EXAMENSARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

A Model for Constructive Set Theory in Intuitionistic Type Theory

av

Noa Hermele

2007 - No 3

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET, 10691 STOCKHOLM

A Model for Constructive Set Theory in Intuitionistic Type Theory

Noa Hermele

Examensarbete i matematik 20 poäng, fördjupningskurs

Handledare: Per Martin-Löf

2007

Abstract

Peter Aczel developed a constructive set theory called CZF that is a constructive version of the classical set theory ZF. Aczel showed that CZF can be interpreted in Martin Löf's type theory by considering a type of sets, hence giving CZF a constructive meaning. In this master's thesis we review this interpretation.

Content

Introduction

Background
Set theory is a young discipline in mathematics, starting in the end of the 19th Century, and it was meant to serve as a foundation of mathematics.

Due to the foundational crisis that shook classical mathematics at the same time as the birth of set theory, some mathematicians started to search for a safer ground for mathematics to be built upon. So alongside the rise of set theory a new school of mathematics took form, now known as *constructive mathematics*. Many mathematicians remained sceptical though – once this safer constructive ground was established, how much of classical mathematics could actually be rebuilt on it?[1]

Not too much, it seemed, at least not until 1967, when Bishop published his book *Foundations of Constructive Analysis*[2]. Here Bishop answered the sceptics by showing how one actually could develop constructive mathematics in the field of mathematical analysis. Bishop's book was the start of a renewed interest in constructive mathematics and a call for continuing the task of reformulating all the fields of mathematics. One field left to rebuild constructively was set theory. Myhill presented, in 1975, a constructive set theory that had the advantage of being very close to classical set theory, but it lacked a clear explanation of the constructive notion of set.[3]

In this paper we will consider a constructive set theory, called CZF, formulated by Peter Aczel. Aczel introduced it in a series of three papers where he showed that CZF has an interpretation in Martin-Löf's type theory, thus giving CZF a constructive meaning.[4] To review this interpretation will be the main task of this master's thesis.

It might help if the reader has some previous knowledge of first-order classical and intuitionistic logic as well as of classical set theory, otherwise the presentation is meant to be self-contained.

A note on constructive mathematics
Constructive mathematics is a very broad and heterogeneous field. Here are two features of constructive mathematics that will be used in this paper.

Intuitionism
According to intuitionism a statement cannot be said to be true or false independently of our knowledge concerning the statement. A statement is true if there is a proof for it and a statement is false if the assumption that the statement has a proof leads to a contradiction. Therefore we cannot state for an arbitrary statement that it is either true or false. The law of the excluded third is therefore not valid in intuitionism.[5]

---

[1] Troelstra and van Dalen 1988.
[2] Bishop 1967
[3] Myhill 1975.
[4] Aczel 1978, Aczel 1982, Aczel 1986.
[5] Troelstra and van Dalen, 1988, vol. I, p. 4.

Predicativity
Definitions should be predicative, i.e. it is not permissible to define an object d by referring to a collection D of which the object d is to be an element. This means that quantification over D in defining d is not permitted.[6]

Disposition
In Chapter 1 we present the axioms of classical set theory and briefly discuss them from a constructive point of view. Then we present the axioms of constructive set theory as formulated by Aczel. The justification of the axioms will partly come later when we give a constructive model for Aczel's set theory.

In Chapter 2 and 3 we present the type theory of Martin-Löf, in which we will give the model for Aczel's set theory. Chapter 4 proves that there is, in type theory, a model for intuitionistic predicate logic. Then, in Chapter 5, we prove some results in type theory that will be used later in the interpretation of CZF.

Chapter 6 introduces the type of sets in type theory. And in Chapter 7 we construct a model for Aczel's set theory. This is the justification of the axioms of CZF.

[6] Troelstra and van Dalen, 1988, vol. I, p. 2.

1 Set Theories

Classical set theory
Classical Zermelo-Fraenkel axiomatic set theory, ZF, is formulated in first order classical logic. The binary predicates $\in$ and $=$ are the only non-logical symbols. The axioms of ZF are pairing, union, powerset, infinity, extensionality, foundation, separation and replacement.

There are several axioms in ZF that are problematic from a constructive point of view. In order to obtain the constructive set theory CZF we make the following changes:

1. Use intuitionistic logic instead of classical logic.
2. Use the subset collection scheme instead of the power set axiom.
The power set axiom is impredicative.
3. Use the set induction scheme instead of the foundation axiom.
The foundation axiom implies the law of the excluded third. Set induction scheme is a contrapositive of the foundation axiom.[7]
4. Use the restricted separation scheme instead of the separation scheme.
The separation scheme is impredicative.
5. Use the strong collection scheme instead of the replacement scheme.
The strong collection scheme implies the replacement scheme.[8] The replacement scheme is not non-constructive but we need the strong collection scheme when using restricted separation.

So the set theoretical axioms of CZF are pairing, union, infinity, extensionality, set induction, restricted separation, subset collection and strong collection. Now let's formulate it explicitly.

The axiom system CZF
The language, L, of CZF is the standard first-order language for set theory having $\in$ and $=$ as its only non-logical symbols. The system is based on intuitionistic first-order logic with the logical operators $\bot$, $\wedge$, $\vee$, $\rightarrow$, $\forall$, $\exists$, $(\forall x \in y)$ and $(\exists x \in y)$. As usual $\neg F$ and $F \leftrightarrow P$ will abbreviate $F \rightarrow \bot$ and
$(F \rightarrow P) \wedge (P \rightarrow F)$ respectively.

Definition
A formula is *restricted* if all its quantifiers are restricted, i.e. it has been built up only using $\bot$, $\wedge$, $\vee$, $\rightarrow$, $\in$, $=$, $(\forall x \in y)$ and $(\exists x \in y)$.

CZF is axiomatized using a standard axiomatization of intuitionistic predicate logic. The remaining axioms are as follows:

Restricted quantifiers
$(\forall x \in y)F(x) \leftrightarrow \forall x(x \in y \rightarrow F(x))$
$(\exists x \in y)F(x) \leftrightarrow \exists x(x \in y \wedge F(x))$
for every formula F(x) of L.

---

[7] See Troelstra and van Dalen, 1988, vol. II, p. 622 for a proof.
[8] See Troelstra and van Dalen, 1988, vol. II, p. 622 for a proof.

Extensionality
i) $\forall x \forall y \forall z((x = y \wedge y \in z) \rightarrow x \in z)$
ii) $\forall x \forall y(x = y \leftrightarrow \forall z(z \in x \leftrightarrow z \in y))$

Set induction
$\forall x((\forall y \in x)F(y) \rightarrow F(x)) \rightarrow \forall x F(x)$
for all formulae F(x) of L.

Pairing
$\forall x \forall y \exists z \forall u(u \in z \leftrightarrow u = x \vee u = y)$

Union
$\forall x \exists z \forall u(u \in z \leftrightarrow (\exists y \in x)(u \in y))$

Restricted separation
$\forall x \exists z \forall y(y \in z \leftrightarrow y \in x \wedge F(y))$
for all restricted formulae F(y) of L.

Strong collection
$\forall u((\forall x \in u)\exists y F(x,y) \rightarrow \exists v F'(u,v))$
for all formulae F(x,y) of L, where F'(u,v) abbreviates
$(\forall x \in u)(\exists y \in v)F(x,y) \wedge (\forall y \in v)(\exists x \in u)F(x,y)$.

Subset collection
$\forall v \forall w \exists t \forall u((\forall x \in v)(\exists y \in w)F(x,y) \rightarrow (\exists z \in t)F'(v,z))$
for all formulae F(x,y) of L.

Infinity
$\exists z((\exists x \in z)(\forall y \in x)\bot \wedge (\forall x \in z)(\exists y \in z)suc(x,y))$
where suc(x,y) is
$x \in y \wedge (\forall u \in x)(u \in y) \wedge (\forall u \in y)(u \in x \vee u = x)$

Remarks
CZF with classical logic has the same theorems as ZF.[9]

Many aspects of the informal development of classical set theory still apply when working informally in CZF. For example, natural numbers, ordered pairs, relations and functions can be defined just as in classical set theory.[10]

Now we will turn to Martin-Löf's type theory in order to find a constructive justification of the axioms of CZF.

[9] A proof of this is to be found in Aczel 1978, or in Troelstra and van Dalen, 1988, vol. II, p. 624.
[10] To see how this is developed see Aczel and Rathjen, 2000/2001.

2 Introducing type theory

Martin-Löf's type theory (from now on referred to as 'type theory') is meant to give us a theory that formulates a constructive foundation of mathematics. Unlike most other formalizations of mathematics, type theory is not based on first-order predicate logic. Instead, predicate logic is interpreted within type theory through the correspondence of propositions and types. A proposition is interpreted as a type whose elements represent the proofs of the proposition. Hence, a false proposition is interpreted as an empty type and a true proposition as a non-empty type. This will be further discussed in Chapter 3.

We will restrict the presentation of type theory to what is needed for the interpretation of CZF.[11]

Language and meaning

Judgements in type theory
The basic units of type theory are judgements, possibly depending on certain assumptions. There are four fundamental judgements in type theory

$A \in \text{type}$	A is a type
$A = B$	A and B are equal types
$a \in A$	a is an element of the type A
$a = b \in A$	a and b are equal elements of the type A

Note that the judgement $a \in A$ is only meaningful if A is a type, i.e. it presupposes the judgement $A \in \text{type}$. The judgement $a = b \in A$ presupposes the judgements $a \in A$ and $b \in A$. And $A = B$ presupposes $A \in \text{type}$ and $B \in \text{type}$.

The semantics of type theory will explain what these judgements mean.

Contexts
Each judgement has a finite, possibly empty, sequence of assumptions, called *context*, of the form

$$x_1 \in A_1, x_2 \in A_2(x_1), \ldots, x_n \in A_n(x_1, \ldots, x_{n-1})$$

Judgements with an empty context are called *categorical judgements* and judgements with a non-empty context are called *hypothetical judgements*. We will write the context in parentheses after the judgement. The empty context () will be left out.

Propositions and judgements
Since we have judgements as basic units and not propositions, the distinction between proposition and judgement becomes essential. What we combine by means of logical operators and hold to be true are propositions. When we hold a proposition to be true, we make a judgement. The premises and conclusion of a logical inference are judgements.

[11] We mainly follow the presentation of type theory in Martin-Löf 1984 and Nordström, Petterson and Smith 1990, see also Ranta 1994.
.

The semantics of the judgement forms
The semantic explanation of a judgement is given by stating the conditions under which the judgement is justified.

The meaning of $A \in$ type
To know that A is a type is to know how the canonical elements of A are formed as well as how two equal canonical elements of A are formed.

The equality relation between canonical elements must always be defined in such a way as to be reflexive, symmetric and transitive.

The meaning of $a \in A$
Suppose we know A to be a type. To know that a is an element of a type A is to know that a is a method which, when executed, yields a canonical element of A as a result.

We assume the notion of method to be primitive.

The meaning of $a = b \in A$
To know that two arbitrary elements a, b of a type A are equal is to know that when executed, a and b yield equal canonical elements of A as results.

The meaning of $A = B$
Suppose we know A and B to be types. To know that two types A and B are equal is to know that a canonical element of A is also a canonical element of B and vice versa, and further more, equal canonical elements of A are also equal canonical elements of B and vice versa.

From this we get the semantics of the categorical judgements in type theory.

Hypothetical judgements with one assumption
The first form of judgement is generalized to the hypothetical form

$B(x) \in \text{type}\ (x \in A)$

which says that B(x) is a type under the assumption $x \in A$. The meaning of the judgement is that B(a) is a type whenever a is an element of A. And also that B(a) and B(c) are equal types whenever a and c are equal elements of A. We will call B(x) *a family of types over A*.

The second form of judgement is generalized to the hypothetical form

$f(x) \in B(x)\ (x \in A)$

which says that f(x) is an element of the type B(x) under the assumption $x \in A$. The meaning of the judgement is that whatever element a of A is substituted for x in f(x), an element f(a) of B(a) results. And that substitutions of equal elements a and b of A result in equal elements of B(a).

The hypothetical judgements $f(x) = g(x) \in B(x)\ (x \in A)$ and $B(x) = D(x)\ (x \in A)$ are treated in a similar way.

Hypothetical judgements with more than one assumption
Hypothetical judgements with more than one assumption are defined by means of induction.[12]

From this we get the semantics of the judgements in type theory.

General rules
The rules of type theory are given of form

$$\frac{J_1 \ \ldots \ J_n}{J}$$

where $J_1, \ldots, J_n$, J are all judgements. In stating the rules we suppress mention of a context that is common to both the premises and the conclusion of the rule.

A rule of inference is justified by explaining the conclusion on the assumption that the premises are known.

First we have some general rules concerning equality and substitution. These rules are justified by the semantics of type theory.

Rules of equality
The rules of reflexivity, symmetry and transitivity are valid for all elements and types, since they are valid for canonical ones, and each non-canonical element can be computed into canonical form. Furthermore the meanings of $A = B$, $a \in A$ and $a = b \in A$ justify the rules of *type equality*.

$$\frac{a \in A \qquad A = B}{a \in B} \quad \text{type eq 1}$$

$$\frac{a = b \in A \qquad A = B}{a = b \in B} \quad \text{type eq 2}$$

Substitution rules
The meanings of the four hypothetical judgements yield four groups of substitution rules.

The first hypothetical judgement $B(x) \in \text{type}$ $(x \in A)$ yields the rules of *substitution in types*.

$$\frac{a \in A \qquad \begin{array}{c}(x \in A)\\ B(x) \in \text{type}\end{array}}{B(a) \in \text{type}} \quad \text{sub type 1}$$

$$\frac{a = b \in A \qquad \begin{array}{c}(x \in A)\\ B(x) \in \text{type}\end{array}}{B(a) = B(b)} \quad \text{sub type 2}$$

In the rules, the hypotheses are written above the corresponding judgements. The second hypothetical judgement $f(x) \in B(x)$ $(x \in A)$ yields the rules of *substitution in elements*.

[12] See Nordström, Petterson and Smith 1990 pp. 29-33 for a detailed account.

$$\frac{\begin{array}{l}(x \in A)\\ f(x) \in B(x) \quad a \in A\end{array}}{f(a) \in B(a)} \quad \text{sub el 1}$$

$$\frac{\begin{array}{l}(x \in A)\\ f(x) \in B(x) \quad a = b \in A\end{array}}{f(a) = f(b) \in B(a)} \quad \text{sub el 2}$$

The third and fourth hypothetical judgments yield the substitution rules, *substitution in equal elements* and *substitution in equal types*, in a similar way. All substitution rules are generalized to the case of more than one assumption.

3 Operators and their rules

For each operator we use in type theory we have four rules: formation, introduction, elimination and equality rules.

The *formation rules* say that we can form a certain type from certain other types or families of types.

The *introduction rules* say what are the canonical elements and equal canonical elements of the type, thus giving its meaning. The *constructors* by which canonical elements are formed are introduced in these rules.

The *elimination rules* are a kind of induction rules. The *selectors*, which are introduced in these rules, make it possible to do recursion and define functions on the type defined by the introduction rules.

The *equality rules* relate the introduction and elimination rules by showing how the selector operates on the canonical elements.

To each rule of formation, introduction and elimination there corresponds an additional equality rule, which we will leave implicit. What it means is that it allows us to substitute equals for equals. E.g. the equality rules corresponding to the introduction rules say that the operator produces equal elements from equal arguments and only from equal arguments.

Cartesian product of a family of types

The operator Π that forms the *cartesian product of a family of types*, generalizes the usual notion of functions from A to B by allowing B to depend on an element of A.

$$\frac{A \in \text{type} \qquad \begin{matrix}(x \in A)\\ B(x) \in \text{type}\end{matrix}}{(\Pi x \in A)B(x) \in \text{type}} \quad \Pi\text{-formation}$$

We introduce the constructor λ abstraction which is used to form the elements of the function type.

$$\frac{\begin{matrix}(x \in A)\\ b(x) \in B(x)\end{matrix}}{(\lambda x)b(x) \in (\Pi x \in A)B(x)} \quad \Pi\text{-introduction}$$

The selector ap is used for *applying* an element of $(\Pi x \in A)B(x)$ to an argument $a \in A$. The application of $(\lambda x)b(x)$ to an argument a is computed by substituting a for x in b(x).

$$\frac{c \in (\Pi x \in A)B(x) \qquad a \in A}{ap(c,a) \in B(a)} \quad \Pi\text{-elimination}$$

$$\frac{\begin{matrix}(x \in A)\\ b(x) \in B(x)\end{matrix} \quad a \in A}{ap((\lambda x)b(x),a) = b(a) \in B(a)} \quad \Pi\text{-equality}$$

If $B(x) = B$ does not depend on x, then $(\Pi x \in A)B(x)$ becomes the type of functions between A and B, denoted $A \to B$. We get the rules for → by replacing $(\Pi x \in A)B(x)$ with $A \to B$ in the Π rules.

Disjoint union of a family of types
The operator Σ that forms the *disjoint union of a family of types* generalizes the cartesian product of two types A and B by allowing B to depend on a variable $x \in A$.

$$\frac{A \in \text{type} \qquad \begin{array}{c}(x \in A)\\ B(x) \in \text{type}\end{array}}{(\Sigma x \in A)B(x) \in \text{type}}\ \Sigma\text{-formation}$$

Σ has one constructor, the pairing operator. The canonical elements of $(\Sigma x \in A)B(x)$ will be of the form (a,b) where $a \in A$ and $b \in B(a)$.

$$\frac{a \in A \qquad b \in B(a)}{(a,b) \in (\Sigma x \in A)B(x)}\ \Sigma\text{-introduction}$$

$$\frac{c \in (\Sigma x \in A)B(x) \qquad \begin{array}{c}(x \in A,\ y \in B(x))\\ d(x,y) \in C((x,y))\end{array}}{E(c,d) \in C(c)}\ \Sigma\text{-elimination}$$

$$\frac{a \in A \qquad b \in B(a) \qquad \begin{array}{c}(x \in A,\ y \in B(x))\\ d(x,y) \in C((x,y))\end{array}}{E((a,b),d) = d(a,b) \in C((a,b))}\ \Sigma\text{-equality}$$

There are two special cases of E that give us the selectors p and q, the left and right projections. Let $p(c) = E(c,(x,y)x) \in A$ and $q(c) = E(c,(x,y)y) \in B(p(c))$, then

$$\frac{c \in (\Sigma x \in A)B(x)}{p(c) \in A}\ \Sigma\text{-elimination 1}$$

$$\frac{c \in (\Sigma x \in A)B(x)}{q(c) \in B(p(c))}\ \Sigma\text{-elimination 2}$$

$$\frac{a \in A \qquad b \in B(a)}{p((a,b)) = a \in A}\ \Sigma\text{-equality 1}$$

$$\frac{a \in A \qquad b \in B(a)}{q((a,b)) = b \in B(a)}\ \Sigma\text{-equality 2}$$

If $B(x) = B$ does not depend on x, then $(\Sigma x \in A)B(x)$ becomes the cartesian product of two types, denoted $A \times B$. We get the rules of × by replacing $(\Sigma x \in A)B(x)$ with $A \times B$ in the Σ rules.

Disjoint union of two sets
The canonical elements of the *disjoint union of two types*, A + B, are canonical injections of the elements of the constituent types. The elimination rule says that a function can be defined on A + B if it can be defined separately on each of the types A and B.

$$\frac{A \in \mathrm{type} \qquad B \in \mathrm{type}}{A + B \in \mathrm{type}} \quad +\text{-formation}$$

$$\frac{a \in A}{i(a) \in A + B} \quad +\text{-introduction 1}$$

$$\frac{b \in B}{j(b) \in A + B} \quad +\text{-introduction 2}$$

$$\frac{c \in A + B \qquad \begin{array}{c}(x \in A)\\ d(x) \in C(i(x))\end{array} \qquad \begin{array}{c}(y \in B)\\ e(y) \in C(j(y))\end{array}}{D(c,d,e) \in C(c)} \quad +\text{-elimination}$$

$$\frac{a \in A \qquad \begin{array}{c}(x \in A)\\ d(x) \in C(i(x))\end{array} \qquad \begin{array}{c}(y \in B)\\ e(y) \in C(j(y))\end{array}}{D(i(a),d,e) = d(a) \in C(i(a))} \quad +\text{-equality 1}$$

$$\frac{b \in B \qquad \begin{array}{c}(x \in A)\\ d(x) \in C(i(x))\end{array} \qquad \begin{array}{c}(y \in B)\\ e(y) \in C(j(y))\end{array}}{D(j(b),d,e) = e(b) \in C(j(b))} \quad +\text{-equality 2}$$

Identity types

Since the judgemental equality $a = b \in A$ cannot be used for constructing new types, we need an identity type that expresses that two elements are equal.

$$\frac{A \in \mathrm{type} \qquad a \in A \qquad b \in A}{I(A,a,b) \in \mathrm{type}} \quad \text{I-formation}$$

The type $I(A,a,a)$ will have the canonical element $r(a)$.

$$\frac{a \in A}{r(a) \in I(A,a,a)} \quad \text{I-introduction}$$

$$\frac{c \in I(A,a,b) \qquad \begin{array}{c}(x \in A)\\ d(x) \in C(x,x,r(x))\end{array}}{J(c,d) \in C(a,b,c)} \quad \text{I-elimination}$$

$$\frac{a \in A}{J(r(a),d) = d(a) \in C(a,a,r(a))} \quad \text{I-equality}$$

Finite types

We introduce the enumerated types N_0, N_1, N_2.[13] The finite types N_k have the canonical elements $0_k,\ldots,(k-1)_k$ for $k = 0,1,2$. The formation rules have no premises.

[13] We leave the rest of the N_k's out since we do not need them in this paper. See e.g. Nordström 1990, p. 41-42, for the general case.

$N_0 \in$ type $\quad$ N_0-formation

$N_1 \in$ type $\quad$ N_1-formation

$N_2 \in$ type $\quad$ N_2-formation

There is an introduction rule without premises for each of the enumerated constants $0_k,\ldots,(k-1)_k$. So N_0 has no element, N_1 the single element 0_1 and N_2 the elements 0_2 and 1_2.

$0_1 \in N_1$ $\quad$ N_1-introduction

$0_2 \in N_2,\ 1_2 \in N_2$ $\quad$ N_2-introduction

The elimination rule has the major premise $c \in N_k$, and a minor premise for each of the elements $0_k,\ldots,(k-1)_k$.

$$\frac{c \in N_0}{R_0(c) \in C(c)} \quad N_0\text{-elimination}$$

$$\frac{c \in N_1 \qquad c_1 \in C(0_1)}{R_1(c,c_1) \in C(c)} \quad N_1\text{-elimination}$$

$$\frac{c \in N_2 \qquad c_1 \in C(0_2) \qquad c_2 \in C(1_2)}{R_2(c,c_1,c_2) \in C(c)} \quad N_2\text{-elimination}$$

Thus there is an equality rule for each of the elements $0_k,\ldots,(k-1)_k$.

$$\frac{c_1 \in C(0_1)}{R_1(0_1,c_1) = c_1 \in C(0_1)} \quad N_1\text{-equality}$$

and

$$\frac{c_1 \in C(0_2) \qquad c_2 \in C(1_2)}{R_2(0_2,c_1,c_2) = c_1 \in C(0_2)} \quad N_2\text{-equality1}$$

$$\frac{c_1 \in C(0_2) \qquad c_2 \in C(1_2)}{R_2(1_2,c_1,c_2) = c_2 \in C(1_2)} \quad N_2\text{-equality 2}$$

Natural numbers

So far we have no means of constructing infinite types. Now we introduce N, the type of natural numbers, by the following formation rule.

$N \in$ type $\quad$ N-formation

The introduction rules for N say that canonical natural numbers are 0 and those of the successor form s(a).

$$0 \in N \qquad \text{N-introduction 1}$$

$$\frac{a \in N}{s(a) \in N} \qquad \text{N-introduction 2}$$

The elimination rule for N can be seen both as giving a proof of the proposition C(c), for an arbitrary c ∈ N, by induction, and as a rule for defining a function on N by recursion.

$$\frac{c \in N \qquad d \in C(0) \qquad \begin{array}{c}(x \in N,\ y \in C(x))\\ e(x,y) \in C(s(x))\end{array}}{R(c,d,e) \in C(c)} \qquad \text{N-elimination}$$

Finally we have the equality rules for N.

$$\frac{d \in C(0) \qquad \begin{array}{c}(x \in N,\ y \in C(x))\\ e(x,y) \in C(s(x))\end{array}}{R(0,d,e) = d \in C(0)} \qquad \text{N-equality 1}$$

$$\frac{a \in N \quad d \in C(0) \qquad \begin{array}{c}(x \in N,\ y \in C(x))\\ e(x,y) \in C(s(x))\end{array}}{R(s(a),d,e) = e(a,R(a,d,e)) \in C(s(a))} \qquad \text{N-equality 2}$$

N-induction
By suppressing proofs we get

$$\frac{c \in N \qquad C(0)\ \text{true} \qquad \begin{array}{c}(x \in N,\ C(x)\ \text{true})\\ C(s(x))\ \text{true}\end{array}}{C(c)\ \text{true}} \qquad \text{N-induction}$$

The following statement can be proved by N-induction, one ∧ introduction, two ∀ introductions and two → introductions (see Chapter 4 for ∧, ∀ and → rules):
$C(0) \wedge ((\forall x \in N)C(x) \to C(s(x))) \to (\forall x \in N)C(x)$

Well-orderings
We will now introduce the well-ordering type (Wx ∈ A)B(x), whose elements are well-founded trees. To form a canonical element of (Wx ∈ A)B(x), we must say which way the tree is formed and what the parts are. If we have an element a ∈ A, i.e. if we have a particular form we want the tree to have, and if we have a function from B(a) to (Wx ∈ A)B(x), i.e. if we have a collection of subtrees, then we may form the tree sup(a,b).

$$\frac{A \in \text{type} \qquad \begin{array}{c}(x \in A)\\ B(x) \in \text{type}\end{array}}{(Wx \in A)B(x) \in \text{type}} \qquad \text{W-formation}$$

$$\frac{a \in A \qquad b \in B(a) \to (Wx \in A)B(x)}{\sup(a,b) \in (Wx \in A)B(x)} \qquad \text{W-introduction}$$

$$\frac{c \in (Wx \in A)B(x) \qquad \begin{matrix}(x \in A,\ y \in B(x) \rightarrow (Wx \in A)B(x),\ z \in (\Pi v \in B(x))C(ap(y,v)))\\ d(x,y,z) \in C(sup(x,y))\end{matrix}}{T(c,d) \in C(c)} \quad \text{W-elimination}$$

$$\frac{a \in A \qquad b \in B(a) \rightarrow (Wx \in A)B(x) \qquad \begin{matrix}(x \in A,\ y \in B(x) \rightarrow (Wx \in A)B(x),\ z \in (\Pi v \in B(x))C(ap(y,v)))\\ d(x,y,z) \in C(sup(x,y))\end{matrix}}{T(sup(a,b),d) = d(a,b,(\lambda v)T(ap(b,v),d)) \in C(sup(a,b))} \quad \text{W-equality}$$

Universes

Now we arrive at types of types. So let us define a universe as the least type closed under certain specified type forming operations. So far we have constructed types from N_0, N_1, N_2, N by means of the operations Π, Σ, +, I(A,b,c) and W. We now consider the universe U of all types generated from N_0, N_1, N_2, N with closure with respect to Π, Σ, +, I(A,b,c) and W.

$U \in \text{type}$ U-formation

$$\frac{A \in U}{A \in \text{type}} \quad \text{U-formation}$$

$$\frac{A \in U \qquad \begin{matrix}(x \in A)\\ B(x) \in U\end{matrix}}{(\Pi x \in A)B(x) \in U} \quad U_\Pi\text{-introduction}$$

$$\frac{A \in U \qquad \begin{matrix}(x \in A)\\ B(x) \in U\end{matrix}}{(\Sigma x \in A)B(x) \in U} \quad U_\Sigma\text{-introduction}$$

$$\frac{A \in U \qquad B \in U}{A + B \in U} \quad U_+\text{-introduction}$$

$$\frac{A \in U \qquad b,c \in A}{I(A,b,c) \in U} \quad U_I\text{-introduction}$$

$N_0 \in U$ $N_1 \in U$ $N_2 \in U$ $N \in U$ U_N-introduction

$$\frac{A \in U \qquad \begin{matrix}(x \in A)\\ B(x) \in U\end{matrix}}{(Wx \in A)B(x) \in U} \quad U_W\text{-introduction}$$

U itself is not an element of U. We say that a type A is *small* if A ∈ U. Now U is closed under Π, Σ, +, I(A,b,c), N_0,…, N. U is a type but it is not small.

The type theory presented in this chapter is enough for giving a model of CZF. There is a second level of type theory, which is not presented here since it is not needed for our model, see e.g. Nordström 1990 for a presentation.

4 Propositions as types

We now start with the interpretation of the axioms of CZF. The first thing we have to show is that intuitionistic predicate logic can be interpreted in type theory. In order to prove this, we will return to the relationship between propositions and types.

The intuitionistic notion of proposition
Intuitionistically, a proposition is defined by laying down what counts as a proof of the proposition and a proposition is true if it has a proof, i.e. if a proof of it can be given. Thus truth is identified with provability. The proofs of complex propositions are defined as certain complexes of proofs of their constituent propositions.

The propositions-as-types principle
Since types are defined by prescribing how its canonical elements are formed and propositions are defined by laying down how its canonical proofs are formed, we identify types and propositions, i.e. we treat them as one and the same notion. This is the propositions-as-types interpretation on which type theory is based.[14]

So, propositions are types and proofs are elements. That a proposition is true means that the type has an element. Now we can reread the judgements of type theory in the following way:

$A \in$ type	A is a type	A is a proposition (abbreviated $A \in$ prop)
$a \in A$	a is an element of A	a is a proof of the proposition A
A true	A has an element	A is true

Making no distinction between propositions and types we can use $A \in$ type and $A \in$ prop interchangeably. A true is an abbreviation of $a \in A$ obtained by suppressing the proof a.

Logical operators
The next step is to look at the logical operators. Here is the explanation of the logical operators according to Brouwer-Heyting-Kolmogorov.

proposition	is proved by
$\perp$	---
$A \wedge B$	a proof of A and a proof of B
$A \vee B$	a proof of A or a proof of B
$A \rightarrow B$	a method for obtaining a proof of B from any proof of A
$\neg A$	a method for obtaining a proof of $\perp$ from any proof of A
$(\forall x \in A)B(x)$	a method for obtaining a proof of B(a) from any $a \in A$
$(\exists x \in A)B(x)$	an element $a \in A$ and a proof of B(a)

Using the language of type theory we can rewrite the list above as follows:

proposition	is proved by
$\perp$	---
$A \wedge B$	a pair (a,b) where $a \in A$ and $b \in B$
$A \vee B$	a canonical injection i(a) where $a \in A$ or j(b) where $b \in B$

[14] See Howard 1969.

$A \rightarrow B$	a lambda abstract $(\lambda x)b(x)$ where $b(x) \in B$ $(x \in A)$
$\neg A$	a lambda abstract $(\lambda x)b(x)$ where $b(x) \in \perp$ $(x \in A)$
$(\forall x \in A)B(x)$	a lambda abstract $(\lambda x)b(x)$ where $b(x) \in B(x)$ $(x \in A)$
$(\exists x \in A)B(x)$	a pair (a,b) where $a \in A$ and $b \in B(a)$

Applying the proposition-as-types principle we get the following list of definitions:

proposition	is defined by the type
$\perp$	N_0
$A \wedge B$	$(\Sigma x \in A)B$, where B does not depend on x, i.e. $A \times B$
$A \vee B$	$A + B$
$A \rightarrow B$	$(\Pi x \in A)B$, where B does not depend on x, i.e. $A \rightarrow B$
$\neg A$	$(\Pi x \in A)N_0$, i.e. $A \rightarrow N_0$
$(\forall x \in A)B(x)$	$(\Pi x \in A)B(x)$
$(\exists x \in A)B(x)$	$(\Sigma x \in A)B(x)$

We can hereby use terminology from the two sides of the list interchangeably.

Now we are ready to prove the following theorem:

Fundamental theorem
For each natural-deduction derivation in intuitionistic predicate calculus, there is a corresponding proof in type theory that is obtained by filling in the proof objects.

Proof
The rules of intuitionistic predicate calculus of the form of Gentzen's natural deduction are obtained by suppressing proofs in the type theoretical rules.

Universal quantification
The $\forall$ rules are obtained from the Π rules.

$$\frac{A \in \text{prop} \qquad \begin{matrix}(x \in A)\\ B(x) \in \text{prop}\end{matrix}}{(\forall x \in A)B(x) \in \text{prop}} \ \forall\text{-formation}$$

$$\frac{\begin{matrix}(x \in A)\\ B(x)\ \text{true}\end{matrix}}{(\forall x \in A)B(x)\ \text{true}} \ \forall\text{-introduction}$$

$$\frac{(\forall x \in A)B(x)\ \text{true} \qquad a \in A}{B(a)\ \text{true}} \ \forall\text{-elimination}$$

Implication
The $\rightarrow$ rules are also obtained from the Π rules.

$$\frac{A \in \text{prop} \qquad B \in \text{prop}}{A \rightarrow B \in \text{prop}} \ \rightarrow\text{-formation}$$

$$\frac{\begin{matrix}(A \text{ true}) \\ B \text{ true}\end{matrix}}{A \to B \text{ true}} \quad \to\text{-introduction}$$

$$\frac{A \to B \text{ true} \quad A \text{ true}}{B \text{ true}} \quad \to\text{-elimination}$$

Existential quantification

The ∃ rules are obtained from the Σ rules.

$$\frac{A \in \text{type} \qquad \begin{matrix}(x \in A) \\ B(x) \in \text{prop}\end{matrix}}{(\exists x \in A)B(x) \in \text{prop}} \quad \exists\text{-formation}$$

$$\frac{a \in A \qquad B(a) \text{ true}}{(\exists x \in A)B(x) \text{ true}} \quad \exists\text{-introduction}$$

$$\frac{(\exists x \in A)B(x) \text{ true} \qquad \begin{matrix}(x \in A, B(x) \text{ true}) \\ C \text{ true}\end{matrix}}{C \text{ true}} \quad \exists\text{-elimination}$$

Conjunction

The ∧ rules are also obtained from the Σ rules, (using the rules Σ-elimination 1 and Σ-elimination 2).

$$\frac{A \in \text{prop} \qquad B \in \text{prop}}{A \wedge B \in \text{prop}} \quad \wedge\text{-formation}$$

$$\frac{A \text{ true} \qquad B \text{ true}}{A \wedge B \text{ true}} \quad \wedge\text{-introduction}$$

$$\frac{A \wedge B \text{ true}}{A \text{ true}} \quad \wedge\text{-elimination 1}$$

$$\frac{A \wedge B \text{ true}}{B \text{ true}} \quad \wedge\text{-elimination 2}$$

Disjunction

The ∨ rules are obtained from the + rules.

$$\frac{A \in \text{prop} \qquad B \in \text{prop}}{A \vee B \in \text{prop}} \quad \vee\text{-formation}$$

$$\frac{A \text{ true}}{A \vee B \text{ true}} \quad \vee\text{-introduction 1}$$

$$\frac{B \text{ true}}{A \vee B \text{ true}} \quad \vee\text{-introduction 2}$$

$$\frac{A \vee B \text{ true} \quad \begin{array}{c}(A \text{ true})\\ C \text{ true}\end{array} \quad \begin{array}{c}(B \text{ true})\\ C \text{ true}\end{array}}{C \text{ true}} \qquad \vee\text{-elimination}$$

Absurdity
The absurdity rules are obtained from the N_0 rules.

$\perp \in$ prop $\quad \perp$-formation

$$\frac{\perp \text{ true}}{C \text{ true}} \qquad \perp\text{-elimination}$$

This concludes the proof that type theory validates intuitionistic logic.

5 Some results

Here are some results in type theory that are needed in interpreting CZF.

The axiom of choice (AC)
The axiom of choice is provable in type theory, i.e.
$(\forall x \in A)(\exists y \in B(x))C(x,y) \rightarrow (\exists f \in (\Pi x \in A)B(x))(\forall x \in A)C(x,ap(f,x))$ true
where $A \in$ type, $B(x) \in$ type $(x \in A)$, $C(x,y) \in$ type $(x \in A,\ y \in B(x))$.

Proof
Let $A \in$ set, $B(x) \in$ set $(x \in A)$, $C(x,y) \in$ set $(x \in A,\ y \in B(x))$ and assume
$z \in (\Pi x \in A)(\Sigma y \in B(x))C(x,y)$. Let $x \in A$. Then

$$
\begin{array}{lll}
 & & \underline{\overset{2.}{z \in (\Pi x \in A)(\Sigma y \in B(x))C(x,y)} \quad \overset{1.}{x \in A}} \\
\Pi\text{-elimination} & & \underline{ap(z,x) \in (\Sigma y \in B(x))C(x,y)} \\
\Sigma\text{-elimination 1} & & \underline{p(ap(z,x)) \in B(x)} \qquad\qquad\qquad (x \in A, y \in B(x)) \\
\Pi\text{-equality, 1.} & & \underline{ap((\lambda x)p(ap(z,x)),x) = p(ap(z,x)) \in B(x) \qquad C(x,y) \in \text{set}} \\
\text{sub type 2} & \text{i)} & C(x,ap((\lambda x)p(ap(z,x)),x)) = C(x,p(ap(z,x)))
\end{array}
$$

$$
\begin{array}{lll}
 & & \underline{\overset{2.}{z \in (\Pi x \in A)(\Sigma y \in B(x))C(x,y)} \quad \overset{1.}{x \in A}} \\
\Pi\text{-elimination} & & \underline{ap(z,x) \in (\Sigma y \in B(x))C(x,y)} \\
\Sigma\text{-elimination 2} & \text{ii)} & q(ap(z,x)) \in C(x,p(ap(z,x)))
\end{array}
$$

Now combining i) and ii)

$$
\begin{array}{ll}
 & \underline{q(ap(z,x)) \in C(x,p(ap(z,x))) \quad C(x,ap((\lambda x)p(ap(z,x)),x)) = C(x,p(ap(z,x)))} \\
\text{type eq 1} & \underline{q(ap(z,x)) \in C(x,ap((\lambda x)p(ap(z,x)),x))} \\
\Pi\text{-introduction, 1.} & \text{iii) } (\lambda x)q(ap(z,x)) \in (\Pi x \in A)C(x,ap((\lambda x)p(ap(z,x)),x))
\end{array}
$$

$$
\begin{array}{lll}
 & & \underline{\overset{2.}{z \in (\Pi x \in A)(\Sigma y \in B(x))C(x,y)} \quad \overset{1.}{x \in A}} \\
\Pi\text{-elimination} & & \underline{ap(z,x) \in (\Sigma y \in B(x))C(x,y)} \\
\Sigma\text{-elimination 1} & & \underline{p(ap(z,x)) \in B(x)} \\
\Pi\text{-introduction, 1.} & \text{iv)} & (\lambda x)p(ap(z,x)) \in (\Pi x \in A)B(x)
\end{array}
$$

Now combining iii) and iv)

$$
\begin{array}{l}
\underline{\text{iii)} \qquad\qquad \text{iv)}} \\
\Sigma\text{-introduction} \\
\underline{((\lambda x)p(ap(z,x)),(\lambda x)q(ap(z,x))) \in (\Sigma f \in (\Pi x \in A)B(x))(\Pi x \in A)C(x,ap(f,x))} \\
\rightarrow\text{-introduction, 2.} \\
(\Pi x \in A)(\Sigma y \in B(x))C(x,y) \rightarrow (\Sigma f \in (\Pi x \in A)B(x))(\Pi x \in A)C(x,ap(f,x)) \text{ true}
\end{array}
$$

Notation: We will from now on write ap(f,x) simply as f(x), since the distinction is not essential in the remainder of this paper.

Σ-, +-, N_0-, N_1- and N_2-existence

i) *Σ-existence.*

Let $B(x) \in$ type $(x \in A)$, then

$(\exists z \in (\Sigma x \in A)B(x))F(z) \leftrightarrow (\exists x \in A)(\exists y \in B(x))F((x,y))$

ii) *+-existence.*

Let $A \in$ type and $B \in$ type, then

$(\exists z \in A + B)F(z) \leftrightarrow (\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y))$

iii) *N_0-, N_1- and N_2-existence.*

$(\exists z \in N_0)F(z) \leftrightarrow \bot$

$(\exists z \in N_1)F(z) \leftrightarrow F(0_1)$

$(\exists z \in N_2)F(z) \leftrightarrow F(0_2) \vee F(1_2)$

Proof

The implication from right to left is given by the introduction rules and the opposite direction by the elimination rules.

i) Assume $(\exists x \in A)(\exists y \in B(x))F((x,y))$ true. Then, by $\exists$-elimination, we have $x \in A$, $y \in B(x)$ and $F((x,y))$ true. Σ-introduction gives us $(x,y) \in (\Sigma x \in A)B(x)$ and $F((x,y))$ true. Then $(\exists z \in (\Sigma x \in A)B(x))F(z)$ true.

Now let $z \in (\Sigma x \in A)B(x))$. We have $p(z) \in A$, and $q(z) \in B(p(z))$. If $F(p(z),q(z))$, then $(\exists x \in A)(\exists y \in B(x))F((x,y))$.

So $F(p(z),q(z)) \rightarrow (\exists x \in A)(\exists y \in B(x))F((x,y))$ true.

Σ-elimination gives us $F(z) \rightarrow (\exists x \in A)(\exists y \in B(x))F((x,y))$ true.

Hence $(\forall z \in (\Sigma x \in A)B(x))(F(z) \rightarrow (\exists x \in A)(\exists y \in B(x))F((x,y)))$ true.

Hence $(\exists z \in (\Sigma x \in A)B(x))F(z) \rightarrow (\exists x \in A)(\exists y \in B(x))F((x,y))$ true.

ii) Assume $(\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y))$ true.

$(\exists x \in A)F(i(x))$ true gives $a \in A$ and $F(i(a))$ true. +-introduction gives $i(a) \in A + B$.

Thus $(\exists z \in A + B)F(z)$ true.

Similarly $(\exists y \in B)F(j(y))$ true gives $b \in B$ and $F(j(b))$ true. +-introduction gives $j(b) \in A + B$.

Thus $(\exists z \in A + B)F(z)$ true.

Hence $(\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y)) \rightarrow (\exists z \in A + B)F(z)$ true.

Now if $a \in A$ and $F(i(a))$ true then $(\exists x \in A)F(i(x))$ true.

And then $(\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y))$ true.

Hence $F(i(x)) \rightarrow (\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y))$ true.

Similarly if $b \in B$ and $F(j(b))$ true then $(\exists y \in B)F(j(y))$ true.

And then $(\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y))$ true.

Hence $F(j(y)) \rightarrow (\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y))$ true.

Assume $z \in A + B$. Then +-elimination gives $F(z) \rightarrow (\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y))$ true.

So $(\forall z \in A + B)(F(z) \rightarrow (\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y)))$ true.

Hence $(\exists z \in A + B)F(z) \rightarrow (\exists x \in A)F(i(x)) \vee (\exists y \in B)F(j(y))$ true.

iii) $(\exists z \in N_0)F(z)$ is never true, thus equivalent to $\bot$.

Let $F(0_1)$ true. $0_1 \in N_1$ by N_1-introduction. $\exists$-introduction gives $(\exists z \in N_1)F(z)$ true.

Let $(\exists z \in N_1)F(z)$ true. $\exists$-elimination gives $F(0_1)$ true.

Let $F(0_2) \vee F(1_2)$ true. $0_2 \in N_2$, $1_2 \in N_2$ by N_2-introduction. Assume 0_2 and 1_2. $\exists$-introduction gives $(\exists z \in N_2)F(z)$ true and $(\exists z \in N_2)F(z)$ true. $\vee$-elimination gives $(\exists z \in N_2)F(z)$ true.

Let $(\exists z \in N_2)F(z)$ true. We have $F(0_2) \rightarrow F(0_2) \vee F(1_2)$ true and $F(1_2) \rightarrow F(0_2) \vee F(1_2)$ true.
Assume $z \in N_2$. N_2–elimination gives $F(z) \rightarrow F(0_2) \vee F(1_2)$ true.
Hence $(\forall z \in N_2)(F(z) \rightarrow F(0_2) \vee F(1_2)$ true.
Hence $(\exists z \in N_2)F(z) \rightarrow F(0_k) \vee F(1_2)$ true.

6 A type of sets

In order to give a constructive meaning to the set theoretical notions of CZF we will explain its interpretation in type theory.

The iterative notion of set – a constructive version
We will now discuss informally the motivation for the interpretation. The classical iterative notion of set has been used to explain the meaning of classical set theory. The idea is to seek a constructive version of the iterative notion of set. With 'iterative notion of set' we mean the notion that arises by iterating the notion 'set of' in order to get sets, sets of sets, sets of sets of sets, etc.

Assume that we had a notion of 'set of objects', that we could apply to an arbitrary domain of objects. Then the universe of iterative sets might be viewed as the domain of objects that is inductively constructed by the single rule

if A is a *set of* iterative sets then A is an iterative set.

We also need a suitable notion of 'set of objects' for an arbitrary given domain of objects. In general let us take a set of objects from some domain to be the result of combining into a whole the selection of those objects from the domain that are to be the elements of the set. The set may be written $\{a_i\}_i$ where the a_i's are understood to be the selected elements of the set. Sets are to be treated extensionally. Two sets $\{a_i\}_i$ and $\{b_j\}_j$ are extensionally equal if every a_i is equal to some b_j and every b_j is equal to some a_i. Notice that we have used a variable i to index the selections of the elements a_i of the set $\{a_i\}_i$. What can be the range of i? This needs consideration if we are to avoid circularity. It is no good to let i range over any set. An independent notion is needed. Now we turn to type theory and we consider the notion of type. So let i range over a type I and the set $\{a_i\}_i$ should be written more explicitly as $\{a_i\}_{i\in I}$. The iterative sets are now inductively constructed using the rule that for each type I

if a_i is an iterative set for $i \in I$ then $\{a_i\}_{i\in I}$ is an iterative set.

This seems to be acceptable as a rule of construction. But in order to use it in a type theoretical framework and hence give an interpretation of the set theoretical language, we need a type of iterative sets. If all types I are allowed in forming iterative sets, then the iterative sets themselves cannot be expected to form a type. So we need to restrict the type I. If I is required to be a small type in forming the iterative set, then we obtain a relativised notion of iterative set over the type U of small types, and we can have the type V of iterative sets over U. It is this type V that we use to give an interpretation of constructive set theory in Martin-Löf's type theory.

Equality between two sets $\{a_i\}_{i\in I}$ and $\{b_j\}_{j\in J}$, denoted by $\approx$, will be explained as: $(\forall i \in I)(\exists j \in J)(a_i \approx b_j) \wedge (\forall j \in J)(\exists i \in I)(a_i \approx b_j)$. That x is a member of a set $\{a_i\}_{i\in I}$, denoted by ε, will be explained as: $(\exists i \in I)(a_i \approx x)$. This is an informal explanation of $\approx$ and ε, the definitions will follow in Chapter 7.

The iterative type V
By considering the type of sets, defined by $V = (Wx \in U)x$, we will see that we obtain a constructive interpretation of CZF. This version is a constructive version of the classical conception of the cumulative hierarchy of sets, as explained informally in the previous section.

Definition of V
Let V = (Wx ∈ U)x.

Notation: We use Greek lower case letters $\alpha,\beta,\gamma,\delta,\eta$ for elements of V.

The rules for W-types give us the following V rules:

V ∈ type   V-formation

$$\frac{A \in U \qquad b \in A \to V}{\sup(A,b) \in V} \quad \text{V-introduction}$$

Notation: We shall also write (supx ∈ A)b(x) for sup(A,b).

$$\frac{\alpha \in V \qquad \begin{array}{c}(x \in U,\ y \in x \to V,\ z \in (\forall v \in x)C(y(v)))\\ d(x,y,z) \in C(\sup(x,y))\end{array}}{T(\alpha,d) \in C(\alpha)} \quad \text{V-elimination}$$

$$\frac{A \in U \qquad b \in A \to V \qquad \begin{array}{c}(x \in U,\ y \in x \to V,\ z \in (\forall v \in x)C(y(v)))\\ d(x,y,z) \in C(\sup(x,y))\end{array}}{T(\sup(A,b),d) = d(A,b,(\lambda v)T(b(v),d)) \in C(\sup(A,b))} \quad \text{V-equality}$$

V-induction
By supressing proofs in V-elimination we get.

$$\frac{\alpha \in V \qquad \begin{array}{c}(x \in U,\ y \in x \to V,\ (\forall v \in x)C(y(v))\ \text{true})\\ C(\sup(x,y))\ \text{true}\end{array}}{C(\alpha)\ \text{true}} \quad \text{V-induction}$$

The following statement can be proved by V-induction, two → introductions and three ∀ introductions:

$$(\forall x \in U)(\forall y \in x \to V)((\forall v \in x)C(y(v)) \to C(\sup(x,y))) \to (\forall x \in V)C(x)$$

An important property of the type V, proved in the following lemma, is that we can always for each element $\alpha \in V$ recover the branching index $-\alpha \in U$ and the corresponding mapping $\sim\alpha \in -\alpha \to V$.

Lemma 1
Let $\alpha \in V$. There are one-place functions $-$ and $\sim$ such that $-\alpha \in U$ and $\sim\alpha \in -\alpha \to V$. And for $\alpha = \sup(A,b)$ we have $-\sup(A,b) = A$ and $\sim\sup(A,b) = b$.

Proof
Let $\alpha \in V$, i.e. $\alpha \in (Wx \in U)x$. Let $-\alpha = T(\alpha,(x,y,z)x)$.
So $-\alpha \in U$. Now let $\sim\alpha = T(\alpha,(x,y,z)y)$ be defined by

$$\frac{\alpha \in V \qquad \begin{matrix}(x \in U,\ y \in x \to V)\\ y \in -\mathrm{sup}(x,y) \to V\end{matrix}}{T(\alpha,(x,y,z)y) \in -\alpha \to V}\qquad \text{V-elimination}$$

Then $\sim\alpha \in -\alpha \to V$. And V-equality gives

$$\frac{A \in U \qquad b \in A \to V \qquad \begin{matrix}(x \in U)\\ x \in U\end{matrix}}{T(\mathrm{sup}(A,b),(x,y,z)x) = A \in U}\qquad \text{V-equality}$$

$$\frac{A \in U \qquad b \in A \to V \qquad \begin{matrix}(x \in U,\ y \in x \to V)\\ y \in x \to V\end{matrix}}{T(\mathrm{sup}(A,b),(x,y,z)y) = b \in A \to V}\qquad \text{V-equality}$$

So $-\mathrm{sup}(A,b) = A$ and $\sim\mathrm{sup}(A,b) = b$.

Remark 1
We have $I(V,\mathrm{sup}(-\alpha,\sim\alpha),\alpha)$ true.

Proof
Let $\alpha = \mathrm{sup}(A,b)$ and $g = (\lambda\alpha)\mathrm{sup}(-\alpha,\sim\alpha) \in V \to V$. We have $g(\alpha) = \mathrm{sup}(-\alpha,\sim\alpha) = \mathrm{sup}(-\mathrm{sup}(A,b),\sim\mathrm{sup}(A,b)) = \mathrm{sup}(A,b) = \alpha$. So now we have $r(\mathrm{sup}(x,y)) \in I(V,g(\mathrm{sup}(x,y)),\mathrm{sup}(x,y))$ for $x \in U$ and $y \in A \to V$.

$$\frac{\alpha \in V \qquad \begin{matrix}(x \in U,\ y \in x \to V)\\ r(\mathrm{sup}(x,y)) \in I(V,g(\mathrm{sup}(x,y)),\mathrm{sup}(x,y))\end{matrix}}{T(\alpha,r) \in I(V,g(\alpha),\alpha)}\qquad \text{V-elimination}$$

Which gives us $I(V,\mathrm{sup}(-\alpha,\sim\alpha),\alpha)$ true.

Let us summarize the basic idea of the interpretation: $V = (Wx \in U)x$ is the domain of our model. Each canonical element of V is of the form $\mathrm{sup}(A,b)$ where $A \in U$ and $b(x) \in V$ for $x \in A$. That is, the elements of V are constructed inductively as families of sets indexed by the elements of a small type. Alternatively, the elements of V are well-founded trees where the successors of a node are always indexed by the elements of a small type. The membership relation corresponds to the successor relation on such trees.

For any element $\alpha \in V$, $\sim\alpha(x)$ for $x \in \alpha$ play the role of elements. The small types play the same role as the ordinals do in ZF.

7 A model

CZF has the language L with the relations = and $\in$. An intuitionistic model for CZF is a triple $\mathfrak{M} = (M,\approx,\varepsilon)$ where $\approx$ has the same number of arguments as =, ε has the same number of arguments as $\in$ and M is an intuitionistically meaningful domain. Let $A(=,\in)$ be a sentence in L. $\mathfrak{M}$ is a model for $A(=,\in)$ if and only if $A^{\mathfrak{M}}(\approx,\varepsilon)$ holds intuitionistically, where $A^{\mathfrak{M}}(\approx,\varepsilon)$ is obtained from $A(=,\in)$ by replacing all occurrences of = and $\in$ with $\approx$ and ε respectively, and relativizing all quantifiers in $A(=,\in)$ to $\mathfrak{M}$ (i.e. $\forall x$ and $\exists x$ are replaced by $(\forall x \in M)$ and $(\exists x \in M)$ respectively).[15]

So what we want to show is that $\mathfrak{V} = (V,\approx,\varepsilon)$ is a model of CZF. That is, we want to show that the axioms of CZF interpreted in $\mathfrak{V}$ are valid.

Let $L^{\mathfrak{V}}$ denote the language L where $\approx$, ε, $(\forall x \in V)$, $(\exists x \in V)$, $(\forall x\ \varepsilon\ y)$ and $(\exists x\ \varepsilon\ y)$ replaced by =, $\in$, $\forall x$, $\exists x$, $(\forall x \in y)$ and $(\exists x \in y)$ respectively. Here follow the axioms of CZF interpreted in $\mathfrak{V}$:

Restricted quantifier axioms
$(\forall x\ \varepsilon\ y)F(x) \leftrightarrow (\forall x \in V)(x\ \varepsilon\ y \rightarrow F(x))$
$(\exists x\ \varepsilon\ y)F(x) \leftrightarrow (\exists x \in V)(x\ \varepsilon\ y \wedge F(x))$
for every formula $F(x)$ in $L^{\mathfrak{V}}$.

Extensionality axioms
i) $(\forall x \in V)(\forall y \in V)(\forall z \in V)((x \approx y \wedge y\ \varepsilon\ z) \rightarrow x\ \varepsilon\ z)$
ii) $(\forall x \in V)(\forall y \in V)(x \approx y \leftrightarrow (\forall z \in V)(z\ \varepsilon\ x \leftrightarrow z\ \varepsilon\ y))$

Set induction
$(\forall x \in V)((\forall y\ \varepsilon\ x)F(y) \rightarrow F(x)) \rightarrow (\forall x \in V)F(x)$
for all formulae $F(x)$ in $L^{\mathfrak{V}}$.

Pairing
$(\forall x \in V)(\forall y \in V)(\exists z \in V)(\forall u \in V)(u\ \varepsilon\ z \leftrightarrow u \approx x \vee u \approx y)$

Union
$(\forall x \in V)(\exists z \in V)(\forall u \in V)(u\ \varepsilon\ z \leftrightarrow (\exists y\ \varepsilon\ x)(u\ \varepsilon\ y))$

Restricted separation
$(\forall x \in V)(\exists z \in V)(\forall y \in V)(y\ \varepsilon\ z \leftrightarrow y\ \varepsilon\ x \wedge F(y))$
for all restricted formulae $F(y)$ in $L^{\mathfrak{V}}$.

Strong collection
$(\forall u \in V)((\forall x\ \varepsilon\ u)(\exists y \in V)F(x,y) \rightarrow (\exists v \in V)F'(u,v))$
for all formulae $F(x,y)$ in $L^{\mathfrak{V}}$, where $F'(u,v)$ abbreviates
$(\forall x\ \varepsilon\ u)(\exists y\ \varepsilon\ v)F(x,y) \wedge (\forall y\ \varepsilon\ v)(\exists x\ \varepsilon\ u)F(x,y)$.

[15] Troelstra and van Dalen, 1988, vol. I, p. 75.

Subset collection
$(\forall v \in V)(\forall w \in V)(\exists t \in V)(\forall u \in V)((\forall x\ \varepsilon\ v)(\exists y\ \varepsilon\ w)F(x,y) \rightarrow (\exists z\ \varepsilon\ t)F'(v,z))$
for all formulae $F(x,y)$ in $L^{\mathfrak{v}}$.

Infinity
$(\exists z \in V)((\exists x\ \varepsilon\ z)(\forall y\ \varepsilon\ x)\perp \wedge (\forall x\ \varepsilon\ z)(\exists y\ \varepsilon\ z)succ(x,y))$
where $succ(x,y)$ is
$x\ \varepsilon\ y \wedge (\forall u\ \varepsilon\ x)(u\ \varepsilon\ y) \wedge (\forall u\ \varepsilon\ y)(u\ \varepsilon\ x \vee u \approx x)$

Proofs of the axioms
Now we proceed to prove the axioms.[16]

Definition 1 $(\forall x\ \varepsilon\ \alpha)F(x)$ and $(\exists x\ \varepsilon\ \alpha)F(x)$
Let $\alpha \in V$ and let $F(x)$ be a proposition for $x \in V$.
$(\forall x\ \varepsilon\ \alpha)F(x) = (\forall x \in -\alpha)F(\sim\alpha(x))$
$(\exists x\ \varepsilon\ \alpha)F(x) = (\exists x \in -\alpha)F(\sim\alpha(x))$

Theorem 1 Set induction
$(\forall x \in V)((\forall y\ \varepsilon\ x)F(y) \rightarrow F(x)) \rightarrow (\forall y \in V)F(y)$

Proof
Assume $h \in (\forall x \in V)((\forall y\ \varepsilon\ x)F(y) \rightarrow F(x))$.
Then $h \in (\forall x \in V)((\forall y \in -x)F(\sim x(y)) \rightarrow F(x))$.
So $h(x) \in ((\forall y \in -x)F(\sim x(y)) \rightarrow F(x))$ for $x \in V$.
So $h(x)(z) \in F(x)$ for $x \in V$ and $z \in (\forall y \in -x)F(\sim x(y))$.
In order to use V-elimination we have to express h through $-x$ and $\sim x$.
So let $h' = (\lambda -x)(\lambda\sim x)h(sup(-x,\sim x))$
Then $h'(-x)(\sim x)(z) \in F(sup(-x,\sim x))$ for $-x \in V, \sim x \in -x \rightarrow V$ and $z \in (\forall y \in -x)F(\sim x(y))$.
Now let $\alpha \in V$.
V-elimination gives $T(\alpha,h') \in F(\alpha)$.
Hence $(\lambda y)T(y,h') \in (\forall y \in V)F(y)$.
Hence $(\lambda x)(\lambda y)T(y,x') \in (\forall x \in V)((\forall y\ \varepsilon\ x)F(y) \rightarrow F(x)) \rightarrow (\forall y \in V)F(y)$, i.e.
$(\forall x \in V)((\forall y\ \varepsilon\ x)F(y) \rightarrow F(x)) \rightarrow (\forall y \in V)F(y)$ true.

Lemma 2
There is a type $\approx$ such that, for $A,B \in U$ and $f,g \in U \rightarrow V$,
$sup(A,f) \approx sup(B,g) = (\forall x \in A)(\exists y \in B)(f(x) \approx g(y)) \wedge (\forall y \in B)(\exists x \in A)(f(x) \approx g(y))$, where
$sup(A,f) \approx sup(B,g)$ stands for $(sup(A,f),sup(B,g)) \in \approx$ and such that
$sup(A,f) \approx sup(B,g)$ is a small type for all $sup(A,f),sup(B,g) \in V$, i.e. a type in U.

Proof
We define
$G_1(u,v,w) = (\forall x \in u)(\exists y \in -v)w(x)(\sim v(y)) \in U$
$G_2(u,v,w) = (\forall y \in -v)(\exists x \in u)w(x)(\sim v(y)) \in U$
and $G(u,z,w) = (\lambda v)(G_1(u,v,w) \wedge G_2(u,v,w)) \in V \rightarrow U$
where $u \in U$, $v \in V$, $z \in u \rightarrow V$ and $w \in u \rightarrow (V \rightarrow U)$ which is another way of writing

[16] We mainly follow Aczel 1982 and Troelstra and van Dalen 1988.

$w \in (\forall x \in u)(V \to U)$. NB. $G(u,z,w)$ does not actually depend on z.
Now by V-elimination we get $T(\alpha,G) \in V \to U$, for $\alpha \in V$. So we define, for $\sup(A,f),\sup(B,g) \in V$
$\sup(A,f) \approx \sup(B,g) = T(\sup(A,f),G)(\sup(B,g)) \in U$.
From V-equality we get $T(\sup(A,f),G) = G(A,f,(\lambda u)T(f(u),G)) \in V \to U$, where $A \in U$ and $f \in A \to V$. Let $b = (\lambda u)T(f(u),G)$.
Now $\sup(A,f) \approx \sup(B,g) = T(\sup(A,f),G)(\sup(B,g)) = G(A,f,(\lambda u)T(f(u),G))(\sup(B,g)) = G(A,f,b)(\sup(B,g)) = G_1(A,\sup(B,g),b) \wedge G_2(A,\sup(B,g),b)$.
But $G_1(A,\sup(B,g),b) = (\forall x \in A)(\exists y \in B)b(x)(g(y)) = (\forall x \in A)(\exists y \in B)T(f(x),G)(g(y)) = (\forall x \in A)(\exists y \in B)(f(x) \approx g(y))$.
And $G_2(A,\sup(B,g),b) = (\forall y \in B)(\exists x \in A)b(x)(g(y)) = (\forall y \in B)(\exists x \in A)T(f(x),G)(g(y)) = (\forall y \in B)(\exists x \in A)(f(x) \approx g(y))$.
Hence
$\sup(A,f) \approx \sup(B,g) = (\forall x \in A)(\exists y \in B)(f(x) \approx g(y)) \wedge (\forall y \in B)(\exists x \in A)(f(x) \approx g(y))$.

Remark 2
Let $\alpha = \sup(A,f) \in V$ and $\beta = \sup(B,g) \in V$. Then the definition of $\approx$ can be reformulated in the following way:
$\alpha \approx \beta = (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)(x \approx y) \wedge (\forall y\ \varepsilon\ \beta)(\exists x\ \varepsilon\ \alpha)(x \approx y)$

Lemma 3 The equivalence relation $\approx$
For $\alpha,\beta,\gamma \in V$
i) $\alpha \approx \alpha$
ii) $\alpha \approx \beta \to \beta \approx \alpha$
iii) $\alpha \approx \beta \wedge \beta \approx \gamma \to \alpha \approx \gamma$

Proof
i) Let $\alpha \in V$. We have
$(\forall x\ \varepsilon\ \alpha)(x \approx x)$
$\to (\forall x \in -\alpha)(\sim\alpha(x) \approx \sim\alpha(x))$.
$\to (\forall x \in -\alpha)(\exists y \in -\alpha)(\sim\alpha(x) \approx \sim\alpha(y))$.
$\to (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \alpha)(x \approx y)$.
Similarly, by renaming $(\forall x\ \varepsilon\ \alpha)(x \approx x)$ we get
$(\forall y\ \varepsilon\ \alpha)(y \approx y)$
$\to (\forall y \in -\alpha)(\sim\alpha(y) \approx \sim\alpha(y))$.
$\to (\forall y \in -\alpha)(\exists x \in -\alpha)(\sim\alpha(x) \approx \sim\alpha(y))$.
$\to (\forall y\ \varepsilon\ \alpha)(\exists x\ \varepsilon\ \alpha)(x \approx y)$.
So we have $(\forall x\ \varepsilon\ \alpha)(x \approx x) \to (\alpha \approx \alpha)$.
And Theorem 1 gives us $(\forall\alpha \in V)(\alpha \approx \alpha)$.

ii) Let $F(x) = (\forall y \in V)(x \approx y \to y \approx x)$. Let $\alpha \in V$ such that $(\forall x\ \varepsilon\ \alpha)F(x)$. Let $\beta \in V$.
$\alpha \approx \beta$
$\to (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)(x \approx y)$
$\to (\forall x\ \varepsilon\ \alpha)(\exists y \in -\beta)(x \approx \sim\beta(y))$
$\to F(x)$ for $x\ \varepsilon\ \alpha$ and $\sim\beta(y) \in V$ for $y \in -\beta$
$(\forall x\ \varepsilon\ \alpha)(\exists y \in -\beta)(\sim\beta(y) \approx x)$
$\to (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)(y \approx x)$.

Similarly $\alpha \approx \beta$
$\rightarrow (\forall y\ \varepsilon\ \beta)(\exists x\ \varepsilon\ \alpha)(x \approx y)$
$\rightarrow (\forall y \in -\beta)(\exists x\ \varepsilon\ \alpha)(x \approx {\sim}\beta(y))$
$\rightarrow$ F(x) for x ε α and $\sim\beta(y) \in V$ for $y \in -\beta$
$(\forall y \in -\beta)(\exists x\ \varepsilon\ \alpha)({\sim}\beta(y) \approx x)$
$\rightarrow (\forall y\ \varepsilon\ \beta)(\exists x\ \varepsilon\ \alpha)(y \approx x)$
Hence $\alpha \approx \beta \rightarrow \beta \approx \alpha$.
Hence $(\forall x\ \varepsilon\ \alpha)F(x) \rightarrow F(\alpha)$.
Hence $(\forall x \in V)F(x)$, i.e. $(\forall x \in V)(\forall y \in V)(x \approx y \rightarrow y \approx x)$.

iii) Let $F(x) = (\forall y \in V)(\forall z \in V)(x \approx y \wedge y \approx z \rightarrow x \approx z)$. Let $\alpha, \beta, \gamma \in V$ and $(\forall x\ \varepsilon\ \alpha)F(x)$.
$\alpha \approx \beta \wedge \beta \approx \gamma$
$\rightarrow (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)(x \approx y) \wedge (\forall y\ \varepsilon\ \beta)(\exists z\ \varepsilon\ \gamma)(y \approx z)$
$\rightarrow (\forall x\ \varepsilon\ \alpha)(\exists y \in -\beta)(x \approx {\sim}\beta(y)) \wedge (\forall y \in -\beta)(\exists z \in -\gamma)({\sim}\beta(y) \approx {\sim}\gamma(z))$
$\rightarrow (\forall x\ \varepsilon\ \alpha)(\exists y \in -\beta)(\exists z \in -\gamma)(x \approx {\sim}\beta(y) \wedge {\sim}\beta(y) \approx {\sim}\gamma(z))$
$\rightarrow$ F(x) for x ε α, $\sim\beta(y) \in V$ for $y \in -\beta$ and $\sim\gamma(z) \in V$ for $z \in -\gamma$
$(\forall x \in -\alpha)(\exists z \in -\gamma)({\sim}\alpha(x) \approx {\sim}\gamma(z))$
$\rightarrow (\forall x\ \varepsilon\ \alpha)(\exists z\ \varepsilon\ \gamma)(x \approx z)$.
Similarly $\alpha \approx \beta \wedge \beta \approx \gamma \rightarrow \beta \approx \gamma \wedge \alpha \approx \beta$
$\rightarrow (\forall z\ \varepsilon\ \gamma)(\exists y\ \varepsilon\ \beta)(y \approx z) \wedge (\forall y\ \varepsilon\ \beta)(\exists x\ \varepsilon\ \alpha)(x \approx y)$
$\rightarrow (\forall z \in -\gamma)(\exists y \in -\beta)({\sim}\beta(y) \approx {\sim}\gamma(z)) \wedge (\forall y \in -\beta)(\exists x\ \varepsilon\ \alpha)(x \approx {\sim}\beta(y))$
$\rightarrow (\forall z \in -\gamma)(\exists y \in -\beta)(\exists x\ \varepsilon\ \alpha)({\sim}\beta(y) \approx {\sim}\gamma(z) \wedge x \approx {\sim}\beta(y))$
$\rightarrow (\forall z \in -\gamma)(\exists y \in -\beta)(\exists x\ \varepsilon\ \alpha)(x \approx {\sim}\beta(y) \wedge {\sim}\beta(y) \approx {\sim}\gamma(z))$
$\rightarrow$ F(x) for x ε α, $\sim\beta(y) \in V$ for $y \in -\beta$ and $\sim\gamma(z) \in V$ for $z \in -\gamma$
$(\forall z \in -\gamma)(\exists x\ \varepsilon\ \alpha)(x \approx {\sim}\gamma(z))$
$\rightarrow (\forall z\ \varepsilon\ \gamma)(\exists x\ \varepsilon\ \alpha)(x \approx z)$.
Hence $\alpha \approx \beta \wedge \beta \approx \gamma \rightarrow \alpha \approx \gamma$.
Hence $(\forall x\ \varepsilon\ \alpha)F(x) \rightarrow F(\alpha)$.
Hence $(\forall x \in V)F(x)$, i.e. $(\forall x \in V)(\forall y \in V)(\forall z \in V)(x \approx y \wedge y \approx z \rightarrow x \approx z)$.

Definition 2 $\alpha\ \varepsilon\ \beta$ *and* $\alpha \subseteq \beta$
For $\alpha,\beta \in V$, let
$\alpha\ \varepsilon\ \beta = (\exists y\ \varepsilon\ \beta)(\alpha \approx y)$
$\alpha \subseteq \beta = (\forall x\ \varepsilon\ \alpha)(x\ \varepsilon\ \beta)$

Remark 3
i) $\alpha \subseteq \beta = (\forall x\ \varepsilon\ \alpha)(x\ \varepsilon\ \beta) = (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)(x \approx y)$
ii) $\alpha\ \varepsilon\ \beta$ is a small type since $\alpha \approx \beta$ is small.
iii) $\alpha \subseteq \beta$ is a small type since $\alpha\ \varepsilon\ \beta$ is small.

Definition 3
Let $x \in V$. F(x) is *extensional* over V if
$(\forall x \in V)(\forall y \in V)(x \approx y \wedge F(x) \leftrightarrow F(y))$

Remark 4
An implication of Definition 3 is
$(\forall y \in V)((\forall x \in V)(x \approx y \rightarrow F(x)) \leftrightarrow F(y))$.

Lemma 4
If F is extensional over V then, for $\alpha \in V$
i) $(\forall x\ \varepsilon\ \alpha)F(x) \leftrightarrow (\forall x \in V)(x\ \varepsilon\ \alpha \rightarrow F(x))$
ii) $(\exists x\ \varepsilon\ \alpha)F(x) \leftrightarrow (\exists x \in V)(x\ \varepsilon\ \alpha \wedge F(x))$

Proof
i) $(\forall x \in V)(x\ \varepsilon\ \alpha \rightarrow F(x))$
$\leftrightarrow (\forall x \in V)((\exists y\ \varepsilon\ \alpha)(x \approx y) \rightarrow F(x))$
$\leftrightarrow (\forall x \in V)((\exists y \in -\alpha)(x \approx {\sim}\alpha(y)) \rightarrow F(x))$
$\leftrightarrow (\forall x \in V)(\forall y \in -\alpha)(x \approx {\sim}\alpha(y) \rightarrow F(x))$
$\leftrightarrow (\forall y \in -\alpha)(\forall x \in V)(x \approx {\sim}\alpha(y) \rightarrow F(x))$
$\leftrightarrow$ by Remark 4
$(\forall y \in -\alpha)F({\sim}\alpha(y))$
$\leftrightarrow (\forall y\ \varepsilon\ \alpha)F(y)$
$\leftrightarrow (\forall x\ \varepsilon\ \alpha)F(x)$
ii) $(\exists x \in V)(x\ \varepsilon\ \alpha \wedge F(x))$
$\leftrightarrow (\exists x \in V)((\exists y\ \varepsilon\ \alpha)(x \approx y) \wedge F(x))$
$\leftrightarrow (\exists x \in V)((\exists y \in -\alpha)(x \approx {\sim}\alpha(y)) \wedge F(x))$
$\leftrightarrow (\exists x \in V)(\exists y \in -\alpha)(x \approx {\sim}\alpha(y) \wedge F(x))$
$\leftrightarrow (\exists y \in -\alpha)(\exists x \in V)(x \approx {\sim}\alpha(y) \wedge F(x))$
$\leftrightarrow$ by Definition 3
$(\exists y \in -\alpha)F({\sim}\alpha(y))$
$\leftrightarrow (\exists y\ \varepsilon\ \alpha)F(y)$
$\leftrightarrow$ change of bound variable
$(\exists x\ \varepsilon\ \alpha)F(x)$.

Theorem 2 Extensionality
For $\alpha,\beta,\gamma \in V$
i) $(\alpha \approx \beta \wedge \beta\ \varepsilon\ \gamma) \rightarrow \alpha\ \varepsilon\ \gamma$
ii) $\alpha \approx \beta \leftrightarrow (\forall x \in V)(x\ \varepsilon\ \alpha \leftrightarrow x\ \varepsilon\ \beta)$

Proof
i) $\alpha \approx \beta \wedge \beta\ \varepsilon\ \gamma$
$\rightarrow \alpha \approx \beta \wedge (\exists z\ \varepsilon\ \gamma)(\beta \approx z)$
$\rightarrow \alpha \approx \beta \wedge (\exists z \in -\gamma)(\beta \approx {\sim}\gamma(z))$
$\rightarrow (\exists z \in -\gamma)(\alpha \approx \beta \wedge \beta \approx {\sim}\gamma(z))$
$\rightarrow$ by Lemma 3 iii)
$(\exists z \in -\gamma)(\alpha \approx {\sim}\gamma(z))$
$\rightarrow (\exists z\ \varepsilon\ \gamma)(\alpha \approx z)$
$\rightarrow \alpha\ \varepsilon\ \gamma$
Hence $F(x) = (x\ \varepsilon\ \alpha)$ is extensional over V.
ii) Now
$\alpha \subseteq \beta$
$\leftrightarrow (\forall x\ \varepsilon\ \alpha)(x\ \varepsilon\ \beta)$
$\leftrightarrow$ by Lemma 4 i)
$(\forall x \in V)(x\ \varepsilon\ \alpha \rightarrow x\ \varepsilon\ \beta)$
We also have $\beta \subseteq \alpha$

$\leftrightarrow (\forall x\ \varepsilon\ \beta)(x\ \varepsilon\ \alpha)$
$\leftrightarrow$ by Lemma 4 i)
$(\forall x \in V)(x\ \varepsilon\ \beta \rightarrow x\ \varepsilon\ \alpha)$
Now $\alpha \approx \beta = (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)(x \approx y) \wedge (\forall y\ \varepsilon\ \beta)(\exists x\ \varepsilon\ \alpha)(x \approx y)$
and $\alpha \approx \beta \rightarrow \beta \approx \alpha$
and $\alpha \subseteq \beta = (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)(x \approx y)$ give
$\alpha \approx \beta$
$\leftrightarrow (\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)(x \approx y) \wedge (\forall y\ \varepsilon\ \beta)(\exists x\ \varepsilon\ \alpha)(x \approx y)$
$\leftrightarrow \alpha \subseteq \beta \wedge \beta \subseteq \alpha$
$\leftrightarrow (\forall x \in V)(x\ \varepsilon\ \alpha \rightarrow x\ \varepsilon\ \beta) \wedge (\forall x \in V)(x\ \varepsilon\ \beta \rightarrow x\ \varepsilon\ \alpha)$
$\leftrightarrow (\forall x \in V)(x\ \varepsilon\ \beta \leftrightarrow x\ \varepsilon\ \alpha)$.

Lemma 5
i) $(\alpha\ \varepsilon\ \beta \wedge \beta \approx \gamma) \rightarrow \alpha\ \varepsilon\ \gamma$
ii) $(\alpha\ \varepsilon\ \beta \wedge \beta\ \varepsilon\ \alpha) \rightarrow \perp$

Proof
i) Assume $\alpha\ \varepsilon\ \beta \wedge \beta \approx \gamma$. $\beta \approx \gamma$ implies that $x\ \varepsilon\ \gamma$ whenever $x\ \varepsilon\ \beta$, by Theorem 2 ii). Hence $\alpha\ \varepsilon\ \gamma$.
ii) Let $\alpha,\beta \in V$ and let $F(x) = (\forall y \in V)((x\ \varepsilon\ y \wedge y\ \varepsilon\ x) \rightarrow \perp)$. Assume $(\forall x\ \varepsilon\ \alpha)F(x)$.
Now $(\alpha\ \varepsilon\ \beta \wedge \beta\ \varepsilon\ \alpha)$
$\rightarrow (\beta\ \varepsilon\ \alpha \wedge \alpha\ \varepsilon\ \beta)$
$\alpha \in V$, $\beta\ \varepsilon\ \alpha$ and $(\forall x\ \varepsilon\ \alpha)F(x)$ give
$\rightarrow \perp$
Hence by Set induction, $(\alpha\ \varepsilon\ \beta \wedge \beta\ \varepsilon\ \alpha) \rightarrow \perp$.

Lemma 6
If $F \in L^{\upsilon}$, then F is extensional.

Proof
By induction on the construction of F.
Lemma 3 iii) takes care of $\approx$, and Theorem 2 i) takes care of ε, for example.

Theorem 3 Restricted quantifier
i) $(\forall x\ \varepsilon\ \alpha)F(x) \leftrightarrow (\forall x \in V)(x\ \varepsilon\ \alpha \rightarrow F(x))$
ii) $(\exists x\ \varepsilon\ \alpha)F(x) \leftrightarrow (\exists x \in V)(x\ \varepsilon\ \alpha \wedge F(x))$
By Lemmata 4 and 6

Theorem 4 Pairing
If $\alpha,\beta \in V$, then there is $\gamma \in V$ such that for all $\eta \in V$
$\eta\ \varepsilon\ \gamma \leftrightarrow (\eta \approx \alpha \vee \eta \approx \beta)$

Proof
Let $\alpha,\beta \in V$ and let $\gamma = (\sup z \in N_2)(\lambda z)R_2(z,\alpha,\beta) \in V$. Then for $\eta \in V$
$\eta\ \varepsilon\ \gamma$
$\leftrightarrow (\exists z\ \varepsilon\ \gamma)(\eta \approx z)$
$\leftrightarrow (\exists z \in -\gamma)(\eta \approx \sim\gamma(z))$
$\leftrightarrow (\exists z \in N_2)(\eta \approx R_2(z,\alpha,\beta))$

$\leftrightarrow$ by N_2-existence $(\exists z \in N_2)F(z) \leftrightarrow F(0_2) \vee F(1_2)$
$\eta \approx R_2(0_2,\alpha,\beta) \vee \eta \approx R_2(1_2,\alpha,\beta)$
$\leftrightarrow$ N_2-equality
$\eta \approx \alpha \vee \eta \approx \beta$

Theorem 5 Union
If $\alpha \in V$, then there is $\gamma \in V$ such that for all $\eta \in V$
$\eta \,\varepsilon\, \gamma \leftrightarrow (\exists x \,\varepsilon\, \alpha)(\eta \,\varepsilon\, x)$

Proof
Let $\alpha \in V$ and let $\gamma = (\sup z \in (\Sigma x \in -\alpha){-}{\sim}\alpha(x)){\sim}({\sim}\alpha(p(z)))(q(z)) \in V$. Now for $\eta \in V$
$\eta \,\varepsilon\, \gamma$
$\leftrightarrow (\exists z \in -\gamma)(\eta \approx {\sim}\gamma(z))$
$\leftrightarrow (\exists z \in (\Sigma x \in -\alpha){-}{\sim}\alpha(x))(\eta \approx {\sim}({\sim}\alpha(p(z)))(q(z)))$
$\leftrightarrow$ by Σ-existence $(\exists z \in (\Sigma x \in A)B(x))F(z) \leftrightarrow (\exists x \in A)(\exists y \in B(x))F((x,y))$
$(\exists x \in -\alpha)(\exists y \in {-}{\sim}\alpha(x))(\eta \approx {\sim}({\sim}\alpha(p((x,y))))(q((x,y))))$
$\leftrightarrow (\exists x \in -\alpha)(\exists y \in {-}{\sim}\alpha(x))(\eta \approx {\sim}({\sim}\alpha(x))(y))$
$\leftrightarrow (\exists x \in -\alpha)(\exists y \,\varepsilon\, {\sim}\alpha(x))(\eta \approx y)$
$\leftrightarrow (\exists x \,\varepsilon\, \alpha)(\exists y \,\varepsilon\, x)(\eta \approx y)$
$\leftrightarrow (\exists x \,\varepsilon\, \alpha)(\eta \,\varepsilon\, y)$

Lemma 7
If $F(x) \in L^{\mathfrak{v}}$ for $x \in V$ and $F(x)$ is restricted, then $F(x)$ is small.

Proof
By induction on the construction of $F(x)$.
$F \rightarrow G$, $F \wedge G$, $F \vee G$, $\perp$ are small if F and G are small, by definition and since U is closed with respect to Π, Σ, $+$, N_0.
$(\forall x \,\varepsilon\, y)$ and $(\exists x \,\varepsilon\, y)$ are small by Definition 1 and since U is closed with respect to Π and Σ.
$\alpha \approx \beta$ is small by Lemma 2.
$\alpha \,\varepsilon\, \beta$ is small by Remark 3.

Theorem 6 Restricted separation
If $\alpha \in V$, $F(x) \in L^{\mathfrak{v}}$ for $x \in V$ and $F(x)$ is restricted, then there is $\gamma \in V$ such that for all $\eta \in V$
$\eta \,\varepsilon\, \gamma \leftrightarrow \eta \,\varepsilon\, x \wedge F(\eta)$

Proof
Let $\alpha \in V$, $F(x) \in L^{\mathfrak{v}}$ for $x \in V$ and $F(x)$ be restricted . Then $F(x)$ is a small type, by Lemma 7. Let $\gamma = (\sup z \in (\Sigma x \in -\alpha)F({\sim}\alpha(x))){\sim}\alpha(p(z)) \in V$. Now if $\eta \in V$
$\eta \,\varepsilon\, \gamma$
$\leftrightarrow (\exists z \,\varepsilon\, \gamma)(\eta \approx z)$
$\leftrightarrow (\exists z \in -\gamma)(\eta \approx {\sim}\gamma(z))$
$\leftrightarrow (\exists z \in (\Sigma x \in -\alpha)F({\sim}\alpha(x)))(\eta \approx {\sim}\alpha(p(z)))$
$\leftrightarrow$ by Σ-existence $(\exists z \in (\Sigma x \in A)B(x))F(z) \leftrightarrow (\exists x \in A)(\exists y \in B(x))F((x,y))$
$(\exists x \in -\alpha)(\exists y \in F({\sim}\alpha(x)))(\eta \approx {\sim}\alpha(p((x,y))))$
$\leftrightarrow (\exists x \in -\alpha)(\exists y \in F({\sim}\alpha(x)))(\eta \approx {\sim}\alpha(x))$
$\leftrightarrow (\exists x \in -\alpha)(F({\sim}\alpha(x)) \wedge \eta \approx {\sim}\alpha(x))$

$\leftrightarrow (\exists x \mathbin{\varepsilon} \alpha)(F(x) \wedge \eta \approx x)$
$\leftrightarrow$ by Lemma 6
$(\exists x \mathbin{\varepsilon} \alpha)(F(\eta) \wedge \eta \approx x)$
$\leftrightarrow (\exists x \mathbin{\varepsilon} \alpha)(x \approx \eta) \wedge F(\eta)$
$\leftrightarrow \eta \mathbin{\varepsilon} \alpha \wedge F(\eta)$

Notation
Let $F(x,y)$ be a type for $x,y \in V$. Then let $F'(x,y)$ be given by
$F'(x,y) = (\forall u \mathbin{\varepsilon} x)(\exists v \mathbin{\varepsilon} y)F(u,v) \wedge (\forall v \mathbin{\varepsilon} y)(\exists u \mathbin{\varepsilon} x)F(u,v)$.

Lemma 8
If $\alpha, \beta \in V$ such that $-\alpha = -\beta$ then
$(\forall x \in -\alpha)F(\sim\alpha(x),\sim\beta(x)) \rightarrow F'(\alpha,\beta)$

Proof
Let $\alpha,\beta \in V$ such that $-\alpha = -\beta$, then we have
$(\forall x \in -\alpha)F(\sim\alpha(x),\sim\beta(x))$
$\rightarrow (\forall x \in -\alpha)(\exists y \in -\alpha)F(\sim\alpha(x),\sim\beta(y))$
$\rightarrow$ substitution $-\alpha = -\beta$
$(\forall x \in -\alpha)(\exists y \in -\beta)F(\sim\alpha(x),\sim\beta(y))$
$\rightarrow (\forall x \mathbin{\varepsilon} \alpha)(\exists y \mathbin{\varepsilon} \beta)F(x,y)$
Similarly
$(\forall x \in -\alpha)F(\sim\alpha(x),\sim\beta(x))$
$\rightarrow$ renaming
$(\forall y \in -\alpha)F(\sim\alpha(y),\sim\beta(y))$
$\rightarrow (\forall y \in -\beta)(\exists x \in -\beta)F(\sim\alpha(y),\sim\beta(x))$
$\rightarrow$ substitution $-\alpha = -\beta$
$(\forall y \in -\beta)(\exists x \in -\alpha)F(\sim\alpha(x),\sim\beta(y))$
$\rightarrow (\forall y \mathbin{\varepsilon} \beta)(\exists x \mathbin{\varepsilon} \alpha)F(x,y)$
So we have $(\forall x \in -\alpha)F(\sim\alpha(x),\sim\beta(x)) \rightarrow (\forall x \mathbin{\varepsilon} \alpha)(\exists y \mathbin{\varepsilon} \beta)F(x,y) \wedge (\forall y \mathbin{\varepsilon} \beta)(\exists x \mathbin{\varepsilon} \alpha)F(x,y)$
Hence $(\forall x \in -\alpha)F(\sim\alpha(x),\sim\beta(x)) \rightarrow F'(\alpha,\beta)$.

Theorem 7 Strong collection
If $\alpha \in V$ then
$(\forall x \mathbin{\varepsilon} \alpha)(\exists y \in V)F(x,y) \rightarrow (\exists v \in V)F'(\alpha,v)$

Proof
Let $\alpha \in V$, then
$(\forall x \mathbin{\varepsilon} \alpha)(\exists y \in V)F(x,y)$
$\rightarrow (\forall x \in -\alpha)(\exists y \in V)F(\sim\alpha(x),y)$
AC $(\forall x \in A)(\exists y \in B(x))C(x,y) \rightarrow (\exists f \in (\Pi x \in A)B(x))(\forall x \in A)C(x,f(x))$
AC implies that there is $f \in -\alpha \rightarrow V$ such that $(\forall x \in -\alpha)F(\sim\alpha(x),f(x))$
Let $\beta = \sup(-\alpha,f) \in V$. We have $-\beta = -\alpha$ and $\sim\beta(x) = f(x)$ for $x \in -\alpha$, by construction.
$(\forall x \in -\alpha)F(\sim\alpha(x),f(x))$
$\rightarrow$ substitution $f(x) = \sim\beta(x)$
$(\forall x \in -\alpha)F(\sim\alpha(x),\sim\beta(x))$
So by Lemma 8, $F'(\alpha,\beta)$
$\rightarrow (\exists v \in V)F'(\alpha,v)$

Theorem 8 Subset Collection
If $\alpha,\beta \in V$ then there is $\gamma \in V$ such that
$(\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)F(x,y) \rightarrow (\exists z\ \varepsilon\ \gamma)F'(\alpha,z)$

Proof
Let $\alpha,\beta \in V$ and let $\gamma = (\sup z \in -\alpha \rightarrow -\beta)(\sup x \in -\alpha)\sim\beta(z(x)) \in V$.
We have $-\gamma = -\alpha \rightarrow -\beta \in U$ and $\sim\gamma(z) = (\sup x \in -\alpha)\sim\beta(z(x)) \in V$ for $z \in -\gamma$.
Assume $(\forall x\ \varepsilon\ \alpha)(\exists y\ \varepsilon\ \beta)F(x,y)$
$\rightarrow (\forall x \in -\alpha)(\exists y \in -\beta)F(\sim\alpha(x),\sim\beta(y))$
AC $(\forall x \in A)(\exists y \in B(x))C(x,y) \rightarrow (\exists f \in (\Pi x \in A)B(x))(\forall x \in A)C(x,f(x))$
By AC there is $f \in -\alpha \rightarrow -\beta$ such that $(\forall x \in -\alpha)F(\sim\alpha(x),\sim\beta(f(x))$.
Let $\delta = (\sup x \in -\alpha)\sim\beta(f(x)) \in V$. We have $-\delta = -\alpha \in U$. And for $x \in -\alpha$ we have $f(x) \in -\beta$ and $\sim\delta(x) = \sim\beta(f(x)) \in V$.
Now $\sim\gamma(f) = (\sup x \in -\alpha)\sim\beta(f(x)) = \delta$. Hence $\delta \approx \sim\gamma(f))$, since $x \approx x$.
And since $f \in -\alpha \rightarrow -\beta$, i.e. $f \in -\gamma$, we have
$(\exists x \in -\gamma)(\delta \approx \sim\gamma(x))$
So $(\exists x\ \varepsilon\ \gamma)(\delta \approx x)$
So $\delta\ \varepsilon\ \gamma$.
Also $-\delta = -\alpha$ and
$(\forall x \in -\alpha)F(\sim\alpha(x),\sim\beta(f(x)))$
$\rightarrow (\forall x \in -\alpha)F(\sim\alpha(x),\sim\delta(x))$
So Lemma 8 implies $F'(\alpha,\delta)$.
Hence $(\exists z\ \varepsilon\ \gamma)F'(\alpha,z)$.

Lemma 9
Let $\varnothing = \sup(N_0,(\lambda x)R_0(x)) \in V$ and for $\alpha \in V$ let $\alpha' = (\sup x \in -\alpha + N_1)D(x,\sim\alpha,(y)\alpha) \in V$.
Then
i) for $\eta \in V$
$\eta\ \varepsilon\ \varnothing \leftrightarrow \perp$
ii) for $\alpha \in V$, $\alpha' \in V$ and for $\eta \in V$
$\eta\ \varepsilon\ \alpha' \leftrightarrow (\eta\ \varepsilon\ \alpha \vee \eta \approx \alpha)$
iii) for $\alpha\ \varepsilon\ V$
$\alpha\ \varepsilon\ \alpha'$
iv) for $\alpha \in V$
$\alpha' \approx \varnothing \rightarrow \perp$
v) for $\alpha,\beta \in V$
$\alpha' \approx \beta' \rightarrow \alpha \approx \beta$

Proof
i) For $\eta \in V$
$\eta\ \varepsilon\ \varnothing$
$\leftrightarrow (\exists x\ \varepsilon\ \varnothing)(\eta \approx x)$
$\leftrightarrow (\exists x \in -\varnothing)(\eta \approx \sim\varnothing(x))$
$\leftrightarrow (\exists x \in N_0)(\eta \approx R_0(x))$
$\leftrightarrow$ by N_0-existence $(\exists z \in N_0)F(z) \leftrightarrow \perp$
$\perp$

ii) Let $\alpha \in V$. If $\eta \in V$ then
$\eta \,\varepsilon\, \alpha'$
$\leftrightarrow (\exists x \,\varepsilon\, \alpha')(\eta \approx x)$
$\leftrightarrow (\exists x \in -\alpha')(\eta \approx \sim\alpha'(x))$
$\leftrightarrow (\exists x \in -\alpha + N_1)(\eta \approx D(x,\sim\alpha,(y)\alpha)$
$\leftrightarrow$ +-existence $(\exists x \in A + B)F(x) \leftrightarrow (\exists x_1 \in A)F(i(x_1)) \vee (\exists x_2 \in B)F(j(x_2))$
$(\exists x_1 \in -\alpha)(\eta \approx D(i(x_1),\sim\alpha,(y)\alpha) \vee (\exists x_2 \in N_1)(\eta \approx D(j(x_2),\sim\alpha,(y)\alpha)$
$\leftrightarrow (\exists x_1 \in -\alpha)(\eta \approx D(i(x_1),\sim\alpha,(y)\alpha) \vee (\eta \approx D(j(0_1),\sim\alpha,(y)\alpha)$
$\leftrightarrow (\exists x_1 \in -\alpha)(\eta \approx \sim\alpha(x_1)) \vee (\eta \approx \alpha)$
$\leftrightarrow (\exists x_1 \,\varepsilon\, \alpha)(\eta \approx x_1) \vee (\eta \approx \alpha)$
$\leftrightarrow (\eta \,\varepsilon\, \alpha) \vee (\eta \approx \alpha)$
iii) Let $\alpha \in V$
$\rightarrow \approx$ reflexive
$\alpha \approx \alpha$
$\rightarrow$ by ii) where $\eta = \alpha$
$\alpha \,\varepsilon\, \alpha'$
iv) Let $\alpha' \approx \varnothing$
$\rightarrow$ by iii) and Lemma 5 i)
$\alpha \,\varepsilon\, \varnothing$
$\rightarrow$ by i)
$\perp$
v) Let $\alpha,\beta \in V$
We have $\alpha \,\varepsilon\, \alpha'$ and $\beta \,\varepsilon\, \beta'$ by iii)
$\alpha' \approx \beta'$
$\rightarrow \beta' \approx \alpha'$
$\rightarrow$ Lemma 5 i)
$\alpha \,\varepsilon\, \beta' \wedge \beta \,\varepsilon\, \alpha'$
$\rightarrow$ by ii)
$(\alpha \,\varepsilon\, \beta \vee \alpha \approx \beta) \wedge (\beta \,\varepsilon\, \alpha \vee \beta \approx \alpha)$
$\rightarrow$ by Lemma 5 ii)
$\alpha \approx \beta$

Remark 5
By Lemma 9 iii) we have $\alpha \,\varepsilon\, \alpha'$.
By ii) we have $\eta \,\varepsilon\, \alpha' \rightarrow \eta \,\varepsilon\, \alpha \vee \eta \approx \alpha$.
By ii) we have $\eta \,\varepsilon\, \alpha \rightarrow \eta \,\varepsilon\, \alpha'$.
Thus succ(α,α').

Theorem 9 Infinity
Let $\Delta(n) = R(n,\varnothing,(x,y)y') \in V$ for $n \in N$.
Let $\omega = (\sup x \in N)\Delta(x) \in V$. Then
i) $\varnothing \,\varepsilon\, \omega$
ii) $(\forall x \,\varepsilon\, \omega)(x' \,\varepsilon\, \omega)$

Proof
Note that $\alpha \,\varepsilon\, \omega \leftrightarrow (\exists x \,\varepsilon\, \omega)(\alpha \approx x) \leftrightarrow (\exists x \in -\omega)(\alpha \approx \sim\omega(x)) \leftrightarrow (\exists x \in N)(\alpha \approx \Delta(x))$ and
$(\forall x \,\varepsilon\, \omega)F(x) \leftrightarrow (\forall x \in -\omega)F(\sim\omega(x)) \leftrightarrow (\forall x \in N)F(\Delta(x))$

i) We have $\Delta(0) = R(0,\varnothing,(x,y)y') = \varnothing$. So $\Delta(0) \approx \varnothing$. So $\varnothing \varepsilon \omega$.
ii) Let $\alpha \varepsilon \omega$. Hence $\alpha \approx \Delta(n)$ for some $n \in N$. Then $\alpha' = (\Delta(n))'$.
$\Delta(s(n)) = R(s(n),\varnothing,(x,y)y') = (x,y)y'(n,R(n,\varnothing,(x,y)y')) = (x,y)y'(n,\Delta(n)) = (\Delta(n))'$.
So $\alpha' \approx \Delta(s(n))$ for some $s(n) \in N$.
Hence $\alpha' \varepsilon \omega$. Hence $(\forall x \varepsilon \omega)(x' \varepsilon \omega)$.

Conclusion

Theorem
The axioms of CZF are valid in $\mathfrak{V} = (V, \approx, \varepsilon)$.

Proof
Chapter 4 proves that the axioms of intuitionistic logic are valid in type theory. Theorems 1 – 9 in Chapter 7 prove the axioms of Set induction, Extensionality, Restricted quantification, Pairing, Union, Restricted separation, Strong collection, Subset collection and Infinity.

We have now shown that $\mathfrak{V}$ is a model for CZF, i.e. that there is an interpretation of CZF in type theory. This gives us what we were searching for: a constructive justification of CZF.

References

Peter Aczel, The type theoretical interpretation of constructive set theory, *Logic Colloquium '77*, eds. A. Macintyre, L. Pacholski and J. Paris, North-Holland, Amsterdam, 1978, pp. 55 – 66.

Peter Aczel, The type theoretical interpretation of constructive set theory: choice principles, *The L.E.J. Brouwer Centenary Symposium*, eds. A.S. Troelstra and D. van Dalen, North-Holland, Amsterdam, 1982, pp. 1 – 40.

Peter Aczel, The type theoretical interpretation of constructive set theory: inductive definitions, *Logic, Methodology and Philosophy of Science VII*, eds. Barcan Marcus et al., North-Holland, Amsterdam, 1986, pp. 17 – 49.

Peter Aczel and M. Rathjen, Notes on constructive set theory, Institut Mittag-Leffler, report no. 40, 2000/2001.

Errett Bishop, *Foundations of Constructive Analysis*, McGraw-Hill, New York, 1967.

W. A. Howard, The formulae-as-types notion of construction, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, London, 1980, (1969), pp. 479 – 490.

Per Martin-Löf, *Intuitionistic Type Theory*, Bibliopolis, Naples, 1984.

John Myhill, Constructive set theory, *Journal of Symbolic Logic*, volume 40, 1975, pp. 347 – 382.

Bengt Nordström, Kent Petterson and Jan Smith, *Programming in Martin-Löf's Type Theory. An Introduction*, Clarendon Press, Oxford, 1990.

Aarne Ranta, *Type-Theoretical Grammar*, Clarendon press, Oxford, 1994.

A.S. Troelstra and D. van Dalen, *Constructivism in Mathematics, Volumes I, II*, North-Holland, Amsterdam, 1988.