) *$a = b; a = c \vdash b = c$.*

Axiom 2.4.2 (Sab) *$a = c; b = d \vdash ab = cd$.*

Axiom 2.4.3 (S$\hat{x}$) *$a = b \vdash \hat{x}a. = \hat{x}b.$.*

2.5 Truth and pairing

In MT, the word *map* is used almost as a synonym for the word *function*. There are, however, two maps that are not functions. One of these maps is denoted $\boxed{\mathsf{T}}$ and is, among other, used to represent *truth*. The major property of T is that it is not a function.

In MT, all functions f satisfy $f = \hat{x}fx.$. The two maps that are not functions do not share this property. A term of form $\hat{x}a.$ always denotes a function.

MT has a "pairing" function $\boxed{\mathsf{P}}$ with the property that $\mathsf{P}abx$ equals a if $x = \mathsf{T}$ and $\mathsf{P}abx$ equals b if x is a function. This can be expressed by two simple axioms:

Axiom 2.5.1 (PT) $\mathsf{P}ab\mathsf{T} = a$.

Axiom 2.5.2 ($\mathsf{P}\hat{x}$) $\mathsf{P}ab\hat{x}c. = b$

Since T is not a function, it does not make sense to apply it to an argument. Nevertheless, $\mathsf{T}a$ is a syntactically valid term and it is convenient, purely as a matter of convention, to define $\mathsf{T}a$ to have value T:

Axiom 2.5.3 ($\mathsf{T}a$) $\mathsf{T}a = \mathsf{T}$.

2.6 Terms as equations

Whenever a term a occurs in a place where an equation is expected, then a is to be interpreted as the equation $a = \mathsf{T}$.

2.7 Programming

The constructs introduced so far allow to express arbitrary computer programs (i.e. the constructs are computationally complete [11]). The following constructs are convenient for expressing programs:

$$
\begin{array}{lcll}
\boxed{\mathsf{F}} & = & \hat{x}\mathsf{T}. & \textit{falsehood} \\
\boxed{x \left\{ \begin{array}{l} a \\ b \end{array} \right.} & = & \mathsf{P}abx & \textit{selection} \text{ (or } \textit{if-then-else}\text{)} \\
\boxed{\mathsf{Y}} & = & \hat{f}\hat{x}f(xx).\hat{x}f(xx).. & \textit{recursion} \text{ (or } \textit{fixed point} \text{ operator)} \\
\boxed{a, b} & = & \hat{x}\mathsf{P}abx. & \textit{pair} \\
\boxed{x^H} & = & x\mathsf{T} & \\
\boxed{x^T} & = & x\mathsf{F} &
\end{array}
$$

The construct $x \left\{ \begin{array}{l} y \\ z \end{array} \right.$ is the *McCarthy conditional* ([13], p.54).

Pairs are right associative so that a, b, c means $a, (b, c)$. Later, Lemma A.6.23 proves that $\mathsf{P}ab$ is a function for all a and b and, therefore, $a, b = \mathsf{P}ab$ holds for all a and b. However, the slightly more complicated definition $a, b = \hat{x}\mathsf{P}abx.$ is convenient when proving Lemma A.6.20 which is a prerequisite for proving Lemma A.6.23.

In MT, any function *weakly* represents falsehood whereas the map F *strongly* represents falsehood. In general, any concept has one strong and many weak representations, and the strong representation is also one of the weak ones.

A weak and a strong representation together define a Galois connection. As examples, F strongly represents falsehood and

$$x \left\{ \begin{array}{l} a \\ b \end{array} \right. = b$$

whenever x weakly represents falsehood. In particular,

$$\mathsf{F} \left\{ \begin{array}{l} a \\ b \end{array} \right. = b$$

The map T strongly represents truth and is at the same time the only map that weakly represents truth. The map T satisfies

$$\mathsf{T} \left\{ \begin{array}{l} a \\ b \end{array} \right. = a$$

The pair, head and tail operations satisfy

$$\begin{array}{lcl} (x,y)^H & = & x \\ (x,y)^T & = & y \end{array}$$

The fixed point function Y satisfies

$$\mathsf{Y}f = \hat{x}f(xx).\hat{x}f(xx). = f(\hat{x}f(xx).\hat{x}f(xx).) = f(\mathsf{Y}f)$$

where all three equations are instances of Axiom $\hat{x}b$. The fixed point function Y allows to state recursive definitions. As an example, the definition

$$f = \hat{x}x, f(x+1).$$

can be taken as shorthand for

$$f = \mathsf{Y}\hat{f}\hat{x}x, f(x+1)..$$

The function f satisfies

$$f0 = (0, f1) = (0, 1, f2) = (0, 1, 2, f3) = \cdots$$

Hence, $f0$ is the infinite list $(0, 1, 2, 3, \ldots)$.

2.8 Perpetuation

Terms built up from the constructs presented so far are machine computable. More precisely, given a term a, a computer can determine whether $a = \mathsf{T}$ or a is a function. Some terms a, however, causes a computer to *perpetuate* (i.e. *loop indefinitely*, but it is wrong to say that functional programs "loop", and the words "perpetuate", "perpetual" and "perpetuation" tend to produce simpler sentences than "loop indefinitely"). In MT, all perpetual terms (i.e. terms that cause a computer to perpetuate) are equal to the map $\perp$ where $\perp$ is defined shortly. As an example, consider *Russells paradox*:

$$\begin{array}{lcl} \boxed{\neg x} & = & x\left\{\begin{array}{l}\mathsf{F}\\ \mathsf{T}\end{array}\right. \\ R & = & \hat{x}\neg xx.\hat{x}\neg xx. \end{array}$$

The construct $\neg x$ expresses logical negation in that it maps truth to falsehood and vice versa. It has priority below functional application so that $\neg xy$ means $\neg(xy)$. (See Appendix D.2 for a summary of priority rules).

Axiom $\hat{x}b$ gives

$$R = \hat{x}\neg xx.\hat{x}\neg xx. = \neg\hat{x}\neg xx.\hat{x}\neg xx. = \neg R$$

Hence, if R is T then R is a function and vice versa, which shows that R is neither T nor a function. Computation of R by machine leads to the computation

$$R = \neg R = \neg\neg R = \neg\neg\neg R = \cdots$$

which never ends, so R equals $\perp$ in MT.

The definition of $\perp$ reads

$$\boxed{\perp} = \mathsf{Y}\hat{x}x.$$

Section 2.13 introduces a partial ordering of all maps, and Lemma A.6.8 and A.6.9 prove that $\perp$ is the unique minimal map with respect to this ordering.

In MT, any map x is either T, $\perp$ or a function, so any map x satisfies exactly one of the following:

$$\begin{array}{lcll} x & = & \mathsf{T} & \text{or} \\ x & = & \perp & \text{or} \\ x & = & \hat{y}xy. & . \end{array}$$

This is expressed in MT by the *quartum non datur* axiom:

Axiom 2.8.1 (QND) $a\mathsf{T} = b\mathsf{T}; a\perp = b\perp; a\hat{y}xy. = b\hat{y}xy. \vdash ax = bx.$

Computation of $\mathsf{P}abx$ is done by first computing x and then selecting either a or b. Hence, if computation of x never ends, then computation of $\mathsf{P}abx$ never ends either:

Axiom 2.8.2 ($\mathsf{P}\perp$) $\mathsf{P}ab\perp = \perp$.

Likewise, computation of fa is done by first computing f to the extent where it can be seen whether $f = \mathsf{T}$ or f is a function. If that computation never ends, then computation of fa never ends either:

Axiom 2.8.3 ($\perp a$) $\perp a = \perp$.

2.9 T is not the opposite of $\perp$

The similarity of T and $\perp$ is accidental. The construct $\perp$ reads "bottom" whereas T reads "tee" and stands for "truth", not "top".

2.10 Logical connectives

It is convenient to have two versions of logical conjunction:

$$\begin{aligned}
\boxed{x \mathbin{\tilde{\wedge}} y} &= x \left\{ \begin{array}{l} y \\ \mathsf{F} \end{array} \right. \\
\boxed{x \wedge y} &= x \left\{ \begin{array}{l} y \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array} \right. \\ y \left\{ \begin{array}{l} \mathsf{F} \\ \mathsf{F} \end{array} \right. \end{array} \right.
\end{aligned}$$

These two logical conjunctions have different properties. As an example, $\mathsf{F} \mathbin{\tilde{\wedge}} \perp = \mathsf{F}$ whereas $\mathsf{F} \wedge \perp = \perp$. As another example, $x \wedge y = y \wedge x$ holds for all maps x and y (by Axiom QND). In contrast, $x \mathbin{\tilde{\wedge}} y = y \mathbin{\tilde{\wedge}} x$ does not hold e.g. for $x = \mathsf{F}$ and $y = \perp$.

The construct $\boxed{x \rightarrow y}$ is shorthand for the equation

$$x \mathbin{\tilde{\wedge}} y = x \mathbin{\tilde{\wedge}} \mathsf{T}$$

The equation $a \rightarrow b$ holds regardless of b if a is a function or $a = \perp$. If $a = \mathsf{T}$, then $a \rightarrow b$ holds if and only if $b = \mathsf{T}$. Hence, the equation $a \rightarrow b$ expresses "if $a = \mathsf{T}$ then $b = \mathsf{T}$". Likewise, if $\mathcal{E}$ is an equation $y = z$ then $x \rightarrow \mathcal{E}$ is shorthand for the equation

$$x \mathbin{\tilde{\wedge}} y = x \mathbin{\tilde{\wedge}} z$$

and expresses "if $x = \mathsf{T}$ then $\mathcal{E}$".

Some further logical connectives read:

$$\boxed{x \mathrel{\tilde{\vee}} y} = x \begin{cases} \mathsf{T} \\ y \end{cases}$$

$$\boxed{x \vee y} = x \begin{cases} y \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ y \begin{cases} \mathsf{F} \\ \mathsf{F} \end{cases} \end{cases}$$

$$\boxed{x \mathrel{\tilde{\Rightarrow}} y} = x \begin{cases} y \\ \mathsf{T} \end{cases}$$

$$\boxed{x \Rightarrow y} = x \begin{cases} y \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ y \begin{cases} \mathsf{T} \\ \mathsf{T} \end{cases} \end{cases}$$

$$\boxed{x \Leftrightarrow y} = x \begin{cases} y \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ y \begin{cases} \mathsf{F} \\ \mathsf{T} \end{cases} \end{cases}$$

The unary operator, $!x$ is defined by

$$!x = x \begin{cases} \mathsf{T} \\ \mathsf{T} \end{cases}$$

The construct $!x$ equals T unless $x = \bot$ in which case $!x = \bot$.

In the following, a term will be called a *logical term* if it is built up from variables using only the connectives $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$. Any logical term is also a term of classical propositional calculus. Those logical terms that are provable in propositional calculus will be called *tautologies* in the following (a more general definition of tautologies is given in Appendix A.3. Many theorems can be formulated about provability of tautologies. One theorem is: A logical term $\mathcal{A}$ is a tautology if and only if $\mathcal{A} = {!x_1} \wedge \cdots \wedge {!x_n}$ is provable in MT where $x_1, \ldots, x_n$ are the free variables of $\mathcal{A}$. Another theorem is: If $\mathcal{A}$ and $\mathcal{B}$ are logical terms in which exactly the same variables occur free in the two and which both contain at least one logical connective, then $\mathcal{A} \Leftrightarrow \mathcal{B}$ is a tautology iff $\mathcal{A} = \mathcal{B}$ is provable in MT. A third theorem is: If $\mathcal{A}$ and $\mathcal{B}$ are logical terms and if all free variables in $\mathcal{B}$ also occur free in $\mathcal{A}$ then $\mathcal{A} \Rightarrow \mathcal{B}$ is a tautology iff $\mathcal{A} \rightarrow \mathcal{B}$ is provable in MT. All these theorems are consequences of Axiom QND.

2.11 Roots

Now define

$$\boxed{\wr\, \mathrm{x}} = x \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array} \right.$$

For all maps x, $\wr x$ will be referred to as the "root" of x. Hence, the root of T is T, the root of $\perp$ is $\perp$, and the root of any function f is F. The construct $\wr x$ has the same priority as $\neg x$ so that $\wr f x$ means $\wr(f x)$.

The graphical representation presented in the following may aid the intuitive understanding of maps. As an example, consider the map

$$a = \hat{x} x \mathsf{T}.$$

Among other, this map has the following properties:

$$\begin{array}{lclclclcl}
\wr a & = & \mathsf{F} & & & & & & \\
\wr a \mathsf{T} & = & \wr \mathsf{T}\mathsf{T} & = & \wr \mathsf{T} & = & \mathsf{T} & & \\
\wr a \perp & = & \wr \perp \mathsf{T} & = & \wr \perp & = & \perp & & \\
\wr a \hat{u} \mathsf{F}. & = & \wr \hat{u} \mathsf{F}.\mathsf{T} & = & \wr \mathsf{F} & = & \mathsf{F} & & \\
\wr a \hat{u} \mathsf{F}.\perp & = & \wr \mathsf{F} \perp & = & \wr \hat{x} \mathsf{T}.\perp & = & \wr \mathsf{T} & = & \mathsf{T} \\
\wr a \hat{u} \mathsf{F}.\mathsf{T} & = & \wr \mathsf{F}\mathsf{T} & = & \wr \hat{x} \mathsf{T}.\mathsf{T} & = & \wr \mathsf{T} & = & \mathsf{T} \\
\wr a \hat{u} a. & = & \wr \hat{u} a.\mathsf{T} & = & \wr a = \mathsf{F} & & & & \\
\wr a \hat{u} a.\hat{u} a. & = & \wr a & = & \mathsf{F} & & & & \\
\wr a \hat{u} a.\hat{u} a.\hat{u} a. & = & \wr a & = & \mathsf{F} & & & &
\end{array}$$

These properties can be illustrated graphically as follows:

This figure is a *trie* [10], i.e. a tree that has a root and in which nodes as well as edges have labels. Nodes can have the labels T, $\perp$ and F whereas edges can have arbitrary maps as labels.

Such a trie can display the root of $a z_1 z_2 \cdots z_n$ for arbitrary $n \geq 0$ and arbitrary maps $z_1, z_2, \ldots, z_n$. As a mental picture, a map can be interpreted as an infinitely large trie, namely the one that displays the root of $a z_1 z_2 \cdots z_n$ for all $n \geq 0$ and all maps $z_1, z_2, \ldots, z_n$.

Two maps p and q are equal if their trie representations are equal, i.e. if

$$\wr p z_1 z_2 \cdots z_n = \wr q z_1 z_2 \cdots z_n$$

for all $n \geq 0$ and all maps $z_1, z_2, \ldots, z_n$. This can be expressed by an axiom of *extensionality* that says that if $\wr fxyz_1z_2\cdots z_n = \wr gxyz_1z_2\cdots z_n$ for all $n \geq 0$ and all maps $x, y, z_1, z_2, \ldots, z_n$, then $fxy = gxy$ for all maps x and y:

Axiom 2.11.1 (E) *$\wr fxy = \wr gxy; fxyz = fab; gxyz = gab \vdash fxy = gxy$ if x, y and z are not free in f and g.*

(Note that $\neg fx; \neg gx; fxy = gxy \vdash f = g$ so in many applications one may conclude $f = g$ right after using Axiom E.

To see how the axiom works, suppose the terms f, g, a and b satisfy

$$\begin{aligned} \wr fxy &= \wr gxy \\ fxyz &= fab \\ gxyz &= gab \end{aligned}$$

Now let $u = \hat{x}\hat{y}\hat{z}a...$ and $v = \hat{x}\hat{y}\hat{z}b...$. With these definitions, $uxyz = a$ and $vxyz = b$, so

$$\begin{aligned} fxyz &= f(uxyz)(vxyz) \\ gxyz &= g(uxyz)(vxyz) \end{aligned}$$

To see, e.g., that $\wr fxyz_1z_2z_3 = \wr gxyz_1z_2z_3$, proceed as follows. First define

$$\begin{array}{lllllll} x_1 &=& uxyz_1 & \quad & y_1 &=& vxyz_1 \\ x_2 &=& ux_1y_1z_2 & & y_2 &=& vx_1y_1z_2 \\ x_3 &=& ux_2y_2z_3 & & y_3 &=& vx_2y_2z_3 \end{array}$$

With these definition,

$$\begin{aligned} \wr fxyz_1z_2z_3 &= \wr fx_1y_1z_2z_3 = \wr fx_2y_2z_3 = \wr fx_3y_3 \\ &= \wr gx_3y_3 = \wr gx_2y_2z_3 = \wr gx_1y_1z_2z_3 = \wr gxyz_1z_2z_3 \end{aligned}$$

In general, $\wr fxyz_1z_2\cdots z_n = \wr gxyz_1z_2\cdots z_n$ for all $n \geq 0$ which entails $fxy = gxy$.

2.12 Computability

An important property of MT is *computability*. A term $\mathcal{A}$ will be called *computable* if it does not contain ε. Computable terms $\mathcal{A}$ have the property that $\wr\mathcal{A}$ is computable in the following sense: There exists a computer program which, given a computable term $\mathcal{A}$, returns T if $\wr\mathcal{A} = \mathsf{T}$, returns F if $\wr\mathcal{A} = \mathsf{F}$, and loops indefinitely if $\wr\mathcal{A} = \perp$. The computation can be done e.g. by normal order reduction [2].

2.13 Order

Any map has a root which is either T, F of $\perp$. These roots are *ordered* by the relation $\boxed{x \preceq y}$ as follows:

The diagram says that $\perp \preceq \mathsf{T}$ and $\perp \preceq \mathsf{F}$ whereas $\mathsf{T} \not\preceq \mathsf{F}$ and $\mathsf{F} \not\preceq \mathsf{T}$. As one could expect, $\mathsf{T} \preceq \mathsf{T}$, $\mathsf{F} \preceq \mathsf{F}$, $\perp \preceq \perp$, $\mathsf{T} \not\preceq \perp$ and $\mathsf{F} \not\preceq \perp$.

The $\preceq$ relation is extended to all maps as follows: $f \preceq g$ iff

$$\wr f z_1 z_2 \cdots z_n \preceq \wr g z_1 z_2 \cdots z_n$$

for all $n \geq 0$ and all maps $z_1, z_2, \ldots, z_n$ (c.f. Section 2.11). It is straightforward to see the following:

$$\begin{array}{l} a \preceq a \\ a \preceq b; b \preceq a \vdash a = b \\ a \preceq b; b \preceq c \vdash a \preceq c \end{array}$$

These properties are provable from the definition of $\preceq$ which is stated in a moment.

Any two maps a and b have a greatest lower bound $a \downarrow b$ w.r.t. $\preceq$. This greatest lower bound can be defined as follows:

$$\boxed{x \downarrow y} = x \begin{cases} y \begin{cases} \mathsf{T} \\ \perp \end{cases} \\ y \begin{cases} \perp \\ \hat{z} x z \downarrow y z. \end{cases} \end{cases}$$

The construct $a \preceq b$ is taken to be shorthand for the equation $a = a \downarrow b$. As shown later, $x \downarrow x = x$, $x \downarrow y = y \downarrow x$ and $x \downarrow (y \downarrow z) = (x \downarrow y) \downarrow z$ follow from axiom E, and these three equations entail $x \preceq x$, $x \preceq y; y \preceq x \vdash x = y$ and $x \preceq y; y \preceq z \vdash x \preceq z$, respectively. The monotonicity property $x \preceq y \vdash fx \preceq fy$ follows from the stability axiom stated later.

2.14 Minimal fixed points

Axiom $\hat{x}b$ allows to prove $f(\mathsf{Y}f) = \mathsf{Y}f$. However, $\mathsf{Y}f$ is not just an arbitrary fixed point of f, i.e. an arbitrary solution x to $fx = x$. Rather, $\mathsf{Y}f$ is the *minimal fixed point*:

Axiom 2.14.1 (Y) $fa \preceq a \vdash \mathsf{Y}f \preceq a$.

As an example, $\hat{x}x.a = a$ so $\mathsf{Y}\hat{x}x. \preceq a$ for all a. Since $\perp$ is defined by $\perp = \mathsf{Y}\hat{x}x.$ this shows

$$\perp \preceq a$$

for all a, so $\perp$ is minimal among all maps w.r.t. $\preceq$.

The definition of classicality stated in Section 3.12 is recursive and, hence, implicitly includes Y in its definition. As shown in Theorem A.7.1 in Appendix A, transfinite induction follows from the definition of classicality and Axiom Y.

2.15 Committedness

The stability property of the stable version was originally inspired by concepts from parallel programming. However, stability turned out to simplify MT dramatically, and this is the reason why the stable version is of interest.

A parallel program may start up several computations in parallel. Such parallel computations will be referred to as *processes*.

If a program P starts up a number of processes, then P will be said to be *committed* if it waits for all processes to terminate before P itself terminates (i.e. if P commits itself to wait for all processes it starts). On the contrary, P will be said to be *speculative* if it may terminate and deliver its result without waiting for all processes to terminate. Stability in the stable version is a formalisation and generalisation of committedness.

As an example, consider parallel committed disjunction $x \vee_c y$ and parallel speculative disjunction ("parallel or") $x \vee_s y$. Committed disjunction $x \vee_c y$ computes x and y in parallel. When the truth values of both x and y are known, the construct returns their disjunction. If computation of x or y or both perpetuates, then the construct never returns a value. In other words, the truth table of committed disjunction looks as follows:

$\vee_c$	T	F	$\perp$
T	T	T	$\perp$
F	T	F	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$

Speculative disjunction also computes x and y in parallel, but if one of the two yields T, then the construct returns T without waiting for the other. Hence, the truth table is as follows:

$\vee_s$	T	F	$\perp$
T	T	T	T
F	T	F	$\perp$
$\perp$	T	$\perp$	$\perp$

Speculative parallelism makes it difficult or impossible to predict the amount of computing resources consumed by a program. As an example, consider two expressions u and v, where u evaluates to T and v consumes more and more computer memory without ever terminating. Speculative disjunction of u and v yields T if u manages to complete before v consumes all of the memory of the computer, and the disjunction cannot complete otherwise. In contrast, committed disjunction will never complete when applied to u and v which is more deterministic.

For this and other reasons, speculative parallelism is often avoided in programming languages. (Formally, programming languages with and without speculative parallelism are equally powerful). The stability axiom stated later is a formalisation and generalisation of committedness, and it rules out constructs with speculative parallelism like $x \vee_s y$.

The committed disjunction $x \vee_c y$ equals the construct $x \vee y$ introduced earlier. That construct was defined by

$$x \vee y = x \begin{cases} y \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ y \begin{cases} \mathsf{F} \\ \mathsf{F} \end{cases} \end{cases}$$

Computation of $x \vee y$ using this definition would be sequential in that it would cause first x and then y to be computed. On the contrary, $x \vee_c y$ causes x and y to be computed in parallel. Nevertheless, $x \vee y$ and $x \vee_c y$ are semantically equal in the sense that they yield the same result in all cases. Concepts like computation order are syntactical of nature. The stability axiom stated later is a purely semantical one, but its motivation is somewhat syntactical of nature in that it considers computation processes.

2.16 Stable lazy semantics

For all tuples $\underline{u} = \langle u_1, u_2, \ldots, u_n \rangle$ of maps let $a\underline{u}$ denote $a u_1 u_2 \cdots u_n$. According to Axiom E, two maps a and b are equal if $\wr a\underline{u} = \wr b\underline{u}$ for all tuples $\underline{u}$ of maps. From now on, $\underline{u}$ and $\underline{v}$ tacitly range over tuples of maps.

Consider two maps f and x and let $y = fx$. The map x is fully determined by the values of $\wr x\underline{u}$ for all $\underline{u}$. Likewise, y is fully determined by $\wr y\underline{v}$ for all $\underline{v}$. Hence, the map f can be interpreted as a mapping which, given values for $\wr x\underline{u}$ can compute the values of $\wr y\underline{v}$.

In general, the value of $\wr y\underline{v}$ for some $\underline{v}$ will depend on the value of $\wr x\underline{u}$ for some but not all $\underline{u}$. As an example, for the ε construct introduced in Section 3, εx depends on $\wr x\underline{u}$ for all singleton tuples $\underline{u} = \langle u_1 \rangle$ for which u_1 is classical and depends on $\wr x\underline{u}$ for no other $\underline{u}$.

The map f can be interpreted as a hypercomputer that accesses the value of $\wr x\underline{u}$ for a range of tuples $\underline{u}$ in order to compute $\wr y\underline{v}$. This range of tuples will be referred to as the "inner" range of f. The "limited size" property of classical maps essentially says that the inner range of classical maps have limited size. The inner range depends not only on f but also on x and $\underline{v}$. The computer is "hyper" in the sense that the range of tuples may be infinitely large which is more than real world computers can cope with. As an example, ε is more powerful than any real world computer.

The semantics of MT is committed in the sense that if f accesses the value of $\wr x\underline{u}$ in order to computer $\wr y\underline{v}$ and if $\wr x\underline{u} = \bot$ then $\wr y\underline{v} = \bot$. In other words, if f starts up computation of $\wr x\underline{u}$, then it also commits itself to wait for the result, even if the result never emerges. The semantic equivalent and generalisation of committedness stated in a while will be referred to as *stability*.

Stability rules out certain constructs (like the speculative parallel disjunction $\vee_s$ mentioned in the previous section) but does not weaken MT. Rather, stability makes MT clearer and cleaner than it would otherwise be.

Note that stability does not rule out a construct like P. The value of $\wr\mathsf{P}abx$ depends on $\wr a$ and $\wr x$ if $\wr x = \mathsf{T}$ and depends on $\wr b$ and $\wr x$ if $\wr x = \mathsf{F}$. The function P first computes $\wr x$ and then computes either $\wr a$ or $\wr b$ in order to compute $\wr\mathsf{P}abx$. This does not contradict stability. It is just an example of an inner range that depends on the argument x.

Stability can be expressed thus:

Axiom 2.16.1 (C) $a \preceq c; b \preceq c \vdash fa \downarrow fb = f(a \downarrow b)$.

The connection between stability and the axiom is as follows: Suppose $a \preceq c$ and $b \preceq c$, let f be a map and let $\underline{v}$ be a tuple of maps. We shall now make $\wr(fa \downarrow fb)\underline{v} = \wr f(a \downarrow b)\underline{v}$ plausible.

Computation of $\wr f(a \downarrow b)\underline{v}$ requires computation of $(a \downarrow b)\underline{u}$ for a range S of tuples $\underline{u}$. This range S is the inner range of f for the arguments $a \downarrow b$ and $\underline{v}$. There are two possibilities: $\wr(a \downarrow b)\underline{u} \neq \bot$ for all $\underline{u} \in S$ or $\wr(a \downarrow b)\underline{u} = \bot$ for some $\underline{u} \in S$.

(1) If $\wr(a \downarrow b)\underline{u} \neq \bot$ for all $\underline{u} \in S$ then $\wr a\underline{u} = \wr b\underline{u} = \wr c\underline{u} = \wr(a \downarrow b)\underline{u}$ for all $\underline{u} \in S$ because $a \downarrow b \preceq a \preceq c$ and $a \downarrow b \preceq b \preceq c$. In this case, computation of $fx\underline{v}$ proceeds the same way for $x = a$, $x = b$, $x = c$ and $x = a \downarrow b$ because $fx\underline{v}$ only depends on $\wr x\underline{u}$ for values of $\underline{u}$ where $\wr a\underline{u} = \wr b\underline{u} = \wr c\underline{u} = \wr(a \downarrow b)\underline{u}$. Hence, $\wr fa\underline{v} = \wr fb\underline{v} = \wr fc\underline{v} = \wr f(a \downarrow b)\underline{v}$ which entails $\wr(fa \downarrow fb)\underline{v} = \wr f(a \downarrow b)\underline{v}$ (because $(fa \downarrow fb)\underline{v} = fa\underline{v} \downarrow fb\underline{v}$ when $a \preceq c$ and $b \preceq c$).

(2) If $\wr(a \downarrow b)\underline{u} = \bot$ for some $\underline{u} \in S$, then $\wr a\underline{u} = \bot$ or $\wr b\underline{u} = \bot$ (or both). Without loss of generality assume $\wr a\underline{u} = \bot$. Since $a \downarrow b \preceq a$, computation of $\wr fx\underline{v}$ proceeds the same way for $x = a$ and $x = a \downarrow b$. Hence, $\wr fx\underline{v}$ depends on $\wr x\underline{u}$ both for $x = a$ and $x = a \downarrow b$, so it follows from the stability that $\wr fa\underline{v} = \bot$ and $\wr f(a \downarrow b)\underline{v} = \bot$ from which $\wr(fa \downarrow fb)\underline{v} = \wr f(a \downarrow b)\underline{v}$ follows.

To conclude: maps in MT are stable and this is expressed by Axiom C. Trivially, maps are also “lazy” [8]. This requires no further axioms since it is expressed in Axiom PT, $\mathsf{P}\hat{x}$ and $\hat{x}b$.

Two maps a and b are *compatible* if there is a map c such that $a \preceq c$ and $b \preceq c$. Hence, the stability axiom can be stated: if a and b are compatible, then $fa \downarrow fb = f(a \downarrow b)$.

2.17 Monotonicity

A particularly important consequence of stability is monotonicity:

Theorem 2.17.1 $a \preceq b \vdash fa \preceq fb$.

Proof. By stability, $a \preceq b$ and $b \preceq b$ imply $f(a \downarrow b) = fa \downarrow fb$. Further, $a \preceq b$ means $a = a \downarrow b$ so $fa = f(a \downarrow b) = fa \downarrow fb$, and $fa = fa \downarrow fb$ is exactly what $fa \preceq fb$ means. $\diamondsuit$

A more formal proof may be found in Appendix A where it is also proved that $f \preceq g; a \preceq b \vdash fa \preceq gb$ and $a \preceq b \vdash \hat{x}a. \preceq \hat{x}b.$.

3 Quantification

3.1 Introduction

Section 3 introduces a distinction between *classical* and *non-classical* maps and studies the properties of the collection $\boxed{\mathcal{C}}$ of all classical maps. Furthermore, Section 3 introduces the choice construct ε, the universal quantifier $\forall$ and the existential quantifier $\exists$ which all quantify over $\mathcal{C}$.

All "classical" mathematical objects like integers, real numbers, sets and differentiable manifolds just to name a few are representable by classical maps. The scheme of representation of all sets of ZFC given in [6] also works in the stable version.

As examples, T and F are classical maps whereas $\perp$, $\hat{x}x.$, Y and maps that represent proper classes and non-well-founded sets are non-classical. Classical maps satisfy Tertium Non Datur: Any classical map a is either T or a function.

The collection $\mathcal{C}$ will be defined as the smallest collection of maps which contains T and which satisfies a certain closure property. A definition of this kind tacitly assumes that such a collection exists, which is no innocent assumption. The assumption that $\mathcal{C}$ exists has consistency power greater than or equal to that of ZFC. If the conjectures in Appendix B hold then the consistency power is less than or equal that of ZFC+SI where $\boxed{\text{SI}}$ assumes the existence of an inaccessible ordinal (so, in particular, stable MT is consistent if ZFC+SI is consistent).

3.2 Auxiliary concepts

The term *collection* will be used about any collection of maps whatsoever whereas the terms *class* and *set* will be reserved for particular kinds of collections.

For the time being, the collection $\mathcal{C}$ of all classical maps will be taken as granted. For the time being it is sufficient to know that T is classical and $\perp$ is not.

In the following, a number of auxiliary concepts are introduced:

Moderate. A map f is said to be *moderate* if fx is classical whenever x is classical.

Defined. A map u is said to be *defined* if $u \neq \perp$.

Total. A map f is said to be *total* if fx is defined for all classical x.

Satisfiable. A map f is said to be *satisfiable* if $fx = \mathsf{T}$ for some classical x.

Three-valued. A map u is said to be *three-valued* if u equals T, F or $\perp$.

Two-valued. A map u is said to be *two-valued* if u equals T or $\perp$.

Three-valued function. A map f is a *three-valued function* if f is a function (i.e. not T and $\perp$) and fx is three-valued for all maps x.

Two-valued function. A map f is a *two-valued function* if f is a function and fx is two-valued for all maps x.

The lemmas and proofs stated in the following are somewhat informal of nature. Appendix A contains the corresponding formal proofs.

Lemma 3.2.1 If f is not total then $f\perp = \perp$.

Proof. If f is not total then $fx = \perp$ for some classical x. Now $\perp \preceq x$ so monotonicity gives $f\perp \preceq fx \preceq \perp$ which implies $f\perp = \perp$. $\diamondsuit$

Lemma 3.2.2 If u and v are two-valued and $u = \mathsf{T}$ iff $v = \mathsf{T}$ then $u = v$.

Proof. If $u = \mathsf{T}$ then $u = \mathsf{T} = v$. If $u \neq \mathsf{T}$ then $u = \perp = v$. $\diamondsuit$

Lemma 3.2.3 If f and g are two-valued functions and if $fx = \mathsf{T}$ iff $gx = \mathsf{T}$ then $f = g$.

Proof. $fx = gx$ follows from the previous lemma and then $f = g$ by extensionality. $\diamondsuit$

Now recall that

$$\begin{array}{rcl} \wr x & = & x\left\{\begin{array}{l}\mathsf{T}\\ \mathsf{F}\end{array}\right. \\ !x & = & x\left\{\begin{array}{l}\mathsf{T}\\ \mathsf{T}\end{array}\right. \end{array}$$

Further define

$$\boxed{?x} = x\left\{\begin{array}{l}\mathsf{T}\\ \perp\end{array}\right.$$

The following are trivial:

Lemma 3.2.4 $\wr x$ is three-valued.

Lemma 3.2.5 u is three-valued iff $u = \wr u$.

Lemma 3.2.6 $?x$ is two-valued.

Lemma 3.2.7 u is two-valued iff $u = ?u$.

Lemma 3.2.8 u is defined iff $!u = \mathsf{T}$.

3.3 Choice and quantification

The choice construct $\boxed{\varepsilon}$ in MT satisfies:

- $\varepsilon f = \bot$ if f is not total.
- εf is classical if f is total.
- $f(\varepsilon f) = \mathsf{T}$ if f is total and satisfiable.

The ε construct is Hilbert's ε-operator [9]. The existential quantifier $\exists$ in MT satisfies:

- $\exists f = \bot$ if f is not total.
- $\exists f = \mathsf{T}$ if f is total and satisfiable.
- $\exists f = \mathsf{F}$ if f is total but not satisfiable.

In MT, ε is a fundamental concept whereas $\exists$ is a defined one:

$$\boxed{\exists} = \hat{f} \wr f(\varepsilon f).$$

Using the lemmas in the previous section it is straightforward to verify that the properties of $\exists$ listed above follow from this definition and the properties of ε. (To see that $\exists f = \bot$ if f is not total it is also necessary to use that $\bot$ is not classical).

The universal quantifier $\forall$ in MT satisfies:

- $\forall f = \bot$ if f is not total.
- $\forall f = \mathsf{T}$ if f is total and $fx = \mathsf{T}$ for all classical x.
- $\forall f = \mathsf{F}$ if f is total and $fx \neq \mathsf{T}$ for some classical x.

In MT, $\forall$ is a defined concept:

$$\boxed{\forall} = \hat{f} \neg \exists \hat{x} \neg fx..$$

It seems reasonable to read terms like $\varepsilon \hat{x} \mathcal{A}.$, $\exists \hat{x} \mathcal{A}.$ and $\forall \hat{x} \mathcal{A}.$ as follows:

- $\varepsilon \hat{x} \mathcal{A}.$ reads "an x such that $\mathcal{A}$".
- $\exists \hat{x} \mathcal{A}.$ reads "there exists an x such that $\mathcal{A}$".
- $\forall \hat{x} \mathcal{A}.$ reads "for all x, $\mathcal{A}$".

In reading the constructs this way it is tacitly understood that x ranges over classical maps and that $\hat{x}\mathcal{A}.$ is total.

One could imagine an existential quantifier $\tilde{\exists}$ for which $\tilde{\exists} f = \mathsf{T}$ iff f is satisfiable. However, such a quantifier would not be stable and has no place in stable MT. See sections 3.15 and 4.2 for further discussion of $\tilde{\exists}$.

3.4 Covariant collections

In MT, collections are represented by maps. Two different schemes of representation will be used, and these two schemes will be referred to as the *covariant* and *contravariant* representations, respectively. The covariant representation is the simplest and is the one normally used when representing collections in lambda calculus (c.f. [15, 16]). In MT, contravariant representation will be used for representing sets and is the key to modelling ZFC in MT. The covariant representation is treated below whereas the contravariant one is treated in Section 3.9.

The *co-range* of a map f is the collection of maps x for which $fx = \mathsf{T}$. If the collection S is the co-range of the map f then f is said to be a covariant representation of S.

As examples, T covariantly represents the collection of all maps and so does F (since $\mathsf{F} = \hat{x}\mathsf{T}.$). The map $\perp$ represents the empty collection and so does e.g. $\hat{x}\perp.$ and $\hat{x}\mathsf{F}.$. In the covariant representation we have:

$$\begin{array}{ll} fx & \text{models } x \in S \\ \hat{x}\mathcal{A}. & \text{models } \{x \mid \mathcal{A}\} \end{array}$$

Not all collections have covariant representations. As an example, the collection $\{\perp\}$ that merely contains $\perp$ has no covariant representation since, by monotonicity, no map f satisfies $fx = \mathsf{T}$ iff $x = \perp$.

A collection can have many covariant representations, but at most one that is a two-valued function. If f is a two-valued function and S is its co-range, then f is said to be the *characteristic function* of S. The characteristic function f satisfies

$$fx = \begin{cases} \mathsf{T} & \text{if } x \text{ is in } S \\ \perp & \text{otherwise} \end{cases}$$

If f is a covariant representation of S, then $\hat{x}?fx.$ is the unique characteristic function of S.

The collection $\mathcal{C}$ of classical maps has a covariant representation ℓ, which is defined later. Actually, ℓ will become the definition of "classical".

Since not all collections have covariant representations, there is a tacit assumption in saying that $\mathcal{C}$ has a covariant representation. This assumption has several important consequences. One is that if x is classical and $x \preceq y$ then y is classical by monotonicity. The argument is that if $x \preceq y$ then $\ell x \preceq \ell y$ by monotonicity, so if $\ell x = \mathsf{T}$ then $\ell y = \mathsf{T}$. Another consequence is that if x and y are compatible classical maps then $x \downarrow y$ is classical by stability. The argument is that stability gives $\ell(x \downarrow y) = \ell x \downarrow \ell y$ so if $\ell x = \mathsf{T}$ and $\ell y = \mathsf{T}$ then $\ell(x \downarrow y) = \mathsf{T}$.

As will be seen later, ℓ is a three-valued function. The characteristic function $\tilde{\ell}$ of $\mathcal{C}$ can be defined by

$$\tilde{\ell} = \hat{f}?\ell f.$$

To say that f is a covariant collection formally just means that f is a map. Informally, however, it in addition means that f should be interpreted as a covariant representation of a collection S. Now define

$$\begin{array}{lcl} \boxed{\mathsf{total}} & = & \hat{f}\forall\hat{x}!fx. \\ \boxed{\mathsf{moderate}} & = & \hat{f}\forall\hat{x}\ell(fx). \end{array}$$

The maps total and $\mathsf{moderate}$ are covariant collections of the total and moderate maps, respectively. The map total is a two-valued function so it is the characteristic function of the total maps. The map $\mathsf{moderate}$ is a three-valued function.

Now let $x\dot{\in}S$ denote that x belongs to the covariant collection S:

$$\boxed{x\dot{\in}S} = Sx$$

With these definitions,

$$\begin{array}{lcl} f\dot{\in}\mathsf{total} & = & \forall\hat{x}!fx. \\ f\dot{\in}\mathsf{moderate} & = & \forall\hat{x}\ell(fx). \end{array}$$

3.5 Quantification axioms

εf is classical iff f is total:

$$\ell(\varepsilon f) = f\dot{\in}\mathsf{total}$$

Written out using $\hat{x}a.$ for f this reads:

Axiom 3.5.1 (Q3) $\ell(\varepsilon\hat{x}a.) = \forall\hat{x}!a.$

$\forall\hat{x}fx.$ is defined if f is total:

$$!\forall\hat{x}fx. = f\dot{\in}\mathsf{total}$$

Written out using $\hat{x}a.$ for f this reads:

Axiom 3.5.2 (Q4) $!\forall\hat{x}a. = \forall\hat{x}!a.$

If a is classical and fx is true for all classical x then fa is true:

$$\ell a \wedge \forall\hat{x}fx. \rightarrow fa$$

Written out using $b = \hat{x}fx.$ this reads:

Axiom 3.5.3 (Q1) $\ell a \wedge \forall b \rightarrow ba$

The value of εf only depends on the root of fx and only for classical x. Since fx and $\ell x \wedge fx$ have the same root for all classical x this implies

$$\varepsilon\hat{x}fx. = \varepsilon\hat{x}\ell x \wedge fx.$$

Written out using $a = fx$ this reads:

Axiom 3.5.4 (Q2) $\varepsilon\hat{x}a. = \varepsilon\hat{x}\ell x \wedge a.$

Axiom Q2 both expresses Ackermann's axiom ([5], p.244) and also expresses that ε quantifies over no more than $\mathcal{C}$. On the contrary, Axiom Q1 expresses that ε quantifies over no less than $\mathcal{C}$. Axiom Q1 also expresses that εf, when possible, finds a classical x such that $fx = \mathsf{T}$. Axiom Q3 and Q4 express that εf depends on $\wr fx$ for all classical x, i.e. that if $fx = \perp$ for some classical x then $\varepsilon f = \perp$. Furthermore, Axiom Q3 expresses that εf chooses classical maps whenever $\varepsilon f \neq \perp$.

3.6 Well-foundedness

Let S be a collection of maps. For convenience, a map x will be called an S-*map* in the following if x belongs to S. Furthermore, *with respect to* will be abbreviated *w.r.t.*

A map f is said to be *well-founded* w.r.t. a collection S if the following condition holds:

- For any infinite sequence $x_1, x_2, \ldots$ of S-maps there exists a non-negative integer n such that $fx_1x_2\cdots x_n = \mathsf{T}$.

As an example, the covariant collection $\mathcal{C}^\circ$ of maps f that are well-founded w.r.t. $\mathcal{C}$ can be defined recursively as follows:

$$\mathcal{C}^\circ = \hat{f}f\left\{\begin{array}{l}\mathsf{T}\\ \forall\hat{x}fx\dot{\in}\mathcal{C}^\circ\end{array}\right. .$$

All classical maps happen to be well-founded w.r.t. $\mathcal{C}$ as expressed by the theorem of transfinite induction (Theorem A.7.1).

3.7 Equivalence

Two maps f and g are said to be *equivalent* w.r.t. a collection S if f and g are well-founded w.r.t. S and the following condition holds:

> For any finite (possibly empty) sequence $x_1, x_2, \ldots, x_n$ of S-maps, $\wr fx_1x_2\cdots x_n = \wr gx_1x_2\cdots x_n$.

Equivalence w.r.t. S is an equivalence relation on all maps that are well-founded w.r.t. S.

The requirement that f and g are well-founded ensures that $\wr fx_1\cdots x_n = \wr gx_1\cdots x_n$ could also be written e.g. $fx_1\cdots x_n \Leftrightarrow gx_1\cdots x_n$.

3.8 Classical equality

Equivalence w.r.t. $\mathcal{C}$ is definable in MT:

$$f \sim g = f \begin{cases} g \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ g \begin{cases} \mathsf{F} \\ \forall \hat{x} fx \sim gx. \end{cases} \end{cases}$$

The relation $\sim$ is an equivalence relation over $\mathcal{C}^\circ$. Since $\mathcal{C}$ is a subcollection of $\mathcal{C}^\circ$, $f \sim g$ is also an equivalence relation over all classical maps as expressed by Lemma A.9.1, A.9.3 and A.9.4 in Appendix A:

$$\forall \hat{x} x \sim x.$$
$$\forall \hat{x} \forall \hat{y} x \sim y \Rightarrow y \sim x..$$
$$\forall \hat{x} \forall \hat{y} \forall \hat{z} x \sim y \wedge y \sim z \Rightarrow x \sim z...$$

The collection $\mathcal{C}$ of classical maps happens to be closed under $\sim$ in the sense that if x is classical and $x \sim y$ then y is classical.

The relation $f \sim g$ will be referred to as *classical equality*, and f and g will be said to be classically equal if $f \sim g$. Informally speaking, classical mathematics is obtained by restricting all variables to be classical maps and by identifying classically equal maps. This will be elaborated a bit in Section 3.11.

3.9 Contravariant collections

The *range* of a map f is the collection of maps x for which $x \sim fy$ for some classical y. If the collection S is the range of the map f then f is said to be a *contravariant* representation of S. As an exception to this, the range of T is defined to be the empty collection $\emptyset$ so T contravariantly represents $\emptyset$:

$$\boxed{\emptyset} = \mathsf{T}$$

The range of the identity function $\hat{x}x.$ contains all classical maps and no more, so $\hat{x}x.$ is a contravariant representation of $\mathcal{C}$. From now on, $\mathcal{C}$ will be taken to be this contravariant representation:

$$\boxed{\mathcal{C}} = \hat{x}x.$$

The membership relation is easy to express in MT:

$$\boxed{x \in f} = f \begin{cases} \mathsf{F} \\ \exists \hat{y} x \sim fy. \end{cases}$$

As can be seen, $x \in f$ equals T iff x belongs to the range of f, i.e. iff x belongs to the contravariant collection f.

To say that f is a contravariant collection formally just means that f is a map. Informally, however, it in addition means that f should be interpreted as a contravariant representation of a collection.

A *set* is a contravariant collection which is also a classical map, and a *class* is a contravariant collection which is also a moderate map. As an example, T and thereby $\emptyset$ is classical so $\emptyset$ is a set. Likewise, $\hat{x}x.y$ is classical whenever y is classical so $\hat{x}x.$ and thereby $\mathcal{C}$ are moderate. Hence, the collection $\mathcal{C}$ of classical maps form a class.

Universal and existential quantification restricted to a class S is defined as follows:

$$\begin{array}{rcl} \forall x{\in}S\colon \mathcal{A}. & = & \hat{y}\forall \hat{x} x{\in}y \mathrel{\tilde{\Rightarrow}} \mathcal{A}..S \\ \exists x{\in}S\colon \mathcal{A}. & = & \hat{y}\exists \hat{x} x{\in}y \mathrel{\tilde{\wedge}} \mathcal{A}..S \end{array}$$

If S does not contain x free then the above definitions reduce to

$$\begin{array}{rcl} \forall x{\in}S\colon \mathcal{A}. & = & \forall \hat{x} x{\in}S \mathrel{\tilde{\Rightarrow}} \mathcal{A}. \\ \exists x{\in}S\colon \mathcal{A}. & = & \exists \hat{x} x{\in}S \mathrel{\tilde{\wedge}} \mathcal{A}. \end{array}$$

Equivalence $\sim_S$ w.r.t. a class S can be defined thus:

$$f \sim_S g = f \begin{cases} g \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ g \begin{cases} \mathsf{F} \\ \forall x \in S\colon fx \sim_S gx. \end{cases} \end{cases}$$

The definitions of $\mathcal{C}$, $f \sim g$ and $f \sim_S g$ fit together in the sense that $f \sim g = f \sim_{\mathcal{C}} g$ as one should expect since both $f \sim g$ and $f \sim_{\mathcal{C}} g$ express equivalence w.r.t. $\mathcal{C}$.

3.10 Union sets

Now consider the construct $x{\in}{\in}S$ given by

$$\boxed{x{\in}{\in}S} = \exists \hat{y} x{\in}y \wedge y{\in}S.$$

If S is a contravariant class of contravariant sets then $x{\in}{\in}S$ states that x is a member of the union of S.

In the following it is convenient to work with $x{\in}{\in}S$ instead of the concept of a union because the existence of unions has not yet been touched.

The following constructs allow to quantify over the union of S (regardless of whether or not that union is a class):

$$\begin{array}{rcl} \forall x{\in}{\in}S\colon \mathcal{A}. & = & \hat{y}\forall \hat{x} x{\in}{\in}y \mathrel{\tilde{\Rightarrow}} \mathcal{A}..S \\ \exists x{\in}{\in}S\colon \mathcal{A}. & = & \hat{y}\exists \hat{x} x{\in}{\in}y \mathrel{\tilde{\wedge}} \mathcal{A}..S \end{array}$$

Likewise, equivalence $f \sim^2_S g$ w.r.t. the union of S can be defined thus:

$$f \sim^2_S g = f \begin{cases} g \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ g \begin{cases} \mathsf{F} \\ \forall x \dot{\in} \dot{\in} S\colon fx \sim^2_S gx. \end{cases} \end{cases}$$

3.11 Extensionality

If $\sim_1$ and $\sim_2$ are equivalence relations over $\mathcal{C}$, then a map f will be said to be *extensional* from $\sim_1$ to $\sim_2$ if

$$\forall \hat{x} \forall \hat{y} x \sim_1 y \Rightarrow fx \sim_2 fy..$$

The covariant collection $\mathcal{E}(\sim_1, \sim_2)$ of maps that are extensional from $\sim_1$ to $\sim_2$ can be defined by

$$\boxed{\mathcal{E}(\sim_1, \sim_2)} = \hat{f} \forall \hat{x} \forall \hat{y} x \sim_1 y \Rightarrow fx \sim_2 fy...$$

With this definition, the term

$$f \dot{\in} \mathcal{E}(\sim_1, \sim_2)$$

expresses that f is extensional from $\sim_1$ to $\sim_2$.

As an example, Lemma A.14.3 in Appendix A proves that moderate maps map classically equal maps to classically equal maps. With the notation introduced so far this can be expressed by

$$f \dot{\in} \mathsf{moderate} \rightarrow f \dot{\in} \mathcal{E}(\sim, \sim)$$

All classical maps are moderate so, in particular, classical maps map classically equal maps to classically equal maps. As mentioned in Section 3.8, classical mathematics is obtained by restricting to classical maps and identifying classically equal maps. The result above shows that this is a quotient construction.

3.12 Classicality

A map f will be said to be *limited* if there exists a set S such that $f \dot{\in} \mathcal{E}(\sim^2_S, \sim)$. In other words, the covariant collection of limited maps reads:

$$\boxed{\mathsf{limited}} = \hat{f} \exists \hat{S} f \dot{\in} \mathcal{E}(\sim^2_S, \sim)..$$

Limitation is very closely related to the notion of limited size of sets in ZFC. Limitation finally allows to express the closure property of the collection $\mathcal{C}$ of classical maps:

(1) T is classical.

(2) If f is moderate and limited, then f is classical.

(3) No other maps than those from (1) and (2) are classical.

Informally, the limitation requirement in (2) ensures consistency and avoids Burali-Forti's paradox. The moderation requirement corresponds to the axiom of restriction in ZFC but plays a more active role in the development of MT than restriction does in ZFC. In stable MT, it is possible to define a class W of non-well-founded maps which has the same closure properties as $\mathcal{C}$ above except that "moderate" is deleted in (2), and it is particularly easy to obtain semantics analogous to the AFA version of Aczel's theory of non-well-founded sets [1]. However, attempts to base MT on non-well-founded maps from the beginning without dealing with classical maps first have not yet been successful: moderation is central in proving the theorem of transfinite induction and transfinite induction is needed also when dealing with non-well-founded maps.

It is now possible to define the covariant collection ℓ of all maps. According to the closure properties, ℓ must satisfy

$$f \dot{\in} \ell = f \tilde{\vee} f \dot{\in} \mathsf{moderate} \wedge f \dot{\in} \mathsf{limited}$$

The covariant collection $\mathsf{moderate}$ satisfies

$$f \dot{\in} \mathsf{moderate} = \forall \hat{x} \ell(fx).$$

so ℓ can be defined recursively thus:

$$\ell = \hat{f} f \tilde{\vee} \forall \hat{x} \ell(fx). \wedge f \dot{\in} \mathsf{limited}.$$

The map ℓ does not occur anywhere in the definition of $\mathsf{limited}$. Writing out the recursive definition using Y yields

$$\ell = \mathsf{Y} \hat{g} \hat{f} f \tilde{\vee} \forall \hat{x} g(fx). \wedge f \dot{\in} \mathsf{limited}..$$

As will be seen later, the minimality of Y and the definition of ℓ together imply the theorem of transfinite induction (Theorem A.7.1).

Written out with $f \dot{\in} \mathsf{limited}$ expanded but without Y, the definition of ℓ reads

$$\ell = \hat{f} f \left\{ \begin{array}{l} \mathsf{T} \\ \forall \hat{x} \ell(fx). \wedge \exists \hat{S} \forall \hat{x} \forall \hat{y} x \sim_S^2 y \Rightarrow fx \sim fy... \end{array} \right. .$$

This is the definition of ℓ recorded in the list of definitions in D.4.

The definition of ℓ contains explicit recursion in that the definition of ℓ contains ℓ once in the right hand side of the definition. Furthermore, the definition contains implicit recursion in that all quantifiers quantify over the collection of classical maps. For this reason, it would be unfair to say that the notion of classicality can be defined recursively, since the recursive definition tacitly assumes that the collection is already known in its entirety. Hence, it is more fair to say that classicality is defined by a closure property which (as conjectured in Appendix B) happens to be consistent. On the contrary, the notion of well-foundedness in original MT can, with some right, be claimed to be defined recursively. This typically makes original MT more appealing to a trained logician than stable MT, but the simplicity of stable MT makes it much more easy to teach e.g. to first year students.

Each explicit occurrence of ℓ in the definition of ℓ strengthens the theorem of transfinite induction (Theorem A.7.1), so it is a pity that ℓ only occurs once. This is discussed in Section 3.15 and 4.3. Section 4.3 also continuous the discussion of implicit versus explicit recursion.

3.13 Generalisation of stability

The greatest lower bound $u \downarrow v$ of two maps u and v has already been defined. This is now generalised to the greatest lower bound $\downarrow g$ of all values of gx where x ranges over $\mathcal{C}$:

$$\boxed{\downarrow g} = \forall \hat{x} gx. \begin{cases} \mathsf{T} \\ \forall \hat{x} \neg gx. \begin{cases} \hat{y} \downarrow \hat{x} gxy.. \\ \perp \end{cases} \end{cases}$$

This allows to generalise the stability axiom to infinitely many maps:

Axiom 3.13.1 (C') $\ell x \rightarrow (a \preceq c) \vdash \downarrow \hat{x} fa. = f(\downarrow \hat{x} a.)$ *if x is not free in c and f.*

Axiom C follows from Axiom C' and Axiom C' is a generalisation of Axiom C. The models in Appendix B are conjectured to satisfy both axioms. Merely Axiom C as needed to verify that all of ZFC can be developed in stable MT which is why merely Axiom C is assumed in Appendix A.

3.14 The union set axiom

It is the hope that the use of $x \sim_S^2 y$ in the definition of ℓ allows to prove the existence of union sets:

$$\forall \hat{V} \exists \hat{U} \forall \hat{x} x \in U \Leftrightarrow x \in\in V...$$

This has not yet succeeded, however, so the existence of union sets has to be stated as an axiom:

Axiom 3.14.1 (U) $\forall \hat{V} \exists \hat{U} \forall \hat{x} x \in U \Leftrightarrow x \in\in V...$.

See section 3.15 for further discussion of Axiom U.

3.15 Discussion of axioms

At this point, stable MT has been introduced. Appendix D gives a summary of syntax, definitions and axioms.

Axiom $\mathsf{P}\bot$ and $\bot a$ say $\mathsf{P}ab\bot = \bot$ and $\bot a = \bot$. This ought to follow from stability. At present, $\bot \preceq a$ can be proved both from Axiom Y and from Axiom $\mathsf{P}\bot$ together with the definition of $\downarrow$ which indicates redundancy. It is an open question whether Axiom $\mathsf{P}\bot$ and $\bot a$ are independent or follow from the other axioms.

The axioms Q1, Q2, Q3 and Q4 are not particularly satisfactory in that they require $\forall$ to be defined before they are stated and because they do not express the three fundamental properties of ε directly (see the beginning of Section 3.3 for the fundamental properties of ε).

It is the hope that the union set axiom follows from the other axioms. As a conjecture, it is necessary to define the concept of ordinal numbers inside MT and to prove that the union of any set of ordinal numbers is an ordinal number before the union set axiom can be proved.

It is tempting to dispense for stability and introduce a non-stable existential quantifier $\tilde{\exists}$ with the properties

$$\begin{array}{lcll} \tilde{\exists} & = & \mathsf{T} & \text{if } fx = \mathsf{T} \text{ for some } x \\ \tilde{\exists} & = & \bot & \text{otherwise} \end{array}$$

This would allow to define ℓ "more recursively":

$$\ell = \hat{f} f \begin{cases} \mathsf{T} \\ \forall \hat{x} \ell(fx). \wedge \tilde{\exists} \hat{S} \ell S \wedge \forall \hat{x} \forall \hat{y} x \sim_S^2 y \Rightarrow fx \sim fy... \end{cases}$$

This would in turn allow to strengthen the transfinite induction theorem stated later. As a conjecture, this would allow to prove that inaccessible ordinals are non-classical. This would make the definition of "classical" much more rigid than the definition of "set" in set theory. The definition of "classical" is already quite more rigid than the definition of "set" because of the explicit countermeasure against Burali-Fortis paradox.

Note that the conjecture does not say anything about whether or not inaccessible ordinals exist. It merely states that if they exist, then they are not classical. For further discussion, see Section 4.3.

4 Further versions of map theory

The original version of MT allows to quantify over all "well-founded" maps whereas the stable version allows to quantify over classical maps. The three versions given in the following allow to quantify over many different ranges of maps.

The three new versions will be referred to as the *non-monotonic*, *unstable* and *pure* versions, respectively. The non-monotonic version is an extension of the stable version in the sense that all theorems and inference rules of the stable version are also theorems and inference rules of the non-monotonic version. On the contrary, the stability axiom does not hold in the unstable and pure versions. The unstable version is an extension of the pure version.

4.1 The non-monotonic version

The *non-monotonic* version is somewhat like the NBG extension of ZFC, but it is more general. The syntax of the non-monotonic version reads

$$\begin{array}{lcl} \mathsf{v} & ::= & x_1 \mid x_2 \mid x_3 \mid \cdots \\ \mathsf{t} & ::= & \mathsf{v} \mid \mathsf{tt} \mid \hat{\mathsf{v}}\mathsf{t}. \mid \mathsf{T} \mid \mathsf{P} \mid \varepsilon_{\mathsf{t}} \mid \ell \\ \mathsf{e} & ::= & \mathsf{t} = \mathsf{t} \end{array}$$

In constructs of form ε_s, all free variables in s are called "non-monotonic free variables". No term of form $\hat{x}a.$ is allowed to bind a non-monotonic free variable. This restriction is discussed in a moment.

The syntax of the non-monotonic version differs from the syntax of the stable version in three respects: ε is replaced by ε_s, non-monotonic free variables are introduced, and ℓ is no longer a defined concept.

The construct ε_s is similar to ε of the stable version, but it quantifies over the covariant set s instead of the collection $\mathcal{C}$ of classical maps. The properties of ε_s read (c.f. Section 3.3):

- $\varepsilon_s f = \bot$ if fx is undefined for some $x \dot{\in} s$.
- $\varepsilon_s f \dot{\in} s$ if fx is defined for all $x \dot{\in} s$.
- $f(\varepsilon_s f) = \mathsf{T}$ if fx is defined for all $x \dot{\in} s$ and $fx = \mathsf{T}$ for some $x \dot{\in} s$.

In particular, ε_ℓ of the non-monotonic version equals ε of the stable version. For that reason it is convenient to define $\varepsilon = \varepsilon_\ell$ in the non-monotonic version.

All defined concepts in the stable version that refer to ε directly or indirectly has a subscripted and an unsubscripted version in the non-monotonic version. As an example,

$$\exists = \hat{f}\wr f(\varepsilon f).$$

in the stable version corresponds to

$$\exists_s = \hat{f}\wr f(\varepsilon_s f).$$

and

$$\exists = \exists_\ell$$

in the non-monotonic version.

The formula $\hat{y}\varepsilon_y.$ is ill-formed because the abstraction binds the non-monotonic variable y. This is somewhat analogous to the fact that $\hat{\imath}x_i$ is ill-formed:

The collection $x_1, x_2, \ldots$ should be understood as an infinite collection of variables rather than as an operator x which operates on an index i. Likewise, the collection of ε_s for all terms s should be understood as an infinite collection of choice constructs rather than as a single operator ε which operates on s.

The analogy between x_i and $\exists_s$ is not complete since in x_i, i is syntactically restricted to be a non-empty sequence of digits not starting with a zero, whereas s in ε_s is syntactically restricted to be a term.

If s in ε_s contains free variables, then these free variables should be understood as "arbitrary constants" rather than as "variables that vary". An equation like $\varepsilon_x \bot = \bot$ states that $\varepsilon_x \bot = \bot$ holds "for each x" rather than "for all x". The only practical consequence is that $\varepsilon_x \bot = \bot$ does not imply $\hat{x}\varepsilon_x \bot . = \hat{x}\bot .$ (since the latter equation is ill-formed).

Maps are still monotonic as in the stable version. This is important in gaining consistency. The monotonicity property would be ruined if abstractions were allowed to bind non-monotonic free variables.

Axiom Q1, Q2, Q3 and Q4 are modified by replacing ε and ℓ by ε_s and s, respectively. As a consequence, $\forall$ is replaced by $\forall_s$. Furthermore, in Axiom Q3 it is necessary to assume that s is non-empty in the sense that $sb = \mathsf{T}$ for some b.

Axiom 4.1.1 (Q1) $sa \wedge \forall_s b \rightarrow ba$.

Axiom 4.1.2 (Q2) $\varepsilon_s \hat{x} a . = \varepsilon_s \hat{x} sx \wedge a . .$

Axiom 4.1.3 (Q3) $sb \rightarrow s(\varepsilon_s \hat{x} a .) = \forall_s \hat{x} !a . .$

Axiom 4.1.4 (Q4) $!\forall_s \hat{x} a . = \forall_s \hat{x} !a . .$

Since ℓ is a basic construct in the non-monotonic version, it needs an axiom instead of a definition:

Axiom 4.1.5 (ℓ) $\ell = \mathsf{Y}\hat{g}\hat{f}f \left\{ \begin{array}{l} \mathsf{T} \\ \forall \hat{x} g(fx) . \wedge \exists \hat{S} \forall \hat{x} \forall \hat{y} x \sim_S^2 y \Rightarrow fx \sim fy ... \end{array} \right. \quad .. .$

Note that ℓ appears implicitly in the right hand side of Axiom ℓ in that $\forall$, $\exists$, $\sim$ and $\sim_S^2$ depend on ε_ℓ. It is not possible to turn the axiom into a definition by replacing these implicit appearances of ℓ by g since this would violate the restriction on non-monotonic free variables.

All theorems of the stable version are also theorems of the non-monotonic version provided that all occurrences of ε are replaced by ε_ℓ. The structure in Appendix B.2 is conjectured to be a model of the non-monotonic version for the obvious definition of ε_s. It is an open question whether or not the non-monotonic version is a conservative extension of the stable version.

As examples of use of the non-monotonic version define

$$
\begin{array}{rcl}
\ell_1 & = & \hat{f} \forall \hat{x} fx \dot{\in} \ell .. \\
\ell_2 & = & \hat{f} \forall_{\ell_1} \hat{x} fx \dot{\in} \ell_1 .. \\
\ell_\omega & = & \mathsf{Y}\hat{g}\hat{f}f \left\{ \begin{array}{l} \mathsf{T} \\ \forall \hat{x} fx \dot{\in} \ell_\omega . \end{array} \right. \quad ..
\end{array}
$$

With these definitions, ε_{ℓ_1} quantifies over all NBG-classes and ε_{ℓ_2} quantifies over all classes of NBG-classes. Furthermore, $\varepsilon_{\ell_\omega}$ quantifies over all hereditarily "NBG-small" well-founded classes where a class is NBG-small if it has cardinality less than or equal to the class of all sets in NBG.

Non-well-founded sets can be handled in the stable version because it is possible to define a surjective map from well-founded to non-well-founded sets. Non-well-founded classes can be handled in the non-monotonic version.

4.2 The unstable version

The *unstable* version is not an extension of the stable version since the stability axiom does not hold in the unstable version. The unstable version has two new constructs compared to the non-monotonic version, namely $\exists^{r,s}$ and $\varepsilon^{r,s,t}$. All free variables in r, s and t in $\exists^{r,s}$ and $\varepsilon^{r,s,t}$ are non-monotonic free variables.

Now let r, s and t be terms and let S denote the (naive) set of all maps x for which $rx = sx$, i.e. let $S = \{x \mid rx = sx\}$. The properties of $\exists^{r,s}$ and $\varepsilon^{r,s,t}$ read:

1. $\exists^{r,s}t = \mathsf{T}$ iff $tx = \mathsf{T}$ for some $x \in S$.
2. $t\varepsilon^{r,s,t} = \mathsf{T}$ and $\varepsilon^{r,s,t} \in S$ if $tx = \mathsf{T}$ for some $x \in S$.

These new constructs are described by the following three axioms:

Axiom 4.2.1 (E1) $\exists^{r,s}t \to t\varepsilon^{r,s,t}$.

Axiom 4.2.2 (E2) $\exists^{r,s}t \to (r\varepsilon^{r,s,t} = s\varepsilon^{r,s,t})$.

Axiom 4.2.3 (E3) $h \to (ra = sa) \vdash h \wedge ta \to \exists^{r,s}t$.

Axiom E1 and E2 say that if there is an x such that $rx = sx$ and $tx = \mathsf{T}$ then $\varepsilon^{r,s,t}$ is such an x. Axiom E3 says that if $ra = sa$ and $ta = \mathsf{T}$ then there exists an x such that $rx = sx$ and $tx = \mathsf{T}$. The h in Axiom E3 is merely included to ensure that the deduction theorem can be proved.

In the unstable version, stability is replaced by monotonicity:

Axiom 4.2.4 (M) $b \preceq c \vdash ab \preceq ac$.

All theorems of ZFC are provable in the unstable version. It is an open question whether or not the unstable version is consistent.

4.3 A more recursive definition of ℓ

Now consider the existential quantifier $\tilde{\exists}$ given by

$$\boxed{\tilde{\exists}} = \exists^{\mathsf{T},\mathsf{T}}$$

It quantifies over all maps and has the following property:

$\tilde{\exists}f = \mathsf{T}$ iff $fx = \mathsf{T}$ for some map x.

As discussed in Section 3.15, $\tilde{\exists}$ allows to define ℓ "more recursively". In the unstable version, the definition of ℓ would be an axiom and would read as follows:

$$\ell = \mathsf{Y}\hat{g}\hat{f}f \begin{cases} \mathsf{T} \\ \forall_\ell \hat{x} g(fx). \wedge \tilde{\exists}\hat{S} g S \wedge \forall_\ell \hat{x} \forall_\ell \hat{y} x \sim^2_S y \Rightarrow fx \sim fy \ldots \end{cases} \quad ..$$

Note how this axiom is recursive in two different ways: Occurrences of g as well as occurrences of ℓ on the right hand side are recursive occurrences of ℓ, but only the occurrences of g can be used in transfinite induction. It is not possible to replace the occurrences of ℓ in the right hand side by g because that would make g a non-monotonic variable which is not allowed to be bound.

In Section 3.15 it was conjectured that an axiom like that above would allow to prove that inaccessible ordinals are not classical, but that conjecture does not say anything about whether or not inaccessible ordinals exist. One might very well have that the axiom above together with the ability to quantify over arbitrary covariant collections allows to produce models of ZFC+SI.

4.4 The pure version

The *pure* version is the unstable version from which axiom ℓ and U are removed. It is an open question whether or not the pure version is consistent and whether or not all theorems of ZFC are provable in the pure version.

Neither the concept of "well-founded" in original MT nor "classical" in stable MT are built into the pure version.

As an example of use of unstable existential quantification, Appendix C defines the concept of well-foundedness from original MT inside pure MT. It is an open question whether or not the definition allows to prove the axioms of the original version.

4.5 On the axiom of choice

Most theories contain at a debatable axiom: Euclidean geometry contains the parallel axiom and Peano arithmetic contains the axiom of induction. The debatable axiom in the stable version is hidden as a definition, namely the definition of ℓ. This debatable axiom is brought more to light in the non-monotonic and the unstable versions, where it appears as an axiom.

Frege set theory did not seem to contain any debatable axioms, but the axiom of comprehension turned out to be problematic in conjunction with Tertium Non Datur. The pure version seems to contain no debatable axioms either but, contrary to Frege set theory, it contains $\perp$ that falsifies Tertium Non Datur, and it has the restriction on non-monotonic free variables as a guard against paradoxes. For this reason it would be interesting to know whether or not the pure version is consistent and whether or not all of ZFC set theory can be developed in it.

It is not completely true that the pure version has no debatable axioms since it contains Hilberts ε-operator and thereby implicitly contains the axiom of choice. The axiom of choice is debatable in the sense that there has been a lot of debate about it even though the axiom is obvious, useful, necessary for several practical applications, and has absolutely no share in any paradox. The so-called strange consequences of the axiom of choice are really just strange properties of infinity. The author has chosen to disregard this historical debate and to include Hilberts ε-operator in all versions of MT defined so far. Thereby the axiom of choice is hardcoded into all these versions. It is of course possible to define versions with a universal quantifier instead of the ε-operator in which the axiom of choice does not hold.

5 Conclusion

Four new versions of MT have been introduced: The stable version and three more called the non-monotonic, unstable and pure versions, respectively. For the stable version, two axioms, Axiom C and Axiom C', have been proposed where the latter is stronger than the former.

It is shown in Appendix A that all axioms and theorems of [6] except Axiom Well-2 carries over so that, in particular, all theorems of ZFC set theory are provable in the stable version.

The three variants have been presented. The non-monotonic version is somewhat like the NBG-extension of ZFC.

6 Acknowledgement

My thanks are due to Chantal Berline for useful model-theoretic discussions and lots of comments on the manuscript. Responsibility for any errors, in particular in the models in Appendix B, remains mine.

A Formal development

A.1 Goal

This appendix presents some elementary metatheorems of MT and develops it to a point where all theorems of ZFC are provable in MT.

With one exception, stable MT is "compatible" with original MT. The predicate ℓ in the stable version corresponds to ϕ in the original version, but they are not completely identical. The most notable difference is that $\ell\ell = \mathsf{T}$ whereas $\phi\phi = \bot$.

All axioms of the original version except one are provable in the stable version. The exception is Axiom Well-2 in [6] which depends on $\phi\phi = \bot$. Axiom Well-2 is only used to prove two lemmas, Lemma C-K and Lemma C-P in [6], and those two lemmas hold in the stable version.

The goal of this section is to prove all axioms of the original version except Well-2 and to prove Lemma C-K and C-P, so that all results of [6] carry over. In particular, by virtue of [6], this shows that all theorems of ZFC are provable in MT and thereby verifies the expressive power of MT.

The axioms of the original version are verified as follows:

original version	stable version
Axiom Trans	Axiom T
Axiom Sub1	Axiom Sab
Axiom Sub2	Axiom S$\hat{x}$
Axiom Apply1	Axiom $\mathsf{T}a$
Axiom Apply2	Axiom $\hat{x}b$
Axiom Apply3	Axiom $\bot a$
Axiom Select1	Axiom PT
Axiom Select2	Axiom $\mathsf{P}\hat{x}$
Axiom Select3	Axiom $\mathsf{P}\bot$
Axiom Rename	Axiom R
Axiom QND'	Axiom QND
Axiom Quantify1	Axiom Q1
Axiom Quantify2	Axiom Q2
Axiom Quantify3	Axiom Q3
Axiom Quantify4	Lemma A.5.1
Axiom Quantify5	Lemma A.5.5
Axiom Well 1	Lemma A.8.1
Axiom Well 2	Does not carry over
Axiom Well 3	Lemma A.8.2
Axiom C-A	Lemma A.8.4
Axiom C-K'	Special case of lemma A.10.7
Lemma C-K	Lemma A.10.7
Axiom C-P'	Special case of lemma A.10.13
Lemma C-P	Lemma A.10.13
Axiom C-Curry	Lemma A.15.7
Axiom C-Prim	Lemma A.15.11
Axiom C-M1	Lemma A.15.3
Axiom C-M2	Lemma A.15.4
Axiom Induction	Lemma A.7.1

A.2 Elementary lemmas

The stable version of MT has several axioms in common with the original version in [6], and the lemmas below carry over. In the lemmas, variables denote arbitrary terms unless otherwise noted. In a lemma like $\exists \hat{x} a. \rightarrow \ell(\varepsilon \hat{x} a.)$, a and x denote arbitrary terms but, because of the syntax of MT, x is implicitly restricted to be an arbitrary variable.

Lemma A.2.1 (Reflexivity) $a = a$.

Lemma A.2.2 (Commutativity) $a = b \vdash b = a$.

Lemma A.2.3 (Transitivity) $a = b; b = c \vdash a = c$.

Lemma A.2.4 (Substitutivity) $a = a' \vdash b = c$ if c arises from b by replacing a by a' any number of times.

Lemma A.2.5 (Renaming) If a and a' are identical except for naming of bound variables, then $a = a'$.

If $a = b$ according to one of the axioms PT, $\mathsf{P}\hat{x}$, $\mathsf{P}\bot$, $\mathsf{T}a$, $\hat{x}b$ or $\bot a$, and if c and d are identical except that one subterm of c of form a is replaced by b in d, then d is said to be a *reduct* of c. Let $c \xrightarrow{r} d$ denote that either d is a reduct of c or d is identical to c except possibly for renaming of bound variables. Let $c \xrightarrow{*} d$ denote that there exist terms $a_0, \ldots, a_n$ such that a_0 is the term c, a_n is the term d, and $a_{i-1} \xrightarrow{r} a_i$ for $i \in \{1, \ldots, n\}$.

Lemma A.2.6 (Reduction) If $a \xrightarrow{*} c$ and $b \xrightarrow{*} c$ then $a = b$.

A.3 Consequences of Quartum Non Datur

Let $a = b$ be an equation whose free variables are exactly $x_1, \ldots, x_n$. A *logical instance* of $a = b$ is an equation $a' = b'$ where each variable x_i is replaced by T, $\hat{y}x_i y.$ or $\bot$. Hence, $a = b$ has exactly 3^n logical instances. The following lemmas carry over from [6].

Lemma A.3.1 If all logical instances of a formula $a = b$ are provable, then $a = b$.

A formula $a = b$ is a *tautology* if each logical instance is provable by the reduction lemma.

Lemma A.3.2 All tautologies are provable.

If $a = b$ is a tautology, then any equation that can be obtained by replacing free variables in $a = b$ by arbitrary terms is an *instance of a tautology*.

Lemma A.3.3 All instances of tautologies are provable.

As examples, $a \wedge b = b \wedge a$ and $a \wedge a = \wr a$ are instances of tautologies for arbitrary terms a and b.

Now recall that the construct $x \rightarrow y$ is shorthand for the equation $x \mathrel{\tilde{\wedge}} y = x \mathrel{\tilde{\wedge}} \mathsf{T}$ and $x \rightarrow (y = z)$ is shorthand for $x \mathrel{\tilde{\wedge}} y = x \mathrel{\tilde{\wedge}} z$. The equation $a \rightarrow b$ expresses "if $a = \mathsf{T}$ then $b = \mathsf{T}$" and $x \mathrel{\tilde{\wedge}} y = x \mathrel{\tilde{\wedge}} z$ expresses "if $a = \mathsf{T}$ then $b = c$".

Lemma A.3.4 (Modus ponens) $a; a \rightarrow b \vdash b$.

Lemma A.3.5 (Modus ponens) $a; a \rightarrow (b = c) \vdash b = c$.

Lemma A.3.6 $a \rightarrow a$.

Lemma A.3.7 $b = c \vdash a \to (b = c)$.

Lemma A.3.8 (QND) If

$$a_1 \wedge \cdots \wedge a_n \to (b = c)$$

is an instance of a tautology, then

$$a_1; \cdots; a_n \vdash b = c$$

In the QND lemma, the antecedents $a_1; \cdots; a_n$ should be understood as the antecedents $a_1 = \mathsf{T}; \cdots; a_n = \mathsf{T}$, c.f. Section 2.6.

The QND lemma can be used, e.g., to prove

$$a; a \Rightarrow b \vdash b$$

The QND lemma expresses quartum non datur whereas TND below expresses tertium non datur.

Lemma A.3.9 (TND) $!a; a \to (b = c); \neg a \to (b = c) \vdash b = c$.

The following two lemmas are suited for use together with the Deduction Theorem stated in the next section.

Lemma A.3.10 (Indirect proof) $!a; \neg a \to \mathsf{F} \vdash a$.

Lemma A.3.11 (Monotonic deduction) $!a_1; \cdots; !a_n; !b; a_1 \wedge \cdots \wedge a_n \to b \vdash a_1 \wedge \cdots \wedge a_n \Rightarrow b$.

In [6], $\Rightarrow$ and $\to$ are referred to as "monotonic" and "non-monotonic" implication, respectively, which explains the name of the last lemma above.

A.4 The Deduction Theorem

Recall the following two axioms:

Axiom A.4.1 (S$\hat{x}$) $a = b \vdash \hat{x}a. = \hat{x}b.\,.$

Axiom A.4.2 (E) $\wr fxy = \wr gxy; fxyz = fab; gxyz = gab \vdash fxy = gxy$ *if* x, y *and* z *are not free in* f *and* g.

The variable x in Axiom S$\hat{x}$ and the variables x, y and z in Axiom E will be referred to as *quantification variables*.

If h is a term then a *hypothetical proof with hypothesis* h is a sequence of equations where each equation either is the equation $h = \mathsf{T}$ or an instance of a simple axiom or follows from previous equations in the sequence by a complex axiom; furthermore, for each application of Axiom S$\hat{x}$ and E, the quantification variables are not allowed to be variables that occur free in h. The last equation in a hypothetical proof is called the conclusion of the proof.

Theorem A.4.3 (Deduction) If there is a hypothetical proof with hypothesis h and conclusion $a = b$, then there exists an ordinary formal proof of $h \rightarrow (a = b)$.

The special case where b is the term T reads:

Theorem A.4.4 (Deduction) If there is a hypothetical proof with hypothesis h and conclusion a, then there exists an ordinary formal proof of $h \rightarrow a$.

Like in [6], the Deduction Theorem follows from $h \rightarrow h$ and $a = b \vdash h \rightarrow (a = b)$ and one auxiliary lemma for each complex axiom in MT:

Lemma A.4.5 (Deduction/Axiom T**)** $h \rightarrow (a = b); h \rightarrow (a = c) \vdash h \rightarrow (b = c)$.

Proof. See [6]. $\diamondsuit$

Lemma A.4.6 (Deduction/Axiom Sab**)** $h \rightarrow (a = c); h \rightarrow (b = d) \vdash h \rightarrow (ab = cd)$.

Proof. See [6]. $\diamondsuit$

Lemma A.4.7 (Deduction/Axiom S$\hat{x}$**)** $h \rightarrow (a = b) \vdash h \rightarrow (\hat{x}a. = \hat{x}b.)$ if x is not free in h.

Proof. See [6]. $\diamondsuit$

Lemma A.4.8 (Deduction/Axiom QND**)** $h \rightarrow (a\mathsf{T} = b\mathsf{T}); h \rightarrow (a\bot = b\bot); h \rightarrow (a\hat{y}xy. = b\hat{y}xy.) \vdash h \rightarrow (ax = bx)$.

Proof. See [6]. $\diamondsuit$

Lemma A.4.9 $\mathsf{Y}\hat{x}fx. = \mathsf{Y}f$.

Proof. By QND. $\diamondsuit$

Lemma A.4.10 $a \rightarrow (b \preceq c)$ if and only if $a \mathbin{\tilde{\wedge}} b \preceq a \mathbin{\tilde{\wedge}} c$.

Proof. $a \rightarrow (b \preceq c)$ means $a \mathbin{\tilde{\wedge}} b = a \mathbin{\tilde{\wedge}} (b \downarrow c)$ and $a \mathbin{\tilde{\wedge}} b \preceq a \mathbin{\tilde{\wedge}} c$ means $a \mathbin{\tilde{\wedge}} b = (a \mathbin{\tilde{\wedge}} b) \downarrow (a \mathbin{\tilde{\wedge}} c)$, so the lemma follows from $a \mathbin{\tilde{\wedge}} (b \downarrow c) = (a \mathbin{\tilde{\wedge}} b) \downarrow (a \mathbin{\tilde{\wedge}} c)$ which can be proved by QND in a. $\diamondsuit$

Lemma A.4.11 (Deduction/Axiom Y**)** $h \rightarrow (fa \preceq a) \vdash h \rightarrow (\mathsf{Y}f \preceq a)$.

Proof. Let $g = \hat{x}h \mathbin{\tilde{\wedge}} fa.$ (where x is a fresh variable). $g(h \mathbin{\tilde{\wedge}} a) = h \mathbin{\tilde{\wedge}} f(h \mathbin{\tilde{\wedge}} a) = h \mathbin{\tilde{\wedge}} fa \preceq h \mathbin{\tilde{\wedge}} a$ gives $\mathsf{Y}g \preceq h \mathbin{\tilde{\wedge}} a$. Further, $\mathsf{Y}g = g(\mathsf{Y}g) = h \mathbin{\tilde{\wedge}} f(\mathsf{Y}g) = h \mathbin{\tilde{\wedge}} f(\mathsf{Y}\hat{x}h \mathbin{\tilde{\wedge}} fx.) = h \mathbin{\tilde{\wedge}} f(\mathsf{Y}\hat{x}fx.) = h \mathbin{\tilde{\wedge}} f(\mathsf{Y}f) = h \mathbin{\tilde{\wedge}} \mathsf{Y}f$ so $h \mathbin{\tilde{\wedge}} \mathsf{Y}f \preceq h \mathbin{\tilde{\wedge}} a$ which entails $h \rightarrow (\mathsf{Y}f \preceq a)$. $\diamondsuit$

Lemma A.4.12 (Deduction/Axiom C**)** $h \rightarrow (a \preceq c); h \rightarrow (b \preceq c) \vdash h \rightarrow (fa \downarrow fb = f(a \downarrow b))$.

Proof. $h \mathrel{\tilde{\wedge}} (fa \downarrow fb) = h \mathrel{\tilde{\wedge}} (f(h \mathrel{\tilde{\wedge}} a) \downarrow f(h \mathrel{\tilde{\wedge}} b)) = h \mathrel{\tilde{\wedge}} f((h \mathrel{\tilde{\wedge}} a) \downarrow (h \mathrel{\tilde{\wedge}} b)) = h \mathrel{\tilde{\wedge}} f(a \downarrow b)$. $\diamondsuit$

Lemma A.4.13 $a \mathrel{\tilde{\wedge}} b = a \mathrel{\tilde{\wedge}} c$ if and only if $a \mathrel{\tilde{\Rightarrow}} b = a \mathrel{\tilde{\Rightarrow}} c$.

Proof. $a \mathrel{\tilde{\Rightarrow}} b = a \mathrel{\tilde{\Rightarrow}} (a \mathrel{\tilde{\wedge}} b) = a \mathrel{\tilde{\Rightarrow}} (a \mathrel{\tilde{\wedge}} c) = a \mathrel{\tilde{\Rightarrow}} c$ and $a \mathrel{\tilde{\wedge}} b = a \mathrel{\tilde{\wedge}} (a \mathrel{\tilde{\Rightarrow}} b) = a \mathrel{\tilde{\wedge}} (a \mathrel{\tilde{\Rightarrow}} c) = a \mathrel{\tilde{\wedge}} c$. $\diamondsuit$

As a consequence, $a \rightarrow (b = c)$ if and only if $a \mathrel{\tilde{\Rightarrow}} b = a \mathrel{\tilde{\Rightarrow}} c$.

Lemma A.4.14 (Deduction/Axiom E**)** If x, y and z do not occur free in f, g and h then

$$\begin{array}{ll} & h \rightarrow (\wr fxy = \wr gxy) \\ ; & h \rightarrow (fxyz = fab) \\ ; & h \rightarrow (gxyz = gab) \\ \vdash & h \rightarrow (fxy = gxy) \end{array}$$

Proof. Let $f' = \hat{x}\hat{y}h \mathrel{\tilde{\Rightarrow}} fxy..$ and $g' = \hat{x}\hat{y}h \mathrel{\tilde{\Rightarrow}} gxy..$. $h \rightarrow (\wr fxy = \wr gxy)$ entails $\wr f'xy = \wr(h \mathrel{\tilde{\Rightarrow}} fxy) = h \mathrel{\tilde{\Rightarrow}} \wr fxy = h \mathrel{\tilde{\Rightarrow}} \wr gxy = \wr(h \mathrel{\tilde{\Rightarrow}} gxy) = \wr g'xy$. $h \rightarrow (fxyz = fab)$ entails $f'xyz = (h \mathrel{\tilde{\Rightarrow}} fxy)z = h \mathrel{\tilde{\Rightarrow}} fxyz = h \mathrel{\tilde{\Rightarrow}} fab = f'ab$ and likewise for g. Hence, $f'xy = g'xy$ by Extensionality, so $h \mathrel{\tilde{\Rightarrow}} fxy = h \mathrel{\tilde{\Rightarrow}} gxy$ which proves the lemma. $\diamondsuit$

A.5 Quantification

The original version of MT contains two axioms, Quantify 4 and Quantify 5, about quantification that do not occur in the stable version. Hence, before the results of [6] can be imported into the present paper, these two axioms have to be proved.

Lemma A.5.1 (Quantify 4) $\exists \hat{x} a. \rightarrow \ell(\varepsilon \hat{x} a.)$.

Proof. Assume $\exists \hat{x} a.$. The assumption $\exists \hat{x} a.$ gives $!\exists \hat{x} a.$. Hence,

$$\begin{array}{lll} \mathsf{T} & = & !\exists \hat{x} a. & \text{Assumption} \\ & = & !\neg \forall \hat{x} \neg a. & \text{Def of } \forall\text{, QND} \\ & = & !\forall \hat{x} \neg a. & \text{QND} \\ & = & \forall \hat{x} ! \neg a. & \text{Axiom Q4} \\ & = & \forall \hat{x} ! a. & \text{QND} \\ & = & \ell(\varepsilon \hat{x} a.) & \text{Axiom Q3} \end{array}$$

$\diamondsuit$

Three auxiliary lemmas will be proved before proving Quantify 5:

Lemma A.5.2 $\forall \hat{x} a. = \wr \hat{x} a. (\varepsilon \hat{x} \neg a.)$.

Proof. $\forall\hat{x}a. = \neg\exists\hat{x}\neg a. = \neg\wr\hat{x}\neg a.(\varepsilon\hat{x}\neg a.) = \wr\hat{x}a.(\varepsilon\hat{x}\neg a.)$. $\diamondsuit$

Lemma A.5.3 $\varepsilon\hat{x}\neg(\ell x \wedge a). = \varepsilon\hat{x}\neg a.$.

Proof.

$$\begin{array}{rcll} \varepsilon\hat{x}\neg(\ell x \wedge a). & = & \varepsilon\hat{x}\ell x \wedge \neg(\ell x \wedge a). & \text{Axiom Q2} \\ & = & \varepsilon\hat{x}\ell x \wedge \neg a. & \text{QND} \\ & = & \varepsilon\hat{x}\neg a. & \text{Axiom Q2} \end{array}$$

$\diamondsuit$

Lemma A.5.4 $\ell(\varepsilon\hat{x}\neg a.) = !\forall\hat{x}a.$.

Proof.

$$\begin{array}{rcll} \ell(\varepsilon\hat{x}\neg a.) & = & \forall\hat{x}!\neg a. & \text{Axiom Q3} \\ & = & \forall\hat{x}!a. & \text{QND} \\ & = & !\forall\hat{x}a. & \text{Axiom Q4} \end{array}$$

$\diamondsuit$

Lemma A.5.5 (Quantify 5) $\forall\hat{x}a. = \forall\hat{x}\ell x \wedge a.$.

Proof.

$$\begin{array}{rcll} \forall\hat{x}\ell x \wedge a. & = & \wr\hat{x}\ell x \wedge a.(\varepsilon\hat{x}\neg(\ell x \wedge a).) & \text{Lemma A.5.2} \\ & = & \wr\hat{x}\ell x \wedge a.(\varepsilon\hat{x}\neg a.) & \text{Lemma A.5.3} \\ & = & \ell(\varepsilon\hat{x}\neg a.) \wedge \wr\hat{x}a.(\varepsilon\hat{x}\neg a.) & \text{Reduction, QND} \\ & = & !\forall\hat{x}a. \wedge \forall\hat{x}a. & \text{Lemma A.5.2, A.5.4} \\ & = & \forall\hat{x}a. & \text{QND} \end{array}$$

$\diamondsuit$

These lemmas allow to import all results in [6] that depend on the quantification axioms of the original version of MT:

Lemma A.5.6 If a is free for x in b then $\ell a; \forall\hat{x}b. \vdash \langle b \mid x{:=}a\rangle$.

Lemma A.5.7 $\ell x \to a \vdash \forall\hat{x}a.$

Lemma A.5.8 If a is free for x in b then $\ell a; !\exists\hat{x}b.; \langle b \mid x{:=}a\rangle \vdash \exists\hat{x}a.$

Lemma A.5.9 If u is shorthand for $\varepsilon\hat{x}a.$ and u is free for x in a, then

$$\exists\hat{x}a. \vdash \ell u; \langle a \mid x{:=}u\rangle$$

Note that $\mathcal{A} \vdash \mathcal{B}; \mathcal{C}$ is shorthand for two lemmas, one that says $\mathcal{A} \vdash \mathcal{B}$ and one that says $\mathcal{A} \vdash \mathcal{C}$.

Lemma A.5.10 (Ackermann) $\ell x \to a \Leftrightarrow b \vdash \varepsilon\hat{x}a. = \varepsilon\hat{x}b.$.

The above lemma expresses Ackermanns axiom ([5], p.244).

A.6 Order and monotonicity

Theorem A.6.1 (Extensionality) If $n \geq 0$ and if $x_1, \ldots, x_n$, y and z do not occur free in f and g then

$$\begin{array}{lrcl} & \wr f x_1 \cdots x_n y & = & \wr g x_1 \cdots x_n y \\ ; & f x_1 \cdots x_n y z & = & f a_1 \cdots a_n b \\ ; & g x_1 \cdots x_n y z & = & g a_1 \cdots a_n b \\ \vdash & f x_1 \cdots x_n y & = & g x_1 \cdots x_n y \end{array}$$

Proof. Recall the definition of x^H and x^T in Section 2.7. Define

$\underline{x}_i = x^{\overbrace{TT\cdots T}^{i-1}H}$,
$f' = \hat{x}\hat{y}f\underline{x}_1 \cdots \underline{x}_n y..$,
$g' = \hat{x}\hat{y}g\underline{x}_1 \cdots \underline{x}_n y..$,
$a' = \hat{x}_1 \cdots \hat{x}_n a. \cdots .\underline{x}_1 \cdots \underline{x}_n$, and
$b' = \hat{x}_1 \cdots \hat{x}_n b. \cdots .\underline{x}_1 \cdots \underline{x}_n$.

Now apply the extensionality axiom on f', g', a' and b'. Note that the proof also works in the case where $n = 0$. $\diamondsuit$

Lemma A.6.2 $x = x \downarrow x$.

Proof. Let $f = \hat{x}x.$ and $g = \hat{x}x \downarrow x.$. $\wr fx = \wr gx$ follows from QND. Further, $fxy = xy = f(xy)$ and $gxy = (x \downarrow x)y = \text{if}(x, \mathsf{T}, \hat{z}xz \downarrow xz.)y = \text{if}(x, \mathsf{T}, xy \downarrow xy) = xy \downarrow xy = g(xy)$. Now the lemma follows from the Extensionality Theorem (Theorem A.6.1). $\diamondsuit$

Lemma A.6.3 $x \downarrow y = y \downarrow x$.

Proof. Let $f = \hat{x}\hat{y}x \downarrow y..$ and $g = \hat{x}\hat{y}y \downarrow x..$. $\wr fxy = \wr gxy$ follows from QND. Further, $fxyz = (x \downarrow y)z = \text{if}(x \downarrow y, \mathsf{T}, x \downarrow y)z = \text{if}(x \downarrow y, \mathsf{T}, xz \downarrow yz) = \text{if}(x \downarrow y, \mathsf{T}, xz) \downarrow \text{if}(x \downarrow y, \mathsf{T}, yz) = f\text{if}(x \downarrow y, \mathsf{T}, xz)\text{if}(x \downarrow y, \mathsf{T}, yz)$ and likewise for g. Now the lemma follows from the Extensionality Theorem. $\diamondsuit$

Lemma A.6.4 $x \downarrow (y \downarrow z) = (x \downarrow y) \downarrow z$.

Proof. Let $f = \hat{x}\hat{y}\hat{z}x \downarrow (y \downarrow z)...$ and $g = \hat{x}\hat{y}\hat{z}(x \downarrow y) \downarrow z....$ Now $\wr fxyz = \wr gxyz$ by QND. Further, $fxyzu = \text{if}(x \downarrow (y \downarrow z), \mathsf{T}, x \downarrow (y \downarrow z))u = \text{if}((x \downarrow y) \downarrow z, \mathsf{T}, xu \downarrow (yu \downarrow zu)) = \text{if}((x \downarrow y) \downarrow z, \mathsf{T}, xu) \downarrow (\text{if}((x \downarrow y) \downarrow z, \mathsf{T}, yu) \downarrow \text{if}((x \downarrow y) \downarrow z, \mathsf{T}, yu)) = f\text{if}((x \downarrow y) \downarrow z, \mathsf{T}, xu)\text{if}((x \downarrow y) \downarrow z, \mathsf{T}, yu)\text{if}((x \downarrow y) \downarrow z, \mathsf{T}, zu)$ and likewise for g. Now the lemma follows from the Extensionality Theorem. $\diamondsuit$

The following lemmas prove the following: $\preceq$ is a partial order. T is a maximal element and $\perp$ is the unique minimal element for $\preceq$, and $\hat{x}\perp$ is the unique minimal element among all functions. $x \downarrow y$ is the greatest lower bound of x and y and functional application and abstraction are monotonic w.r.t. $\preceq$. Section A.6 ends with an investigation of P and proves that $\mathsf{P}ab$ is a function for all a and b.

Lemma A.6.5 $a \preceq a$

Proof. $a = a \downarrow a$. $\diamondsuit$

Lemma A.6.6 $a \preceq b; b \preceq a \vdash a = b$.

Proof. $a = a \downarrow b = b \downarrow a = b$. $\diamondsuit$

Lemma A.6.7 $a \preceq b; b \preceq c \vdash a \preceq c$.

Proof. $a = a \downarrow b = a \downarrow (b \downarrow c) = (a \downarrow b) \downarrow c = a \downarrow c$. $\diamondsuit$

Lemma A.6.8 $\perp \preceq a$.

Proof. $\hat{y}y.a = a \vdash \mathsf{Y}\hat{y}y. \preceq a \vdash \perp \preceq a$. $\diamondsuit$

Lemma A.6.9 $a \preceq \perp \vdash a = \perp$.

Proof. $a \preceq \perp; \perp \preceq a \vdash a = \perp$. $\diamondsuit$

Lemma A.6.10 $a \preceq b \vdash fa \preceq fb$.

Proof. $a \preceq b; b \preceq b \vdash fa = f(a \downarrow b) = fa \downarrow fb$. $\diamondsuit$

Lemma A.6.11 $a \preceq b \vdash \hat{x}a. \preceq \hat{x}b.$.

Proof. $\hat{x}a. \downarrow \hat{x}b. = \hat{x}a \downarrow b. = \hat{x}a.$. $\diamondsuit$

Lemma A.6.12 $f \preceq g \vdash fa \preceq ga$.

Proof. $f \preceq g \vdash \hat{y}ya.f \preceq \hat{y}ya.g \vdash fa \preceq ga$. $\diamondsuit$

Lemma A.6.13 $f \preceq g; a \preceq b \vdash fa \preceq gb$.

Proof. $fa \preceq fb \preceq gb$. $\diamondsuit$

Lemma A.6.14 $\text{if}(a, \mathsf{T}, \perp) = \mathsf{T} \vdash a = \mathsf{T}$.

Proof. $\text{if}(a, \mathsf{T}, \perp) \to a$ is a tautology. $\diamondsuit$

Lemma A.6.15 $\mathsf{T} \preceq a \vdash a = \mathsf{T}$.

Proof. $\mathsf{T} = \mathsf{T} \downarrow a \vdash \text{if}(a, \mathsf{T}, \perp) = \mathsf{T} \vdash a = \mathsf{T}$. $\diamondsuit$

Lemma A.6.16 $\hat{y}\perp. \preceq f \vdash f = \hat{y}fy.$ if y is not free in f.

Proof. $\hat{y}\perp. \preceq f \vdash \neg\hat{y}\perp. \preceq \neg f \vdash \mathsf{T} \preceq \neg f \vdash \neg f \vdash f = \hat{y}fy.$. $\diamondsuit$

Lemma A.6.17 $f = \hat{y}fy. \vdash \hat{y}\perp. \preceq f$.

Proof. $\perp \preceq fy \vdash \hat{y}\perp. \preceq \hat{y}fy. = f$. $\diamondsuit$

Lemma A.6.18 $a \downarrow b \preceq a$.

Proof. $a \downarrow b = (a \downarrow a) \downarrow b = a \downarrow (a \downarrow b) = (a \downarrow b) \downarrow a$. $\diamondsuit$

Lemma A.6.19 $a \preceq b; a \preceq c \vdash a \preceq b \downarrow c$.

Proof. $a = a \downarrow a = (a \downarrow b) \downarrow (a \downarrow c) = a \downarrow (b \downarrow c)$. $\diamondsuit$

Lemma A.6.20 $(u, v) \downarrow (x, y) = (u \downarrow x, v \downarrow y)$.

Proof. $(u, v) \downarrow (x, y) = \hat{z}(u, v)z \downarrow (x, y)z. = \hat{z}\text{if}(z, u, v) \downarrow \text{if}(z, x, y). = \hat{z}\text{if}(z, u \downarrow x, v \downarrow y). = (u \downarrow x, v \downarrow y)$. $\diamondsuit$

The proof above uses that (x, y) is a function for all x and y. The following lemmas prove that P is also a function for all x and y and thereby prove that $(x, y) = \mathsf{P}xy$ for all x and y.

Lemma A.6.21 $a\bot\bot = ab\bot \downarrow a\bot c$.

Proof. Let $f = \hat{u}a(u\mathsf{T})(u\mathsf{F})$. . This definition entails $f(x, y) = axy$. Now $(\bot, c) \preceq (b, c)$ and $(b, \bot) \preceq (b, c)$ by monotonicity so $a\bot\bot = f(\bot, \bot) = f((b, \bot) \downarrow (\bot, c)) = f(b, \bot) \downarrow f(\bot, c) = ab\bot \downarrow a\bot c$. $\diamondsuit$

Lemma A.6.22 $\mathsf{P}\bot\bot$ is a function (i.e. $\neg\mathsf{P}\bot\bot = \mathsf{T}$).

Proof. For all a and x, $\neg ax \to \neg a$ by QND, so if ax is a function then a is also a function. Hence, $\mathsf{PF}\bot$ is a function because $\mathsf{PF}\bot\mathsf{T} = \mathsf{F}$ is a function and $\mathsf{P}\bot\mathsf{F}$ is a function because $\mathsf{P}\bot\mathsf{FF} = \mathsf{F}$ is a function. Now $\neg\mathsf{P}\bot\bot = \neg\mathsf{PF}\bot \downarrow \neg\mathsf{P}\bot\mathsf{F} = \mathsf{T} \downarrow \mathsf{T} = \mathsf{T}$ proves that $\mathsf{P}\bot\bot$ is a function. $\diamondsuit$

Lemma A.6.23 $\mathsf{P}ab$ is a function for all a and b.

Proof. $\neg\mathsf{P}\bot\bot = \mathsf{T}$ implies $\neg\mathsf{P}ab = \mathsf{T}$ by monotonicity. $\diamondsuit$

Lemma A.6.24 $\mathsf{P}ab = \hat{x}\mathsf{P}abx. = (a, b)$.

Proof. $\neg u \to (u = \hat{x}ux.)$ holds by QND so $\neg\mathsf{P}ab$ implies $\mathsf{P}ab = \hat{x}\mathsf{P}abx.$ which, by definition of (a, b), equals (a, b). $\diamondsuit$

Lemma A.6.25 $fa\bot = \mathsf{T}; f\bot\bot = \bot \vdash f\bot b = \bot$.

Proof. $\mathsf{T} = fa\bot \preceq fab$ gives $fab = \mathsf{T}$. $f\bot b \preceq fab = \mathsf{T}$ gives $f\bot b = f\bot b \downarrow \mathsf{T}$. Hence, $\bot = f\bot\bot = fa\bot \downarrow f\bot b = \mathsf{T} \downarrow f\bot b = f\bot b$. $\diamondsuit$

A.7 Transfinite induction

The theorem of transfinite induction reads:

Theorem A.7.1 (Induction) If x and y are not free in f then

$$f\mathsf{T}; \neg x \wedge \ell x \wedge \forall \hat{y} f(xy). \rightarrow fx \vdash \ell x \rightarrow fx$$

A number of auxiliary lemmas will be proved before proving the Induction Theorem. In the lemmas, $a_1 = b_1; \cdots; a_m = b_m \dashv\vdash c_1 = d_1; \cdots; c_n = d_n$ means that $a_1 = b_1; \cdots; a_m = b_m \vdash c_1 = d_1; \cdots; c_n = d_n$ and $c_1 = d_1; \cdots;$ $c_n = d_n \vdash a_1 = b_1; \cdots; a_m = b_m$.

Definition A.7.2 $\boxed{\text{if}(x, y, z)} = x \left\{ \begin{array}{l} y \\ z \end{array} \right.$.

Lemma A.7.3 $?x = \mathsf{T} \dashv\vdash x = \mathsf{T}$.

Proof. $?x \rightarrow x$ is a tautology so $?x = \mathsf{T}$ implies $x = \mathsf{T}$. If $x = \mathsf{T}$ then $?x = \mathsf{T}$ by the definition of $?x$. $\diamondsuit$

Lemma A.7.4 $a \rightarrow b \dashv\vdash ?a \preceq ?b$.

Proof. Assume $?a \preceq ?b$ and $a = \mathsf{T}$. Now $\mathsf{T} = ?a \preceq ?b$ gives $?b = \mathsf{T}$ which entails $b = \mathsf{T}$ so $a \rightarrow b$. On the contrary, if $a \rightarrow b$ then $?a = ?(a \tilde{\wedge} \mathsf{T}) = ?(a \tilde{\wedge} b) = ?a \tilde{\wedge} ?b$ by QND. $\diamondsuit$

Lemma A.7.5 $a \rightarrow b; b \rightarrow a \dashv\vdash ?a = ?b$.

Proof. Follows from $?a = ?b \dashv\vdash ?a \preceq ?b; ?b \preceq ?a$. $\diamondsuit$

Lemma A.7.6 $\forall \hat{x} ?a. = ?\forall \hat{x} ?a$.

Proof. $\forall \hat{x} ?a. = \wr \hat{x} ?a.(\varepsilon \hat{x} \neg ?a.) = ?\wr \hat{x} ?a(\varepsilon \hat{x} \neg ?a.) = ?\forall \hat{x} ?a.$. $\diamondsuit$

Lemma A.7.7 $\forall \hat{x} a. = \wr \hat{x} a.(\varepsilon \hat{x} \neg a.)$.

Proof. $\forall \hat{x} a. = \neg \exists \hat{x} \neg a. = \neg \wr \hat{x} \neg a.(\varepsilon \hat{x} \neg a.) = \wr \hat{x} a.(\varepsilon \hat{x} \neg a.)$. $\diamondsuit$

Lemma A.7.8 $\forall \hat{x} ?a. = ?\forall \hat{x} a.$.

Proof. Assume $\forall \hat{x} ?a.$. Assume ℓx. Now $?a$ which entails a so $\ell x \rightarrow a$. Hence, $\forall \hat{x} a.$ which proves $?\forall \hat{x} a.$. In conclusion, $\forall \hat{x} ?a. \rightarrow ?\forall \hat{x} a.$.

Now assume $?\forall \hat{x} a.$ and ℓx. The assumptions imply $\forall \hat{x} a.$ and a so $?a$, $\ell x \rightarrow ?a$ and $\forall \hat{x} ?a.$, so $?\forall \hat{x} a. \rightarrow \forall \hat{x} ?a.$.

Using $a \rightarrow b; b \rightarrow a \vdash ?a = ?b$ this gives $?\forall \hat{x} ?a. = ??\forall \hat{x} a.$. Hence, $\forall \hat{x} ?a. = ?\forall \hat{x} ?a. = ??\forall \hat{x} a. = ?\forall \hat{x} a.$. $\diamondsuit$

Lemma A.7.9 $a \rightarrow b; \neg a \rightarrow \neg b \dashv\vdash \wr a \preceq \wr b$.

Proof. $\wr a \downarrow \wr b = \text{if}(a, \mathsf{T}, \mathsf{F}) \downarrow \text{if}(b, \mathsf{T}, \mathsf{F}) = \text{if}(a, \text{if}(b, \mathsf{T}, \bot), \text{if}(b, \bot, \mathsf{F})) = \text{if}(a, \text{if}(a \tilde{\wedge} b, \mathsf{T}, \bot), \text{if}(\neg a \tilde{\wedge} \neg b, \mathsf{F}, \bot)) = \text{if}(a, \text{if}(a \tilde{\wedge} \mathsf{T}, \mathsf{T}, \bot), \text{if}(\neg a \tilde{\wedge} \mathsf{T}, \mathsf{F}, \bot)) = \text{if}(a, \mathsf{T}, \mathsf{F}) = \wr a$. The other direction follows from Lemma A.6.15 and A.6.16. $\diamondsuit$

Lemma A.7.10 $!a \to !b; \neg b \to \neg a \vdash a \to b$.

Proof. Assume $a = \mathsf{T}$. Now $\neg a = \mathsf{F}$ so $!a \to !b; \neg b \to \mathsf{F}$ which entails b so $a \to b$. $\diamondsuit$

Lemma A.7.11 $!\forall \hat{x} a \tilde{\wedge} b. \to !(\forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.)$.

Proof. Assume $!\forall \hat{x} a \tilde{\wedge} b.$ and ℓx. Now $\forall \hat{x} !(a \tilde{\wedge} b).$ so $!(a \tilde{\wedge} b)$ which entails $!a$ so $\ell x \to !a$ and $\forall \hat{x} !a.$ which proves $!\forall \hat{x} a.$. The proof proceeds by TND:

If $\forall \hat{x} a.$ then a so $a \tilde{\wedge} b = b$ which entails $!\forall \hat{x} b.$ and, hence, $!(\forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.)$.

If $\neg \forall \hat{x} a.$ then $!(\forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.) = !\mathsf{F} = \mathsf{T}$.

In any case, $!(\forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.)$ which proves the lemma. $\diamondsuit$

Lemma A.7.12 $\forall \hat{x} a \tilde{\wedge} b. \preceq \forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.$.

Proof. Assume $\forall \hat{x} a \tilde{\wedge} b.$ and ℓx. The assumptions entail $a \tilde{\wedge} b$ which entails a and b. Hence, $\ell x \to a$ and $\ell x \to b$ which proves $\forall \hat{x} a.$ and $\forall \hat{x} b.$. Hence, $\forall \hat{x} a \tilde{\wedge} b. \to \forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.$.

Now assume $\forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.$ and ℓx. The assumptions entail $\forall \hat{x} a.$, $\forall \hat{x} b.$, a , b, and $a \tilde{\wedge} b$ so $\ell x \to a \tilde{\wedge} b$ and $\forall \hat{x} a \tilde{\wedge} b.$. Hence, $\forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b. \to \forall \hat{x} a \tilde{\wedge} b.$ which, combined with Lemma A.7.11 gives $\neg \forall \hat{x} a \tilde{\wedge} b. \to \neg(\forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.)$.

The above and Lemma A.7.9 yields $\wr \forall \hat{x} a \tilde{\wedge} b. \preceq \wr(\forall \hat{x} a. \tilde{\wedge} \forall \hat{x} b.) = \wr \forall \hat{x} a. \tilde{\wedge} \wr \forall \hat{x} b.$ which combined with $\forall f = \wr \forall f$ proves the lemma. $\diamondsuit$

Proof of the Induction Theorem. Define

$$g = \hat{x} \ell x \tilde{\wedge} ?fx.$$

$$L = \hat{h} \hat{x} x \begin{cases} \mathsf{T} \\ \forall \hat{y} h(xy). \wedge \exists \hat{S} \forall \hat{u} \forall \hat{v} u \sim_S^2 v \Rightarrow xu \sim xv... \end{cases} ..$$

Note that the definition of ℓ says

$$\ell = \mathsf{Y} L$$

Now $Lgx \preceq gx$ will be proved by QND in x:

Case $x = \mathsf{T}$: $Lgx = Lg\mathsf{T} = \mathsf{T}$ and $gx = g\mathsf{T} = \mathsf{T}$ so $Lgx \preceq gx$.

Case $x = \bot$: $Lgx = Lg\bot = \bot$ so $Lgx \preceq gx$.

Case $x = \hat{y} xy.$:

$$\begin{array}{rcl} Lgx & = & \forall \hat{y} g(xy). \wedge \exists \hat{S} \forall \hat{u} \forall \hat{v} u \sim_S^2 v \Rightarrow xu \sim xv... \\ & = & \forall \hat{y} \ell(xy) \tilde{\wedge} ?f(xy). \wedge \exists \hat{S} \forall \hat{u} \forall \hat{v} u \sim_S^2 v \Rightarrow xu \sim xv... \\ & \preceq & (\forall \hat{y} \ell(xy). \tilde{\wedge} ?\forall \hat{y} f(xy)) \wedge \exists \hat{S} \forall \hat{u} \forall \hat{v} u \sim_S^2 v \Rightarrow xu \sim xv... \\ & \preceq & (\forall \hat{y} \ell(xy). \wedge \exists \hat{S} \forall \hat{u} \forall \hat{v} u \sim_S^2 \Rightarrow xu \sim xv...) \tilde{\wedge} ?\forall \hat{y} f(xy). \\ & = & \ell x \tilde{\wedge} ?\forall \hat{y} f(xy). \\ & \preceq & \ell x \tilde{\wedge} ?fx \\ & = & gx \end{array}$$

In conclusion, $Lgx \preceq gx$ so $Lg = \hat{x}Lgx. \preceq \hat{x}gx. = g$ which entails $\ell = \mathsf{Y}L \preceq g$ by Axiom Y. Hence, if $\ell x = \mathsf{T}$ then $\mathsf{T} = \ell x \preceq gx$ which entails $gx = \mathsf{T}$ which in turn entails $fx = \mathsf{T}$ so $\ell x \to fx$. $\diamondsuit$

A.8 Simple lemmas about ℓ

The following lemma says that T is classical:

Lemma A.8.1 (Well-1) $\ell\mathsf{T} = \mathsf{T}$.

Proof. Follows from the definition of ℓ. $\diamondsuit$

The following lemma says that $\bot$ is non-classical:

Lemma A.8.2 (Well-3) $\ell\bot = \bot$.

Proof. Follows from the definition of ℓ. $\diamondsuit$

The following lemma says that classical maps differ from $\bot$ and thereby allows to use TND on classical maps.

Lemma A.8.3 $\ell x \to !x$.

Proof. By QND. $\diamondsuit$

The following lemma says that classical maps applied to classical maps yield classical maps.

Lemma A.8.4 (C-A) $\ell x; \ell y \vdash \ell(xy)$.

Proof. By TND in x. $\diamondsuit$

A.9 Properties of $\sim$ and related constructs

Lemma A.9.1 $\forall \hat{x} x \sim x.$.

Proof. $\ell x \to x \sim x$ is proved by transfinite induction in x: $\mathsf{T} \sim \mathsf{T} = \mathsf{T}$ follows from the definition of $\sim$. Now assume $\neg x$, ℓx and $\forall \hat{y} xy \sim xy.$ (this is the "induction hypothesis"). The induction hypothesis yields $x \sim x = \hat{y}xy. \sim \hat{y}xy. = \forall \hat{y} xy \sim xy. = \mathsf{T}$, which concludes the proof. $\diamondsuit$

Lemma A.9.2 $\forall \hat{x} \forall \hat{y} !x \sim y..$.

Proof. $\ell x \to \forall \hat{y} !x \sim y.$ is proved by transfinite induction in x:

If ℓy then $!y$ so $!\mathsf{T} \sim y$ which proves $\forall \hat{y} !\mathsf{T} \sim y.$.

If $\neg x$, ℓx, $\forall \hat{z} \forall \hat{y} !xz \sim y..$, ℓy and ℓz then $\ell(yz)$, $\forall \hat{y} !xz \sim y.$ and $!xz \sim yz$ so $\forall \hat{z} !xz \sim yz.$. If $\neg y$ then $!x \sim y = !\hat{z}xz. \sim \hat{z}yz. = !\forall \hat{z} xz \sim yz. = \forall \hat{z} !xz \sim yz. = \mathsf{T}$. If $y = \mathsf{T}$ then $!x \sim y$ is also T so $\forall \hat{y} !x \sim y.$ in both cases. $\diamondsuit$

Lemma A.9.3 $\forall \hat{x} \forall \hat{y} x \sim y \Rightarrow y \sim x..$.

Proof. $\ell x \rightarrow \forall \hat{y} x \sim y \Rightarrow y \sim x.$ is proved by transfinite induction in x:

If ℓy then $\mathsf{T} \sim y \Rightarrow y \sim \mathsf{T}$ by TND so $\forall \hat{y} \mathsf{T} \sim y \Rightarrow y \sim \mathsf{T}.$.

If $\neg x$, ℓx, $\forall \hat{z} \forall \hat{y} xz \sim y \Rightarrow y \sim xz..$, ℓy and ℓz then $\ell(yz)$, $\forall \hat{y} xz \sim y \Rightarrow y \sim xz.$ and $xz \sim yz \Rightarrow yz \sim xz$. Now assume $x \sim y$. If $\neg y$ then $T = x \sim y = \forall \hat{z} xz \sim yz.$ which implies $\forall \hat{z} yz \sim xz.$ and $y \sim x$. This combined with $!x \sim y$ and $!y \sim x$ yields $x \sim y \Rightarrow y \sim x$. If $y = \mathsf{T}$ then $x \sim y \Rightarrow y \sim x$ by QND. In both cases, $x \sim y \Rightarrow y \sim x$ so $\forall \hat{y} x \sim y \Rightarrow y \sim x.$ and the lemma follows by transfinite induction. $\diamondsuit$

Lemma A.9.4 $\forall \hat{x} \forall \hat{y} \forall \hat{z} x \sim y \wedge y \sim z \Rightarrow x \sim z...$.

Proof. The proof of $\ell x \rightarrow \forall \hat{y} \forall \hat{z} x \sim y \wedge y \sim z \Rightarrow x \sim z..$ by transfinite induction in x is left to the reader. $\diamondsuit$

Lemma A.9.5 $\forall \hat{x} \forall \hat{y} !x \in y..$.

Proof. Assume ℓx and ℓy. If $y = \mathsf{T}$ then $!x \in y = !\mathsf{F} = \mathsf{T}$. If $\neg y$ then assume ℓz. Now $\ell(xz)$ so $!xz \sim y$ and $!x \in y = !\forall \hat{z} xz \sim y. = \mathsf{T}$. $\diamondsuit$

Lemma A.9.6 $\forall \hat{x} \forall \hat{y} !x \in\in y..$.

Proof. Assume ℓx, ℓy and ℓz. Now $!x \in z$ and $!z \in y$ so $!x \in\in y = !\exists \hat{z} x \in z \wedge z \in y. = \mathsf{T}$. $\diamondsuit$

Lemma A.9.7 $\forall \hat{S} \forall \hat{x} \forall \hat{y} !x \sim_S^2 y...$.

Proof. $\ell x \rightarrow \forall \hat{S} \forall \hat{y} !x \sim_S^2 y..$ is proved by transfinite induction in x.

If ℓS and ℓy then $!\mathsf{T} \sim_S^2 y = !\mathrm{if}(y, \mathsf{T}, \mathsf{F}) = \mathsf{T}$ so $\forall \hat{S} \forall \hat{y} !\mathsf{T} \sim_S^2 y..$.

If $\neg x$, ℓx, $\forall \hat{z} \forall \hat{S} \forall \hat{y} !xz \sim_S^2 y...$, ℓS, ℓy and ℓz then $\ell(yz)$ and $!xz \sim_S^2 yz$ so $\forall \hat{z} !xz \sim_S^2 yz.$. If $\neg y$ then $!x \sim_S^2 y = !\forall \hat{z} z \in\in S \dot{\Rightarrow} xz \sim_S^2 yz. = \mathsf{T}$ because $\ell z \rightarrow !z \in\in S$ and $\ell z \rightarrow !xz \sim_S^2 yz$. If $y = \mathsf{T}$ then $!x \sim_S^2 = !\mathrm{if}(x, \mathsf{T}, \mathsf{F}) = \mathsf{T}$ so $!x \sim_S^2 y$ also holds in this case. $\diamondsuit$

Lemma A.9.8 $\forall \hat{S} \forall \hat{x} x \sim_S^2 x..$.

Proof. Analogous to the proof of $\forall \hat{x} x \sim x.$. $\diamondsuit$

Lemma A.9.9 $\forall \hat{S} \forall \hat{x} \forall \hat{y} x \sim_S^2 y \Rightarrow y \sim_S^2 x...$.

Proof. Analogous to the proof of $\forall \hat{x} \forall \hat{y} x \sim y \Rightarrow y \sim x..$. $\diamondsuit$

Lemma A.9.10 $\forall \hat{S} \forall \hat{x} \forall \hat{y} \forall \hat{z} x \sim_S^2 y \wedge y \sim_S^2 z \Rightarrow x \sim_S^2 z....$.

Proof. Analogous to the proof of $\forall \hat{x} \forall \hat{y} \forall \hat{z} x \sim y \wedge y \sim z \Rightarrow x \sim z...$. $\diamondsuit$

A.10 Some set-theoretic constructs

Lemma A.10.1 $\ell x \to \ell(ax); \ell S; \ell u \wedge \ell v \wedge u \sim^2_S v \to au \sim av \vdash \ell a$.

Proof. $\ell x \to \ell(ax)$ and $\ell\mathsf{T}$ yields $\ell(a\mathsf{T})$ which entails $!a$. If $a = \mathsf{T}$ then ℓa. If $\neg a$ then the assumptions plus the definition of ℓ yields ℓa. $\diamondsuit$

Definition A.10.2 $\emptyset = \mathsf{T}$.

Lemma A.10.3 $\ell\emptyset$.

Proof. Follows from $\ell\mathsf{T}$. $\diamondsuit$

Definition A.10.4 $x \notin y = \neg x \in y$.

Lemma A.10.5 $\forall\hat{x} x \notin \emptyset$.

Proof. $x \notin \emptyset = \neg x \in \mathsf{T} = \neg\mathsf{F} = \mathsf{T}$. $\diamondsuit$

Definition A.10.6 $\{x\} = \hat{y}x.$.

Lemma A.10.7 (C-K) $\forall\hat{x}\ell\{x\}.$.

Proof. Assume ℓx. If ℓz then $\ell(\hat{y}x.z) = \ell x = \mathsf{T}$ so $\ell z \to \ell(\{x\}z)$. Now let $S = \emptyset$ and assume ℓu, ℓv and $u \sim^2_S v$. Now $\hat{y}x.u \sim x$ and $\hat{y}x.v \sim x$ so $\{x\}u \sim \{x\}v$. The lemma now follows from Lemma A.10.1. $\diamondsuit$

Lemma A.10.8 $\forall\hat{x}\forall\hat{y} y \in \{x\} \Leftrightarrow y \sim x..$.

Proof. Assume ℓx and ℓy. If $y \in \{x\}$ then let u satisfy $y \sim \{x\}u$. Now $\{x\}u = x$ so $y \sim x$ and $y \in \{x\} \to y \sim x$. On the contrary, if $y \sim x$ then $y \sim \{x\}\mathsf{T}$ so $y \in \{x\}$. $\diamondsuit$

Lemma A.10.9 $\ell\mathsf{F}$.

Proof. $\mathsf{F} = \{\mathsf{T}\}$. $\diamondsuit$

In the following, the sentence "let u satisfy $\mathcal{A}$" is shorthand for "let $u = \varepsilon\hat{u}\mathcal{A}.$". Whenever $\exists\hat{u}\mathcal{A}. = \mathsf{T}$, the sentence "let u satisfy $\mathcal{A}$" has the consequence that ℓu and that $\mathcal{A} = \mathsf{T}$ for that particular u.

Lemma A.10.10 If $a \in b$ then $\neg b$ and $\exists\hat{z} a \sim bz.$.

Proof. Follows from the definition of $a \in b$. $\diamondsuit$

Lemma A.10.11 If ℓa, ℓb, ℓc, $\neg b$ and $a \sim bc$ then $a \in b$.

Proof. ℓa and ℓb yields $\forall\hat{u}!a \sim bu.$ which combined with $a \sim bc$ yields $\exists\hat{z} a \sim bz.$ which combined with $\neg b$ yields $a \in b$. $\diamondsuit$

Definition A.10.12 $\{x, y\} = \mathsf{P}xy$.

Lemma A.10.13 (C-P) $\forall\hat{x}\forall\hat{y}\ell\{x,y\}..$.

Proof. If ℓz then $!z$ so $\ell(\mathsf{P}xyz)$ which proves $\ell z \to \ell(\mathsf{P}xyz)$. Now let $S = \emptyset$ and assume ℓu, ℓv and $u \sim^2_S v$. $u \sim^2_S v$ yields $\wr u = \wr v$ so $\mathsf{P}xyu = \mathsf{P}xyv$ which entails $\mathsf{P}xyu \sim \mathsf{P}xyv$. The lemma now follows from Lemma A.10.1. $\diamondsuit$

Lemma A.10.14 $\forall\hat{x}\forall\hat{y}\forall\hat{z}z \in \mathsf{P}xy \Leftrightarrow z \sim x \vee z \sim y...$.

Proof. Assume ℓx, ℓy and ℓz. If $z \in \mathsf{P}xy$ then let u satisfy ℓu and $z \sim \mathsf{P}xyu$. If $u = \mathsf{T}$ then $z \sim x$ and if $\neg u$ then $z \sim y$. In any case, $z \sim x \vee z \sim y$. If $z \sim x$ then $z \sim \mathsf{P}xy\mathsf{T}$ yields $z \in \mathsf{P}xy$ and if $z \sim y$ then $z \sim \mathsf{P}xy\mathsf{F}$ yields $z \in \mathsf{P}xy$. $\diamondsuit$

A.11 Union sets

Define

$$\begin{array}{lcl} \bigcup & = & \hat{x}\varepsilon\hat{y}\forall\hat{z}z \in y \Leftrightarrow z \in\in x... \\ x \subseteq y & = & \forall\hat{z}z \in x \Rightarrow z \in y. \\ x \cup y & = & \bigcup\{x,y\} \end{array}$$

Lemma A.11.1 If ℓa and ℓb then $\ell(\bigcup b)$ and $a \in \bigcup b \Leftrightarrow a \in\in b$.

Proof. Follows from Axiom U which says $\forall\hat{x}\exists\hat{y}\forall\hat{z}z \in y \Leftrightarrow z \in\in x...$. $\diamondsuit$

Lemma A.11.2 If ℓa then $\emptyset \subseteq a$.

Proof. Assume ℓz. $z \in \emptyset = \mathsf{F}$ gives $z \in \emptyset \Rightarrow z \in a$. $\diamondsuit$

Lemma A.11.3 If ℓa and ℓb then $ab \subseteq \bigcup a$.

Proof. The lemma holds trivially if $a = \mathsf{T}$. Now assume $\neg a$. Further assume ℓz and $z \in ab$. Now $z \in ab$ and $ab \in a$ gives $z \in\in a$ so $z \in \bigcup a$. Hence, $\forall\hat{z}z \in ab \Rightarrow z \in \bigcup a.$. $\diamondsuit$

Lemma A.11.4 If ℓa and ℓb then $a \subseteq \{a,b\}$ and $b \subseteq \{a,b\}$.

Proof. Let $c = \{a,b\}$. Now $a = c\mathsf{T}$ and $b = c\mathsf{F}$ which entail $a \subseteq \bigcup c$ and $b \subseteq \bigcup c$, respectively. $\diamondsuit$

Lemma A.11.5 If ℓa and ℓb then $a \in b \Leftrightarrow a \in\in \{b\}$.

Proof. If $a \in b$ then $a \in b$ and $b \in \{b\}$ gives $a \in\in \{b\}$. If $a \in\in \{b\}$ then choose z such that $a \in z$ and $z \in \{b\}$. Now $z \sim b$ so $a \in b$. $\diamondsuit$

A.12 Equality predicates

The construct $x \sim_S y$ is defined analogous to $x \sim_S^2 y$:

$$x \sim_S y = x \begin{cases} y \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ y \begin{cases} \mathsf{F} \\ \forall x \in S : xz \sim_S yz. \end{cases} \end{cases}$$

Lemma A.9.7, A.9.8, A.9.9 and A.9.10 carry over immediately:

Lemma A.12.1 $\forall \hat{S} \forall \hat{x} \forall \hat{y} ! x \sim_S y \ldots$.

Lemma A.12.2 $\forall \hat{S} \forall \hat{x} x \sim_S x ..$.

Lemma A.12.3 $\forall \hat{S} \forall \hat{x} \forall \hat{y} x \sim_S y \Rightarrow y \sim_S x \ldots$.

Lemma A.12.4 $\forall \hat{S} \forall \hat{x} \forall \hat{y} \forall \hat{z} x \sim_S y \wedge y \sim_S z \Rightarrow x \sim_S z \ldots.$.

The definitions of $x \sim_S y$ and $x \sim_S^2 y$ are recursive and they are shorthand for

$$\begin{array}{rcl} x \sim_S y & = & \mathsf{Y}(FS)xy \\ x \sim_S^2 y & = & \mathsf{Y}(GS)xy \end{array}$$

where

$$\begin{array}{rcl} F & = & \hat{S}\hat{g}\hat{x}\hat{y}x \begin{cases} y \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ y \begin{cases} \mathsf{F} \\ \forall z \in S : g(xz)(yz). \end{cases} \end{cases} \\ G & = & \hat{S}\hat{g}\hat{x}\hat{y}x \begin{cases} y \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ y \begin{cases} \mathsf{F} \\ \forall z \in\in S : g(xz)(yz). \end{cases} \end{cases} \end{array}$$

Lemma A.12.5 If ℓU, ℓV and $\forall \hat{z} z \in U \Leftrightarrow z \in\in V$. then $x \sim_U y = x \sim_V^2 y$.

Proof. The assumptions give $FU = GV$ from which $x \sim_U y = x \sim_V^2 y$ follows. $\diamondsuit$

Corollary A.12.6 If ℓS then $x \sim_{\bigcup S} y = x \sim_S^2 y$ and $x \sim_S y = x \sim_{\{x\}}^2 y$.

Corollary A.12.7 $\forall \hat{U} \exists \hat{V} \forall \hat{x} \forall \hat{y} x \sim_U y \Leftrightarrow x \sim_V^2 y \ldots.$.

Corollary A.12.8 $\forall\hat{V}\exists\hat{U}\forall\hat{x}\forall\hat{y}x \sim_U y \Leftrightarrow x \sim_V^2 y.\dots$.

Lemma A.12.9 If ℓU, ℓV and $U \subseteq V$ then $\forall\hat{x}\forall\hat{y}x \sim_V y \Rightarrow x \sim_U y..$.

Proof. By induction in x. The case where $x = \mathsf{T}$ is trivial. Now assume ℓx, $\neg x$ and $\forall\hat{z}\forall\hat{y}xz \sim_V y \Rightarrow xz \sim_U y..$. Further assume ℓy and $x =_V y$. $\neg x$ and $x \sim_V y$ gives $\neg y$. Further, $x \sim_V y$ gives $\forall z \in V\colon xz \sim_V yz.$ which entails $\forall z \in U\colon xz \sim_V yz.$ by $U \subseteq V$ which in turn entails $\forall z \in U\colon xz \sim_U yz.$ by the induction hypothesis. This proves $x \sim_U y$. $\diamondsuit$

A.13 A simplified limitation predicate

The definition of ℓ is recursive and is shorthand for $\ell = \mathsf{Y}G$ where

$$G = \hat{g}\hat{x}x \begin{cases} \mathsf{T} \\ \forall\hat{y}g(xy). \wedge \exists\hat{S}\forall\hat{u}\forall\hat{v}u \sim_S^2 v \Rightarrow xu \sim xv\dots \end{cases} ..$$

Now define $\bar{\ell} = \mathsf{Y}F$ where

$$F = \hat{g}\hat{x}x \begin{cases} \mathsf{T} \\ \forall\hat{y}g(xy). \wedge \exists\hat{S}\forall\hat{u}\forall\hat{v}u \sim_S v \Rightarrow xu \sim xv\dots \end{cases} ..$$

Lemma A.13.1 $\ell = \bar{\ell}$.

Proof. Follows from $F = G$. $\diamondsuit$

In other words, having the union set axiom available, $u \sim_S^2 v$ can be replaced with $u \sim_S v$ in the definition of ℓ without change of semantics. It is the hope of the author, however, that the union set axiom can be proved from the other axioms when $u \sim_S^2 v$ is used instead of $u \sim_S v$.

Lemma A.13.2 If $\forall\hat{y}\ell(ay).$, ℓS and $\forall\hat{u}\forall\hat{v}u \sim_S v \Rightarrow au \sim av..$ then ℓa.

Proof. Follows from Lemma A.10.1 and the lemma above. $\diamondsuit$

Now define the "inner range" $\mathcal{I}x$ as follows:

$$\mathcal{I}x = \varepsilon\hat{S}\forall\hat{u}\forall\hat{v}u \sim_S v \Rightarrow xu \sim xv\dots$$

Lemma A.13.3 If ℓa then $\ell(\mathcal{I}a)$ and if ℓb, ℓc and $b \sim_{\mathcal{I}a} c$ then $ab \sim ac$.

Proof. ℓa entails $!a$. The lemma is trivial if $a = \mathsf{T}$ and follows from $\ell = \bar{\ell}$ otherwise. $\diamondsuit$

Lemma A.13.4 If ℓb, ℓc, ℓS, ℓa, $b \sim_S c$ and $\mathcal{I}a \subseteq S$ then $ab \sim ac$.

Proof. $b \sim_S c$ and $\mathcal{I}a \subseteq S$ gives $b \sim_{\mathcal{I}a} c$ which entails $ab \sim ac$. $\diamondsuit$

A.14 The Replacement Theorem

Lemma A.14.1 $\forall\hat{u}\forall\hat{v}u \sim v \Rightarrow \ell(u \downarrow v) \wedge u \sim u \downarrow v..\,.$

Proof. By induction in u. If $u = \mathsf{T}$: Assume ℓv and $u \sim v$. We have $v = \mathsf{T}$ so $u \downarrow v = \mathsf{T}$ and $\ell(u \downarrow v) \wedge u \sim u \downarrow v$ holds.

Now assume $\neg u$, ℓu and $\forall\hat{x}\forall\hat{v}ux \sim v \Rightarrow \ell(ux \downarrow v) \wedge ux \sim ux \downarrow v$. We shall prove $\forall\hat{v}u \sim v \Rightarrow \ell(u \downarrow v) \wedge u \sim u \downarrow v$.

Assume ℓv and $u \sim v$. We now have $\neg v$.

If ℓx then $\ell(vx)$ and $ux \sim vx$ so $\ell(ux \downarrow vx)$ and $\ell((u \downarrow v)x)$ which proves $\forall\hat{x}\ell((u \downarrow v)x).\,.$

Let $S = \mathcal{I}u$. Now $\forall\hat{r}\forall\hat{t}r \sim_S t \Rightarrow ur \sim ut..\,.$ We shall prove $\forall\hat{r}\forall\hat{t}(u \downarrow v)r \sim (u \downarrow v)t...$ Assume ℓr, ℓt and $r \sim_S t$. Now $\ell(vr)$ and $ur \sim ut$ so $ur \sim ur \downarrow vr$ and therefor $ur \sim (u \downarrow v)r$. Likewise, $ut \sim (u \downarrow v)t$ which combined with $ur \sim ut$ gives $(u \downarrow v)r \sim (u \downarrow v)t$. Hence, $\exists\hat{S}\forall\hat{r}\forall\hat{t}r \sim_S t \Rightarrow (u \downarrow v)r \sim (u \downarrow v)t....$

This allows to conclude $\ell(u \downarrow v)$.

We shall now prove $u \sim u \downarrow v$. To do so it is sufficient to prove $\forall\hat{x}ux \sim (u \downarrow v)x.\,.$ Assume ℓx. Now $\ell(vx)$ and $ux \sim vx$ so $ux \sim ux \downarrow vx$ which combined with $ux \downarrow vx = (u \downarrow v)x$ gives $ux \sim (u \downarrow v)x$. $\diamondsuit$

Lemma A.14.2 $\ell a; a \preceq b \vdash a \sim b$.

Proof. Assume ℓa and $a \preceq b$. Now $a \sim a = \mathsf{T}$ and $a \sim a \preceq a \sim b$ by monotonicity so $a \sim b = \mathsf{T}$. $\diamondsuit$

Lemma A.14.3 (Extensionality of $\sim$) $\forall\hat{x}\ell(fx) \wedge \ell u \wedge \ell v \wedge u \sim v \rightarrow fu \sim fv$.

Proof. Assume $\forall\hat{x}\ell(fx)$, ℓu, ℓv and $u \sim v$. Now $\ell(u \downarrow v)$ so $\ell(f(u \downarrow v))$ which combined with $f(u \downarrow v) \preceq fu$ gives $f(u \downarrow v) \sim fu$. Likewise, $f(u \downarrow v) \sim fv$ so $fu \sim fv$. $\diamondsuit$

Theorem A.14.4 (Replacement) $\forall\hat{x}\ell(fx).\wedge \ell a \rightarrow \ell\hat{x}f(ax).\,.$

Proof. Let $b = \hat{x}f(ax).\,.$ Assume $\forall\hat{x}\ell(fx).\,.$ If ℓx then $\ell(ax)$ and $\ell(f(ax))$ so $\forall\hat{x}\ell(f(ax)).$ and $\forall\hat{x}\ell(bx).\,.$ Let $S = \mathcal{I}a$ and assume ℓu, ℓv and $u \sim_S v$. Now $au \sim av$ so $f(au) \sim f(av)$. Hence, $\forall\hat{u}\forall\hat{v}u \sim_S v \Rightarrow gu \sim gv..$ which proves ℓb. $\diamondsuit$

A.15 Remaining axioms of the original version

Lemma A.15.1 $\ell a \rightarrow \ell\hat{x}axx.\,.$

Proof. Let $b = \hat{x}axx.\,.$ Assume ℓa. If ℓx then $\ell(ax)$ and $\ell(axx)$ so $\forall\hat{x}\ell(bx).\,.$ Let $c = \hat{x}\mathcal{I}(ax).$ and $d = \mathcal{I}a \cup \bigcup c$. Now ℓc by replacement so ℓd. Assume ℓu, ℓv and $u \sim_d v$. $\mathcal{I}a \subseteq d$ gives $au \sim av$ so $auu \sim avu$. Furthermore, $\mathcal{I}(av) \subseteq d$ gives $avu \sim avv$ so $auu \sim avv$. Hence, $\forall\hat{u}\forall\hat{v}u \sim_d v \Rightarrow bu \sim bv..\,.$ $\diamondsuit$

Lemma A.15.2 $\forall\hat{x}\ell(fx).\wedge \ell a \rightarrow \ell\hat{x}f(ax)x.\,.$

Proof. Let $g = \hat{x}f(ax).$. Now ℓg by replacement so $\ell\hat{x}gxx.$ by the lemma above. Now $\hat{x}gxx. = \hat{x}f(ax)x.$ shows $\ell\hat{x}f(ax)x.$. $\diamondsuit$

Lemma A.15.3 (C-M1) $\forall\hat{y}\ell\hat{x}a.. \rightarrow \forall\hat{y}\ell\hat{x}\hat{y}a.(yx)..$.

Proof. Let $f = \hat{y}\hat{x}a..$ and assume $\forall\hat{y}\ell\hat{x}a..$ and ℓy. Now $\forall\hat{y}\ell\hat{x}fux..$ which entails $\forall\hat{y}\ell(fy).$. Hence, $\ell\hat{x}f(yx)x.$ by the above theorem which proves $\ell\hat{x}\hat{y}a.(yx).$. $\diamondsuit$

Lemma A.15.4 (C-M2) $\forall\hat{y}\ell\hat{x}a.. \rightarrow \forall\hat{y}\ell\hat{x}\hat{x}a.(xy)..$.

Proof. Assume $\forall\hat{y}\ell\hat{x}a..$ and ℓy and let $f = \hat{y}\hat{x}a..$ and $b = fy$. Now $\forall\hat{y}\ell\hat{x}fyx..$ so $\ell\hat{x}fyx.$ and $\ell\hat{x}bx.$ which entails ℓb. We shall prove $\ell\hat{x}fy(xy).$, i.e. we shall prove $\ell\hat{x}b(xy).$. If ℓx then $\ell(b(xy))$ so $\hat{x}\ell(b(xy)).$. Now let $S = \mathcal{I}b \cup \{y\}$ and assume ℓu, ℓv and $u \sim_S v$. $y \in S$ gives $uy \sim_S vy$ and $\mathcal{I}b \subseteq S$ gives $b(uy) \sim b(vy)$ so $\forall\hat{u}\forall\hat{v}u \sim_S v \Rightarrow b(uy) \sim b(vy)..$. $\diamondsuit$

Lemma A.15.5 $\ell x \wedge \ell y \wedge \ell z \wedge \ell S \wedge x \sim_S y \rightarrow (x,z) \sim_S (y,z)$.

Proof. Assume ℓu and $u \in S$. If $u = \mathsf{T}$ then $(x,z)u = x$ and $(y,z)u = y$ so $(x,z)u \sim_S (y,z)u$. If $\neg u$ then $(x,z)u = z$ and $(y,z)u = z$ so $(x,z)u \sim_S (y,z)u$. In any case, $(x,z)u \sim_S (y,z)u$ so $\forall\hat{u} \in S\colon (x,z)u \sim_S (y,z)u.$ which proves $(x,z) \sim_S (y,z)$. $\diamondsuit$

Lemma A.15.6 $\ell x \wedge \ell y \wedge \ell z \wedge \ell S \wedge x \sim_S y \rightarrow (z,x) \sim_S (z,y)$.

Proof. Same as above. $\diamondsuit$

Lemma A.15.7 (C-Curry) $\ell a \rightarrow \ell\hat{x}\hat{y}a(x,y)..$.

Proof. Assume ℓa and let $b = \hat{x}\hat{y}a(x,y)..$ and $S = \mathcal{I}a$. We shall prove ℓb.

Now assume ℓx and let $c = bx$. We now prove ℓc: Assume ℓy. Now $\ell(x,y)$ so $\ell(a(x,y))$ which proves $\ell(cy)$. Hence, $\forall\hat{y}\ell(cy).$. Now assume ℓu, ℓv and $u \sim_S v$. The assumptions give $(x,u) \sim_S (x,v)$ so $a(x,u) \sim a(x,v)$ which proves $\forall\hat{u}\forall\hat{v}cu \sim cv..$. Hence, ℓc. This proves $\forall\hat{x}\ell(bx).$.

Now assume ℓu, ℓv and $u \sim_S v$. We now prove $bu \sim bv$: Assume ℓy. Now $(u,y) \sim_S (v,y)$ so $a(u,y) \sim a(v,y)$ which proves $\forall\hat{y}a(u,y) \sim a(v,y).$ and $\hat{y}a(u,y) \sim \hat{y}a(v,y)$ which is equivalent to $bu \sim bv$.

In conclusion, $\ell b = \mathsf{T}$ which concludes the lemma. $\diamondsuit$

Lemma A.15.8 If ℓa, ℓb, $\forall\hat{x}\ell(fx).$ and

$$cx = x \left\{ \begin{array}{l} a \\ f\hat{z}c(x(bz)). \end{array} \right.$$

then $\forall\hat{x}\ell(cx).$.

Proof. By induction in x. If $x = \mathsf{T}$ then $cx = a$ so $\ell(cx)$.

Now assume $\neg x$, ℓx and $\forall \hat{y} \ell(c(xy))..$ In this case $cx = f\hat{z}c(x(bz))..$ Hence, to prove $\ell(cx)$ it is sufficient to prove $\ell \hat{z}c(x(bz))..$

If ℓz then $\ell(bz)$ so $\ell(c(x(bz)))$ according to $\forall \hat{y} \ell(c(xy))..$ Hence, $\forall \hat{z} \ell(c(x(bz$ $))).$.

Now let $S = \mathcal{I}b$ and assume ℓu, ℓv and $u \sim_S v$. Now $bu \sim bv$ which combined with $\forall \hat{y} \ell(c(xy)).$ gives $c(x(bu)) \sim c(x(bv))$.

Hence, $\ell \hat{z}c(x(bz)).$ and $\ell(cx)$ which concludes the proof. $\diamondsuit$

Lemma A.15.9 If ℓa, ℓb, $\forall \hat{x} \ell(fx).$, $S = \hat{x}bx.$ and

$$cx = x \left\{ \begin{array}{l} a \\ f\hat{z}c(x(bz)). \end{array} \right.$$

then $\forall \hat{u} \forall \hat{v} u \sim_S v \Rightarrow cu \sim cv...$

Proof. By induction in u. If $u = \mathsf{T}$, ℓv and $u \sim_S v$ then $v = \mathsf{T}$ and $cu = cv = a$ so $cu \sim cv$.

Now assume $\neg u$, ℓu and $\forall \hat{w} \forall \hat{v} uw \sim_S v \Rightarrow c(uw) \sim cv...$ Further assume ℓv and $u \sim_S v$. In this case $\neg v$, $cu = f\hat{z}c(u(bz)).$ and $cv = f\hat{z}c(v(bz))..$

Now assume ℓz. $bz \in S$ and $u \sim_S v$ gives $u(bz) \sim_S v(bz)$ which combined with the induction hypothesis gives $c(u(bz)) \sim c(v(bz))$. Hence, $\forall \hat{z} c(u(bz)) \sim c(v(bz)).$ which proves $\hat{z}c(u(vz)). \sim \hat{z}c(v(bz))..$ This combined with $\ell \hat{z}c(u(bz)).$ and $\ell \hat{z}c(v(bz)).$ (which follow from the Replacement Theorem) yields $f\hat{z}c(u(bz)).$ $\sim f\hat{z}c(v(bz)).$ so $cu \sim cv$ as required. $\diamondsuit$

Lemma A.15.10 If ℓa, ℓb, $\forall \hat{x} \ell(fx).$ and

$$cx = x \left\{ \begin{array}{l} a \\ f\hat{z}c(x(bz)). \end{array} \right.$$

then ℓc.

Proof. Follows from the previous two lemmas. $\diamondsuit$

Corollary A.15.11 (C-Prim) $\ell a \wedge \ell b \wedge \forall \hat{x} \ell(fx). \rightarrow \underline{\mathsf{Prim}} fab$ where

$$\underline{\mathsf{Prim}} = \hat{f}\hat{a}\hat{b}\mathsf{Y}\hat{g}\hat{x}x \left\{ \begin{array}{l} a \\ f\hat{z}g(x(bz)). \end{array} \right. \quad .....$$

B Models of stable map theory

This Appendix describes how the models of original MT in [6] and [3], respectively, can be changed into models of stable MT.

B.1 First model

This section describes how the model M_O of original MT in [6] can be changed into a model M_S which is conjectured to be a model of stable MT. Only the changes of the model are listed. The very tedious verification of each and every axiom of stable MT in the model has not yet been carried out. The changes from M_O to M_S are as follows:

In M_O, $\breve{\Phi}$ is a set whose elements are interpreted as well-founded maps. Distinct elements of $\breve{\Phi}$ represent distinct well-founded maps and $(\breve{\Phi}, \leq)$ has a rich structure.

In M_S, the elements of $\breve{\Phi}$ are reinterpreted. If $x \in \breve{\Phi}$ represents a well-founded map y in M_O, then x represents the unique minimal limited map z for which $z \leq y$ in M_S. With this interpretation, $(\breve{\Phi}, \leq)$ is flat in the sense that

$$x = y \Leftrightarrow x \leq y$$

for all $x, y \in \breve{\Phi}$ in M_S. Furthermore, distinct objects of $\breve{\Phi}$ may represent the same minimal limited map.

This change allows to throw out $\breve{Q}$ from the model construction. $\breve{Q}$ is present in M_O in order to allow the semantics of elements of $\breve{\Phi}$ to be defined gradually by transfinite recursion and in order to define $\dot{W}$ (which is the counterpart of ℓ) in M_O. (In M_S, ℓ is a defined rather than a fundamental concept and needs no attention).

Because of the absence of $\breve{Q}$ in M_S, the semantics of elements of $\breve{\Phi}$ has to be introduced at the initial step of the transfinite recursion in Section 11.4 in [6]. More precisely, the changes are:

In Section 11.4, remove $\dot{W}$ and $\breve{Q}$ from the direct sum that defines $\acute{M}$ and remove equations (39), (40) and (43) that give semantics to $\dot{W}$ and $\breve{Q}$.

In Section 11.4, to enforce the reinterpretation of elements of $\breve{\Phi}$ change equation (42) to

$$\acute{r}'(v)(\acute{m}(w,u)) = w\langle\langle i \in u^d \mapsto j \in w^{dRD} \mapsto X(v,u,i,j)\rangle\rangle$$

where

$$X(v,u,i,j) = \begin{cases} v(\acute{m}(u(i),j)) & \text{if } \forall j \in \breve{\Phi}^*: v(\acute{m}(u(i),j)) \neq \tilde{\bot} \\ \tilde{\bot} & \text{otherwise} \end{cases}$$

In Section 11.4, to introduce the semantics of $\breve{\Phi}$ at the beginning of the transfinite recursion, change the definition of $\acute{r}''(0)$ to

$$\begin{array}{lcll} \acute{r}''(0)(\acute{m}(w,u)) & = & w\langle\langle u\rangle\rangle & \text{for } w{\in}\breve{\Phi},\ u{\in}\acute{M}^* \\ \acute{r}''(0)(x) & = & \tilde{\bot} & \text{if } x{\in}\acute{M},\ \neg\exists w{\in}\breve{\Phi}\exists u{\in}\acute{M}^*: x = \acute{m}(w,u) \end{array}$$

To take account of the change of $\acute{r}''(0)$ in the proofs, change $\acute{M} \rightarrow L$ to $\acute{M} \tilde{\rightarrow} L$ everywhere i Section 11.5 and 11.6 where

$$\acute{M} \tilde{\rightarrow} L = \{v \in \acute{M} \rightarrow L \mid \acute{r}''(0) \leq v\}$$

The relations $x \dot{=}^{\Phi}_{v} y$ and $x \dot{=}^{Q}_{v,w} y$ no longer play a role in the model construction because of the removal of $\dot{W}$ and $\breve{Q}$. It might help to know, however, that the analogy of $x \dot{=}^{\Phi}_{v} y$ could be defined by

$$x \dot{=}^{\Phi}_{v} y \Leftrightarrow \forall z \in \breve{\Phi}^{*}\colon a(x)(z) = v(\acute{m}(y, z))$$

With this definition of $x \dot{=}^{\Phi}_{v} y$, and if $\dot{W}$ is kept in the model, then $\dot{W}$ obtains the semantics of ℓ.

B.2 Second model

The stability axiom and the change from well-founded to limited maps made possible by stability are the central differences between original and stable MT. The corresponding changes to the model in [3] are outlined in the following.

For all regular ordinals κ, [3] defines the notion of "κ-Scott domain", "κ-compact" and so on as generalisations of "Scott domain", "compact" and so on, respectively, where the notions coincide for $\kappa = \omega$.

Furthermore, [3] defines a κ-Scott domain

$$\mathcal{D} = (D_{\mathcal{D}}, \leq_{\mathcal{D}})$$

a set Φ of well-founded maps

$$\Phi \subseteq D_{\mathcal{D}}$$

and two κ-continuous isomorphisms $\tilde{A}$ and $\tilde{\lambda}$ where

$$\begin{array}{lcl} \tilde{A} & \in & \mathcal{D} \to ((\mathcal{D} \to \mathcal{D}) \oplus_{\perp} \{\mathsf{T}\}) \\ \tilde{\lambda} & \in & ((\mathcal{D} \to \mathcal{D}) \oplus_{\perp} \{\mathsf{T}\}) \to \mathcal{D} \end{array}$$

and where $\tilde{A}$ and $\tilde{\lambda}$ are the inverses of each other. The structures $\mathcal{D}$, Φ, A and Tr together are used for defining a model of original MT where $D_{\mathcal{D}}$ is the domain of the model, A is used for defining functional application, Tr is used for lambda abstraction and Φ is used for defining ϕ and ε.

Now define

$$\mathcal{D}' = (D_{\mathcal{D}}, \sim_{\mathcal{D}}, \leq_{\mathcal{D}})$$

where $\sim_{\mathcal{D}} \subseteq D_{\mathcal{D}} \times D_{\mathcal{D}}$ satisfies

$$x \sim_{\mathcal{D}} y \Leftrightarrow \exists z \in D_{\mathcal{D}}\colon x \leq_{\mathcal{D}} z \wedge y \leq_{\mathcal{D}} z$$

The structure $\mathcal{D}'$ is a PCS ([3]). For all PCS's $X = (D_X, \sim_X, \leq_X)$ and sets $Y \subseteq D_X$ define

$$X|_Y = (D_X \cap Y, \sim_X \cap Y^2, \leq_X \cap Y^2)$$

Furthermore, for all PCS's X define $u =_X v$ by

$$u =_X v \Leftrightarrow u \leq_X v \wedge v \leq_X u$$

and define $X_=$ as the quotient

$$X_= = X/{=_X}$$

Let

$$X \simeq Y$$

denote that the PCS's X and Y are isomorphic.

Reference [3] defines a PCS $C = (D, \sim, \leq)$ such that

$$\mathcal{D}' \mid_{\mathcal{D}_p \setminus \{\perp\}} \simeq C_=$$

where $\mathcal{D}_p$ is the set of prime elements of $\mathcal{D}$. Furthermore, [3] defines C^* as the set of coherent, κ-small subsets of C. C^* can be organised as a PCS

$$C' = (C^*, \sim_{C^*}, \leq_{C^*})$$

where

$$\begin{array}{ll} x \sim_{C^*} y & \text{if } x \cup y \text{ is coherent in } C \\ x \leq_{C^*} y & \text{if } \bar{x} \subseteq \bar{y} \end{array}$$

where $\bar{x}$ and $\bar{y}$ are the initial segments in C generated by x and y, respectively.

The PCS C' satisfies

$$\mathcal{D}' \mid_{\mathcal{D}_c} \simeq C'_=$$

where $\mathcal{D}_c$ is the set of κ-compact elements of $\mathcal{D}$.

In [3], κ is assumed to be regular and to satisfy $\sigma < \kappa$ for some inaccessible ordinal σ. In the model presented here, the ordinal κ is furthermore assumed to be inaccessible itself. It is likely that this extra assumption can be dropped.

For all sets X of sets define the set $\mathrm{ch}(X)$ of choice functions over X by

$$\mathrm{ch}(X) = \{f \in X \rightarrow \bigcup X \mid \forall x \in X\colon fx \in x\}$$

Note that $\mathrm{ch}(X)$ is empty if X contains the empty set and that $\mathrm{ch}(X)$ is κ-small if X is κ-small (the inaccessibility of κ is used here).

C has the property that any non-empty, κ-small subset X of C has an infimum $\inf_C(X)$ in C except if $t \in X$ and X contains at least one more element (the exception is due to the absence of $\perp$ in the sense that $C_= \simeq \mathcal{D}|_{\mathcal{D}_p \setminus \{\perp\}}$). In particular, any non-empty, κ-small coherent subset X of C has an infimum $\inf_C(X)$ in C. The equivalent holds for C^*: For all non-empty, κ-small coherent subsets Y of C^*, $\inf_{C^*}(Y)$ can be defined as

$$\inf_{C^*}(Y) = \{\inf_C(\{fx \mid x \in Y\}) \mid f \in \mathrm{ch}(Y)\}$$

That $\mathrm{ch}(Y)$ is κ-small ensures that $\inf_{C^*}(Y)$ belongs to C^*.

For $X \subseteq C$ and $Y \subseteq C^*$ define $\underline{\inf}_C(X)$ and $\underline{\inf}_{C^*}(Y)$ by

$$\begin{array}{lcll} \underline{\inf}_C(X) & = & \{\inf_C(X)\} & \text{if } X \text{ has an infimum} \\ \underline{\inf}_C(X) & = & \emptyset & \text{otherwise} \\ \underline{\inf}_{C^*}(Y) & = & \{\inf_{C^*}(Y)\} & \text{if } Y \text{ is coherent} \\ \underline{\inf}_{C^*}(Y) & = & \emptyset & \text{otherwise} \end{array}$$

As can be seen, $\underline{\inf}_C(X)$ and $\underline{\inf}_{C^*}(Y)$ are unit sets or empty.

In [3], $C = (D, \sim, \leq)$ satisfies

$$D = (C^* \times D) \cup \{t, f\}$$

where t and f are not pairs. Now, for all $X \subseteq D$ define

$$\begin{array}{lcl} \mathrm{dom}(X) & = & \{x \mid \exists y\colon (x, y) \in X\} \\ \mathrm{rng}(X) & = & \{y \mid \exists x\colon (x, y) \in X\} \\ \mathrm{com}(X) & = & \underline{\inf}_{C^*}(\mathrm{dom}(X)) \times \underline{\inf}_C(\mathrm{rng}(X)) \end{array}$$

Note that $\mathrm{com}(X)$ is a unit set or empty.

A set $X \subseteq D$ is now defined to be *stable* if

$$Y \subseteq X \Rightarrow \mathrm{com}(Y) \subseteq X$$

It is now possible to state how the conjectured model for stable MT differs from that in [3]: In Section 8.2 in [3], define C^* as the set of coherent, <u>stable</u>, κ-small subsets of D, and define $\mathcal{D}$ as the set of coherent, <u>stable</u> initial segments of D ordered by inclusion.

Further, define the set L of limited maps by

$$L = \uparrow\{\inf(X) \mid X \subseteq \Phi \wedge X \neq \emptyset \wedge X \text{ is coherent}\}$$

Using this L instead of Φ in the definition of ε is conjectured to give a model of stable MT.

C Well-foundedness in the pure version

C.1 Representation of equations

In the pure version (and therefore also in the unstable one) it is possible to let $\exists^{r,s}$ quantify over any collection S of maps that can be defined equationally. Note that, in particular, if S is defined by an inconsistent equational system then S is empty and $\exists^{r,s} a = \bot$ for all a. The following conventions are convenient for handling equational systems:

For all equations $a = b$ let $*(a = b)$ be shorthand for the term (a, b) and for all terms t let $\mathcal{I}t$ be shorthand for the equation $t\mathsf{T} = t\mathsf{F}$. With these conventions, $\mathcal{I}(*(a = b))$ is equivalent to $a = b$. The term $*(a = b)$ will be said to represent the equation $a = b$ and the equation $\mathcal{I}t$ will be said to be the interpretation of the term t as an equation.

For all equations $a = b$ and $c = d$ let $a = b \wedge^\circ c = d$ be shorthand for the equation $(a, c) = (b, d)$. The equation $a = b \wedge^\circ c = d$ holds if and only if $a = b$ and $c = d$ both hold.

For all equations $a = b$ and variables x let $\forall^\circ x : a = b$ be shorthand for the equation $\hat{x}a. = \hat{x}b.$. The equation $\forall^\circ x : a = b$ holds if and only if $a = b$ holds for all maps x.

For all equations $a = b$ let $\exists^{a=b} x : c.$ be shorthand for $\exists^{\hat{x}a.,\hat{x}b.} \hat{x}c.$. The construct $\exists^{a=b} x : c.$ states that there exists an x such that $a = b$ and $c = \mathsf{T}$.

For all terms t let $\exists^t x : c.$ be shorthand for $\exists^{\mathcal{I}t} x : c.$.

C.2 Quantification

As an example of use of these conventions, the equation $\mathrm{All}(a, q)$ will be defined such that $\mathrm{All}(a, q)$ holds if q is the universal quantifier that quantifies over the naive set $\{x \mid ax = T\}$. In particular, $\mathrm{All}(a, \forall_a)$ holds for all terms a.

$$
\begin{array}{lcl}
\mathrm{All}_1(a, q) & \equiv & \forall^\circ x \forall^\circ f : ax \wedge qf \rightarrow fx \\
\mathrm{All}_2(a, q) & \equiv & \forall^\circ f : qf = q\hat{x}ax \wedge fx. \\
\mathrm{All}_3(a, q) & \equiv & \forall^\circ f : q\hat{x}!fx. = !q\hat{x}fx. \\
\mathrm{All}_4(a, q) & \equiv & q\hat{x}\mathsf{T}. = \mathsf{T} \\
\mathrm{All}(a, q) & \equiv & \mathrm{All}_1(a, q) \wedge^\circ \mathrm{All}_2(a, q) \wedge^\circ \mathrm{All}_3(a, q) \wedge^\circ \mathrm{All}_4(a, q)
\end{array}
$$

C.3 Well-foundedness w.r.t. a set

Let g be a map and let $G = \{x \mid gx = \mathsf{T}\}$. In original MT, a map f is said to be *well-founded* with respect to G if, for all $x_1, x_2, \dots \in G$ there exists an n such that $fx_1 \cdots x_n = \mathsf{T}$.

For all sets G let G° denote the set of maps that are well-founded w.r.t. G. (As a special case and somewhat an exception let $\emptyset^\circ$ denote the set of all maps except $\perp$ where $\emptyset$ is the empty set). Now define the map $\underline{w}$ by

$$
\underline{w} = \hat{q}\hat{f}f \left\{ \begin{array}{l} \mathsf{T} \\ q\hat{x}\underline{w}q(fx). \end{array} \right. ..
$$

With these definitions we have that $\underline{w}\forall_g f = \mathsf{T}$ if and only if f is well-founded w.r.t. G, i.e. if $f \in G^\circ$.

C.4 Decorations

A "decoration" of a map m is a map d which contains all information about m and also contains a "comment" for all maps $x_1, \dots, x_n$ and all n for which $mx_1 \cdots x_n$ is a function.

A decoration d of a function m is represented as a pair (f, c) where c is a comment on m and where fx is a decoration of mx for all x. The maps T and $\perp$ are their own decorations. When used recursively, this shows that

$d\mathsf{T}x_1\mathsf{T}x_2\cdots\mathsf{T}x_n\mathsf{F}$ is the comment assigned to $mx_1\cdots x_n$ and that the root of $d\mathsf{T}x_1\mathsf{T}x_2\cdots\mathsf{T}x_n$ equals the root of $mx_1\cdots x_n$. Now define the map $\underline{m}$ by

$$\underline{m} = \hat{d}d\left\{\begin{array}{l}\mathsf{T}\\ \hat{x}\underline{m}(d\mathsf{T}x)\end{array}\right.$$

The map $\underline{m}$ takes a decoration d of a map m and returns m itself. In other words, $\underline{m}$ strips away all the comments contained in d.

C.5 An informal definition of well-foundedness

In original MT, the collection of well-founded maps is informally defined to be the least collection that satisfies the following rules:

1. T is well-founded.
2. If G is a collection of *limited size* of well-founded maps, and if fx is well-founded for all $x \in G^\circ$, then f is well-founded.

What it means for a collection G to be of "limited size" is defined together with the notion of well-foundedness as follows: A well-founded map f is said to "contain" all those well-founded maps that have to be proven well-founded before f can be proven well-founded. Let f^c denote the contents of f. A collection is said to be of limited size if it is a subset of f^c for some well-founded f.

As an example, the well-foundedness of T does not depend on the well-foundedness of any other map, so T contains no maps. Hence, the empty set is of limited size. If g is well-founded, if fx is well-founded for all $x \in (g^c)^\circ$, and if this is used to prove f well-founded, then f contains g, f contains fx for all $x \in (g^c)^\circ$ and f contains all maps contained by those maps.

C.6 Decorated well-founded maps

As mentioned above, a well-founded map f is said to contain all those well-founded maps that have to be proven well-founded before f can be proven well-founded. This is an ambiguous definition of containment because it depends on the order in which maps are proven to be well-founded.

To overcome this problem, the notion of "decorated, well-founded maps" is introduced. A decorated, well-founded map d is a decoration of a well-founded map m that records the contents of m as comments. In particular, $m = \underline{m}d$.

A decorated, well-founded map d is either T or a function. If it is a function, then it must have the form $(f, (q, d'))$ where (q, d') is the comment assigned to m. For $d = (f, (q, d'))$ to be a decorated, well-founded map, it must satisfy the following:

1. d' must be a decorated, well-founded map.

2. q must be a universal quantifier which quantifies over G° where G is the collection of maps contained in d'.

3. fx must be a decorated, well-founded map for all $x \in G^\circ$.

C.7 Quantification over contents

Define

$$\begin{array}{lcl} \underline{q}dg & = & d\left\{\begin{array}{l} \mathsf{T} \\ \underline{q}_1 dg \end{array}\right. \\ \underline{q}_1(f,(q,d'))g & = & \underline{q}_2 d'g \wedge q\hat{x}\underline{q}_2(fx)g. \\ \underline{q}_2 dg & = & g(\underline{m}d) \wedge \underline{q}dg \end{array}$$

The equation $\underline{q}_1(f,(q,d'))q = \cdots$ means that $\underline{q}_1$ has to be defined such that the equation holds. An obvious definition would be $\underline{q}_1 = \hat{d}\hat{g}\underline{q}_2(d\mathsf{FF})g \wedge (d\mathsf{FT})\hat{x}\underline{q}_2((d\mathsf{T})x)g\ldots$.

If d is a decorated, well-founded map then $\underline{q}dg = \mathsf{T}$ if $gx = \mathsf{T}$ for all well-founded x contained in d. In other words, $\underline{q}d$ is a universal quantifier which quantifies over the contents of d.

C.8 Formalisation of decorated, well-founded maps

Define

$$\begin{array}{lcl} \overline{w}d & = & d\left\{\begin{array}{l} \mathsf{T} \\ \overline{w}_1 d \end{array}\right. \\ \overline{w}_1(f,(q,d')) & = & \overline{w}d' \wedge q\hat{x}\overline{w}(fx). \\ \overline{d}d & = & *(\overline{w}d = \mathsf{T} \wedge^\circ \neg d \rightarrow \mathcal{I}(\overline{d}_1 d) \\ \overline{d}_1(f,(q,d')) & = & *(\mathcal{I}(\overline{c}(q,d'))) \wedge^\circ \forall^\circ x : \underline{w}(\underline{q}d')x \rightarrow \mathcal{I}(\overline{d}(fx))) \\ \overline{c}(q,d') & = & *(\mathrm{All}(\underline{w}(\underline{q}d'),q) \wedge^\circ \mathcal{I}(\overline{d}d')) \end{array}$$

The equation $\mathcal{I}(\overline{d}d)$ holds if d is a decorated, well-founded map. The equation $\mathcal{I}(\overline{d}_1(f,(q,d')))$ holds if $(f,(q,d'))$ is a decorated well-founded map distinct from T.

The equation $\mathcal{I}(\overline{c}(q,d'))$ holds if (q,d') is a valid comment in a decorated, well-founded map. To be a valid comment, d' has to be a decorated, well-founded map. This is expressed by $\overline{d}d'$ in the definition of $\overline{c}$. Let G denote the contents of d'. Now $\underline{q}d'$ is a universal quantifier over G. Further, $\underline{w}(\underline{q}d')x$ holds if $x \in G^\circ$. To be a valid comment, q has to be a universal quantifier which quantifies over G°. This is expressed by $\mathrm{All}(\underline{w}(\underline{q}d'),q)$ in the definition of $\overline{c}$.

As mentioned, the equation $\mathcal{I}(\overline{d}_1(f,(q,d')))$ holds if $(f,(q,d'))$ is a decorated, well-founded map distinct from T. To be such a map, (q,d') has to be a valid comment as expressed by $\overline{c}(q,d')$. Further, fx must be a decorated, well-founded map for all $x \in G^\circ$ where G is the contents of d' as expressed by $\underline{w}(\underline{q}d')x \rightarrow \mathcal{I}(\overline{d}(fx))$.

A decorated, well-founded map d is either T or a decorated, well-founded map distinct from T. This could lead to the definition

$$\overline{d}d = *(!d = \mathsf{T} \wedge^{\circ} \neg d \to \overline{d}_1 d)$$

With this definition, however, one could imagine an infinitely deep d of form

$$d = (f, (q, d))$$

to be well-founded, contradicting the intuitive definition of well-foundedness. Ill-foundedness of this kind leads to Burali-Forti's paradox.

One could also imagine an infinitely deep d of form

$$d = (\hat{x}d., (q, d'))$$

to be well-founded. Such a decorated, ill-formed d would correspond to the ill-founded map $m = \hat{x}m.$ which represents the set $a = \{a\}$ in Aczels AFA set theory [1].

The definition of $\overline{d}$ uses $\overline{w}$ to avoid these two kinds of ill-foundedness. A minor modification of $\overline{d}$ would allow the second kind of of ill-foundedness and still avoid the first. This would lead to a definition of non-well-founded maps.

C.9 Formalisation of well-founded maps

A map m is well-founded if there is a map d such that $m = \underline{m}d$ and $\mathcal{I}(\overline{d}d)$. The following equivalent definition is more suited to formalisation, however: A map m is well-founded if there is a valid comment (q, d') such that $\underline{w}qm = \mathsf{T}$. Now define

$$\phi m = ?\exists^{\overline{c}c} c : \underline{w}(c\mathsf{T})m.$$

Intuitively, this ϕ is the ϕ of [6], but it is an open question whether or not all axioms of [6] are provable in the pure version.

D Summary of stable map theory

D.1 Syntax

$$\begin{array}{lcl} \mathsf{v} & ::= & x_1 \mid x_2 \mid x_3 \mid \cdots \\ \mathsf{t} & ::= & \mathsf{v} \mid \mathsf{tt} \mid \hat{\mathsf{v}}\mathsf{t}. \mid \mathsf{T} \mid \mathsf{P} \mid \varepsilon \\ \mathsf{e} & ::= & \mathsf{t} = \mathsf{t} \end{array}$$

D.2 Priority

The priority of operators is important in reading terms and equations correctly. As an example, according to the priority rules below,

$$\neg ab \rightarrow c \preceq d$$

means

$$(\neg(ab)) \rightarrow (c \preceq d)$$

According to the definition of $\preceq$ this means

$$(\neg(ab)) \rightarrow (c = c \downarrow d)$$

which, according to the definition of $\rightarrow$ is shorthand for the equation

$$((\neg(ab)) \tilde{\wedge} c) = ((\neg(ab)) \tilde{\wedge} (c \downarrow d))$$

The priority is as follows. Functional application fx is the tightest binding of all operators and appears at the top of the table below. Operators on the same line have the same priority.

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

D.3 Associativity

x, y is right associative so that (x, y, z) means $(x, (y, z))$. fx, $x \wedge y$, $x \tilde{\wedge} y$, $x \vee y$ and $x \tilde{\vee} y$ are left associative.

$x \sim y$, $x \in y$, $x \in\in y$, $x \sim_S^2 y$, $x \Rightarrow y$, $x \tilde{\Rightarrow} y$, and $x \Leftrightarrow y$ 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 $(x \Rightarrow y) \wedge (y \Rightarrow z)$.

D.4 Definitions used in axioms

$$
\begin{array}{ll}
\mathsf{F} & = \hat{x}\mathsf{T}. \\
x \left\{ \begin{array}{l} a \\ b \end{array} \right. & = \mathsf{P}abx \\
\mathsf{Y} & = \hat{f}\hat{x}f(xx).\hat{x}f(xx).. \\
\neg x & = x \left\{ \begin{array}{l} \mathsf{F} \\ \mathsf{T} \end{array} \right. \\
\wr x & = x \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array} \right. \\
!x & = x \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{T} \end{array} \right. \\
x \mathrel{\tilde{\wedge}} y & = x \left\{ \begin{array}{l} y \\ \mathsf{F} \end{array} \right. \\
x \mathrel{\tilde{\Rightarrow}} y & = x \left\{ \begin{array}{l} y \\ \mathsf{T} \end{array} \right. \\
x \wedge y & = x \left\{ \begin{array}{l} y \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array} \right. \\ y \left\{ \begin{array}{l} \mathsf{F} \\ \mathsf{F} \end{array} \right. \end{array} \right. \\
x \Rightarrow y & = x \left\{ \begin{array}{l} y \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array} \right. \\ y \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{T} \end{array} \right. \end{array} \right. \\
x \Leftrightarrow y & = (x \Rightarrow y) \wedge (y \Rightarrow x) \\
x \rightarrow y & \text{means } x \mathrel{\tilde{\wedge}} y = x \mathrel{\tilde{\wedge}} \mathsf{T} \\
\exists & = \hat{f}\wr f(\varepsilon f). \\
\forall & = \hat{f}\neg\forall\hat{x}\neg fx.. \\
f \sim g & = f \left\{ \begin{array}{l} g \left\{ \begin{array}{l} \mathsf{T} \\ \mathsf{F} \end{array} \right. \\ g \left\{ \begin{array}{l} \mathsf{F} \\ \forall\hat{x}fx \sim gx. \end{array} \right. \end{array} \right. \\
x \in y & = y \left\{ \begin{array}{l} \mathsf{F} \\ \exists\hat{z}xz \sim y. \end{array} \right. \\
x \in\in y & = \exists\hat{z}x \in z \wedge z \in y. \\
\forall x{\in\in}a\colon b. & = \hat{y}\forall\hat{x}x \in\in y \mathrel{\tilde{\Rightarrow}} b..a
\end{array}
$$

$$f \sim^2_S g \quad = f\begin{cases} g\begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases} \\ g\begin{cases} \mathsf{F} \\ \forall x{\in}{\in}S\colon fx \sim^2_S gx. \end{cases} \end{cases}$$

$$\ell \quad = \hat{f}f\begin{cases} \mathsf{T} \\ \forall \hat{x}\ell(fx). \wedge \exists\hat{S}\forall\hat{x}\forall\hat{y}x \sim^2_S y \Rightarrow fx \sim fy... \end{cases} .$$

$$x \downarrow y \quad = x\begin{cases} y\begin{cases} \mathsf{T} \\ \perp \end{cases} \\ y\begin{cases} \perp \\ \hat{z}xz \downarrow yz. \end{cases} \end{cases}$$

$x \preceq y$ means $x = x \downarrow y$

$$\downarrow g \quad = \forall\hat{x}gx.\begin{cases} \mathsf{T} \\ \forall\hat{x}\neg gx.\begin{cases} \hat{y}\downarrow\hat{x}gxy.. \\ \perp \end{cases} \end{cases}$$

D.5 Axioms

T	Transitivity	$a = b; a = c \vdash b = c.$
Sab	Substitutivity	$a = c; b = d \vdash ab = cd.$
S$\hat{x}$	Substitutivity	$a = b \vdash \hat{x}a. = \hat{x}b.\ .$
PT	Pairing	$\mathsf{P}ab\mathsf{T} = a.$
$\mathsf{P}\hat{x}$	Pairing	$\mathsf{P}ab\hat{x}c. = b$
$\mathsf{P}\perp$	Pairing $\perp$	$\mathsf{P}ab\perp = \perp.$
$\mathsf{T}a$	Application	$\mathsf{T}a = \mathsf{T}.$
$\hat{x}b$	Application	$\hat{x}a.b = \langle a \mid x{:=}b \rangle$
		if b is free for x in a.
$\perp a$	Application	$\perp a = \perp.$
R	Renaming	$\hat{x}\langle a \mid y{:=}x \rangle. = \hat{y}\langle a \mid x{:=}y \rangle.$
		if x is free for y in a and vice versa.
QND	Quartum Non Datur	$a\mathsf{T} = b\mathsf{T}; a\perp = b\perp; a\hat{y}xy. = b\hat{y}xy.$
		$\vdash ax = bx.$
E	Extensionality	$\wr fxy = \wr gxy; fxyz = fab; gxyz = gab$
		$\vdash fxy = gxy$
		if x, y and z are not free in f and g.
Q1	Quantification	$\ell a \wedge \forall b \rightarrow ba$
Q2	Quantification	$\varepsilon \hat{x}a. = \varepsilon \hat{x} \ell x \wedge a.$
Q3	Quantification	$\ell(\varepsilon \hat{x}a.) = \forall \hat{x}!a.$
Q4	Quantification	$!\forall \hat{x}a. = \forall \hat{x}!a.$
Y	Minimality	$fa \preceq a \vdash \mathsf{Y}f \preceq a.$
C	Stability	$a \preceq c; b \preceq c \vdash fa \downarrow fb = f(a \downarrow b).$
C'	Stability	$\ell x \rightarrow (a \preceq c) \vdash \downarrow \hat{x}fa. = f(\downarrow \hat{x}a.).$
		if x is not free in c and f.
U	Union set	$\forall \hat{V} \exists \hat{U} \forall \hat{x} x \in U \Leftrightarrow x \in\in V \ldots\ .$

In axioms, x, y, z, U and V denote arbitrary variables and a, b, c, d, f and g denote arbitrary terms.

Axiom C' is a generalisation of Axiom C and Axiom C follows from Axiom C'. The models in Appendix B are conjectured to satisfy both axioms. Only one of the axioms should be included in the theory. Appendix A merely uses the weaker of the two axioms.

Index

References

[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 and K. Grue. A κ-denotational semantics for Map Theory in ZFC+SI. To Appear.

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

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

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

[7] K. Grue. Mathematics and computation. Available from the author, contact grue@diku.dk, 1994.

[8] P. Henderson. *Functional Programming*. Prentice-Hall, 1980.

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

[10] A. Horowitz and S. Sahni. *Fundamentals of Data Structures in Pascal*. Computer Science Press, 1984.

[11] B. MacLennan. *Functional Programming*. Addison-Wesley, 1990.

[12] J. McCarthy. Recursive functions of symbolic expressions and their computation by machine. *Communications of the ACM*, pages 184–195, 1960.

[13] J. McCarthy. A basis for a mathematical theory of computation. In P. Brafford and D. Hirschberg, editors, *Computer Programming and Formal Systems*, pages 33–70. North-Holland, 1963.

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

[15] D. Scott. *λ-Calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, chapter Combinators and classes, pages 1–26. Springer-Verlag, 1975.

[16] D. Scott. *λ-Calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, chapter Some philosophical issues concerning theories of combinators, pages 346–366. Springer-Verlag, 1975.