An Axiomatic Theory for Partial Functions

Jan Kuper

University of Twente, Department of Computer Science
P.O.Box 217, 7500 AE Enschede, The Netherlands

e-mail: jankuper@cs.utwente.nl

Abstract

We describe an axiomatic theory for the concept of one-place, partial function, where function is taken in its extensional sense. The theory is rather general, i.e., concepts like natural numbers and sets are definable, and topics as non-strictness and self application can be handled. It contains a model of the (extensional) lambda calculus, and commonly applied mechanisms (like currying, inductive definability) are possible. Furthermore, the theory is equi-consistent with and equally powerful as ZF-Set Theory.

The theory (called Axiomatic Function Theory, AFT) is described in the language of classical first order predicate logic with equality and one non-logical predicate symbol for function application. By means of some notational conventions, we describe a method within this logic to handle undefinedness in a natural way.

1 Introduction

Recently, the interest in the concept of partial function has grown, especially within theoretical computer science and the theory of computing (cf. Moggi (1988) for references). An important reason for that is, that partial functions arise naturally when we are modeling computer programs. Parallel to this growing interest in the concept of partial function, interest in a logic of partial terms or a logic of (un)definedness has also increased.

In this paper we want to give the basic outline of an axiomatic theory about partial functions, which can be considered as a general and precise framework for reasoning about partial functions. We also sketch a possibility to handle undefinedness within classical first order logic, without being forced to change the logical rules.
The concept of function seems to be equally fundamental as the concept of set, e.g., notions like natural number, set, ordered $n$-tuple, can easily be defined, using only the concepts of partial function and function application. Furthermore, a naive way of defining functions and operating on them leads to the Russell Paradox. To avoid the Russell Paradox, we develop a Zermelo-Fraenkel-like system of axioms, which gives us an intuitively very reasonable axiomatization of the concept of partial function. We want the theory to be rather general, so we allow self application and related topics (though not for all functions), we indicate a natural way to handle non-strict functions, and we add a functional form of the Axiom of Choice.

Functions are taken in their extensional sense, i.e., they are completely determined by what they do. In other words, they are completely determined by the set of ordered pairs of their arguments and corresponding results. However, functions are not identified with these sets of ordered pairs, but are considered as objects in their own right. Within the theory presented here, the concept of function is the basic concept, and sets can be defined as specific functions. So, the theory gives us a possibility to reason formally about functions in a more direct way than by means of sets of ordered pairs. The penalty of course is, that we can only reason indirectly about sets.

We describe a universe $\mathbb{F}$, in which all objects are one-place functions. All functions have functions as arguments, and functions as results, i.e., all functions are partial functions from $\mathbb{F}$ to $\mathbb{F}$. So from the beginning, all functions are higher order functions. There is one basic ternary relation $\mathcal{A}$, called the relation of application. The elementary formula $\mathcal{A}(x, y, z)$ is interpreted as: the function $x$ applied to the argument $y$ yields the result $z$.

The theory which we present here is expressed in the language of first order predicate logic with equality, and $\mathcal{A}$ as the only non-logical symbol. There is no need for a “bottom element” to represent undefinedness. When a function $f$ is undefined for an argument $x$, this can simply be expressed as $\neg \exists y (\mathcal{A}(f, x, y))$. That is, (un)definedness is a binary relation between a function $f$ and a (possible) argument $x$. Treating (un)definedness as a property of the result of applying $f$ to $x$ gives us a problem in case this result does not exist. In such a case there is nothing to have a property, not even the property of being undefined (this is a problem of a formal language, not of intuition). That is to say, if we want to consider (un)definedness as a property of the result of applying $f$ to $x$, we are forced to introduce one or more elements (“bottom elements”) to represent this result when $f$ is not defined for $x$. We do not want to do that, so within the given language we define (un)definedness as a binary predicate. In fact, this convenient expressibility of undefinedness in a first order language is the reason to prefer
this ternary \textit{relation} of application $\mathfrak{A}$ over a binary \textit{operation} of application.

However, there is one major drawback of this ternary relation: there are no terms. Formulas must be constructed using only variables, $=$, and $\mathfrak{A}$, and the ordinary logical connectives and quantifiers. As a consequence, formulas are hard to read, especially, when we want to express some more complicated situations, like iterated applications which we would like to write as, e.g., $f(x(y))$ (application is understood to associate to the left). For easy readability of the ternary relation of application, we will use \textit{infix notation}. Thus $\mathfrak{A}(f, x, y)$ will be notated as $f x \simeq y$, or as $f \cdot x \simeq y$. The negation $\neg \mathfrak{A}(f, x, y)$ will be notated as $f x \not\simeq y$. In Section 3 we will elaborate this infix notation further.

In Section 2 we will mention some other theories on functions, and in Section 4 we describe the Russell Paradox in a functional setting. In Section 5 we formulate the axioms of the theory, and some concepts and consequences that a general theory of functions should include at the very least. A complete list of the axioms can be found at the end of Section 5. In Section 6 we prove the (relative) consistency of the theory by interpreting it in ZF-Set Theory without the Axiom of Foundation, but including Boffa’s Axiom of Universality and the Axiom of Choice, whereas in Section 7 we interpret this variant of set theory within the theory of functions. Together, Sections 6 and 7 prove that ZF-Set Theory and the theory presented here are equi-consistent and equally powerful.

2 Other Theories on Functions

There are different theories in which we can speak formally about functions. The most well known of these theories is set theory, in which a function is a set of ordered pairs answering the property of uniqueness. As we already remarked, this is an indirect way to handle functions, whereas in this paper we describe a direct way to handle them. There is a wealth of literature on set theory; we only mention \textsc{Fraenkel et al} (1973), \textsc{Van Dalen et al} (1978).

A second important theory, in which the concept of function takes a central place, is the (untyped) lambda calculus. However, the lambda calculus is not a theory about the general concept of function, but about \textit{expressions} to denote functions. Illustrative to this point is, that in the lambda calculus function application is identified with substitution. We show that the theory presented here contains a model for the untyped lambda calculus (Theorem 5.8.4). Some important references on lambda calculus are
Closely related to the untyped lambda calculus are several variants of typed lambda calculi, constructive type theory and generalized type systems. Here the same remark applies as made with respect to the untyped lambda calculus. We hope that the theory presented in this paper can be a general framework for the semantics of different typed lambda calculi. For references on these subjects, cf. Barendregt and Hemerik (1990), Barendregt (1997).

As a third theory about functions, we mention category theory. Recently, category theory has become more and more important in the theory of computing. However, the way category theory handles the concept of function is very abstract, and rather remote from our daily intuition about functions. We feel that our theory is more concrete in this sense. Standard references for category theory are MacLane (1971), Arbib and Manes (1975), Barr and Wells (1990).

There is a fourth theory about functions, which resembles our way of looking at functions rather strongly. This theory is presented by John von Neumann in two articles, in which he formulates an axiomatic theory for sets based on the concept of function (cf. Von Neumann (1925, 1928), Robinson (1937)). Von Neumann’s work has been very influential on the foundations of set theory, mainly because he introduced the concept of class, and because he gave a finite axiomatization of set theory. However, the fact that he took the concept of function as the basic concept, remained without almost any influence. Historically, his choice of taking the concept of function as the basic concept, was judged as “reasonable [...but ...] rather clumsy” (cf. Fraenkel et al (1973), page 135), and as not very “manageable” (cf. Van Dalen and Monna (1972), pages 46–47). Bernays and Gödel reformulated Von Neumann’s theory directly in terms of sets and classes (cf. Bernays (1937, 1941), Gödel (1940)).

As important reasons for this historical judgement we mention that Von Neumann himself intended his work as a foundation for sets, and so his foundation was an indirect one from the very outset. In the second place, his axiomatic system is rather complicated, containing approximately twenty axioms, some of which are very hard to read. Furthermore, Von Neumann’s work was interpreted from the point of view of set theory, and not from the point of view of computation theory. It is exactly this point of view from which the concept of (partial) function has become more and more important.
3 Pseudo Terms

In this section we will describe some fairly simple notational conventions, in order to get a more common notation for function application, such that formulas can be read in a natural way. If necessary, the resulting formulas can always be translated into the underlying traditional formulas, which contain only $A$ as non-logical symbol. These underlying formulas are rather clumsy and highly unreadable, especially when the situation expressed in a formula is a little bit complicated. The intention is, that when we are working within the theory about partial functions presented in Section 5, we can forget about these underlying formulas. However, in the present section we only describe some conventions and examples, but we do not give any proofs.

The contents of this section belong to the field of logic of undefinedness, logic of partial terms, etc. (for references, see e.g. Beeson (1985), Moggi (1988), Troelstra and Van Dalen (1988)). The approach we describe here resembles Feferman’s approach (cf. Feferman (1975)).

In Section 1 we agreed to write $A(f,x,y)$ as $fx \simeq y$, or as $f \cdot x \simeq y$.

This infix notation looks like the familiar notation for function application, but there are some important differences. “$\cdot$” (mostly not written) is not a binary operation, and “$\simeq$” is not a binary relation, but together they form a ternary relation. We will call “$\cdot$” a pseudo operation (of application), “$\simeq$” a pseudo relation (of equivalence), and “$fx$” (or “$f \cdot x$”) a pseudo term.

In order to handle the more complicated situations of iterated application (as in $fx(gy)$), we generalize the concept of pseudo terms inductively as follows:

- Variables $x, y, z, \ldots$ are pseudo terms,
- If $\sigma, \tau$ are pseudo terms, then $\sigma \cdot \tau$ is a pseudo term.

Variables are also terms in the more usual sense. To avoid misunderstandings, we allow for brackets in the standard way. Furthermore, we will assume that “$\cdot$” associates to the left.

Infix formulas, i.e., formulas expressed in infix notation, are simply defined as follows:

- When $\sigma, \tau$ are pseudo terms, then $\sigma \simeq \tau$ and $\sigma \not\simeq \tau$ are infix formulas,
- We may construct more complicated infix formulas by the usual logical connectives and quantifiers.
The only well-formed formula, that can be constructed by means of “=”, is the elementary formula \( x = y \), where \( x \) and \( y \) are variables. So, \( \sigma = \tau \) is not well-formed, when \( \sigma \) or \( \tau \) is not a variable.

We adopt the following notational conventions for infix formulas (\( x, y, z \) stand for variables; \( \sigma, \tau \) stand for pseudo terms; \( x, y \) do not occur in \( \sigma, \tau \)):

\[
\begin{align*}
x \simeq y & \text{ means } x = y, \quad (1) \\
\sigma \tau \simeq z & \text{ means } \exists x, y(\sigma \simeq x \land \tau \simeq y \land xy \simeq z), \quad (2) \\
\sigma \simeq \tau & \text{ means } \forall x(\sigma \simeq x \leftrightarrow \tau \simeq x), \quad (3) \\
\sigma \not\simeq \tau & \text{ means } \neg(\sigma \simeq \tau). \quad (4)
\end{align*}
\]

It is obvious, that by means of these conventions we can always translate an infix formula into a traditional first order formula, which contains “=“ and “\( \exists \)“, but which does not contain “\( \simeq \)“ or “\( \cdot \)“. We will call such a formula a normal form.

In order to avoid ambiguities in translating infix formulas, we agree that conventions 1 and 2 have a higher priority in the translation process than convention 3.

As can be seen from the conventions formulated above, infix formulas can be understood in a very natural way:

- \( \sigma \tau \simeq z \) is true, when both \( \sigma \) and \( \tau \) are defined, and the application of \( \sigma \) to \( \tau \) gives the result \( z \). Otherwise it is false.

- \( \sigma \simeq \tau \) is true, when both \( \sigma \) and \( \tau \) are undefined, and also, when both are defined and have the same value. Otherwise it is false.

We define existence of a pseudo term \( \sigma \), also called: \( \sigma \) is defined, as

\[
E^*(\sigma) \leftrightarrow \exists x(\sigma \simeq x),
\]

where \( x \) may not occur in \( \sigma \). If \( \sigma \) exists, then \( x \) is called the value of \( \sigma \) iff \( \sigma \simeq x \).

We mention without proof, that if \( \sigma \) exists, then its value is unique.

Pseudo terms were introduced to avoid a “bottom element” (as an object), representing undefinedness. However, we may consider “bottom” as a shorthand notation for a non-existing pseudo term. Let \( e \) denote the empty function (such a function exists, cf. Section 5.2). We agree to abbreviate \( e \cdot e \) as \( \bot \) (“bottom”), and for all \( \sigma \) we have

\[
\neg E^*(\sigma) \leftrightarrow \sigma \simeq \bot.
\]
It follows from the conventions above, that all functions are strict, i.e., when \( f \) is a function and \( \sigma \) is an undefined pseudo term, then \( f\sigma \) is also undefined. In general we have that \( \sigma \tau \simeq \perp \), whenever \( \sigma \simeq \perp \) or \( \tau \simeq \perp \).

Note, that existence is not a predicate, since according to the conventions described above the right hand side of its definition may well translate to different normal forms when it is applied to different pseudo terms. For example, \( E^*(fx) \) evaluates to \( \exists z(fx \simeq z) \), i.e., to

\[
\exists z(\mathfrak{A}(f,x,z)),
\]

whereas \( E^*(fxy) \) evaluates to \( \exists z(fxy \simeq z) \), which is (by convention 2) equivalent to \( \exists z,u(fx \simeq u \land uy \simeq z) \), i.e., to

\[
\exists z,u(\mathfrak{A}(f,x,u) \land \mathfrak{A}(u,y,z)).
\]

So, \( E^*(fx) \) and \( E^*(fxy) \) are totally different formulas, that is, we have a series of existence predicates, which we could denote as \( E_1, E_2 \), etc.. However, we prefer to denote them schematically by \( E^* \) and we call \( E^* \) a predicate scheme. We may write \( E \), when \( \sigma \) is a variable. Notice, that we have \( \forall x(E(x)) \).

All predicates \( P \) occurring in the remaining part of this paper, are defined using variables, but they generalize in the same way to predicate schemes \( P^* \) (though in practice we will often omit the “\(^*\)”). That is, we may “substitute” pseudo terms for variables. However, this is not ordinary substitution, since it may change a specific predicate into another predicate, though the resulting predicate keeps more or less the intuitive meaning of the original predicate. We will call it substitution\(^*\).

For example, in Section 5.1 we define the binary predicate \( D(f,x) \) as \( \exists y(fx \simeq y) \), i.e., \( D(f,x) \) means that \( f \) is defined for \( x \). When we substitute\(^*\) \( gu \) for \( x \), we get

\[
D^*(f,gu),
\]

and the defining formula becomes

\[
\exists y(f(gu) \simeq y).
\]

By means of convention 2 this translates to

\[
\exists y,v(gu \simeq v \land fv \simeq y),
\]

which is equivalent to

\[
\exists y,v(\mathfrak{A}(g,u,v) \land \mathfrak{A}(f,v,y)).
\]
However, this formula contains three free variables \((f, g\text{ and } u)\), so \(D^*(f, gu)\) is a ternary predicate, whereas \(D(f, x)\) is a binary predicate. The intuitive meaning of \(D^*(f, gu)\) remains in a sense the same: \(f\) is defined for \(gu\). In addition, the existence of \(gu\) is also required.

The same holds for formulas: a formula \(\varphi\) changes under substitution\(^*\) to a different formula \(\varphi^*\) (here too we will mostly omit the “\(^*\)”). In fact, this already is present with the basic predicate \(\mathfrak{A}\). We can look at \(\sigma\tau \simeq \upsilon\) as a substitution\(^*\)-instance of \(xy \simeq z\), i.e., of \(\mathfrak{A}(x, y, z)\). So, \(\sigma\tau \simeq \upsilon\) can be considered as infix notation for \(\mathfrak{A}^*(\sigma, \tau, \upsilon)\).

We have to be careful with substitution\(^*\) of non-existing pseudo terms for variables, since we can not say anything in advance about the truth or falsehood of \(\varphi^*(\sigma_0, \ldots, \sigma_{n-1})\), when one or more of \(\sigma_0, \ldots, \sigma_{n-1}\) do not exist. For example, \(D^*(f, \sigma)\) is false when \(\sigma \simeq \bot\), whereas \(\text{Inj}^*(\sigma)\) is true when \(\sigma \simeq \bot\) (\(\text{Inj}(f)\) means that \(f\) is injective, cf. Definition 5.2.3). It is a matter of choice, if we accept this or not. If we don’t, then in order to show that some pseudo term \(\sigma\) “really” is injective, we separately must prove that \(\sigma\) exists. In general, in \(\varphi^*(\sigma_0, \ldots, \sigma_{n-1})\), we have several choices with respect to the existence of the \(\sigma_i\)'s occurring in it. An example of this point is given after Definitions 5.3.2 and 5.3.3.

Once more we want to remark, that “•” and “\(\simeq\)” are not a real binary operation and relation respectively, they only behave as if they were (except for the case \(x \simeq y\), where “\(\simeq\)” is equivalent to “\(=\)”). That is, intuitively “•” can be understood as a partial operation of application, and “\(\simeq\)” as a relation of equivalence, which indeed is reflexive, transitive and symmetric (we do not prove that here).

From now on we will often omit the prefix “pseudo” and just speak about terms.

4 The Russell Paradox

At first sight, when starting the development of an axiomatic theory about functions, we would like it to be combinatorially complete, i.e., every function that is definable by logical means exists and may be used. In everyday practice of mathematics functions are defined by describing what they are supposed to do with their arguments. In other words, functions are understood to exist, when their behaviour is described.
In the framework of a general first order theory about functions combinatorial completeness thus means, that if we have a first order description of this behaviour, then there exists a function which shows this behaviour. From this point of view an important axiom is a “naive comprehension scheme”:

when \( \varphi(x, y) \) is a (first order) functional formula, i.e., a formula that satisfies the requirement of uniqueness:

\[
\varphi(x, y_1) \land \varphi(x, y_2) \rightarrow y_1 = y_2,
\]

then there exists a function \( f \), which “does as \( \varphi \) says”, i.e.,

\[
fx \simeq y \iff \varphi(x, y)
\]

(\( f \) may not be free in \( \varphi(x, y) \)).

However, such a naive theory cannot exist, since it leads to a functional form of the Russell Paradox: suppose \( a \) is a function such that \( aa \not\simeq a \) (according to the naive comprehension scheme, there surely exists such a function, e.g., the empty function). Now define the “Russell function” \( r \) as follows:

\[
rx \simeq y \leftrightarrow ((xx \simeq x \rightarrow y = a) \land (xx \not= x \rightarrow y = x)).
\]

In this expression, the defining formula of \( r \) indeed is a functional formula. It can easily be seen that

\[
\forall x (rx \simeq x \iff xx \not= x),
\]

i.e., \( x \) is a fixed point of \( r \), if and only if \( x \) is not a fixed point of itself. If we take \( r \) for \( x \), then we have the Russell Paradox for functions:

\[
rr \simeq r \iff rr \not= r.
\]

A solution to the Russell Paradox can be found in the “limitation of size” principle. The result of that is, that a function can only exist, if it is not too “powerful”, i.e., if it is not more powerful than an already existing function.

To express this restriction, suppose we want to prove the existence of a function \( g \) which “does as \( \varphi \) says”, where \( \varphi \) is some functional formula. Then we must restrict the domain of \( g \), such that it is not bigger than the domain of some already existing function \( f \). We can do that by formulating a second functional formula \( \psi(u, x) \), and then require that for every \( x \) in the...
domain of $g$ there is a $u$ in the domain of $f$, such that $\psi(u, x)$ holds. More formally, this can be expressed as

$$\forall f \exists g \forall x, y(gx \simeq y \leftrightarrow \varphi(x, y) \wedge \exists u(\exists v(fu \simeq v) \wedge \psi(u, x)))$$

In Section 5.1 Axiom 3 expresses an alternative and more simple scheme of restricted comprehension. As a consequence of this axiom, all functions are so to say “essentially partial” with respect to the universe $\mathcal{F}$, so within $\mathcal{F}$ there exists no universal identity function, there exist no universal projection functions, etc.

That does not mean, however, that we cannot speak in an intuitive way of such “higher functions” which are defined on the total universe $\mathcal{F}$. For example, in Section 5.3 we will define an intuitive “higher function” $\delta$ from $\mathcal{F}$ to $\mathcal{F}$, such that $\delta f$ is the domain of an arbitrary function $f$ (sets will be defined as specific functions). A second example is composition of functions, which is well-defined for every $f$ and $g$ in $\mathcal{F}$ (though its result may be the empty function). Such “higher functions”, which exist on an intuitive level, and not within $\mathcal{F}$, might be called operations. Operations may be partial with respect to $\mathcal{F}$, and only those operations that are “weak” enough, can exist within $\mathcal{F}$, or, more precisely, can be represented within $\mathcal{F}$. Notice that operations are analogous to classes in ZF Set Theory.

5 Axioms

In this Section the basic outline of the theory about partial functions is formulated. We introduce and describe the eight axioms of the theory, and give some consequences of them. The axioms are described in Sections 5.1, 5.2, 5.4, 5.8 and 5.9, and a complete list of them is given in Section 5.10. In Section 5.3 we define natural numbers and sets as specific one-place functions, and in Section 5.6 we do the same with ordered $n$-tuples and $n$-ary functions. In Section 5.5 we prove that functions may be defined inductively, and in Section 5.7 we prove that the operation of currying on $n$-ary functions is well-defined.

5.1 The Axioms of Uniqueness, Extensionality and Restricted Comprehension

The first two axioms are simple. They formulate uniqueness and extensionality of functions.

Axiom 1 (Uniqueness) $\forall f, x, y_1, y_2(fx \simeq y_1 \land fx \simeq y_2 \rightarrow y_1 = y_2)$.  \hspace{1cm} \Box
The Axiom of Uniqueness says, that a function \( f \) has at most one value for every argument.

**Axiom 2 (Extensionality)** \( \forall f,g (\forall x (fx = gx) \rightarrow f = g) \).

According to the Axiom of Extensionality two functions \( f \) and \( g \) are equal, if they do the same to every argument. Note, that if \( fx \) and \( gx \) both do not exist, then indeed \( f \) and \( g \) do the same to \( x \).

The next axiom describes a scheme of restricted comprehension, as announced in Section 4. Before it can be formulated, we need a few definitions.

**Definition 5.1.1** A function \( f \) is defined for an argument \( x \) (or \( x \) is in the domain of \( f \)), denoted as \( D(f,x) \), iff

\[ \exists y (fx \simeq y) \].

Note, that \( D(f,x) \leftrightarrow E^*(fx) \).

**Definition 5.1.2** \( x \) is a result of \( f \) (or \( x \) is in the range of \( f \)), denoted as \( R(f,x) \), iff

\[ \exists y (fy \simeq x) \].

**Definition 5.1.3** \( x \) belongs to the field of \( f \), denoted as \( Fld(f,x) \), iff

\[ D(f,x) \lor R(f,x) \].

The next axiom (restricted comprehension) actually is an axiom scheme, in which a first order formula \( \varphi \) appears. \( \varphi(x,y) \) is supposed to describe the “behaviour” of the function \( g \) (restricted to the field of \( f \)), whose existence is guaranteed by the Axiom of Restricted Comprehension, so \( \varphi(x,y) \) must be a functional formula, i.e., for all \( x,y \) we have

\[ \varphi(x,y) \land \varphi(x,y') \rightarrow y = y' \]

(cf. Section 4). Parameters are allowed in \( \varphi(x,y) \). In order to prevent circular definitions, \( g \) may not be free in \( \varphi \).

**Axiom 3 (Restricted Comprehension)**

\[ \forall f \exists g \forall x, y (gx \simeq y \leftrightarrow Fld(f,x) \land \varphi(x,y)) \].
Actually, this Axiom of Restricted Comprehension is a little stronger than the one described in Section 4, but given Axioms 4 and 5 their equivalence can easily be proved.

The function $g$ mentioned in Axiom 3 is uniquely determined, which follows immediately from the Axiom of Extensionality. It is denoted by a specific lambda term as

$$\lambda(x \mapsto y \mid \text{Fld}(f, x) \land \varphi(x, y)).$$

The general form of lambda terms is

$$\lambda(x \mapsto y \mid \psi),$$

where $\psi$ is a formula.

As usual, $x$ and $y$ are called bound variables. Variables that are not bound, are called free.

In the literature, lambda terms are usually written as $\lambda x.\sigma$, where $\sigma$ is a term. However, this form is a special case of the lambda terms we introduced here, and we have

$$\lambda x.\sigma \simeq \lambda(x \mapsto y \mid \sigma \simeq y),$$

where $y$ may not occur free in $\sigma$.

Lambda terms may be considered as pseudo terms (cf. Section 3), but we will not elaborate this possibility here. We just remark that

$$\lambda(x \mapsto y \mid \psi) \simeq \bot$$

when $\psi$ is not a functional formula, or when $\psi$ is too “strong”, i.e., for too many $x$’s there is a $y$ such that $\psi(x, y)$ holds.

As an immediate consequence of Axiom 3 we prove that function composition is allowed, i.e., we prove that for any two functions there exists a unique composition of $f$ and $g$.

**Theorem 5.1.4** \(\forall f, g \exists! h \forall x (hx \simeq g(fx)).\)

As usual, the composition $h$ of $f$ and $g$ is denoted as $g \circ f$.

**Proof.** By the Axiom of Restricted Comprehension we can define $h$ as:

$$hx \simeq y \iff \text{Fld}(f, x) \land g(fx) \simeq y.$$  

The uniqueness of $h$ follows from the Axiom of Extensionality. \qed

Clearly, the part $\text{Fld}(f, x)$ in the definition of $h$ in this proof is superfluous. We mentioned it here, to answer the precise pattern of the Axiom of Restricted Comprehension, but in the sequel we will often omit it.
5.2 The Axiom of Infinity

By the Axiom of Restricted Comprehension we can only prove the existence of a function \( g \), *given* the existence of a function \( f \), so we must make a start somewhere. In the next axiom this start is made with a function with an infinite domain. The axiom simply states that there exists a function \( f \) which can be applied iteratively to an argument \( a \), i.e., \( fa \) exists, \( f(fa) \) exists, etc. The only thing to be aware of is, that at least for one \( a \) in the domain of \( f \), all \( a, fa, f(fa), \ldots \) are distinct.

We start with some definitions.

**Definition 5.2.1** \( a \) is a *startpoint* of \( f \), denoted as \( \text{Start}(f,a) \), iff

\[ D(f,a) \land \neg R(f,a). \]

**Definition 5.2.2** \( a \) is an *endpoint* of \( f \), denoted as \( \text{End}(f,a) \), iff

\[ R(f,a) \land \neg D(f,a). \]

**Definition 5.2.3** A function \( f \) is *injective*, denoted as \( \text{Inj}(f) \), iff

\[ \forall x_1, x_2 (\exists y (fx_1 \simeq y \land fx_2 \simeq y) \rightarrow x_1 = x_2). \]

Note, that in Definition 5.2.3 it is not enough to state

\[ fx_1 \simeq fx_2 \rightarrow x_1 = x_2, \]

since \( fx_1 \simeq fx_2 \) holds whenever \( f \) is undefined for \( x_1 \) and \( x_2 \), and then an injective function would have at most one element outside its domain.

**Definition 5.2.4** A function \( f \) is a *successor function*, denoted as \( \text{SF}(f) \), iff

- \( \text{Inj}(f) \),
- \( \exists a(\text{Start}(f,a)) \),
- \( \neg \exists a(\text{End}(f,a)) \).

The Axiom of Infinity now simply says, that there exists a successor function.

**Axiom 4 (Infinity)** \( \exists f(\text{SF}(f)) \).

By means of the Axioms 3 and 4 we can prove the existence of some well-known functions.
Theorem 5.2.5 There exists an empty function: \( \exists f \forall x,y (fx \not\sim y) \).

The empty function is denoted as \( \lambda() \).

**Proof.** Let \( s \) be a successor function. Then by Axiom 3 there is a function \( f \) such that \( fx \sim y \iff Fld(s,x) \land x \neq x \). This \( f \) is the empty function. \( \Box \)

Theorem 5.2.6 Let \( a_0, a_1, \ldots, a_{n-1} \) be pairwise distinct functions. Then for all \( b_0, b_1, \ldots, b_{n-1} \) there exists a function \( f \) such that for all \( x,y \)
\[
fx \sim y \iff ((x = a_0 \land y = b_0) \lor \cdots \lor (x = a_{n-1} \land y = b_{n-1})).
\]
This function \( f \) is denoted as \( \lambda(a_0 \mapsto b_0, \ldots, a_{n-1} \mapsto b_{n-1}) \).

**Proof.** Let \( s \) be a successor function. Define \( E_0(u), E_1(u), \ldots, E_{n-1}(u) \), saying respectively that \( u \) is a startpoint of \( s \), \( u \) is a first “intermediate result” of \( s \), \ldots, \( u \) is a \( (n-1) \)th intermediate result of \( s \), as follows:
\[
E_0(u) \iff Start(s,u),
E_{i+1}(u) \iff \exists v(E_i(v) \land sv \sim u).
\]
By Axiom 3 there exists a function \( g \) such that
\[
\forall u, x(gu \sim x \iff D(s,u) \land ((E_0(u) \land x = a_0) \lor \cdots \lor (E_{n-1}(u) \land x = a_{n-1})).
\]
By Axiom 3 again, there exists a function \( f \) such that
\[
\forall x,y (fx \sim y \iff R(g,x) \land ((x = a_0 \land y = b_0) \lor \cdots \lor (x = a_{n-1} \land y = b_{n-1})).
\]
This function \( f \) is the required function. \( \Box \)

5.3 Natural Numbers and Sets

Using Theorems 5.2.5 and 5.2.6 we can define the natural numbers in many ways. We will define natural numbers in a Von Neumann-like style, namely, a natural number is defined as the identity function on all its predecessors.

**Definition 5.3.1 (Natural Numbers)**

\[
\begin{align*}
0 & \simeq \lambda() \\
1 & \simeq \lambda(0 \mapsto 0) \\
2 & \simeq \lambda(0 \mapsto 0, 1 \mapsto 1) \\
\vdots \\
n' & \simeq \lambda(0 \mapsto 0, 1 \mapsto 1, \ldots, n \mapsto n) \\
\vdots
\end{align*}
\]
In the definition of natural numbers, \( n' \) denotes the successor of \( n \).

Notice, that this definition only defines every natural number separately, by some informal inductive process. However, it does not give us a formal definition of a predicate \( \text{Nat} \), such that \( \text{Nat}(x) \) is \( \text{true} \) iff \( x \) is a natural number. We will give such a formal definition in Section 5.5.

As usual, \( 0 \) and \( 1 \) will model the truth values \( \text{false} \) and \( \text{true} \). Using that, \( \text{sets} \) are often modeled as some characteristic function, which yields \( \text{true} \) for objects inside the set, and \( \text{false} \) for objects outside the set. However, when we do that here, sets would become functions which are defined for the whole universe of functions, and we would have the Russell paradox again. So we define sets as functions which yield \( \text{true} \) for elements of the set, and which are undefined otherwise.

**Definition 5.3.2 (Set)** \( \text{Set}(a) \leftrightarrow \forall x(D(a, x) \rightarrow ax \simeq 1) \).

**Definition 5.3.3 (Membership)** \( x \in a \leftrightarrow \text{Set}(a) \land D(a, x) \).

Notice that \( x \in a \) is meaningful when \( a \) is not a set. In such a case \( x \in a \) is \( \text{false} \).

Notice also, that according to these definitions \( \text{Set}(\bot) \) is \( \text{true} \), but \( x \in \bot \) is \( \text{false} \) for every \( x \). We also have that \( \text{Set}(\lambda()) \) is \( \text{true} \) and \( x \in \lambda() \) is \( \text{false} \). Since \( \lambda() \not= \bot \) we are faced here with the situation of having two different interpretations for the empty set. This is not a real problem, but we may wish to choose (cf. Section 3) that only \( \text{existing} \) pseudo terms can be “real” sets, i.e.,

\[
\text{Set}^*(\sigma) \leftrightarrow \exists z(\sigma \simeq z \land \text{Set}(z))
\]

(where \( z \) may not occur in \( \sigma \)). Then still \( x \in \bot \) is \( \text{false} \) for every \( x \), but \( \bot \) is not a set any more, and thus only the empty function represents the empty set.

We will also use the ordinary set notation with curly braces, defined as specific lambda terms:

\[
\{a, b, c, \ldots\} \quad \text{means} \quad \lambda(a\to 1, b\to 1, c\to 1, \ldots),
\]

\[
\{x \mid \varphi\} \quad \text{means} \quad \lambda(x\to 1 \mid \varphi).
\]

Clearly, the concept of subset can also easily be defined.

**Definition 5.3.4 (Subset)** \( a \subseteq b \leftrightarrow \text{Set}(a) \land \text{Set}(b) \land \forall x(x \in a \rightarrow x \in b) \).
Up to now, we only could express that an object was in the domain of a function, by means of the binary predicate $D$. We could not yet talk formally about the domain itself. The same holds for the range and the field of a function. Using Definitions 5.3.2 and 5.3.3, we can prove that the domain, range and field of a function are sets.

**Theorem 5.3.5** For every function $f$ there exist sets $a$, $b$ and $c$ such that for all $x$

\[
\begin{align*}
x \in a & \iff D(f, x), \\
x \in b & \iff R(f, x), \\
x \in c & \iff Fld(f, x).
\end{align*}
\]

The sets $a$, $b$ and $c$ are called the domain (denoted as $\delta f$), range ($\rho f$) and field ($\Delta f$) of $f$ respectively. Just like lambda terms, these expressions may be considered as pseudo terms. More generally, when $\sigma$ is a pseudo term, then $\delta \sigma$, $\rho \sigma$, $\Delta \sigma$ are also pseudo terms.

Note, that when $\sigma \simeq \bot$ we have to choose between $\delta \bot \simeq \lambda()$ and $\delta \bot \simeq \bot$. For technical reasons, we take $\delta \bot \simeq \bot$ (see e.g. Section 5.7). The same remarks can be made with respect to $\rho$ and $\Delta$.

When in the sequel some operation $F$ (like $\delta$, $\rho$ or $\Delta$) is introduced, we will silently assume that $F$ can be applied to a pseudo term $\sigma$, such that $F \sigma$ is again a pseudo term. A more systematic treatment of pseudo terms falls outside the scope of this paper.

**Proof of Theorem 5.3.5.** By means of Axiom 3, the domain $a$ can be defined as

\[
\begin{align*}
ax & \simeq y \iff D(f, x) \land y \simeq 1.
\end{align*}
\]

Then clearly, $a$ is a set, and for all $x$

\[
x \in a \iff D(f, x).
\]

The proofs of the existence of the range and field of $f$ go the same. □

Note, that when $a$ is a set, $\delta a \simeq a$.

Another easy consequence of the definition of set is: if $\varphi$ is a functional formula, then for every set $a$ there exists a function $f$ which is partial (for a formal definition of partiality, cf. Definition 5.4.9) with respect to $a$, and which “does as $\varphi$ says” (restricted to $a$). The next corollary formulates this property.
Corollary 5.3.6 When $\varphi$ is a functional formula, then for every set $a$ there exists a function $f$ such that for all $x, y$

$$fx \simeq y \iff x \in a \land \varphi(x, y).$$

Proof. Immediate. □

When $a$ is a set, and $\varphi$ is such that for all $x, y$

$$\varphi(x, y) \rightarrow y \simeq 1,$$

then Corollary 5.3.6 gives us the Subset Axiom (Axiom of Separation, Aussonderung) from ZF-Set Theory. As an immediate consequence of that, the intersection $a \cap b$ and the difference $a \setminus b$ of two sets $a$ and $b$ also exist.

Clearly, often the formula $\varphi$ occurring in Corollary 5.3.6 is such, that $\delta f \simeq a$, e.g., for every set $a$ and every $u$ there exists a constant function with $a$ as its domain and $u$ as its constant value. Also, for every set $a$ there exists an identity function $i_a$ with $a$ as its domain. Notice, that by the Axiom of Extensionality for any two sets $a$ and $b$ we have $i_a = i_b \iff a = b$.

Informally, the function $f$ from Corollary 5.3.6 can be denoted as $\varphi \upharpoonright a$, i.e., the restriction of $\varphi$ to $a$. More formally, we can define the restriction of a function to a set.

Definition 5.3.7 A function $g$ is the restriction of a function $f$ to a set $a$, denoted as $f \upharpoonright a$, iff $gx \simeq y \iff x \in a \land fx \simeq y$. □

Theorem 5.3.8 $\forall f, a (\text{Set}(a) \rightarrow \exists g (g \simeq f \upharpoonright a))$

Proof. Immediate. □

Note, that for two sets $a$ and $b$, $a \upharpoonright b$ is the intersection of $a$ and $b$, i.e., $a \cap b \simeq a \upharpoonright b$, and thus in such a case $\upharpoonright$ is commutative.

5.4 The Axioms of Lower and Higher Order

In this section we will describe the Axioms of Lower and Higher Order, and discuss some consequences of them.

The Axiom of Lower Order asserts the existence of a function $g$, which can be applied to all arguments of arguments of a given function $f$. The Axiom of Higher Order guarantees the existence of a function $g$, that can be applied to all functions with the same domain and the same range as a given function $f$.
In both cases it suffices to state the existence of one such lower or higher function, since by virtue of the Axiom of Restricted Comprehension, we can always prove the existence of other lower and higher functions.

Using sets they can be easily formulated.

**Axiom 5 (Lower Order)** \( \forall f \exists g \forall x (x \in \delta g \leftrightarrow \exists u (u \in \delta f \land x \in \delta u)) \). □

**Axiom 6 (Higher Order)** \( \forall f \exists g \forall x (x \in \delta g \leftrightarrow (\delta x \simeq \delta f \land \rho x \simeq \rho f)) \). □

Note, that these two axioms do not specify the function \( g \) exactly. It is possible to add such an exact specification of \( g \) to them, e.g., by letting \( g \) be the identity function on the indicated domain. However, given the Axiom of Restricted Comprehension, this second form of the axioms is equivalent to the first form. We prefer the above given form of the axioms, because of the brevity of the formulation.

In the remaining part of this section, we will discuss some consequences of Axioms 5 and 6. The most important consequence of Axiom 5 is, that we can combine different functions into one single function, and the most important consequence of Axiom 6 is, that when \( a \) and \( b \) are two sets of functions, then there exist functions which are defined for all functions from \( a \) to \( b \). Furthermore, we will define some concepts, and prove the well-definedness of some common operations on sets and functions.

**Theorem 5.4.1** Let \( a \) be a set of sets. Then there exists a set \( b \), such that \( b \simeq \bigcup a \), where \( \bigcup a \simeq \{x \mid \exists y (x \in y \land y \in a)\} \).

As is well known, \( \bigcup a \) is called the sum set of \( a \). As usual, we denote \( \bigcup \{a, b\} \) as \( a \cup b \).

**Proof.** By Axiom 5 and Axiom 3. □

Suppose, we have a set \( a \) of functions, and we want to combine these functions into one single function \( g \), such that the domain of \( g \) is the union of the domains of the functions in \( a \). Then the problem arises, that different functions in \( a \) can be defined for some argument \( u \), but their results for \( u \) differ. In such a case we have to decide which of these functions will determine the value of \( g \) applied to \( u \). In order to make that decision, we introduce an election function \( e \), which elects for every \( u \) a specific \( f \in a \). Then \( gu \simeq euu \).

We have the following definition and theorem.
Definition 5.4.2 Let $a$ be a set of functions. A function $e$ is an election function for $a$, iff

- $\delta e \simeq \bigcup \{\delta f \mid f \in a\}$,
- $\rho e \subseteq a$.

\[\square\]

Theorem 5.4.3 For every set $a$ and election function $e$ for $a$ there exists a combination $g$ of all $f \in a$, such that $\forall u (gu \simeq euu)$.

Proof. By Axiom 3. \[\square\]

Note, that the domain of $g$ need not be equal to the union of the domains of all $f \in a$, since $e$ can choose a “wrong” function $f'$ for a certain $u$, i.e., $u \notin \delta f'$.

As an easy corollary of this theorem we have a special case of it.

Definition 5.4.4 A function $h$ is the result of $f$ updated with $g$, denoted as $f[g]$, iff

- $x \in \delta g \rightarrow hx \simeq gx$,
- $x \notin \delta g \rightarrow hx \simeq fx$.

\[\square\]

When $g$ is given in the form $\lambda (x \mapsto y \mid \varphi)$, we may formulate $f[g]$ without the $\lambda$ and without the parentheses, thus as $f[x \mapsto y \mid \varphi]$. In the same way we have $f[\sigma_0 \mapsto \tau_0, \ldots, \sigma_{n-1} \mapsto \tau_{n-1}]$, with the obvious interpretation.

Note, that the successor $n'$ of a natural number $n$ is $n[n \mapsto n]$. Note also, that if $a$ and $b$ are two sets, then $a[b] \simeq a \cup b$.

Corollary 5.4.5 $\forall f, g \exists h (h \simeq f[g] )$.

Proof. Define the election function $e$ for the set $\{f, g\}$ as follows:

- $ex \simeq g \iff x \in \delta g$,
- $ex \simeq f \iff x \in \delta f \land x \notin \delta g$,

and then apply Theorem 5.4.3. \[\square\]

Now we come to an important consequence of Axiom 6: if $a$ and $b$ are two sets of functions, then there exist functions which are defined for all functions from $a$ to $b$. By Corollary 5.3.6 it is enough to show that the set of all functions from $a$ to $b$ exists, and that is what we will do in the next few pages. Along the way we will define some concepts, and prove the well-definedness of some common operations on sets and functions.
Theorem 5.4.6  Let \( a \) be a set. Then there exists a set \( b \) such that \( b \simeq \mathcal{P}a \), where \( \mathcal{P}a \simeq \{ x \mid x \subseteq a \} \).

As usual, \( \mathcal{P}a \) is called the power set of \( a \).

Proof. Define for all \( b \subseteq a \) the function \( b' \) such that
\[
\begin{align*}
b'x &\simeq 1 \iff x \in b, \\
b'x &\simeq 0 \iff x \in a \land x \notin b, \\
b'x &\simeq \bot \iff x \notin a \land x \notin b.
\end{align*}
\]
Then for all \( b \subseteq a \), with \( b \neq \emptyset \) and \( b \neq a \), we have \( \delta b' \simeq a \) and \( \rho b' \simeq \{0, 1\} \).

By Axiom 6 there exists a function \( f \) such that
\[
\delta f \simeq \{b' \mid b \subseteq a \land b \neq \emptyset \land b \neq a\}.
\]
By Axiom 3 we can further specify \( f \) such that \( fb' \simeq b \).

If there exists no subset \( b \) of \( a \) such that \( b \neq \emptyset \) and \( b \neq a \), then \( \delta f \simeq \emptyset \), so \( f \simeq \lambda() \).

Now
\[
\mathcal{P}a \simeq \rho f \cup \{\emptyset, a\},
\]
and by Theorem 5.4.1 this set exists. \( \Box \)

A generalization of the notion of subset is the concept of subfunction, which we will also need further on.

Definition 5.4.7 (Subfunction) A function \( g \) is a subfunction of a function \( f \), denoted as \( g \subseteq f \), iff
\[
\forall x, y (gx \simeq y \rightarrow fx \simeq y).
\]
Note, that when \( a \) and \( b \) are sets, \( a \subseteq b \leftrightarrow a \subseteq b \).

Theorem 5.4.6 can also be generalized. It appears, that this generalization is an easy consequence of Theorem 5.4.6 itself.

Corollary 5.4.8 For every function \( f \) the set of all subfunctions of \( f \) exists.

Proof. A subfunction of \( f \) is uniquely determined by its domain. By Theorem 5.4.6 the power set of \( \delta f \) exists, so by Axiom 3 the set of all subfunctions of \( f \) also exists. \( \Box \)

In general totality and partiality of functions are relative notions, that is, a function \( f \) is total or partial with respect to a given set \( a \). The definition of these notions is obvious.
Definition 5.4.9

\[
\begin{align*}
\text{Total}(f,a) & \iff \delta f \simeq a, \\
\text{Partial}(f,a) & \iff \delta f \subseteq a,
\end{align*}
\]

According to this definition, totality is a special case of partiality. Instead of saying that \( f \) is partial with respect to \( a \), we may also say that \( a \) is the source set of \( f \), denoted as \( \text{Source}(f,a) \).

Likewise, we can define that a function is surjective with respect to a given set, and that a set is the target set of a function.

Definition 5.4.10

\[
\begin{align*}
\text{Surj}(f,a) & \iff \rho f \simeq a, \\
\text{Target}(f,a) & \iff \rho f \subseteq a.
\end{align*}
\]

Clearly, every function is total with respect to its domain, and surjective with respect to its range. Functions with source set \( a \) and target set \( b \) are called functions from \( a \) to \( b \).

The next theorem, preceded by two lemmas, asserts that the set of all functions from \( a \) to \( b \) exists.

Lemma 5.4.11 For any two sets \( a \) and \( b \) there exists a set \( p \) of all total and surjective functions from \( a \) to \( b \), i.e., for all \( f \)

\[
f \in p \iff \text{Total}(f,a) \land \text{Surj}(f,b).
\]

The set \( p \) is denoted as \( a \Rightarrow b \).

Proof. If there does not exist a function that is total with respect to \( a \) and surjective with respect to \( b \), we have that \( p \simeq \emptyset \). Otherwise, we may apply Axiom 6 and Theorem 5.3.5. \( \Box \)

Note, that when \( a \simeq \emptyset \) and \( b \simeq \emptyset \), then \( a \Rightarrow b \simeq \{\lambda()\} \), and when \( a \not\simeq \emptyset \) and \( b \simeq \emptyset \), then \( a \Rightarrow b \simeq \emptyset \). Also, when \( b \) is "bigger" than \( a \), i.e., when there is no total and injective function from \( b \) to \( a \), then \( a \Rightarrow b \simeq \emptyset \).

Lemma 5.4.12 For every two sets \( a \) and \( b \) there exists a set \( q \) of all total functions from \( a \) to \( b \), i.e., for all \( f \)

\[
f \in q \iff \text{Total}(f,a) \land \text{Target}(f,b).
\]
The set \( q \) is denoted as \( a \rightarrow b \).

**Proof.** By Lemma 5.4.11 for every subset \( u \) of \( b \) the set \( a \Rightarrow u \) exists. Define the function \( g \) with source set \( \mathcal{P}b \), such that \( gu \simeq a \Rightarrow u \). Then \( \bigcup(\rho q) \) is the required set \( q \). □

Note, that when \( a \simeq \emptyset \), then \( a \rightarrow b \simeq \{\lambda()\} \), and when \( a \not\simeq \emptyset \) and \( b \simeq \emptyset \), then \( a \rightarrow b \simeq \emptyset \).

**Theorem 5.4.13** For every two sets \( a \) and \( b \) there exists a set \( r \) of all functions from \( a \) to \( b \), i.e., for all \( f \)

\[
f \in r \leftrightarrow \text{Source}(f,a) \land \text{Target}(f,b).
\]

The set \( r \) is denoted as \( a \Rightarrow b \).

**Proof.** This proof is a variant of the proof of Lemma 5.4.12. By that lemma for every \( u \subseteq a \) the set \( u \rightarrow b \) exists. Define the function \( h \) with source set \( \mathcal{P}a \) such that \( hu \simeq u \rightarrow b \). Then \( \bigcup(\rho h) \) is the required set \( r \). □

Note, that for no \( a \) and \( b \), \( a \rightarrow b \simeq \emptyset \).

### 5.5 Induction

In this section we will show that, within the theory presented here, functions may be defined inductively.

Let \( \varphi(x,y) \) be a functional formula (cf. Section 4) in which parameters may occur. Furthermore, the functionality (in a broader sense) of \( \varphi(x,y) \) may not be impaired when an arbitrary term \( \tau \) is substituted\(^\dagger \) (cf. Section 3) for \( x \) in \( \varphi \). That is to say, either \( \varphi^*(\tau,\bot) \) holds, or there exists a unique \( u \) such that \( \varphi^*(\tau,u) \) holds. Such a formula may be written using infix notation by introducing a symbol \( F \) and writing \( Fx \simeq y \) for \( \varphi(x,y) \)\(^2\).

We will restrict ourselves to the following form of inductive definitions: let \( s \) be some successor function (not every successor function will be allowed) with startpoint \( a \), and let \( \sigma \) be a term. Then we may define a function \( f \), with source set \( \delta s \), as follows:

- \( fa \simeq \sigma \),
- \( f(sx) \simeq F(fx) \).

We start with a definition.

\(^\dagger\) Actually, if this infix notation is used in a more tricky way, then the requirements for \( \varphi \) may be weakened a little. However, we do not employ this technique here.
Definition 5.5.1 (Minimal Successor Function) A function \( m \) is called a **minimal successor function**, denoted as \( MSF(m) \), iff

\[
\begin{align*}
&SF(m), \\
&\exists a(\text{Start}(m,a)), \\
&\forall f(SF(f) \land f \subseteq m \land \forall a(\text{Start}(m,a) \rightarrow \text{Start}(f,a)) \rightarrow m \subseteq f).
\end{align*}
\]

\[ \square \]

Theorem 5.5.2 There exists a minimal successor function.

**Proof.** Let \( s \) be a successor function with startpoint \( a \). Then by Corollaries 5.3.6 and 5.4.8 the set

\[ t \simeq \{ s' \mid SF(s') \land s' \subseteq s \land \text{Start}(s',a) \} \]

exists. Also, \( t \) is non-empty, since \( s \in t \). Now define the function \( m \), such that for all \( x, y \)

\[ mx \simeq y \leftrightarrow \forall s' \in t(s'x \simeq y). \]

Then \( m \) is a successor function with startpoint \( a \) (immediate).

Furthermore, \( m \) has at most one startpoint. For suppose \( a' \) is a second startpoint of \( m \), then by Axiom 3 we can define a successor function \( m' \subseteq m \), such that \( a \) and \( ma \) are startpoints of \( m' \), i.e., \( a' \notin \delta m \). But \( m' \in t \), and so

\[ a' \notin \delta m, \text{ i.e., } a' \text{ is not a startpoint of } m. \]

Finally, for every successor function \( f \) with startpoint \( a \), such that \( f \subseteq m \), we have that \( f \subseteq s \), so \( f \in t \). Thus \( m \subseteq f \).

This proves, that \( m \) is a minimal successor function. \[ \square \]

The importance of the concept of minimal successor function lies in the fact that we may apply mathematical induction on (the domain of) a minimal successor function. More precisely, we have the following theorem.

Theorem 5.5.3 (Mathematical Induction) Let \( \varphi(x) \) be a formula, and \( m \) a minimal successor function with startpoint \( a \). Then

\[ \varphi(a) \land \forall x \in \delta m(\varphi(x) \rightarrow \varphi(mx)) \rightarrow \forall x \in \delta m(\varphi(x)). \]

**Remark.** In accordance with the conventions in Section 3 we write \( \varphi(mx) \) instead of \( \varphi^*(mx) \).

**Proof.** Suppose \( \varphi(a) \) and \( \forall x \in \delta m(\varphi(x) \rightarrow \varphi(mx)) \). Define a function \( f \), such that for all \( x, y \)

\[ fx \simeq y \leftrightarrow mx \simeq y \land \varphi(x), \]

\[ \square \]
then $f \subseteq m$, and $\forall x \in \delta f(\varphi(x))$. Furthermore, $f$ is injective (immediate), $a$ is a startpoint of $f$ (immediate), and $f$ has no endpoint (suppose $y \in \rho f$, i.e., there is an $x$ such that $fx \simeq y$. Then $mx \simeq y$ and $\varphi(x)$. But then $\varphi(y)$, i.e., $y \in \delta f$, so $f$ is a successor function. But $m$ is a minimal successor function, so $f = m$, i.e., $\forall x \in \delta m(\varphi(x))$. □

Notice, that for a successor function $s$ we have that $\delta s \simeq \Delta s$, so Theorem 5.5.3 also holds for the field of a minimal successor function.

Clearly, every minimal successor function $m$ with startpoint $a$ gives us a model for the axioms of Peano Arithmetic, where $a$ plays the role of 0, and $\Delta m$ models the set of all natural numbers.

**Theorem 5.5.4 (Peano)** Let $m$ be a minimal successor function with startpoint $a$. Then

- $a \in \Delta m$,
- $\forall x(x \in \Delta m \rightarrow mx \in \Delta m)$,
- $\forall x(mx \neq a)$,
- $\forall x, y(mx \simeq my \rightarrow x = y)$,
- $\varphi(a) \land \forall x \in \Delta m(\varphi(x) \rightarrow \varphi(mx)) \rightarrow \forall x \in \Delta m(\varphi(x))$.

**Proof.** Immediate. □

We continue with a definition.

**Definition 5.5.5 (Restricted Successor Function)** A function $r$ is a restricted successor function, iff there exists a successor function $s$ such that

$r \subseteq s \land \exists a(\text{Start}(r, a)) \land \exists b(\text{End}(r, b))$. □

**Theorem 5.5.6** Let $m$ be a minimal successor function with startpoint $a$. Then for every $x \in \rho m$ there exists a unique restricted successor function $m_x \subseteq m$ with $a$ as its only startpoint and $x$ as its only endpoint.

Such an $m_x$ is called a minimal restricted successor function. As a degenerate case, take $m_a \simeq \lambda()$.

**Proof.** Define the function $m'$ such that for all $y, z$

$$m'y \simeq z \leftrightarrow my \simeq z \land y \neq a,$$

so $m'$ and $m$ are equal, except for the startpoint of $m$. Clearly, $m'$ is a minimal successor function with startpoint $a' \simeq ma$, and $\delta m' \simeq \rho m$. thus we may proceed by induction on $\delta m'$.
- $m_{a'} \simeq \lambda(a \mapsto a')$ is a restricted successor function with $a$ as its only startpoint and $a'$ as its only endpoint such that $m_{a'} \not\subseteq m$.

- Now suppose $m_x$ is a restricted successor function with $a$ as its only startpoint and $x$ as its only endpoint, such that $m_x \not\subseteq m$. Define
  \[ m_{mx} \simeq m_x[x \mapsto mx], \]
  then clearly $m_{mx}$ is a restricted successor function with $a$ as its only startpoint and $mx$ as its only endpoint, and $m_{mx} \not\subseteq m$.

With respect to the uniqueness of $m_x$, suppose there is a second minimal restricted successor function $\tilde{m}_x$ with $a$ as its only startpoint and $x$ as its only endpoint such that $\tilde{m}_x \neq m_x$ and $\tilde{m}_x \not\subseteq m$. Define the set $d$ as
  \[ d \simeq (\Delta m_x \cup \Delta \tilde{m}_x) \setminus (\Delta m_x \cap \Delta \tilde{m}_x), \]
  then clearly $d \not\subseteq \emptyset$, and $a, x \not\in d$.

Next, define the function $\tilde{m}$ as
  \[ \tilde{m} \simeq m \restriction (\Delta m \setminus d). \]

We will show that $\tilde{m}$ is a successor function with startpoint $a$, $\tilde{m} \not\subseteq m$, and $\tilde{m} \not\subseteq m$, thus contradicting the minimality of $m$.

It is immediately clear that $\tilde{m} \not\subseteq m$, $\tilde{a} \neq m$, a is a startpoint of $\tilde{m}$, and $\tilde{m}$ is injective. So all we have to prove is that $\tilde{m}$ has no endpoint.

Therefore, suppose $z \in \rho \tilde{m}$, and $\tilde{m}y \simeq z$, thus $y \in \Delta m \setminus d$. Then $y \not\in d$, i.e., either
  \[ y \in \Delta m_x \wedge y \in \Delta \tilde{m}_x, \]
  or
  \[ y \not\in \Delta m_x \wedge y \not\in \Delta \tilde{m}_x. \]

In case (1), if $y = x$ then $\tilde{m}y \notin \Delta m_x$ and $\tilde{m}y \notin \Delta \tilde{m}_x$, thus $\tilde{m}y \not\in d$, i.e., $\tilde{m}y \in \delta \tilde{m}$.

If $y \neq x$ then $\tilde{m}y \in \Delta m_x$ and $\tilde{m}y \in \Delta \tilde{m}_x$, thus here too we have $\tilde{m}y \not\in d$.

In case (2) it follows that $\tilde{m}y \notin \Delta m_x$ and $\tilde{m}y \notin \Delta \tilde{m}_x$, since otherwise $\tilde{m}y$ would be a startpoint of $m_x$ or $\tilde{m}_x$, and $\tilde{m}y \neq a$. So again we have $\tilde{m}y \notin d$, i.e., $\tilde{m}y \in \delta \tilde{m}$.

Summarizing, if $z \in \rho \tilde{m}$ then $z \simeq \tilde{m}y \in \delta \tilde{m}$, and thus $\tilde{m}$ is a successor function. \hfill \square

Theorem 5.5.6 justifies the following definition.

25
Definition 5.5.7 If $m$ is a minimal successor function, then

$$
\begin{align*}
  x <_m y & \iff x \in \delta m_y, \\
  x \preceq_m y & \iff x \in \Delta m_y \lor \text{Start}(m, x).
\end{align*}
$$

In this definition, the condition $\text{Start}(m, x)$ is necessary for the case that $x$ and $y$ are both equal to the startpoint of $m$. Then we want $x \preceq_m y$, but $\Delta m_y = \emptyset$.

Now we come to the main theorem of this section. In this theorem $\sigma$ is a term and $\varphi$ a formula. $\varphi$ answers the requirements of functionality (in a broader sense) mentioned at the beginning of this section, and $\varphi(x, y)$ is assumed to be written as $F x \simeq y$.

**Theorem 5.5.8 (Inductive Definability)** Let $m$ be a minimal successor function with startpoint $a$. Then there exists a unique function $f$, such that for all $x \in \delta m$

- $f a \simeq \sigma$,
- $f(mx) \simeq F(f x)$.

**Proof.** Let $m_x$ be the minimal restricted successor function with startpoint $a$ and endpoint $x$, such that $m_x \subseteq m$. Define for every $x \in \delta m$ the function $f_x$ with $\delta f_x \simeq \{a\}$ if $x = a$ and $\delta f_x \simeq \Delta m_x$ otherwise, such that

$$
\begin{align*}
  f_x a & \simeq \sigma, \\
  \forall u <_m x(f_x(u) \simeq F(f_x u)).
\end{align*}
$$

By induction it is easy to prove that for every $x \in \delta m$ there exists a unique $f_x$ such that (1) and (2) hold:

- for the basic case, define $f_a \simeq \lambda(a \mapsto \sigma)$. Since $\delta m_a \simeq \emptyset$, $f_a$ trivially fulfills (1) and (2), and $f_a$ is unique.
- for the induction step, define $f_{mx} \simeq f_x[\lambda mx \mapsto F(f_x x)]$,

then $f_{mx}$ fulfills (1) and (2), and $f_{mx}$ is unique (straightforward).
Now define the function $f$ such that for all $x \in \delta m$

$$fx \simeq f_xx,$$

then $f$ exists and is unique, and we have

$$fa \simeq faa \simeq \sigma,$$

and

$$f(mx) \simeq f_mx(mx) \simeq F(f_xx) \simeq F(fx),$$

which was to be proved. \hfill \Box

An application of Theorem 5.5.8 is, that for a minimal successor function there exists a counting function $c$. We have the following definition and corollary.

**Definition 5.5.9** Let $m$ be a minimal successor function with startpoint $a$. A function $c$ is a counting function of $m$, denoted as $\text{CF}(c,m)$, iff

- $ca \simeq 0$,
- $\forall x \in \Delta m(c(mx) \simeq (cx)[cx\rightarrow cx]), \hfill \Box$

**Corollary 5.5.10** For every minimal successor function there exists a unique counting function $c$.

**Proof.** By Theorem 5.5.8. \hfill \Box

It is clear that $cx$ is a natural number for all $x \in \Delta m$ (cf. Definition 5.3.1), and $(cx)[cx\rightarrow cx]$ denotes the successor of $cx$. So $c$ counts how many applications of $m$, starting at the startpoint of $m$, are needed to reach $x$.

As a consequence, the natural numbers are the elements of the range of such a counting function $c$. So now we can formally define a predicate which indicates that some object is a natural number (cf. Section 5.3).

**Definition 5.5.11** An object $x$ is a natural number, denoted as $\text{Nat}(x)$, iff

$$\exists m, c(\text{MSF}(m) \land \text{CF}(c,m) \land x \in \rho c). \hfill \Box$$
Theorem 5.5.12  The set \( \mathbb{N} \) of the natural numbers and the ordinary successor function \( s_\mathbb{N} \) for the natural numbers both exist and are unique.

**Proof.** Simply take \( \rho_c \) for \( \mathbb{N} \), and define \( s_\mathbb{N} \) by
\[
s_\mathbb{N} n \simeq n' \iff n \in \mathbb{N} \land n' \simeq n[n \mapsto n].
\]
Without proof we mention that \( s_\mathbb{N} \) is a minimal successor function so we may apply mathematical induction on the natural numbers. Furthermore, the natural numbers, together with their successor function, form a model of Peano Arithmetic. We will take this model as the canonical one.

5.6 \( n \text{-ary functions} \)

Up to now we restricted ourselves to arbitrary one-place functions. However, it is easy to model many-place functions as specific one-place functions. To reach this goal, we will first define \( n \)-tuples.

**Definition 5.6.1 (Ordered \( n \)-tuple)** an ordered \( n \)-tuple \( \langle x_0, x_1, \ldots, x_{n-1} \rangle \) is the function
\[
\lambda(0 \mapsto x_0, 1 \mapsto x_1, \ldots, (n-1) \mapsto x_{n-1}).
\]

Note, that that the empty tuple \( () \simeq \lambda() \). Note also, that a natural number \( n \) is the ordered \( n \)-tuple of all its predecessors in increasing order, and that \( \delta n \) (and also \( \rho n \)) is the set of all predecessors of \( n \).

**Theorem 5.6.2** For all sets \( a_0, a_1, \ldots, a_{n-1} \) the set \( c \) of all ordered \( n \)-tuples \( \langle x_0, x_1, \ldots, x_{n-1} \rangle \), with \( x_i \in a_i \ (i = 0, 1, \ldots, n-1) \), exists.

As usual, the set \( c \) is denoted as \( a_0 \times a_1 \times \cdots \times a_{n-1} \), and called the cartesian product of \( a_0, a_1, \ldots, a_{n-1} \). Note, that there is a difference between e.g. \( (a_0 \times a_1) \times a_2 \) and \( a_0 \times (a_1 \times a_2) \), which are understood to contain elements of the form \( \langle \langle x_0, x_1 \rangle, x_2 \rangle \) and \( \langle x_0, x_1, x_2 \rangle \) respectively.

**Proof.** By Lemma 5.4.12 there exists the set
\[
\{0, 1, \ldots, n-1\} \to \bigcup \{a_0, a_1, \ldots, a_{n-1}\},
\]
containing all \( n \)-tuples whose elements are in \( a_0 \) or in \( a_1 \) or in \( \cdots \) or in \( a_{n-1} \). By Corollary 5.3.6 we can separate from this set the subset containing only those ordered \( n \)-tuples \( f \), such that
\[
f0 \in a_0, f1 \in a_1, \ldots, f(n-1) \in a_{n-1}.
\]
This set is the required set $c$. □

**Definition 5.6.3 (n-ary Function)** A function $f$ is an $n$-ary function, iff all arguments of $f$ are of the form $\langle x_0, x_1, \ldots, x_{n-1} \rangle$.

As usual, we will speak of *nullary*, *unary*, *binary*, etc, when $n = 0, 1, 2$ respectively. Note, that the empty function is nullary, unary, binary, etc.

According to this definition, a nullary function is a function whose arguments are all of the form $\langle \rangle$. However, there is only one empty tuple, so a nullary function is a function with the empty tuple as its only argument.

All arguments of unary functions are of the form $\langle x \rangle$, so unary functions are different from our basic concept of one-place functions. All functions in the universe are one-place, not all functions are unary. On the other hand, for every one-place function $f$ there exists a unary function $f'$, such that for all $x$

$$f'(x) \simeq fx,$$

as can be proved easily. Also, the collection of all unary functions, together with a suitably adapted relation of application, forms a model of the theory presented here.

By Definition 5.6.3 an $n$-ary function $f$ is a one-place function whose arguments are ordered $n$-tuples, and so the *elements* of such arguments of $f$ are themselves not arguments of $f$ (at least, in general they are not); they are *values* of the arguments of $f$. We will call an element $x_i$ of an argument of an $n$-ary function $f$ a *quasi argument* of $f$, and say that $f$ is quasi applicable to $x_i$, or $x_i$ belongs to the quasi domain of $f$.

**Definition 5.6.4** $x$ is a quasi argument of an $n$-ary function $f$, denoted as $QD(f, x)$, iff there exist $x_0, x_1, \ldots, x_{n-1}$ such that

$$\langle x_0, x_1, \ldots, x_{n-1} \rangle \in \delta f \land (x = x_0 \lor x = x_1 \lor \cdots \lor x = x_{n-1}).$$

In the same way as we defined the field of a function $f$, we will define the quasi field of an $n$-ary function $f$.

**Definition 5.6.5** $x$ belongs to the quasi field of an $n$-ary function $f$, denoted as $QFld(f, x)$, iff

$$QD(f, x) \lor R(f, x).$$

29
Theorem 5.6.6  For every \(n\)-ary function \(f\) there exist sets \(a\) and \(b\), called the quasi domain and quasi field of \(f\), such that for all \(x\)

\[
\begin{align*}
  x \in a & \iff QD(f,x), \\
  x \in b & \iff QFld(f,x).
\end{align*}
\]

The quasi domain of an \(n\)-ary function \(f\) is denoted as \(\delta_q f\), the quasi field as \(\Delta_q f\).

Proof. The set \(\bigcup \{ \rho x \mid x \in \delta f \} \) is the required quasi domain. The quasi field is simply the union of the quasi domain of \(f\) and the range of \(f\). \(\square\)

In the remaining part of this section, we will indicate how to simulate non-strict functions (cf. Section 4). To do so, we need a generalization of the notion of ordered \(n\)-tuple.

Definition 5.6.7 (Partial \(n\)-tuple) A partial \(n\)-tuple is a function \(p\), such that

\[\operatorname{Source}(p, \delta n).\]

That is, a partial \(n\)-tuple is a function which is partial on \(\{0, 1, \ldots, n-1\}\). So a partial \(n\)-tuple can be viewed as an \(n\)-tuple that may contain holes in it. For example, we can denote a partial 6-tuple \(\lambda(1\rightarrow x, 3\rightarrow y)\) as \(\langle \lambda(1\rightarrow x, 3\rightarrow y) \rangle\).

Since

\[\lambda(1\rightarrow x, 3\rightarrow y) \simeq \lambda(0\rightarrow \perp, 1\rightarrow x, 2\rightarrow \perp, 3\rightarrow y, 4\rightarrow \perp, 5\rightarrow \perp),\]

we may also write this partial 6-tuple as \(\langle \perp, x, \perp, y, \perp \rangle\). Note, that an ordered \(n\)-tuple is a partial \(n\)-tuple. Note also, that a partial \(n\)-tuple is a partial \(m\)-tuple, if \(m \geq n\).

Now we can define non-strict \(n\)-ary functions.

Definition 5.6.8 (Non-Strict \(n\)-ary Function) A function \(f\) is a non-strict \(n\)-ary function, iff all arguments of \(f\) are partial \(n\)-tuples.

Note, that ordinary \(n\)-ary functions are also non-strict \(n\)-ary. In order to avoid misunderstanding, we should require that a non-strict function has in its domain at least one tuple containing holes. Then ordinary \(n\)-ary functions can be called strict. As we already remarked, taken as one-place, all functions are strict (cf. Section 3).

As an example, consider a non-strict unary function \(f\), such that \(\langle \rangle \in \delta f\), and suppose \(\sigma\) is a non-existing term. Then \(\langle \sigma \rangle \simeq \langle \rangle\), and thus \(f(\sigma)\) exists, which is in accordance with the common meaning of non-strict functions.

Within this paper we will not elaborate non-strict functions any further.
5.7 Currying

A well-known operation on \(n\)-ary functions is the operation of currying. Usually, it is introduced in a more or less intuitive way, and only for total functions (cf. e.g. CURRY AND FEYS (1958), STOY (1977), SCHMIDT (1986)). Here we work in a general framework of partial functions, and we will define currying in a precise way. Furthermore, we will show that there exists a unique curried version of every \(n\)-ary function.

We start with the basic definition.

**Definition 5.7.1 (Currying)** A function \(g\) is a curried version of an \(n\)-ary function \(f\), iff for every \(x_0,\ldots,x_{n-1}\)
\[
gx_0\cdots x_{n-1} \simeq f\langle x_0,\ldots,x_{n-1}\rangle.\]
□

However, this definition gives us a problem with respect to the unicity of \(g\). This problem is due to the fact that we are working in a general framework of partial functions. We will illustrate it with an example.

Suppose \(f\) is a ternary function, and
\[
\delta f \simeq \{\langle a_0, b_0, c_0\rangle, \langle a_0, b_1, c_1\rangle, \langle a_1, b_1, c_1\rangle\}.
\]
The curried version \(g\) of \(f\) must be such, that for all \(a, b, c\) we have
\[
gabc \simeq f\langle a, b, c\rangle,
\]
i.e., \(g\) gives “ultimately” the same result as \(f\). Notice however, that for only three triples of values of \(a, b\) and \(c\) this result will be defined.

The problem now is that, e.g., \(ga_1b_0\) can be undefined, but it can also be the empty function, since in both cases \(ga_1b_0c_0\) and \(ga_1b_0c_1\) will be undefined. In fact, the same problem already occurs with \(g\) itself: \(g\) can be defined for some \(a\) (\(\neq a_0, a_1\)), as long as \(gabc\) is undefined for every \(b\) and \(c\).

So, if we want to guarantee the unicity of \(g\), we have to strengthen currying with some “minimality property”.

**Definition 5.7.2** A curried version \(g\) of an \(n\)-ary function \(f\) is minimal, iff for all \(x_0,\ldots,x_{i-1}\) (\(i = 0, 1,\ldots,n-1\)), and for all \(x\)
\[
x \in \delta(gx_0\cdots x_{i-1}) \leftrightarrow
\exists x_{i+1},\ldots,x_{n-1}(\langle x_0,\ldots,x_{i-1}, x, x_{i+1},\ldots,x_{n-1}\rangle \in \delta f). \]
□

In this definition it is assumed that \(\delta\) is strict (as agreed in Section 5.3), i.e., \(\delta(gx_0\ldots x_{n-1}) \simeq \bot\) whenever \(gx_0\ldots x_{n-1} \simeq \bot\). Remember also, that \(\delta(gx_0\ldots x_{n-1}) \simeq \bot\) implies that \(x \in \delta(gx_0\ldots x_{n-1})\) is false for every \(x\).
We will speak of the curried version $g$ of an $n$-ary function $f$, denoted as $\Gamma f$, when $g$ has the minimality property. We have the following theorem.

**Theorem 5.7.3 (Currying)** For every $n$-ary function $f$ a unique function $g$ exists, such that $g$ is the curried version of $f$.

**Remark.** The definition of currying is in fact a definition scheme, i.e., for every $n$ we have a different definition of currying. The same holds for this theorem, and in fact it already holds for the definition of $n$-ary functions. So we have a sequence of predicates 0-ary, 1-ary, 2-ary, etc. To prove the theorem, we will prove it for 0-ary, 1-ary and 2-ary functions, and then proceed by induction on the arity of $f$.

In order to see that this form of induction fits with Theorem 5.5.3, we first define the binary predicate $\text{Ary}(f,n)$, meaning that $f$ is an $n$-ary function, as follows:

$$\text{Ary}(f,n) \leftrightarrow \text{Nat}(n) \land \forall x \in \delta f \forall i \in \delta x (i \in \delta n).$$

Then we must prove (by mathematical induction) the formula

$$\forall f (\text{Ary}(f,n) \rightarrow \exists ! g (g \simeq \Gamma f)),$$

which is precisely what we described above.

**Proof of Theorem 5.7.3.** Let $f$ be a nullary function, i.e., the only argument of $f$ is the empty tuple $\langle \rangle$. Suppose $f \langle \rangle \simeq g$. It is immediately clear, that $g$ is uniquely determined, and that $g$ ultimately (after zero applications) gives the same result as $f$, when $f$ is applied to the tuple of zero arguments. Furthermore, the minimality property becomes void, so it is trivially fulfilled. So, $\Gamma f \simeq g$ when $f$ is a nullary function\(^2\).

Next, let $f$ be a unary function. Define $g$, such that for all $x$

$$gx \simeq f(x).$$

It is routine to check that $g$ is the unique curried version of $f$.

Now let $f$ be a binary function. Define the set $a$ as follows:

$$a \simeq \{ x \mid \exists x_1 (\langle x, x_1 \rangle \in \delta f) \}.$$  

\(^2\) Often nullary functions are considered as constants. If we want to do that in the framework of the theory presented here, it is more correct to take the curried version of a nullary function.

32
Define for every $x$ a function $g_x$ such that $g_x x_1 \simeq f\langle x, x_1 \rangle$. Trivially, $g_x$ exists and is uniquely determined. Note, that $g_x \simeq \lambda()$ iff $x \notin a$.

Finally, define the function $g$ such that $g x \simeq g_x$ for every $x \in a$, and $g$ is undefined for every other $x$. Here too, it is easy to prove that $g$ exists, and that $g$ is uniquely determined.

To conclude that $g$ is the curried version of $f$, we remark that $g$ ultimately gives the same result as $f$, i.e., for every $x_0, x_1$

$$gx_0 x_1 \simeq f\langle x_0, x_1 \rangle,$$

and, furthermore, $g$ has the minimality property, i.e., for every $x_0, x$

\[- x \in \delta g \iff \exists x_1 ((x, x_1) \in \delta f),\]

\[- x \in \delta (gx_0) \iff (x_0, x) \in \delta f.\]

So, $g$ is the unique curried version of $f$, i.e., $\Gamma f \simeq g$.

Now suppose that the curried version of every $n$-ary function exists, and let $f$ be an $(n+1)$-ary function. Define the binary function $f'$, such that all first quasi arguments of $f'$ are $n$–tuples, and such that

$$f'\langle\langle x_0, \ldots, x_{n-1}, x_n \rangle \simeq f\langle x_0, \ldots, x_{n-1}, x_n \rangle.$$

It is easy to prove that $f'$ exists and is uniquely determined. Since $f'$ is binary, we have

$$\Gamma f'\langle x_0, \ldots, x_{n-1} \rangle x_n \simeq f'\langle\langle x_0, \ldots, x_{n-1}, x_n \rangle,$$

where $\Gamma f'$ is an $n$-ary function. So, by the induction hypothesis $\Gamma(\Gamma f')$ exists and is unique. Again, to conclude that $\Gamma(\Gamma f')$ is the curried version of $f$, we must show that $\Gamma(\Gamma f')$ ultimately gives the same result as $f$, and secondly, that $\Gamma(\Gamma f')$ has the minimality property. With respect to the first requirement, notice that for all $x_0, \ldots, x_{n-1}, x_n$ we have

$$\Gamma(\Gamma f')x_0 \cdots x_{n-1} x_n \simeq \Gamma f'\langle x_0, \ldots, x_{n-1} \rangle x_n \simeq f'\langle\langle x_0, \ldots, x_{n-1}, x_n \rangle.$$ 

With respect to the second requirement, notice that for all $x$, and for all $x_0, \ldots, x_{i-1}$ ($i = 0, 1, \ldots, n-1$) we have

$$x \in \delta(\Gamma(\Gamma f'))x_0 \cdots x_{i-1} $$

$$\iff \exists x_{i+1}, \ldots, x_{n-1} ((x_0, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_{n-1}) \in \delta(\Gamma f'))$$

$$\iff \exists x_{i+1}, \ldots, x_{n-1} \exists x_n (((x_0, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_{n-1}), x_n) \in \delta f')$$

$$\iff \exists x_{i+1}, \ldots, x_{n-1}, x_n ((x_0, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_{n-1}, x_n) \in \delta f).$$
Finally, for all \( x_0, \ldots, x_{n-1} \) (i.e., the case that \( i = n \)), and for all \( x \) we have

\[
x \in \delta(\Gamma f')x_0 \cdots x_{n-1} \iff x \in \delta(\Gamma f'(x_0, \ldots, x_{n-1})) \\
\iff \langle x_0, \ldots, x_{n-1}, x \rangle \in \delta f' \\
\iff \langle x_0, \ldots, x_{n-1}, x \rangle \in \delta f
\]

Thus \( \Gamma(\Gamma f') \) fulfills the requirements of Definitions 5.7.1 and 5.7.2 and so \( \Gamma f \simeq \Gamma(\Gamma f') \).

Note, that the empty function has any arity whatsoever, so it is not a priori clear what it means to curry \( \lambda() \). However, whatever arity we consider \( \lambda() \) to have, we always have \( \Gamma(\lambda()) \simeq \lambda() \).

On the other hand, this suggests that currying depends on our way of looking at a function, that is to say, that we might consider an arbitrary function as being partly 0-ary, 1-ary, etc. The operation of currying can be generalized towards arbitrary functions, by indicating which 0-ary, 1-ary, etc. part of a function must be curried.

Often, also the inverse operation of uncurrying a function \( f \) is introduced, but if we want to do that in the present general framework of partial functions, we have to indicate the arity of the uncurried version of \( f \). Finally, when we want to curry non-strict \( n \)-ary functions, we have to change the definition of currying a little.

We don’t examine these aspects of currying here.

### 5.8 The Axiom of Universality

Up to now, all axioms were rather simple. The next axiom is called the Axiom of Universality, and is a more complicated one. It is concerned with functions that are often felt to be counter intuitive in some sense, e.g., like functions that can be applied to themselves (i.e., \( ff \) exists), or like functions which can be mutually applied to each other (i.e., \( fx \) and \( xf \) both exist). As another example, there can exist functions \( a, b \) and \( c \) such that \( ab \simeq c \), \( bc \simeq a \) and \( ca \simeq b \).

From a point of view of generality, we choose to allow for the existence of “funny” functions like these. The Axiom of Universality is a general formulation of our choice. In order to introduce it, we start with an example. Suppose we want to prove the existence of the following functions:

\[
a \simeq \lambda(a \rightarrow a, b \rightarrow b, c \rightarrow c) \\
b \simeq \lambda(a \rightarrow b, b \rightarrow a) \\
c \simeq \lambda(c \rightarrow c)
\]
Note, that these functions need not be unique, there can be different functions answering the same pattern. They are, so to speak, defined up to isomorphism.

Instead of proving the existence of $a$, $b$ and $c$, we may equally well prove the existence of a binary function $g$, which reflects the same pattern, and which must be an apply function, i.e., $g(x, y) \simeq xy$ for all $x \in \Delta_qg$, and for all $y$. In other words, when we formulate $g$ using infix notation, we get the ordinary quasi operation of application restricted to $\Delta_qg$.

In our example we have

$$g \simeq \lambda\langle(a, a)\mapsto a, (a, b)\mapsto b, (a, c)\mapsto c, \quad (b, a)\mapsto b, (b, b)\mapsto a, \quad (c, c)\mapsto c\rangle.$$  

Apart from the strangeness of $a$, $b$ and $c$ there is nothing strange about $g$ itself. So if we take three ordinary (i.e., non-funny) functions $a'$, $b'$ and $c'$ instead of $a$, $b$ and $c$ respectively, we have a perfectly non-strange function $f$, where

$$f \simeq \lambda\langle(a', a')\mapsto a', (a', b')\mapsto b', (a', c')\mapsto c', \quad (b', a')\mapsto b', (b', b')\mapsto a', \quad (c', c')\mapsto c\rangle.$$  

We say that $f$ is isomorphic to $g$. Using infix notation, we can denote $f$ as $\cdot f$, so we get $a' \cdot f a' \simeq a', a' \cdot f b' \simeq b'$, etc.

The Axiom of Universality now proceeds in the opposite direction. Intuitively it says: given $f$, there exists an apply function $g$, which is isomorphic to $f$. As a consequence, all funny functions which are elements of the quasi field of $g$ (in the example) also exist.

There is still one remark to make: $f$ has to meet an additional requirement (called internal extensionality), in order to answer the Axiom of Extensionality. To show why, suppose that we extend the definition of $f$ with an extra line, such that we get

$$f' \simeq f \left[ \langle d', a'\rangle\mapsto b', \langle d', b'\rangle\mapsto a' \right]$$  

$$\simeq \lambda\langle(a', a')\mapsto a', (a', b')\mapsto b', (a', c')\mapsto c', \quad (b', a')\mapsto b', (b', b')\mapsto a', \quad (c', c')\mapsto c', \quad (d', a')\mapsto b', \langle d', b'\rangle\mapsto a'\rangle.$$  

35
Then we have
\[ f'(d', a') \simeq f'(b', a'), \]
\[ f'(d', b') \simeq f'(b', b'). \]
Furthermore, for all \( x \neq a', b' \) we have
\[ f'(d', x) \simeq f'(b', x) \ (\simeq \bot). \]
The consequence of this would be that, because of \( g \), there would exist a function \( d \), such that \( da \simeq ba \), \( db \simeq bb \) and both \( d \) and \( b \) would be undefined for all other \( x \). Thus, by the Axiom of Extensionality, \( d = b \). In order to maintain isomorphism between \( f \) and \( g \) this extension of \( f \) is not allowed.

More precisely, we have the following definitions.

**Definition 5.8.1** A binary function \( f \) is *internally extensional*, denoted as \( \text{IExt}(f) \), iff for all \( x_1, x_2 \in \Delta_q f \)
\[ \forall y (f(x_1, y) \simeq f(x_2, y)) \rightarrow x_1 = x_2. \]

**Definition 5.8.2** Two \( n \)-ary functions \( f \) and \( g \) are *isomorphic*, denoted as \( f \sim g \), iff there is a function \( h \) such that
- \( \text{Inj}(h) \),
- \( \delta h \simeq \Delta_q f \),
- \( \rho h \simeq \Delta_q g \),
- \( \forall x_0, \ldots, x_{n-1} (h(f(x_0, \ldots, x_{n-1})) \simeq g(hx_0, \ldots, hx_{n-1})). \)

**Definition 5.8.3** A binary function \( f \) is called an *apply function*, denoted as \( \text{AF}(f) \), iff
\[ \forall a \in \Delta_q f \forall x (f(a, x) \simeq ax). \]

The Axiom of Universality can now be formulated as follows.

**Axiom 7 (Universality)** \[ \forall f (\text{IExt}(f) \rightarrow \exists g (g \sim f \land \text{AF}(g))). \]

In other words, the property of internal extensionality says that \( f \) can be interpreted as some operation of application, restricted to a certain set (the quasi field of \( f \)). Then by the Axiom of Universality, given such an \( f \), there is a set (the quasi domain of \( g \)), such that the “real” (pseudo) operation
of application (restricted to this set) is isomorphic to $f$. But since $g$ is an apply function, this restriction of the pseudo operation of application coincides with $g$ itself.

From Axiom 7 and the introducing example it is immediately clear, that there exist various sorts of funny functions, like functions that can be applied to themselves, functions that can be applied to each other, etc. For example, there exist functions like $f \simeq \lambda(f \to f)$, or $f_1 \simeq \lambda(f_2 \to f_1)$ and $f_2 \simeq \lambda(f_1 \to f_2)$, etc.. As a more interesting example we mention that there exists a model of the untyped, extensional lambda calculus. Such a model is natural in the sense that all objects of it are functions, and application in the model is just ordinary function application.

**Theorem 5.8.4** There exists a set $d$ of functions which, together with ordinary function application, forms a model for the untyped, extensional lambda calculus.

**Remarks.** We only give a sketch of the proof, since a detailed proof lies outside the scope of this paper. When we use the noun “term”, terms in the sense of the lambda calculus are meant.


**Sketch of the proof.** The terms from the untyped lambda calculus can be coded as natural numbers by means of some standard technique of Gödel numbering. Let $G$ be the set of all Gödel numbers $[X]$ where $X$ is a term. We can define an equivalence relation $\equiv$ on $G$ as follows:

$$[X] \equiv [Y] \text{ iff } X =_{\beta\eta} Y,$$

where $=_{\beta\eta}$ means $\beta\eta$-convertibility in the lambda calculus. Then a model for the lambda calculus can be constructed along the same lines as a term model (cf. Hindley and Seldin (1986), pages 116–117), using equivalence classes of Gödel numbers instead of equivalence classes of terms. Within this model a binary operation of application $\cdot$ exists (modeling application from the lambda calculus), so there exists a binary function $f$ such that

$$f(x, y) \simeq x \cdot_{\lambda} y,$$

where $x$ and $y$ are equivalence classes under $\equiv$ of elements of $G$. Since we consider the *extensional* lambda calculus, $f$ is internally extensional. Thus, according to Axiom 7 there exists an apply function $g$ which is isomorphic to $f$. That is, the quasi field of $g$ with ordinary application is a model for the untyped, extensional lambda calculus. So $d \simeq \Delta_q g$.

Notice that $d \subseteq d \to d$. 

37
5.9 The Axiom of Choice

Now we come to the final axiom of the theory, which is a simple functional form of the Axiom of Choice. Since in the literature (cf. Rubin and Rubin (1963)) the Axiom of Choice is examined extensively, we don’t give any applications of it, but merely mention it.

Axiom 8 (Axiom of Choice) $\forall f \exists g (f \circ g \circ f \simeq f)$. \hfill $\Box$

Notice the concision and elegance of this formulation of the Axiom of Choice in comparison with many other formulations of the Axiom of Choice. It was mentioned in Arbib and Manes (1975), page 9.

5.10 List of Axioms

We conclude the description of the theory of partial functions by giving a complete list of all the axioms.

Axiom 1 (Uniqueness) (cf. Section 5.1)

$\forall f,x,y_1,y_2 (fx \simeq y_1 \land fx \simeq y_2 \rightarrow y_1 = y_2)$.

Axiom 2 (Extensionality) (cf. Section 5.1)

$\forall f,g (\forall x (fx \simeq gx) \rightarrow f = g)$.

Axiom 3 (Restricted Comprehension) (cf. Section 5.1)

$\forall f \exists g \forall x,y (gx \simeq y \leftrightarrow \text{Fld}(f,x) \land \varphi(x,y))$,

where $\varphi(x,y)$ is a functional formula, not containing $g$ free.

Axiom 4 (Infinity) (cf. Section 5.2)

$\exists f (\text{SF}(f))$.

Axiom 5 (Lower Order) (cf. Section 5.4)

$\forall f \exists g \forall x (x \in \delta g \leftrightarrow \exists u (u \in \delta f \land x \in \delta u))$.

Axiom 6 (Higher Order) (cf. Section 5.4)

$\forall f \exists g \forall x (x \in \delta g \leftrightarrow (\delta x \simeq \delta f \land \rho x \simeq \rho f))$.

Axiom 7 (Universality) (cf. Section 5.8)

$\forall f (\text{IExt}(f) \rightarrow \exists g (g \sim f \land \text{AF}(g)))$. 

38
Axiom 8 (Axiom of Choice) (cf. Section 5.9)

\[ \forall f \exists g (f \circ g \circ f \simeq f). \]

From now on, we will call the theory consisting of Axioms 1 to 8: Axiomatic Function Theory, abbreviated as AFT.

6 Consistency

In order to prove the (relative) consistency of AFT, we will construct an inner model in ZF-Set Theory including the Axiom of Choice. For simplicity we use a variant of ZF-Set Theory without the Axiom of Foundation, but including an axiom which asserts the existence of non-well-founded sets, for which we choose the so called Axiom of Universality. The resulting variant of ZF-Set Theory is denoted as ZFC\(^\circ\)U, and known to be relatively consistent with respect to ZFC, i.e., ZF-Set Theory including the Axiom of Foundation and the Axiom of Choice (cf. BOFFA (1969), VON RIMSCHA (1980, 1981), ACZEL (1988)).

The framework of this section is set theoretical, and all concepts (like natural number, ordered \(n\)-tuple, set, function) are to be understood in the sense of set theory, and must not be confused with the same concepts from the other sections of this paper, where they are meant in the sense of AFT.

As usual, an ordered pair \(\langle x, y \rangle\) is defined as the set \(\{\{x\}, \{x, y\}\}\). A function from \(a\) to \(b\) is a relation between \(a\) and \(b\) answering the property of uniqueness, i.e., a function is a specific set of ordered pairs. If a set \(a\) is a function, this is denoted as \(Fnc(a)\). Note, that if \(a\) and \(b\) are two functions, then \(a \subseteq b \iff a \subseteq b\). The domain and range of a function \(f\) are denoted as \(\delta f\) and \(\rho f\) respectively. Clearly, the ternary relation of application (in infix notation, cf. Section 3) is defined as

\[ ax \simeq y \iff Fnc(a) \land \langle x, y \rangle \in a. \]

By \(Tr(a)\) we denote, that a set \(a\) is transitive; \(TC(a)\) denotes the transitive closure of \(a\).

We continue with some basic definitions, which are not completely standard.

**Definition 6.1 (Field)** The field \(\Delta a\) of a set \(a\) is the set

\[ \{x \mid \exists y (\langle x, y \rangle \in a \lor \langle y, x \rangle \in a)\}. \]
Definition 6.2 (Structure) A structure is an ordered pair \( \langle a, r \rangle \) where \( a \) is a set and \( r \) a relation on \( a \), i.e., \( r \subseteq a \times a \).

Definition 6.3 Let \( s \) be a structure \( \langle a, r \rangle \). Then \( s \) is extensional, denoted as \( Es(s) \), iff for all \( x, y, z \in a \)

\[
(x, y) \in r \land \langle x, z \rangle \in r \rightarrow y = z.
\]

Definition 6.4 The internal \( \epsilon \)-structure \( \epsilon(t) \) of a set \( t \) is the structure \( \langle t, e \rangle \), where \( e \) denotes the set

\[
\{ \langle x, y \rangle \mid x, y \in t \land x \in y \}.
\]

Isomorphism of two structures \( s \) and \( t \) is defined as obvious, and denoted as \( s \sim t \).

In set theory the problem of undefinedness also arises naturally, e.g. as in \( \{ x \mid \exists y(x \in y) \} \). It is common practice in set theory to handle this problem in an intuitive way, by agreeing that non-existing terms are not used. For example, \( \{ x \mid x = z \} \) is not used, thus avoiding meaningless formulas like \( \{ x \mid x = x \} = a \). To handle undefinedness within the set theoretical language, we can apply the same technique of pseudo terms and predicate schemes as described in Section 3, where \( \{ x \mid \varnothing \} \) is considered to be a pseudo term. The four notational conventions (cf. Section 3) remain the same, except for convention 2 which has to be replaced by convention 2’ (\( \sigma, \tau \) stand for pseudo terms; \( x, y \) stand for variables not occurring in \( \sigma, \tau \)):

\[
\sigma \in \tau \text{ means } \exists x, y(\sigma \simeq x \land \tau \simeq y \land x \in y).
\] (2’)

Notice, that because of these conventions, \( \in \) becomes a predicate scheme (though we will just write \( \in \), instead of \( \in^* \), cf. Section 3). Though we will not need it, \( \bot \) can be considered as a shorthand notation for e.g. \( \{ x \mid x = x \} \).

We will not further elaborate the technique of pseudo terms for set theory here.

As already remarked, there are set theoretical and function theoretical definitions of many notions. In the consistency proof we will need both definitions of several of these notions. Since the framework of this section is set theoretical, we will denote a notion by its standard formulation, as for example \( 0, 1, x \in a, \langle x, y \rangle \), when the set theoretical definition is meant. When a notion is meant in the sense of AFT, we will place a dot above it, as in \( \hat{0}, \hat{1}, x \in \dot{a}, \langle x, y \rangle \). However, also the function theoretic definitions
must be understood in set theoretical terms, where a function is a specific set of ordered pairs. For example, we have

\[ 0 \simeq \emptyset, \quad 1 \simeq \{\emptyset\}, \]
\[ \hat{0} \simeq \emptyset, \quad \hat{1} \simeq \{\emptyset, \emptyset\}, \]

and

\[ \langle a, b \rangle \simeq \{\{a\}, \{a, b\}\}, \quad \hat{\langle a, b \rangle} \simeq \{\langle x, y \rangle \mid (x \simeq \hat{0} \land y = a) \lor (x \simeq \hat{1} \land y = b)\}. \]

Note, that the following equivalences hold (\(f\) and \(g\) denote functions in the set theoretical sense):

\[ x \in \delta f \leftrightarrow x \in \delta f, \]
\[ x \in \rho f \leftrightarrow x \in \rho f, \]
\[ x \in \Delta f \leftrightarrow x \in \Delta f, \]
\[ f \sim g \leftrightarrow \langle \Delta f, f \rangle \sim \langle \Delta g, g \rangle. \]

In the last equivalence, \(f\) and \(g\) are supposed to be binary functions.

We continue with formulating the axioms of ZFC^\circ U.

**Axiom ZF-1 (Extensionality)** \(\forall a, b(\forall x(x \in a \leftrightarrow x \in b) \rightarrow a = b).\)

**Axiom ZF-2 (Replacement)** \(\forall a \exists b \forall y(y \in b \leftrightarrow \exists x(x \in a \land \varphi(x, y))), \)
where \(\varphi(x, y)\) is a functional formula.

**Axiom ZF-3 (Infinity)** \(\exists a \neq \emptyset \forall x \in a \exists y \in a(x \subseteq y \land x \neq y).\)

**Axiom ZF-4 (Sum Set)** \(\forall a \exists b \forall x(x \in b \leftrightarrow \exists y(x \in y \land y \in a)).\)

**Axiom ZF-5 (Power Set)** \(\forall a \exists b \forall x(x \in b \leftrightarrow x \subseteq a).\)

**Axiom ZF-6 (Universality)** \(\forall s(\text{Es}(s) \rightarrow \exists t(Tr(t) \land e(t) \sim s)).\)

**Axiom ZF-7 (Choice)** \(\forall a(\forall x, y \in a(x = y \lor x \cap y \simeq \emptyset) \rightarrow \exists b \forall x \in a(x \neq \emptyset \rightarrow \exists y(b \cap x \simeq \{y\}))).\)
When we compare these set theoretical axioms with the axioms of AFT, which were formulated in Section 5, there appears to be a striking similarity between AFT and ZFC$^0 U$. All axioms but one correspond rather directly to each other, e.g., the AFT-Axiom Scheme of Restricted Comprehension corresponds to the ZF-Axiom Scheme of Replacement, the AFT-Axioms of Lower and Higher Order correspond to the Sum Set and Power Set Axiom from ZFC$^0 U$ respectively. The only axiom that is present in AFT, but not in ZFC$^0 U$, is the Axiom of Uniqueness.

The elegance of some of the corresponding axioms differs a little, and it is remarkable that some of the less elegant axioms of AFT (to our taste the Axioms of Lower and Higher Order) became a bit more elegant by using sets. The same holds for ZFC$^0 U$, where the Axioms of Infinity and Choice can be made more elegant by using functions.

Of course, as an explanation of the similarities between AFT and ZFC$^0 U$, we must say that ZF-Set Theory was an important source of inspiration during the development of AFT. Nevertheless, it is surprising that the concept of function can be axiomatized in almost the same way as the concept of set, such that the resulting theory about functions is equi-consistent with and equally powerful as the corresponding theory about sets. In the remaining part of this section and in the next section we will prove that fact by interpreting AFT within ZFC$^0 U$ and vice versa.

We continue with some definitions, in order to construct within ZFC$^0 U$ a model for AFT.

**Definition 6.5 (Applicatively Closed)** A set $a$ is **applicatively closed** iff

$$\forall x(x \in a \to \Delta x \subseteq a).$$

**Definition 6.6 (Applicative Closure)** The **applicative closure** of a set $a$ is the smallest applicatively closed set $b$ such that $\Delta a \subseteq b$.

The applicative closure of $a$ is denoted as $\Delta a$. Note, that $\Delta a$ exists for every set $a$, since $\Delta a \subseteq TC(a)$.

**Definition 6.7 (Pure Function)** A function $f$ is a **pure function**, denoted as $PF(f)$, iff $\forall x(x \in \Delta a \to Fnc(x))$.

Note, that $\emptyset$ is a pure function, the empty function. If $f$ is a pure function, then all arguments and values of $f$ are also pure functions.

The next theorem states the consistency of AFT.

**Theorem 6.8 (Consistency of AFT)** The collection of pure functions, together with the ternary relation of application, is a model of AFT.
The proof of this theorem consists of eight lemmas, one for every axiom of AFT. Clearly, all formulas are to be relativized to the collection of pure functions, together with the relation of application, i.e., all quantifiers range over pure functions.

**Lemma 6.9 (Axiom AFT-1)** \( \forall f, x, y_1, y_2 (fx \simeq y_1 \land fx \simeq y_2 \rightarrow y_1 = y_2) \).

**Proof.** Uniqueness of pure functions follows immediately from the definition of functions in set theory. \( \square \)

**Lemma 6.10 (Axiom AFT-2)** \( \forall f, g (\forall x (fx \simeq gx) \rightarrow f = g) \).

**Proof.** Similarly, extensionality of pure functions follows immediately from extensionality of sets. \( \square \)

**Lemma 6.11 (Axiom AFT-3)** \( \forall f \exists g \forall x, y (gx \simeq y \leftrightarrow x \in \Delta f \land \varphi(x, y)) \).

**Proof.** Let \( f \) be a pure function, and \( \varphi(x, y) \) a functional formula which can be expressed using the ternary relation of application as its only non-logical predicate. Clearly, the set \( g \simeq \{ \langle x, y \rangle | x \in \Delta f \land \varphi(x, y) \} \) exists. This is the pure function, whose existence is required. \( \square \)

**Lemma 6.12 (Axiom AFT-4)** \( \exists f (SF(f)) \).

**Proof.** Define the function \( g \), with domain the set \( \omega \) of all natural numbers, recursively as follows:

\[
g0 \simeq \emptyset,
g(s\alpha) \simeq \{ \langle g\alpha, g\alpha \rangle \},
\]

where \( s\alpha \) is the usual set theoretical successor function for ordinal numbers, i.e., \( s\alpha \simeq \alpha \cup \{\alpha\} \).

Then \( g\alpha \) is a pure function for every natural number \( \alpha \). Now define the function \( f \), with the range of \( g \) as its domain, such that \( fx \simeq \{ \langle x, x \rangle \} \). Then \( f \) is a pure function, and, as can be checked easily, \( f \) is a successor function in the sense of AFT. \( \square \)

**Lemma 6.13 (Axiom AFT-5)** \( \forall f \exists g \forall x (x \in \delta g \leftrightarrow \exists u (u \in \delta f \land x \in \delta u)) \).

**Proof.** Let \( f \) be a pure function. Define

\[
g \simeq \{ \langle x, x \rangle | \exists u (u \in \delta f \land x \in \delta u) \}.
\]

Then \( g \) is a pure function, defined for all arguments of all arguments of \( f \). The fact that \( g \) is the identity function on \( \bigcup \{ \delta u | u \in \delta f \} \) does not matter, it just makes the definition of \( g \) easy. \( \square \)
Lemma 6.14 (Axiom AFT-6) \( \forall f \exists g \forall x (x \in \delta g \leftrightarrow (\delta x \simeq \delta f \land \rho x \simeq \rho f)) \).

**Proof.** Let \( f \) be a pure function. Define the set
\[
t \simeq \{ x \in \mathcal{P}(\delta f \times \rho f) \mid \text{Func}(x) \land \delta x \simeq \delta f \land \rho x \simeq \rho f \}.
\]
All elements of \( t \) are pure functions. Now the existence of a pure function \( g \), such that \( \delta g \simeq t \), follows immediately. \( \square \)

Lemma 6.15 (Axiom AFT-7) \( \forall f (\text{IExt}(f) \rightarrow \exists g (g \sim f \land \text{AF}(g))) \).

**Proof.** Suppose \( f \) is a pure function. Suppose also, that \( f \) is a binary function in the sense of AFT, i.e., all arguments of \( f \) are of the form \( \langle x, y \rangle \).

Let
\[
D_q f \simeq \{ x \mid \exists y,z (\langle \langle x, y \rangle, z \rangle \in f \lor \langle \langle y, x \rangle, z \rangle \in f \lor \langle \langle y, z \rangle, x \rangle \in f) \}.
\]
Now suppose, that \( f \) is internally extensional in the sense of AFT, i.e., for all \( x_1, x_2 \in D_q f \)
\[
\forall y,z (\langle \langle x_1, y \rangle, z \rangle \in f \leftrightarrow \langle \langle x_2, y \rangle, z \rangle \in f) \rightarrow x_1 = x_2.
\]
Define for all \( x \in D_q f \) the set \( x' \simeq \{ x, 2 \} \), where \( 2 \simeq \{ \emptyset, \{ \emptyset \} \} \), and define the set
\[
a \simeq \{ x' \mid x \in D_q f \} \cup \{ \{ y', \{ y', z' \}, \langle y', z' \rangle \mid \exists u (\langle \langle u, y' \rangle, z \rangle \in f) \}.
\]
Then for all \( x \in D_q f \) and for all \( u \in a \) we have \( u \notin x' \).

Next, define
\[
r \simeq \{ \langle u, v \rangle \mid u, v \in a \land u \in v \} \cup \{ \langle \langle y', z' \rangle, x' \rangle \mid \langle \langle x, y \rangle, z \rangle \in f \},
\]
then the structure \( \langle a, r \rangle \) is an extensional structure. Thus, by Axiom ZF-6 there exists an isomorphism \( h \) with domain \( a \), such that
- \( \{ hx \mid x \in a \} \) is a transitive set,
- \( hu \in hv \leftrightarrow \langle u, v \rangle \in r \).

Now define the binary function (in the sense of AFT)
\[
g \simeq \{ \langle \langle hx', hy', z' \rangle, \langle x, y \rangle, z \rangle \in f \},
\]
then \( g \) is a pure function such that \( g \sim f \) and furthermore, \( g \) is an apply function in the sense of AFT (straightforward). \( \square \)
Lemma 6.16 (Axiom AFT-8) \( \forall f \exists g (f \circ g \circ f \simeq f) \).

Proof. We start with a well-known equivalent of the Axiom of Choice (cf. Rubin and Rubin (1963), Van Dalen et al (1978)):

\[ \forall f \exists h (h \subseteq f \land \rho f \simeq \rho h \land \text{Inj}(h)). \]

Given that \( f \) is a pure function it is immediately clear that \( h \) is a pure function too. Now let \( h^{-1} \) be the inverse function of \( h \), i.e., \( h^{-1} y \simeq x \) iff \( hx \simeq y \), as usual. By Axiom 3 \( h^{-1} \) exists when \( h \) is injective. Now it is straightforward that \( f \circ h^{-1} \circ f \simeq f \). \( \square \)

7 A Model for Set Theory

In this section we will construct within AFT a model for ZFC^U. The framework of this section is function theoretic again, i.e., all concepts must be understood in the sense of AFT. If necessary, we will place a dot above the notation of a concept, if that concept is meant in the set theoretical sense. So the situation in this section is precisely the opposite of the situation in Section 6.

We start with some definitions and theorems.

Definition 7.1 (Weakly Transitive) A function \( f \) is weakly transitive iff

\[ \forall x \in \delta f (\delta x \subseteq \delta f). \]

As a motivation for the prefix “weakly” in Definition 7.1, we give the following definition.

Definition 7.2 (Transitive) A function \( f \) is transitive, denoted as \( \text{Tr}(f) \), iff

\[ \forall x \in \delta f (x \subseteq \delta f). \]

Note, that a transitive function is also weakly transitive. Note also, that when \( f \) is a set, transitivity of functions coincides with transitivity of sets.

Definition 7.3 (Domain Closure) A set \( a \) is the domain closure of a function \( f \), denoted as \( \delta_c f \), iff \( a \) is the smallest weakly transitive set, such that \( \delta f \subseteq a \).

Theorem 7.4 For every function \( f \), \( \delta_c f \) exists.
Proof. Define inductively the function \( g \), with the set of natural numbers as its domain, such that

\[- g0 \simeq \delta f, \]
\[- g(s \cdot n) \simeq \bigcup \{ \delta x \mid x \in gn \}, \]

where \( s_N \) denotes the successor function for the natural numbers (cf. Section 5.5). Then \( \bigcup (\rho g) \) is the required set \( \delta_c f \).

Now we come to the definition, that is central to the construction of a model for \( \text{ZFC}_U \).

Definition 7.5 (Pure Set) A set \( a \) is a pure set, denoted as \( \text{PS}(a) \), iff

\[ \forall x \in \delta_c a (\text{Set}(x)). \]

The next theorem asserts, that \( \text{ZFC}_U \) can be embedded in AFT.

Theorem 7.6 (Embedding \( \text{ZFC}_U \)) The collection of all pure sets, together with the membership relation, is a model of \( \text{ZFC}_U \).

Within \( \text{ZFC}_U \) well-founded sets can be defined in a well-known way, so \( \text{ZFC} \) can also be embedded in AFT (cf. e.g. Fraenkel et al (1973), pages 93–94).

The proof of Theorem 7.6 consists of seven lemmas, one for each axiom of \( \text{ZFC}_U \). Here too, all formulas must be relativized to the collection of pure sets.

Lemma 7.7 (Axiom ZF-1) \( \forall a, b (\forall x (x \in a \leftrightarrow x \in b) \rightarrow a = b) \).

Proof. Extensionality of pure sets follows immediately from extensionality of functions.

Lemma 7.8 (Axiom ZF-2) \( \forall a \exists b \forall y (y \in b \leftrightarrow \exists x (x \in a \land \varphi(x, y))) \), where \( \varphi(x, y) \) is a functional formula which can be expressed using membership as its only non-logical predicate.

Proof. Let \( a \) be a pure set. By Axiom 3 there exists a function \( f \) such that for all \( x, y \)

\[ fx \simeq y \leftrightarrow x \in a \land \varphi(x, y) \land \text{PS}(y). \]

Then \( \rho f \) is the required pure set \( b \).

Lemma 7.9 (Axiom ZF-3) \( \exists a \not= \emptyset \forall x \in a \exists y \in a (x \subseteq y \land x \not= y) \).
Proof. Define inductively the function \( f \), with domain the set of natural numbers (in the sense of AFT), such that
\[
\begin{align*}
- f^0 &\simeq \emptyset, \\
- f(s_n n) &\simeq \{ fn \}.
\end{align*}
\]
Then \( \rho f \) is a non-empty pure set, such that for all \( x \in \rho f \) there exists a \( y \in \rho f \) with \( x \subseteq y \) and \( x \neq y \). \( \square \)

Lemma 7.10 (Axiom ZF-4) \( \forall a \exists b \forall x(x \in b \iff \exists y(x \in y \land y \in a)) \).

Proof. By Theorem 5.4.1 and the observation that \( \bigcup a \) is a pure set, when \( a \) is a pure set. \( \square \)

Lemma 7.11 (Axiom ZF-5) \( \forall a \exists b \forall x(x \in b \iff x \subseteq a) \).

Proof. By Theorem 5.4.6 and the observation that \( \mathcal{P} a \) is a pure set, when \( a \) is a pure set. \( \square \)

Lemma 7.12 (Axiom ZF-6) \( \forall s \exists t (s \circ t \rightarrow \exists t (Tr(t) \land \epsilon(t) \sim s)) \).

Proof. Suppose \( s \) is an extensional structure in the sense of set theory, i.e., \( s \simeq \langle a, r \rangle \), where \( a \) and \( r \) are pure sets and \( r \subseteq a \times a \), such that for all \( x, y \in a \)
\[
\forall u (\langle u, x \rangle \in r \leftrightarrow \langle u, y \rangle \in r) \rightarrow x = y.
\]
Since \( s \) is extensional, there are two possibilities with respect to an “empty set according to \( r \)”, i.e., either there is exactly one \( \emptyset_r \in a \) such that
\[
\exists x (\langle x, \emptyset_r \rangle \in r),
\]
or there is no such \( \emptyset_r \in a \). If there is not, then let \( \emptyset_r \) denote some function, which is not a pure set, such that \( \emptyset_r \not\in a \) (e.g., \( \emptyset_r \simeq 2 \)).

Now let \( 1_r \not\neq \emptyset_r \) also be a function which is not a pure set, and define the binary function
\[
f \simeq \lambda (\langle x, y \rangle \rightarrow 1_r \mid \langle y, x \rangle \in r)[\langle 1_r, \emptyset_r \rangle \rightarrow \emptyset_r],
\]
then \( f \) is internally extensional in the sense of AFT. By Axiom 7 there exists an apply function \( g \) such that \( g \sim f \), i.e., there is an isomorphism \( h \) with domain \( \Delta_q f \) and range \( \Delta_q g \) such that
\[
h \emptyset_r \simeq \lambda () \text{ and } h 1_r \simeq 1.
\]
Furthermore, for all \( x, y \in a \)

\[
hx \simeq \lambda(\langle y \mapsto 1 \mid \langle y, x \rangle \in r \rangle),
\]

i.e., \( hx \) is a pure set and

\[
hx \simeq \{ hy \mid \langle y, x \rangle \in r \}.
\]

Now define the set

\[
t \simeq \{ hx \mid x \in a \},
\]

then \( t \) is a transitive pure set and the internal \( \epsilon \)-structure (in the sense of set theory) of \( t \) is isomorphic (in the sense of set theory) with \( s \). \( \square \)

**Lemma 7.13 (Axiom ZF-7)**

\[
\forall a(\forall x, y \in a(x = y \lor x \cap y \simeq \emptyset) \to \\
\to \exists b \forall x \in a(x \not\simeq \emptyset \to \exists y(b \cap x \simeq \{ y \}))).
\]

**Proof.** Let the pure set \( f \) be a function in the sense of set theory. Define the function \( (f')' \) in the sense of AFT as follows:

\[
f' \simeq \lambda(x \mapsto y \mid \langle x, y \rangle \in f).
\]

By Axiom 8 there exists a function \( h \), such that \( f \circ h \circ f \simeq f \).

Let \( g' \simeq (h \mid \rho f)^{-1} \), i.e., \( g' \) is the inverse of \( h \mid \rho f \), then clearly

\[
g' \subseteq f' \land \rho f' \simeq \rho g' \land \text{Inj}(g').
\]

Now define

\[
g \simeq \{ \langle x, y \rangle \mid g'x \simeq y \},
\]

then \( g \) is a pure set. Furthermore, \( g \) is a function in the sense of set theory, and we have

\[
g \subseteq f \land \rho f \simeq \rho g \land \text{Inj}(g).
\]

This is a well-known equivalent of the Axiom of Choice, and we refer the reader to the literature (e.g. Rubin and Rubin (1963), Van Dalen et al (1978)). \( \square \)

**Acknowledgements.** I wish to thank Henk Barendregt, Maarten Fokkinga and Gerrit van der Hoeven for their valuable criticism and advice. I also thank the referee of *Information and Computation* for his (her) corrections and suggestions.

**References**


