Noname manuscript No.
(will be inserted by the editor)

Consistency of the intensional level of the Minimalist Foundation with Church's Thesis and Axiom of Choice

Hajime Ishihara · Maria Emilia Maietti · Samuele Maschio · Thomas Streicher

Received: date / Accepted: date

Abstract Consistency with the formal Church's thesis, for short **CT**, and the axiom of choice, for short **AC**, was one of the requirements asked to be satisfied by the intensional level of a two-level foundation for constructive mathematics as proposed by the second author and G. Sambin in 2005.

Here we show that this is the case for the intensional level of the two-level Minimalist Foundation, for short **MF**, completed in 2009 by the second author. The intensional level of **MF** consists of an intensional type theory à la Martin-Löf, called **mTT**.

The consistency of **mTT** with **CT** and **AC** is obtained by showing the consistency with the formal Church's thesis of a fragment of intensional Martin-Löf's type theory, called $\mathbf{MLtt}_1$, where **mTT** can be easily interpreted. Then to show the consistency of $\mathbf{MLtt}_1$ with **CT** we interpret it within Feferman's predicative theory of non-iterative fixpoints $\widehat{ID_1}$ by extending the well known

Hajime Ishihara
School of Information Science.
Japan Advanced Institute of Science and Technology. Nomi. Japan
E-mail: ishihara@jaist.ac.jp

Maria Emilia Maietti
Dipartimento di Matematica.
Università di Padova.
Via Trieste, 63. 35121 Padova. Italy
Tel. + 39 049 8271478
Fax +39 049 8271499
E-mail: maietti@math.unipd.it

Samuele Maschio
Dipartimento di Matematica.
Università di Padova. Italy
E-mail: maschio@math.unipd.it

Thomas Streicher
Fachbereich Mathematik.
Technische Universität Darmstadt. Darmstadt. Germany
E-mail: streicher@mathematik.tu-darmstadt.de

Kleene's realizability semantics of intuitionistic arithmetics so that **CT** is trivially validated.

More in detail the fragment $\mathbf{MLtt}_1$ we interpret consists of first order intensional Martin-Löf's type theory with one universe and with explicit substitution rules in place of usual equality rules preserving type constructors (hence without the so called ξ-rule which is not valid in our realizability semantics).

A key difficulty encountered in our interpretation was to use the right interpretation of lambda abstraction in the applicative structure of natural numbers in order to model all the equality rules of $\mathbf{MLtt}_1$ correctly. In particular the universe of $\mathbf{MLtt}_1$ is modelled by means of $\widehat{ID_1}$-fixpoints following a technique due first to Aczel and used by Feferman and Beeson.

Keywords realizability · type theory · formal Church's Thesis

1 Introduction

Constructive mathematics can be informally defined as mathematics developed with constructive proofs, namely proofs admitting a computational interpretation. Usually a formal definition of constructive proof is given by referring to proofs formalizable in a suitable set-theoretic foundation admitting a computational model. Indeed there is no standard referential foundation for constructive mathematics to which the majority of constructive mathematicians refers to, as it happens for classical mathematics with Zermelo-Fraenkel set theory possibly with the addition of the axiom of choice. Many different foundations for constructive mathematics are available in the literature both in axiomatic set theory à la Zermelo-Fraenkel, such as Aczel's CZF [5, 2–4] or Friedman's IZF [7], or in category theory, such as topoi or pretopoi [17, 16, 18], or in type theory such as Martin-Löf's type theory [26] or Coquand's Calculus of Inductive Constructions [9, 10].

In [23] the second author and G. Sambin embarked on the project of building a minimalist foundation among the most relevant foundations of constructive mathematics. Motivated by this purpose the authors of [23] also argued that a foundation for constructive mathematics should consists of two levels: an intensional level based on an intensional type theory à la Martin-Löf where to make evident the computational contents of mathematical proofs and an extensional level formulated in a language as close as possible to that of present day mathematics and interpreted in the intensional level by means of a quotient model. In particular in [23] it was required that the intensional level should be consistent with the axiom of choice (**AC**) and the formal Church's thesis (**CT**) [31] via a realizability model à la Kleene where to extract the computational contents of proofs. From the well-known fact that **AC**+**CT** is inconsistent with extensionality of functions over Heyting arithmetics extended with finite types [31] it was argued that one single theory can not serve to all the mentioned purposes and that the presence of an extensional level including extensionality of functions as normally assumed in mathematics, is then necessary.

In [19] the second author formulated a two-level system for the Minimalist Foundation, here called **MF**. Both the intensional and the extensional levels of **MF** consist of type systems based on Martin-Löf's type theory [26] but with a primitive notion of propositions in order to avoid the validity of choice principles: the first is called **mTT** and the latter **emTT**.

Here we show that the intensional level **mTT** is consistent with **AC**+**CT** as advocated in [23].

This result is obtained by producing a realizability model for a fragment of Martin-Löf's type theory [26], called $\mathbf{MLtt}_1$, where **mTT**+**AC** can be easily interpreted and where **CT** holds.

Our $\mathbf{MLtt}_1$ is very similar to first order intensional Martin-Löf's type theory with one universe but it differs from Martin-Löf's type theory in the following respects: we drop all congruence laws, in particular the ξ-rule, and instead we postulate the explicit substitution equality rules. Since the main difference between **mTT** and $\mathbf{MLtt}_1$ is the distinction between propositions and sets, to interpret **mTT** in $\mathbf{MLtt}_1$ it sufficies to interpret **mTT**-propositions as $\mathbf{MLtt}_1$-sets according to the well-known Curry-Howard isomorphism and the **mTT**-collection of small propositions as the first universe of $\mathbf{MLtt}_1$.

We then interpret $\mathbf{MLtt}_1$+**CT** by extending Kleene's realizability interpretation of intuitionistic connectives [31] within Feferman's predicative theory of non-iterative fixpoints $\widehat{ID_1}$ in [12, 7]. This means in particular that we interpret the set of functions between natural numbers as the $\widehat{ID_1}$-subset of natural numbers containing their program codes without quotienting them under extensionality so that **CT** turns out to be easily validated.

A key difficulty encountered in producing our realizability interpretation was to use the right interpretation of lambda abstraction in the applicative structure of natural numbers in order to model all the equality rules of $\mathbf{MLtt}_1$ correctly.

Then to interpret the universe of $\mathbf{MLtt}_1$ we adopt the technique of using fixpoints of suitable positive operators due first to Aczel [1] and used by Feferman [12] and Beeson [7] to interpret first order extensional Martin-Löf's type theory with one universe within $\widehat{ID_1}$.

It is worth noting that Beeson's interpretation of first order extensional Martin-Löf's type theory with one universe in [7] does not model $\mathbf{MLtt}_1$+**CT** because it falsifies **CT** since it validates both **AC** and extensionality of functions which both are theorems of extensional Martin-Löf's type theory. Therefore in order to get a version of Martin-Löf's type theory that is consistent with **CT** one has to pass to a version where extensionality of functions is not valid. This can be achieved by dropping the ξ-rule while keeping the other congruence rules as is done in $\mathbf{MLtt}_1$ below. Another option would be to consider the original intensional version of Martin-Löf's type theory in [26] which does not validate extensionality of functions but only a weak form of it expressed by the so-called ξ-rule of lambda terms. But consistency of this version with **CT** is still an open problem. Indeed, the realizability semantics of $\mathbf{MLtt}_1$+**CT** presented here does not work for it, since such a semantics, being based on original Kleene's realizability, does not validate the ξ-rule.

Finally, we want to emphasize, as noticed in [19], that the interpretation of the extensional level of **MF**, including extensionality of function, can be performed within the intensional level **mTT** without the ξ-rule. This is no surprise because partial combinatorial algebras give rise to extensional models and also they do not validate the ξ-rule.

2 The Minimalist Foundation

In [19] a two-level formal system, called *Minimalist Foundation*, for short **MF**, is completed following the design advocated in [23]. The two levels of **MF** are both given by a type theory à la Martin-Löf: the intensional level, called **mTT**, is an intensional type theory including aspects of Martin-Löf's one in [26] (and extending the set-theoretic version in [23] with collections), and its extensional level, called **emTT**, is an extensional type theory including aspects of extensional Martin-Löf's one in [25]. Both type theories include a notion of primitive propositions. The type theory **mTT** of the intensional level can be considered a *predicative version* of Coquand's Calculus of Constructions in [9]. Then a quotient model of setoids à la Bishop [8,13,6,28] over the intensional level is used in [19] to interpret the extensional level in the intensional one. A categorical study of this quotient model has been carried on in [21,20,22] and related to the construction of Hyland's effective topos [14,15].

MF has been designed to be constructive and to be minimalist, that is compatible with (or interpretable in) most relevant constructive and classical foundations for mathematics in the literature. According to these desiderata it has the following features as described in [24]:

- **MF has two types of entities: sets and collections.** A minimalist foundation which is compatible with most of predicative constructive theories in the literature, including Martin-Löf's one in [26], should be certainly predicative and based on intuitionistic predicate logic, including at least the axioms of Heyting arithmetic. For instance it could be a many-sorted logic, such as Heyting arithmetic of finite types [31], where sorts, that we call *types*, include the basic sets we need to represent our mathematical entities.
  However, if we want to develop topology in an intuitionistic and predicative way, we need to equip our foundation with two kinds of entities: *sets* and *collections*. The main reason is that the *power of a non-empty set*, namely the discrete topology over a non-empty set, fails to be a set in a predicative foundation, and it is *only a collection*.

- **MF has two types of propositions.** If a predicative theory is equipped with sets and collections, where the latter include the representation of power-collections of subsets, then, it should also distinguish two types of propositions to remain predicative: those closed under quantifications on sets, called *small propositions* in [19], from those closed under any kind of quantification, called *propositions* in [19].

– **MF has two types of functions.** It is a well-known phenomenon that the *addition of the principle of excluded middle* can turn a predicative theory, as Aczel's CZF or Martin-Löf's type theory, into an impredicative one where *power-collections become sets.* For such theories this happens because the collection of functions from a set A to the boolean set $\{0, 1\}$, i.e. the exponentiation of the boolean set over A, forms a set, too. Hence, a predicative theory compatible with classical theories where the power of a non-empty set is not a set as in Feferman's predicative theories [11], can not validate exponentiation of functions. A drastic solution is to avoid any form of exponentiation in the predicative theory. The solution adopted in **MF** as well as in Feferman's theories [11], is to allow exponentiation of certain functions, called *operations* in [24], which are defined primitively as functional terms $f(x) \in B\ [x \in A]$ in a set B with a free variable in the set A. These operations can be defined as *type-theoretic functions* of a type theory, like in Martin-Löf's type theories [26,25]. Clearly any type theoretic function $f(x) \in B\ [x \in A]$ gives rise to a functional relation $f(x) =_B y\ [x \in A, y \in B]$. But we do not postulate the *axiom of unique choice* which would allow us to identify functional relations with type theoretic functions.
  A proof that **mTT** does not validate the axiom of unique choice can be obtained by interpreting **mTT** in Coquand's Calculus of Constructions which has a model not validating the axiom of unique choice even for natural numbers (see [29]).

2.1 The intensional level of the Minimalist Foundation

Here we describe the intensional level of **MF** in [19], which is represented by a dependent type theory called **mTT**. This type theory is written in the style of Martin-Löf's type theory [26] by means of the following four kinds of judgements:

$$A\ type\ [\Gamma] \qquad A = B\ type\ [\Gamma] \qquad a \in A\ [\Gamma] \qquad a = b \in A\ [\Gamma]$$

that is the type judgement (expressing that something is a specific type), the type equality judgement (expressing when two types are equal), the term judgement (expressing that something is a term of a certain type) and the term equality judgement (expressing the *definitional equality* between terms of the same type), respectively, all under a context Γ.

The word *type* is used as a meta-variable to indicate four kinds of entities: collections, sets, propositions and small propositions, namely

$$type \in \{coll, set, prop, prop_s\}$$

Therefore, in **mTT** types are actually formed by using the following judgements:

$$A\ set\ [\Gamma] \qquad B\ coll\ [\Gamma] \qquad \phi\ prop\ [\Gamma] \qquad \psi\ prop_s\ [\Gamma]$$

saying that A is a set, that B is a collection, that ϕ is a proposition and that ψ is a small proposition.

Here, contrary to [19] where we use only capital latin letters as meta-variables for all types, we use greek letters ψ, ϕ as meta-variables for propositions and capital latin letters A, B as meta-variables for set or collections, and small latin letters a, b, c as meta-variables for terms, i.e. elements of the various types.

Observe that for a set A, when we say that

$$a \in A \ [\Gamma]$$

is derivable in **mTT**, we actually mean that the term *a is an element of the set A under the context* Γ and hence the symbol $\in$ stands for a *set membership.*

In **mTT** as well as in intensional Martin-Löf type theory, one distinguishes between judgemental equality which is decidable and propositional equality which in general is not.

We now proceed by briefly describing the various kinds of types in **mTT**. There is a most general notion of type called *collection* and there is a more restrictive notion called *set. Propositions* are particular collections. Those propositions which are sets are called *small propositions.* Sets are closed under the usual type-forming operations in Martin-Löf type theory without universes. Propositions are closed under usual connectives, quantification over collections and a propositional equality type with an elimination rule which is more restrictive than Martin-Löf's one. Small propositions form a collection $prop_s$ and are closed under the same operations as propositions with the exception of quantifiers and propositional equality type which are admissible only for sets. Collections are closed under Martin-Löf's dependent sum types and the collection of small propositional functions on a set, besides of course the collection $prop_s$.

For a precise formulation containing all the rules see [19].

mTT distinguishes between sets and collections in a way similar to the distinction between set and class in axiomatic set theory. However, all types of **mTT**, i.e. small propositions, propositions, sets and collections, are predicative entities. Indeed, **mTT** can be interpreted into Martin-Löf's type theory (see [26]) by simply identifying **mTT**-collections with Martin-Löf's sets, the collection of small propositions with the first universe, **mTT**-sets with the sets in Martin-Löf's first universe, and propositions are interpreted following Curry-Howard isomorphism.

It is worth noting that, whilst all **mTT**-types are inductively generated by a finite number of rules, the corresponding induction principle is available as an elimination only for all **mTT**-sets with the single exception of dependent product sets where we just have application.

As explained in [23] the distinction between propositions and sets is crucial to avoid the validity of choice principles.

Finally, it is worth noting that in **mTT** usual congruence rules preserving term constructors are restricted to explicit substitution equality rules, called here *replacement rules*, of the form

$$\text{repl)}\ \frac{c(x_1,\ldots,x_n)\in C(x_1,\ldots,x_n)\ \ [\,x_1\in A_1,\ \ldots,\ x_n\in A_n(x_1,\ldots,x_{n-1})\,] \quad a_1=b_1\in A_1\ \ldots\ a_n=b_n\in A_n(a_1,\ldots,a_{n-1})}{c(a_1,\ldots,a_n)=c(b_1,\ldots,b_n)\in C(a_1,\ldots,a_n)}$$

This restriction is crucial to prove consistency of **mTT** with **AC**+**CT** - where **CT** is the usual formulation of formal Church's thesis in first order arithmetics (see [31]) and **AC** is the following form of *axiom of choice*

$$(\mathbf{AC})\ (\forall x\in A)\,(\exists y\in B)\,\rho(x,y)\rightarrow(\exists f\in(\Pi x\in A)\,B)\,(\forall x\in A)\,\rho(x,\mathsf{Ap}(f,x))$$

with A and B generic collections and $\rho(x,y)$ any relation - by means of a realizability semantics we present in the next sections. Indeed, such a semantics, being based on original Kleene realizability in [31], does not validate the ξ-rule[1] of lambda-terms

$$\xi\ \frac{c=c'\in C\ [x\in B]}{\lambda x^B.c=\lambda x^B.c'\in(\Pi x\in B)C}$$

which is instead valid in [26]. However, this restriction does not affect the possibility of adopting **mTT** as the intensional level of a two-level constructive foundation as intended in [23], since its term equality rules suffice to interpret an extensional level including extensionality of functions, as that represented by **emTT**, by means of the quotient model as introduced in [19] and studied abstractly in [20–22].

3 The version $\mathbf{MLtt_1}$ of Martin-Löf's type theory

We here briefly describe the fragment of intensional Martin-Löf's type theory in [26] that we call $\mathbf{MLtt}_1$. In $\mathbf{MLtt}_1$, as in all versions of Martin-Löf's type theory, all entities are sets and the system is described by just using the judgements

$$A\ set\ [\Gamma]\qquad A=B\ set\ [\Gamma]\qquad a\in A\ [\Gamma]\qquad a=b\in A\ [\Gamma]$$

$\mathbf{MLtt}_1$-sets include first order sets with the addition of a single universe U (formulated à la Russell):

$$A\ set\ \equiv\quad N_0\ \ |\ N_1\ \ |\ List(A)\ \ |\ \Sigma_{x\in A}B(x)\ \ |\ A+B\ \ |\ \Pi_{x\in A}\ B(x)\ \ |\ \mathsf{Id}(A,a,b)\ \ |\ U$$

where U is the first universe of first order (or small) sets. We call $\mathbf{MLtt}_0$ the first order set-theoretic part without U.

Then, as for **mTT**, in $\mathbf{MLtt}_0$ we postulate just the replacement rule repl) in place of the usual congruence rules which would include the ξ-rule.

[1] Notice that a trivial instance of the ξ-rule is derivable from repl) when c and c' don't depend on x^B.

This restriction is crucial to get consistency of $\mathbf{MLtt_1}$ with **CT**. Indeed, it is still an open problem whether the original intensional version of Martin-Löf's type theory in [26], including the ξ-rule of lambda terms, is consistent with **CT**.

4 The interpretation of mTT in $\mathbf{MLtt_1}$

The translation of **mTT** into $\mathbf{MLtt_1}$ is almost the identity, as first observed in [19]. We interpret collections of **mTT** as sets of $\mathbf{MLtt_1}$ and sets of **mTT** as sets in the universe U. Propositions of **mTT** are interpreted as particular sets of $\mathbf{MLtt_1}$ and small propositions of **mTT** are interpreted as particular sets in U. The collection of small propositions is interpreted as U.

Notice that this translation is not conservative as it validates the axiom of choice which is not derivable in **mTT**.

5 The theory $\widehat{ID_1}$

Consider the language of second-order arithmetic given by a countable list of individual variables $x_1, ..., x_n...$, a countable list of set variables $X_1, ..., X_n...$, a constant 0, a unary successor functional symbol *succ*, an n-ary functional symbol for every n-ary (definition of a) primitive recursive function, the equality predicate $=$ between individuals, the membership predicate ϵ between individuals and sets, connectives $\wedge, \vee, \rightarrow, \neg$ and quantifiers $\exists$ and $\forall$.

In particular atomic formulas of this language are $t = s$ and $t \,\epsilon\, X$ for individual terms t and s and set variables X.

Let X be a set variable, its occurrence in the atomic formula $t \,\epsilon\, X$ is positive, while an occurrence of X in a non-atomic formula φ is *positive* (*negative* resp.) if one of the following conditions holds (see [7]):

1. φ is $\psi \wedge \rho$ or $\psi \vee \rho$ and the occurrence is positive (negative) in ψ or in ρ;
2. φ is $\psi \rightarrow \rho$ and the occurrence is positive (negative) in ρ or it is negative (positive) in ψ;
3. φ is $\neg\psi$ and the occurrence is negative (positive) in ψ;
4. φ is $\exists x\psi$ or $\forall x\psi$ and the occurrence is positive (negative) in ψ;

A second-order formula $\varphi(x, X)$ is *positive* if it does not contain set quantifiers and it has at most one free individual variable x and at most one free set variable X and all the occurrences of the variable X are positive.

Let's now define the system $\widehat{ID_1}$ introduced in [12] and here called theory of non-iterative fixpoints. It is a first-order classical theory whose language has a countable list of individual variables $x_1, ..., x_n...$, a constant 0, a unary successor functional symbol *succ*, an n-ary functional symbol for every n-ary (definition of a) primitive recursive function, a unary predicate symbol

P_φ for every positive second-order formula $\varphi(x, X)$, the equality predicate $=$, connectives $\wedge, \vee, \rightarrow, \neg$ and quantifiers $\exists$ and $\forall$. [2]

The axioms of $\widehat{ID_1}$ include the axioms of Peano arithmetic (presented with symbols for primitive recursive functions) plus the following axioms schemas:

1. *Induction principle* for every formula of the language of $\widehat{ID_1}$;
2. *Fixpoint schema*: for every positive second-order formula $\varphi(x, X)$ we have

$$P_\varphi(x) \leftrightarrow \varphi(x, P_\varphi)$$

where $\varphi(x, P_\varphi)$ is the formula of $\widehat{ID_1}$ obtained by substituting in $\varphi(x, X)$ all the subformulas $t \,\epsilon\, X$ for some t with $P_\varphi(t)$.

Applicative terms of $\widehat{ID_1}$ and interpretation of λ-abstraction

Recall that *numerals* are defined to be the terms of the form $succ(....(succ(0))...)$.

Definition 1 *Applicative terms* of $\widehat{ID_1}$ are defined according to the following clauses (see [30]):

1. every variable x of $\widehat{ID_1}$ and every numeral $\mathbf{n}$ is an applicative term;
2. if τ and σ are applicative terms, then $\{\tau\}(\sigma)$ is an applicative term.

Definition 2 It τ is an applicative term and $\overline{\sigma}$ is a list of applicative terms, then $\{\tau\}(\overline{\sigma})$ is the applicative term defined according to the following clauses: $\{\tau\}(\,)$ is τ and $\{\tau\}(\sigma_1, ..., \sigma_{n+1})$ is $\{\{\tau\}(\sigma_1, ..., \sigma_n)\}(\sigma_{n+1})$.

Definition 3 Free variables in applicative terms are defined as follows:

1. A variable x' is free in x if and only if it coincides with x;
2. every numeral has no free variables;
3. a variable is free in $\{\tau\}(\sigma)$ if and only if it is free in τ or in σ.

It is well known that in HA (and so a fortiori in $\widehat{ID_1}$) one can encode Kleene application through the so called Kleene predicate $\mathsf{T}(x, y, z)$ and the primitive recursive function U.

Definition 4 For every proposition $P(x)$ if we write $P(\{\tau\}(\sigma))$ for τ and σ applicative terms (that is the result of formally substituting $\{\tau\}(\sigma)$ instead of x in P), this is an abbreviation for the proposition $\exists y(\mathsf{T}(\tau, \sigma, y) \wedge P(\mathsf{U}(y)))$ where y is a fresh variable.

This definition is well given as one can pedantically show by defining an adequate rank for formulas. For example the expression $succ(\{\mathbf{n}\}(\{\mathbf{m}\}(0))) = 0$ stands for $\exists y(\mathsf{T}(\mathbf{n}, \{\mathbf{m}\}(0), y) \wedge succ(\mathsf{U}(y)) = 0)$ which in turn stands for $\exists z(\mathsf{T}(\mathbf{m}, 0, z) \wedge \exists y(\mathsf{T}(\mathbf{n}, \mathsf{U}(z), y) \wedge succ(\mathsf{U}(y)) = 0))$.

Definition 5 Definedness of applicative terms is specified as follows:

[2] We define $\bot$ and $\top$ as abbreviations for $0 = succ(0)$ and $0 = 0$ respectively.

1. $x\downarrow$ and $\mathbf{n}\downarrow$ are $\top$
2. $\{\sigma\}(\sigma')\downarrow$ is $\exists x \mathsf{T}(\sigma,\sigma',x)$

If τ and τ' are applicative terms, then $\tau \simeq \tau'$ is an abbreviation for $\tau\downarrow \vee \tau'\downarrow \rightarrow \tau = \tau'$.

The following lemma is an immediate consequence of the s-m-n lemma (see e. g. [27])

Lemma 1 *There are numerals* $\mathbf{k}$ *and* $\mathbf{s}$ *such that in* $\widehat{ID_1}$ *the following hold*

1. $\{\mathbf{k}\}(x,y) = x$
2. $\{\mathbf{s}\}(x,y)\downarrow$ *and* $\{\mathbf{s}\}(x,y,z) \simeq \{x\}(z,\{y\}(z))$

Definition 6 If x is a variable and τ is an applicative term, then we define the applicative term $\Lambda x.\tau$ as follows:

1. if τ is x, then $\Lambda x.\tau$ is $\{\mathbf{s}\}(\mathbf{k},\mathbf{k})$;
2. if τ does not contain x as free variable, then $\Lambda x.\tau$ is $\{\mathbf{k}\}(\tau)$;
3. otherwise if τ is $\{\sigma\}(\sigma')$ for some applicative terms σ, σ', then $\Lambda x.\tau$ is $\{\mathbf{s}\}(\Lambda x.\sigma, \Lambda x.\sigma')$.

Notice that in general $\Lambda x.\tau\downarrow$ does not hold; however this will not be a problem as it will hold for the interpretations of terms of $\mathbf{MLtt}_1$.

Definition 7 Suppose τ and σ are applicative terms and x is a variable, then we define $\tau[\sigma/x]$ according to the following clauses:

1. if τ is x, then $\tau[\sigma/x]$ is σ
2. if τ is a variable x' different from x or a numeral, then $\tau[\sigma/x]$ is τ
3. if τ is $\{\tau'\}(\tau'')$ for some applicative terms, then $\tau[\sigma/x]$ is $\{\tau'[\sigma/x]\}(\tau''[\sigma/x])$

Notice that the abstraction algorithm defined on p.475 in [31] is different from the one given above. Our choice is motivated by the fact that the abstraction algorithm on p.475 of [31] does not[3] validate the following two lemmas which later turn out as crucial for verifying the correctness of our interpretation of $\mathbf{MLtt}_1$.

Lemma 2 *Let* x, x' *be two distinct variables and let* τ *and* σ *be applicative terms with* x *not free in* σ*. The applicative terms* $(\Lambda x.\tau)[\sigma/x']$ *and* $\Lambda x.(\tau[\sigma/x'])$ *coincide.*

Lemma 3 *If* τ *and* σ *are applicative terms and* x *is a variable, then*

$$\widehat{ID_1} \vdash \sigma\downarrow \rightarrow \{\Lambda x.\tau\}(\sigma) \simeq \tau[\sigma/x].$$

[3] When employing the abstraction algorithm on p.475 of [31] (as suggested on p.609 of *loc.cit.*), one cannot prove Prop. 6.5 on p. 611 of *loc.cit.* because it does not validate the judgement $\mathsf{Ap}(\lambda x.\lambda y.x, \lambda z.0) = \lambda y.\lambda z.0 \in \mathsf{N} \rightarrow (\mathsf{N} \rightarrow \mathsf{N})$ which is derivable in $\mathbf{MLtt}_0$ by β-equality. In order to validate the unrestricted β-rule as required by $\mathbf{MLtt}_0$ one has to use our abstraction algorithm which is a mild variation of the one on p.449 of *loc.cit.*

Notice that in $\widehat{ID_1}$ one can encode pairs of natural numbers through a bijective binary primitive recursive function p with primitive recursive projections p_1 and p_2. Moreover lists of natural numbers can be encoded in such a way that 0 encodes the empty list and the append function $cons$, the length function lh and the component function $(-)_-$ are primitive recursive.

Lemma 4 *There are numerals* $\mathbf{p}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{case}, \mathbf{cons}, \mathbf{rec}$ *such that in* $\widehat{ID_1}$*:*

1. $\{\mathbf{p}\}(x,y) = p(x,y)$, $\{\mathbf{p}_1\}(x) = p_1(x)$ *and* $\{\mathbf{p}_2\}(x) = p_2(x)$;
2. $\{\mathbf{case}\}(0,x,y) = x$ *and* $\{\mathbf{case}\}(1,x,y) = y$;
3. $\{\mathbf{cons}\}(x,y) = cons(x,y)$;
4. $\{\mathbf{rec}\}(x,y,0) = x$ *and* $\{\mathbf{rec}\}(x,y,\{\mathbf{cons}\}(z,u)) \simeq \{y\}(z,u,\{\mathbf{rec}\}(x,y,z))$.

5.1 A universe of sets in $\widehat{ID_1}$

In order to encode a universe of codes of sets closed under standard type constructors of type theory in $\widehat{ID_1}$, we proceed as follows. Codes of sets will be particular natural numbers while a family of sets depending on a set will be encoded by a Gödel number for a recursive function sending elements of the indexing set to codes of sets. The set of codes, the set of families of codes indexed by a set, the membership predicate $x \,\overline{\varepsilon}\, y$ and its auxiliary formal negation $x \not\overline{\varepsilon}\, y$ (needed to keep the clauses positive) are specified by a fixpoint in $\widehat{ID_1}$ given by the subsequent clauses (1)−(21).

In order to make the formulation of these clauses more readable we write n_0 for $p(1,0)$, n_1 for $p(1,1)$, $\sigma(a,b)$ for $p(2,p(a,b))$, $\pi(a,b)$ for $p(3,p(a,b))$, $a \oplus b$ for $p(4,p(a,b))$, $\mathsf{list}(a)$ for $p(5,a)$ and $\mathsf{i}(a,b,c)$ for $p(6,p(a,p(b,c)))$.

1. $\mathsf{Fam}(b,a)$ if and only if $\mathsf{Set}(a) \wedge \forall x\,(x \not\overline{\varepsilon}\, a \vee \mathsf{Set}(\{b\}(x)))$
2. $\mathsf{Set}(\mathsf{n}_0)$ and $\mathsf{Set}(\mathsf{n}_1)$ always hold
3. $\mathsf{Set}(\sigma(a,b))$ if and only if $\mathsf{Fam}(b,a)$
4. $\mathsf{Set}(\pi(a,b))$ if and only if $\mathsf{Fam}(b,a)$
5. $\mathsf{Set}(a \oplus b)$ if and only if $\mathsf{Set}(a) \wedge \mathsf{Set}(b)$
6. $\mathsf{Set}(\mathsf{list}(a))$ if and only if $\mathsf{Set}(a)$
7. $\mathsf{Set}(\mathsf{i}(a,b,c))$ if and only if $\mathsf{Set}(a) \wedge b \,\overline{\varepsilon}\, a \wedge c \,\overline{\varepsilon}\, a$
8. $x \,\overline{\varepsilon}\, \mathsf{n}_0$ never holds
9. $x \,\overline{\varepsilon}\, \mathsf{n}_1$ if and only if $x = 0$
10. $x \,\overline{\varepsilon}\, \sigma(a,b)$ if and only if $\mathsf{Fam}(b,a) \wedge p_1(x) \,\overline{\varepsilon}\, a \wedge p_2(x) \,\overline{\varepsilon}\, \{b\}(a)$
11. $x \,\overline{\varepsilon}\, \pi(a,b)$ if and only if $\mathsf{Fam}(b,a) \wedge \forall y\,(y \not\overline{\varepsilon}\, a \vee \{x\}(y) \,\overline{\varepsilon}\, \{b\}(y))$
12. $x \,\overline{\varepsilon}\, a + b$ if and only if
    $\mathsf{Set}(a) \wedge \mathsf{Set}(b) \wedge ((p_1(x) = 0 \,\wedge\, p_2(x) \,\overline{\varepsilon}\, a) \vee (p_1(x) = 1 \,\wedge\, p_2(x) \,\overline{\varepsilon}\, b))$
13. $x \,\overline{\varepsilon}\, \mathsf{list}(a)$ if and only if $\mathsf{Set}(a) \wedge \forall n\,(n \geq lh(x) \vee (x)_n \,\overline{\varepsilon}\, a)$
14. $x \,\overline{\varepsilon}\, \mathsf{i}(a,b,c)$ if and only if $\mathsf{Set}(a) \wedge b \,\overline{\varepsilon}\, a \wedge c \,\overline{\varepsilon}\, a \wedge x = b \wedge b = c$
15. $x \not\overline{\varepsilon}\, \mathsf{n}_0$ always holds
16. $x \not\overline{\varepsilon}\, \mathsf{n}_1$ if and only if $x > 0$
17. $x \not\overline{\varepsilon}\, \sigma(a,b)$ if and only if $\mathsf{Fam}(b,a) \wedge (p_1(x) \not\overline{\varepsilon}\, a \vee p_2(x) \not\overline{\varepsilon}\, \{b\}(a))$
18. $x \not\overline{\varepsilon}\, \pi(a,b)$ if and only if $\mathsf{Fam}(b,a) \wedge \exists y\,(y \,\overline{\varepsilon}\, a \wedge (\neg\{x\}(y)\downarrow \vee \{x\}(y) \not\overline{\varepsilon}\, \{b\}(y)))$

19. $x \bar{\not\varepsilon}\, a+b$ if and only if
$\mathsf{Set}(a) \wedge \mathsf{Set}(b) \wedge (\neg p_1(x) = 0 \vee p_2(x) \bar{\not\varepsilon}\, a) \wedge (\neg p_1(x) = 1 \vee p_2(x) \bar{\not\varepsilon}\, b)$
20. $x \bar{\not\varepsilon}\, \mathsf{list}(a)$ if and only if $\mathsf{Set}(a) \wedge \exists n\, (n < lh(x) \wedge (x)_n \bar{\not\varepsilon}\, a)$
21. $x \bar{\not\varepsilon}\, \mathsf{i}(a, b, c)$ if and only if $\mathsf{Set}(a) \wedge b\, \bar{\varepsilon}\, a \wedge c\, \bar{\varepsilon}\, a \wedge (\neg x = b \vee \neg b = c)$

Since $\widehat{ID_1}$ allows one only to formulate (not necessarily least) fixpoints we can not prove that the auxiliary formal negation $\bar{\not\varepsilon}$ coincides with the negation of $\bar{\varepsilon}$, i. e.

$$\mathsf{Set}(y) \nvdash_{\widehat{ID_1}} x \bar{\not\varepsilon}\, y \leftrightarrow \neg x\, \bar{\varepsilon}\, y$$

Hence for interpreting the first universe we will restrict to those y in Set for which $\forall x(x \bar{\not\varepsilon}\, y \leftrightarrow \neg x\, \bar{\varepsilon}\, y)$ holds since this collection will turn out to satisfy the required closure properties.

6 Partial interpretation of $\mathbf{MLtt_1}$ syntax

We will interpret types as sets of natural numbers given in terms of predicates formulated in $\widehat{ID_1}$. Types A in a context of length n will be interpreted as sets A^I depending on n-tuples of natural numbers again given by formulas of $\widehat{ID_1}$ in $n+1$ variables. Terms t will be interpreted as applicative terms t^I representing partially defined Gödel numbers for recursive functions between such subsets. A judgement $t \in A\, [x_1 \in A_1, ..., x_k \in A_k]$ holds if $t^I[\overline{n}/\overline{x}]$ is in A^I for all $\overline{n}$ in Γ^I. A judgement $t = s \in A\, [x_1 \in A_1, ..., x_k \in A_k]$ holds if $t^I[\overline{n}/\overline{x}] = s^I[\overline{n}/\overline{x}] \in A^I$ for all $\overline{n}$ in Γ^I. Thus equality judgements are interpreted in an *extensional* way. It will turn out later that functional abstraction is *not validated* by our interpretation.

Notice that we will give a partial interpretation of raw terms and types expressions. Interpretation of types expressions will always be defined whereas interpretation of raw terms will be defined whenever they are well-formed.

Since we have a substitution lemma for applicative terms (see section 5) we also have a substitution lemma for interpretation of raw terms.

6.1 Raw syntax of Martin-Löf type theory

In this section we are going to present the raw syntax of $\mathbf{MLtt}_1$.

Definition 8 Suppose A, B are presets, a, b, c, d, e are preterms and x, y, z are variables then

1. x is a term;
2. N_0 is a preset and $\mathsf{emp}_0(c)$ is a preterm;
3. N_1 is a preset, $\star$ and $\mathsf{El}_{\mathsf{N}_1}(c, e)$ are preterms;
4. $(\Sigma x \in A)B$ is a preset, $\langle a, b\rangle$ and $\mathsf{El}_\Sigma(c, (x, y)\, e)$ are preterms;
5. $(\Pi x \in A)B$ is a preset, $\lambda x.b$ and $\mathsf{Ap}(c, a)$ are preterms;
6. $A + B$ is a preset, $\mathsf{inl}(a)$, $\mathsf{inr}(b)$ and $\mathsf{El}_+(c, (x)\, d, (y)\, e)$ are preterms;
7. $\mathsf{List}(A)$ is a preset, ϵ, $\mathsf{cons}(b, a)$ and $\mathsf{El}_{\mathsf{List}}(c, d, (x, y, z)e)$ are preterms;

8. $\mathsf{Id}(A,a,b)$ is a preset, $\mathsf{id}(a)$ and $\mathsf{El}_{\mathsf{Id}}(a,(x)\,d)$ are preterms;
9. U is a preset, $\widehat{\mathsf{N}}_0$, $\widehat{\mathsf{N}}_1$, $(\widehat{\Sigma}x \in a)b$, $(\widehat{\Pi}x \in a)b$, $a \widehat{+} b$, $\widehat{\mathsf{List}}(a)$ and $\widehat{\mathsf{Id}}(a,b,c)$ are preterms;
10. $\mathsf{T}(a)$ is a preset.

A *precontext* Γ is a list $[x_1 \in A_1, ..., x_n \in A_n]$ where $x_1, ..., x_n$ are pairwise distinct variables and $A_1, ..., A_n$ are presets.

Prejudgements of $\mathbf{MLtt}_1$ are expressions of one of the following forms:

1. $A\, set\, [\Gamma]$
2. $A = B\, set\, [\Gamma]$
3. $a \in A\, [\Gamma]$
4. $a = b \in A\, [\Gamma]$

6.2 The partial interpretation of preterms of $\mathbf{MLtt}_1$

Definition 9 Every preterm of $\mathbf{MLtt}_1$ is interpreted as an applicative term of $\widehat{ID_1}$ as follows:

1. x^I is defined as x
2. $(\mathsf{emp}_0(c))^I$ is defined as 0
3. $\star^I$ is defined as 0 and $(\mathsf{El}_{\mathsf{N}_1}(c,e))^I$ is defined as e^I
4. $(\langle a,b\rangle)^I$ is defined as $\{\mathbf{p}\}(a^I, b^I)$ and $(\mathsf{El}_{\Sigma}(c,(x,y)e))^I$ is defined as $e^I[\{\mathbf{p}_1\}(c^I)/x, \{\mathbf{p}_2\}(c^I)/y]$
5. $(\lambda x.b)^I$ is defined as $\Lambda x.b^I$ and $(\mathsf{Ap}(c,a))^I$ is defined as $\{c^I\}(a^I)$
6. $(\mathsf{inl}(a))^I$ is defined as $\{\mathbf{p}\}(0, a^I)$ and $(\mathsf{inr}(b))^I$ is defined as $\{\mathbf{p}\}(1, b^I)$ and $(\mathsf{El}_{+}(c,(x)d,(y)e))^I$ is defined as $\{\mathbf{case}\}(\{\mathbf{p}_1\}(c^I), d^I[\{\mathbf{p}_2\}(c^I)/x], e^I[\{\mathbf{p}_2\}(c^I)/y])$
7. ϵ^I is defined as 0, $(\mathsf{cons}(b,a))^I$ is defined as $\{\mathbf{cons}\}(b^I, a^I)$ and $(\mathsf{El}_{\mathsf{List}}(c,d,(x,y,z)e))^I$ is defined as $\{\mathbf{rec}\}(d^I, \Lambda x.\Lambda y.\Lambda z.e^I, c^I)$
8. $(\mathsf{id}(a))^I$ is defined as a^I and $(\mathsf{El}_{\mathsf{Id}}(a,(x)d))^I$ is defined as $d^I[a^I/x]$
9. $(\widehat{\mathsf{N}}_0)^I$, $(\widehat{\mathsf{N}}_1)^I$ are defined as $\{\mathbf{p}\}(1,0)$ and $\{\mathbf{p}\}(1,1)$ respectively,
   $((\widehat{\Sigma}x \in a)b)^I$ is defined as $\{\mathbf{p}\}(2, \{\mathbf{p}\}(a^I, \Lambda x.b^I))$
   $((\widehat{\Pi}x \in a)b)^I$ is defined as $\{\mathbf{p}\}(3, \{\mathbf{p}\}(a^I, \Lambda x.b^I))$,
   $(a\widehat{+}b)^I$ is defined as $\{\mathbf{p}\}(4, \{\mathbf{p}\}(a^I, b^I))$,
   $(\widehat{\mathsf{List}}(a))^I$ is defined as $\{\mathbf{p}\}(5, a^I)$ and
   $(\widehat{\mathsf{Id}}(a,b,c))^I$ is defined as $\{\mathbf{p}\}(6, \{\mathbf{p}\}(a^I, \{\mathbf{p}\}(b^I, c^I)))$

Thanks to our coding of lambda-abstraction in terms of applicative terms (see lemma 2), one can easily prove by means of a straightforward verification the following substitution lemma.

Lemma 5 *If a and b are terms and x is a variable which is not bounded in a, then the terms $(\,a[b/x]\,)^I$ and $a^I[\,b^I/x^I]$ coincide.*

6.3 The interpretation of presets and precontexts of $\mathbf{MLtt}_1$

Now we give the interpretation of presets of $\mathbf{MLtt}_1$.

We will interpret presets of $\mathbf{MLtt}_1$ as definable classes of $\widehat{ID_1}$, i. e. formal expressions

$$A = \{x|\,\phi\}$$

where ϕ is a formula of $\widehat{ID_1}$ and x is a variable. If τ is an applicative term of $\widehat{ID_1}$, we will define $\tau\,\varepsilon\,A$ as an abbreviation for $\phi[\tau/x]$.

Definition 10 We will interpret presets of $\mathbf{MLtt}_1$ as follows:

1. $\mathsf{N}_0^I := \{x|\,\bot\}$
2. $\mathsf{N}_1^I := \{x|\,x = 0\}$
3. $((\Sigma y \in A)B)^I := \{x|\,\{\mathbf{p}_1\}(x)\,\varepsilon\,A^I \wedge (\{\mathbf{p}_2\}(x)\,\varepsilon\,B^I)[\{\mathbf{p}_1\}(x)/y]\}$
4. $((\Pi y \in A)B)^I := \{x|\,\forall y\,(y\,\varepsilon\,A^I \rightarrow \{x\}(y)\,\varepsilon\,B^I)\}$
5. $(A{+}B)^I := \{x|\,(\{\mathbf{p}_1\}(x) = 0 \wedge \{\mathbf{p}_1\}(x)\,\varepsilon\,A^I) \vee (\{\mathbf{p}_1\}(x) = 1 \wedge \{\mathbf{p}_1\}(x)\,\varepsilon\,B^I)\}$
6. $(\mathsf{List}(A))^I := \{x|\,\forall i\,(i < lh(x) \rightarrow (x)_i\,\varepsilon\,A^I)\}$
7. $(\mathsf{Id}(A,a,b))^I := \{x|\,x = a^I \wedge a^I = b^I\ \wedge\ a^I\,\varepsilon\,A^I\}$
8. $U^I := \{x|\,\mathsf{Set}(x) \wedge \forall y(y\,\overline{\varepsilon}\,x \leftrightarrow \neg y\,\overline{\not\varepsilon}\,x)\}$
9. $\mathsf{T}(a)^I := \{x|\,x\,\overline{\varepsilon}\,a^I\}$

Precontexts of $\mathbf{MLtt}_1$ will be interpreted as conjunctions of formulas of $\widehat{ID_1}$ as follows.

Definition 11 We define the interpretation of precontexts according to the following clauses:

1. $[\,]^I$ is the formula $\top$;
2. $[\Gamma, x \in A]^I$ is the formula $\Gamma^I\ \wedge\ x^I\,\varepsilon\,A^I$.

6.4 Validity of judgements

Definition 12 We now define validity of judgements J of $\mathbf{MLtt}_1$ in our model:

1. $A\,set\,[\Gamma]$ always holds
2. $A = B\,set\,[\Gamma]$ holds if $\Gamma^I \vdash_{\widehat{ID_1}} \forall x\,(x\,\varepsilon\,A^I \leftrightarrow x\,\varepsilon\,B^I)$
3. $a \in A\,set\,[\Gamma]$ holds if $\Gamma^I \vdash_{\widehat{ID_1}} a^I\,\varepsilon\,A^I$
4. $a = b \in A\,set\,[\Gamma]$ holds if $\Gamma^I \vdash_{\widehat{ID_1}} a^I\,\varepsilon\,A^I \wedge a^I = b^I$

7 Validity theorem

Theorem 1 *If J is a derivable judgement in* $\mathbf{MLtt}_1$*, then J holds in our model.*

Proof The proof proceeds by induction of the structure of derivations in $\mathbf{MLtt}_1$. Most of the cases are trivial. The β-equality judgment follows from lemmas 2 and 3. Generally, equality judgements are immediate from properties of Gödel numbers since equality judgements for terms hold precisely if the denoted numbers are equal. Type equality judgements are verified by showing that the corresponding sets contain the same elements. For working with the universe it is useful to reformulate fixpoint formulas in terms of implications which is possible because $\neg a \vee b$ is equivalent to $a \to b$ due to classical logic in $\widehat{ID_1}$ and $\forall y(y \,\overline{\varepsilon}\, x \leftrightarrow \neg y \not\overline{\varepsilon}\, x)$ for all x in U^I.

Theorem 2 *There exists a numeral* $\mathbf{n}$ *such that* $\widehat{ID_1} \vdash \mathbf{n} \,\varepsilon\, \mathbf{CT}^I$.

Proof This follows from the fact that (the interpretation of) first-order arithmetic in $\mathbf{MLtt}_1$ gets essentially interpreted in our semantics as in Kleene realizability (up to a primitive recursive bijection between different encodings of natural numbers, since natural numbers in $\mathbf{MLtt}_1$ are identified with $\mathsf{List}(\mathsf{N}_1)$).

From these results our main theorem follows immediately.

Theorem 3 *Consistency of* $\widehat{ID_1}$ *implies the consistency of* $\mathbf{MLtt}_1$+**CT** *and of* **mTT+AC+CT**.

Notice that, unfortunately, our realizability model for $\mathbf{MLtt}_1$+**CT** can not be understood as a categorical model although one might think that it correspond to the category of partitioned assemblies (see e. g. [32]). That this is not the case follows from the fact that the operation of Λ-abstraction does not preserves equality since $\{n\}(x) = \{m\}(x)$ might hold for all natural numbers x even if n and m are different natural numbers.

8 Conclusions

We have given an interpretation of **mTT+AC+CT** in $\widehat{ID_1}$ in two steps. First, we translate **mTT+AC+CT** to Martin-Löf's type theory $\mathbf{MLtt}_1$ without the ξ-rule but with Formal Church's thesis **CT**. Then, we interpret the latter within $\widehat{ID_1}$ which amounts to a version of a number realizability interpretation formalized in $\widehat{ID_1}$.

This realizability is very similar to the one in [31] but there is a new contribution. After making clear that the interpretation of λ-abstraction validating the unrestricted β-rule is essentially that on pag. 449 in [31], we have added the interpretation of one universe using the fixpoints provided by $\widehat{ID_1}$ in a way similar to what Beeson did in [7] for an extensional version of Martin-Löf's type theory which, however, is inconsistent with **CT**.

But notice that our interpretation actually validates the equality reflection rule equating propositional and judgemental equality which, however, is weaker than extensional type theory due to the absence of the ξ-rule. Unfortunately, the consistency of Church's Thesis with full Martin-Löf's is still an open problem which to answer presumably is quite difficult.

Other open issues related to this work are the following.

From [7] we are confident that $\mathbf{MLtt}_1$ and $\widehat{ID_1}$ have the same proof-theoretic strength, since the absence of the ξ-rule should not have an impact on this. However it is open whether the strength of **mTT** is weaker and what it precisely is.

Presumably our model of **mTT**+**AC**+**CT** can be extended to an impredicative version, i. e. Inductive Calculus of Constructions+**AC**+**CT** (see [10]), by interpreting collections as partitioned assemblies instead of as subsets of natural numbers.

Acknowledgements The first author thanks the Japan Society for the Promotion of Science (JSPS), Core-to-Core Program (A. Advanced Research Networks) for supporting the research. The second author acknowledges very helpful discussions with Per Martin-Löf and Giovanni Sambin on the topics of this paper. This work was supported by the "National Group for Algebraic and Geometric Structures, and their Applications" (GNSAGA-INDAM) which funded a monthly visit of the last author at the University of Padova, and by the project CPDA157295 funded by the University of Padova

References

1. Aczel, P.: The strength of Martin-Löf's intuitionistic type theory with one universe. In: S. Mietti, S. Vaananen (eds.) Proceedings of Symposia in Mathematical Logic, Oulu, 1974, and Helsinki, 1975, Report No.2, University of Helsinki, Department of philosophy, pp. 1–32 (1977)
2. Aczel, P.: The type theoretic interpretation of constructive set theory. In: Logic Colloquium '77 (Proc. Conf., Wrocław, 1977), *Stud. Logic Foundations Math.*, vol. 96. North-Holland, Amsterdam-New York (1978)
3. Aczel, P.: The type theoretic interpretation of constructive set theory: choice principles. In: D.v.D. Anne Troelstra (ed.) The L.E.J. Brouwer Centenary Symposium (Noordwijkerhout, 1981), *Stud. Logic Foundations Math.*, vol. 110. North-Holland, Amsterdam-New York (1982)
4. Aczel, P.: The type theoretic interpretation of constructive set theory: inductive definitions. In: Logic, methodology and philosophy of science, VII (Salzburg, 1983), *Stud. Logic Foundations Math.*, vol. 114. North-Holland, Amsterdam-New York (1986)
5. Aczel, P., Rathjen, M.: Notes on constructive set theory. (2001). Mittag-Leffler Technical Report No.40
6. Barthes, G., Capretta, V., Pons, O.: Setoids in type theory. J. Funct. Programming **13**(2), 261–293 (2003). Special issue on "Logical frameworks and metalanguages"
7. Beeson, M.: Foundations of Constructive Mathematics. Springer-Verlag, Berlin (1985)
8. Bishop, E.: Foundations of Constructive Analysis. McGraw-Hill Book Co. (1967)
9. Coquand, T.: Metamathematical investigation of a calculus of constructions. In: P. Odifreddi (ed.) Logic in Computer Science, pp. 91–122. Academic Press (1990)
10. Coquand, T., Paulin-Mohring, C.: Inductively defined types. In: P. Martin-Löf, G. Mints (eds.) Proceedings of the International Conference on Computer Logic (Colog '88), *Lecture Notes in Computer Science*, vol. 417, pp. 50–66. Springer, Berlin, Germany (1990)
11. Feferman, S.: Constructive theories of functions and classes. In: M.B. et al. (ed.) Logic Colloquium '78, pp. 159–224. North-Holland (1979)
12. Feferman, S.: Iterated inductive fixed-point theories: application to Hancock's conjecture. In: Patras Logic Symposion, pp. 171–196. North Holland (1982)
13. Hofmann, M.: Extensional Constructs in Intensional Type Theory. Distinguished Dissertations. Springer (1997)

14. Hyland, J.M.E.: The effective topos. In: The L.E.J. Brouwer Centenary Symposium (Noordwijkerhout, 1981), *Stud. Logic Foundations Math.*, vol. 110, pp. 165–216. North-Holland, Amsterdam-New York, (1982)
15. Hyland, J.M.E., Johnstone, P.T., Pitts, A.M.: Tripos theory. Bull. Austral. Math. Soc. **88**, 205–232 (1980)
16. Joyal, A., Moerdijk, I.: Algebraic set theory, *Lecture Note Series*, vol. 220. Cambridge University Press (1995)
17. MacLane, S., Moerdijk, I.: Sheaves in Geometry and Logic. A first introduction to Topos theory. Springer Verlag (1992)
18. Maietti, M.: Modular correspondence between dependent type theories and categories including pretopoi and topoi. Mathematical Structures in Computer Science **15**(6), 1089–1149 (2005)
19. Maietti, M.E.: A minimalist two-level foundation for constructive mathematics. Annals of Pure and Applied Logic **160**(3), 319–354 (2009)
20. Maietti, M.E., Rosolini, G.: Elementary quotient completion. Theory and Applications of Categories **27**(17), 445–463 (2013)
21. Maietti, M.E., Rosolini, G.: Quotient completion for the foundation of constructive mathematics. Logica Universalis **7**(3), 371–402 (2013)
22. Maietti, M.E., Rosolini, G.: Unifying exact completions. Applied Categorical Structures **23**, 43–52 (2015)
23. Maietti, M.E., Sambin, G.: Toward a minimalist foundation for constructive mathematics. In: L. Crosilla and P. Schuster (ed.) From Sets and Types to Topology and Analysis: Practicable Foundations for Constructive Mathematics, no. 48 in Oxford Logic Guides, pp. 91–114. Oxford University Press (2005)
24. Maietti, M.E., Sambin, G.: Why topology in the Minimalist Foundation must be pointfree. Logic and Logical Philosophy **22**(2), 167–199 (2013)
25. Martin-Löf, P.: Intuitionistic Type Theory. Notes by G. Sambin of a series of lectures given in Padua, June 1980. Bibliopolis, Naples (1984)
26. Nordström, B., Petersson, K., Smith, J.: Programming in Martin Löf's Type Theory. Clarendon Press, Oxford (1990)
27. Odifreddi, P.: Classical recursion theory., *Studies in Logic and the Foundations of Mathematics*, vol. 125. North-Holland Publishing Co. (1989)
28. Palmgren, E.: Bishop's set theory (2005). Slides for lecture at the TYPES summer school
29. Streicher, T.: Independence of the induction principle and the axiom of choice in the pure calculus of constructions. Theoretical Computer Science **103**(2), 395–408 (1992)
30. Troelstra, A.S.: Metamathematical Investigation of Intuitionistic Arithmetic and Analysis. Springer (1973)
31. Troelstra, A.S., van Dalen, D.: Constructivism in mathematics, an introduction, vol. I and II. In: Studies in logic and the foundations of mathematics. North-Holland (1988)
32. van Oosten, J.: Realizability: an introduction to its categorical side, *Studies in Logic and Foundations of Mathematics*, vol. 152. Elsevier (2008)