Università degli studi di Padova

Dipartimento di Matematica
Corso di Laurea Magistrale in Matematica

Hedberg's Theorem in the Minimalist Foundation

19th April 2019

Student:
Riccardo Borsetto
Matricola 1159950

Supervisor:
Maria Emilia Maietti

Contents

Chapter 1

Abstract

The Minimalist Foundation is a foundation for constructive mathematics obtained from Martin-Löf's type theory by defining primitive propositions so that choice principles are not validated.

It was ideated by Maria Emilia Maietti and Giovanni Sambin in 2005 ([8]) and completed to a two-level system by Maietti in 2009 ([7]).

In this thesis we show that in the intensional level of the Minimalist Foundation (for short mTT) we can prove Hedberg's theorem stating that types whose propositional equality is decidable enjoy the principle of uniqueness of identity proofs.

The proof is obtained by adapting the original proof by Hedberg in [13] to the Minimalist Foundation. As a consequence we show that no choice principle is needed to prove the theorem.

It is important to have shown this theorem for mTT because it is expected to imply - as in Homotopy Type Theory - that the classical principle of excluded middle is not valid in the extension of the Minimalist Foundation with the addition of a collection of sets and Vladimir Voevodsky's Univalent Axiom - stating an equivalence between isomorphisms on a set with the set of identity proofs of the equality of the set with itself (thought of as a code of a universe).

This thesis is divided into two parts: a cultural section (chapters 2, 3 and 4), written by Maietti and taken from [4], explaining the origins and motivations of type theory and its different versions and a section (chapter 5) introducing the preliminaries to then demonstrate Hedberg's theorem inspired by what was done in Homotopy Type Theory (see [13]).

Chapter 2

The Type Theory

Today's type theory is a branch of both mathematics and computer science. The main object of study of the type theory is the formal system which classify the mathematical entities through the "types" and their elements called "terms". It was originally introduced by Bertrand Russell in early twentieth century as a reliable foundation of mathematics in response to the contradictions noted in some set formulations.

2.1 Distinctive properties of the type theory

We underline some characteristics that distinguish the type theories from the theories of axiomatic sets in the style of Zermelo-Fraenkel.

Distinction between types and their elements. The main novelty of the type theory with respect to the theories of axiomatic Zermelo-Fraenkel ensembles, including those in constructive version, is that while in the latter both the mathematical entities and their elements are without distinction sets (for example the number "3" is a set as much as the set of natural numbers), in type theory the elements of the types generally are not types and are described separately, and evenly the equalities between types and between terms are described separately, even if in a mutually recursive way between them.

Primitive definition of function of Church lambda-calculus.

Another very relevant aspect that distinguishes many of the current type theories from set theory is the adoption of the notation of Church Lambda-calculus that allows to define functions primitively as lambda-terms by associating them with a type that establishes the domain and codomain. If the type theory has enough constructs to interpret the logic of the first order with the possibility of defining functional relations, then one can ask whether any functional relation $R(x, y)$ $(x \in A, y \in B)$ definable in the theory between two types A, B describes the graph of a lambda-function with domain A and codomain B. The principle that affirms that every functional relation is the graph of a lambda-function goes under the name of axiom of unique choice and it is not derivable in all the type theories (for example it is not derivable in the "Minimalist Foundation" described later.

Multiple interpretations of type theory: set, computation, logic. Some type theories admit more than one interpretation which explains the nature of their types and terms:

- A set interpretation according to which types represent sets and their terms the corresponding elements. This interpretation allows to consider the type theory as a set theory.
- A computational interpretation according to which types are seen as types of. programming language data and their terms as programs which produce outputs of their data type. This interpretation allows to consider the type theory as a programming language.
- A logical interpretation according to which types represent propositions and their terms encodings of their demonstrations. This interpretation allows to consider the type theory as a logical calculation.

Typically the type theories admit computational interpretation for all types while only some types are seen as sets and still others like propositions.

Propositions as types. The reading mentioned in the previous entry according to which logical propositions can be represented as types of theirs demonstrations coded by appropriate terms was introduced by H. Curry. It allows us to represent logical connectives like type builders. For example the implication corresponds to the type of lambda-calculus functions between the type of the antecedent and that of the consequent. In addition, the abstraction operation of a proposition in a proof that allows us to deduce an implication corresponds to the lambda-abstraction of the Church lambda-calculus relative to the term encoding the initial derivation.

The correspondence between propositions and types and between proofs of a proposition and typed terms is very natural if the Prawitz natural deduction formalism is adopted as a formal system for propositional deductions.

Predicates as types.

An important novelty of Russell's type theory with respect to set theory it was to adopt Frege's idea of representing a predicate $P(x)$ as a propositional function from a domain of definition of the variable x, represented as a type, with values in a type of all the propositions.

If the type theory admits the definition of a single proposition as a type in the sense of Curry, then the type of propositions is actually a type of types, sometimes called "universe" of propositions.

Further developing this idea, W. Howard and N. de Bruijn first and P. Martin-Löf then introduced in type theory the representation of predicates as dependent types. This representation has in turn inspired a great innovation inconceivable in the formalisms of logic and set theory then known. This novelty consists in defining predicates that depend on types that are themselves predicates or propositions or predicates dependent on proofs of propositions/predicates coded as terms.

Consistent with the representation of predicates as dependent types also the universal and existential quantifications and even propositional equality can be defined in type theory as constructors of dependent types. In particular the intensional representation of the type of propositional equality in Martin-Löf's Type Theory allowed to establish important links between the type theory and the theory of topographical homotopia that we will briefly deal with in following.

Chapter 3

Martin-Löf's Type Theory

In the 1970s, Per Martin-Löf introduced a type theory called **Intuitionistic Type Theory**. Currently this theory is simply called **Martin-Löf's Type Theory** and in the following we will refer to it with **MLTT**.

A relevant aspect of the **MLTT** theory that has attracted the interest of both computer scientists and mathematicians as well as philosophers and linguists, and in recent years also of a medal winner Fields as Vladimir Voevodsky, is described in a publication by Martin-Löf from 1982 entitled "Computer programming and constructive mathematics" in which the author proposes his type theory of both as a paradigm of a programming language and at the same time as a set theory suitable for formalizing constructive mathematics, for example constructive analysis developed by E. Bishop, with the possibility of extracting the computational content of constructive demonstrations through programs. This double identity of **MLTT** as a "foundation for mathematics" and "(functional) programming language" inspired the introduction of programs called proof-assistant able to help a user to formalize a math demonstration within computer using the **MLTT** language.

Description of types and terms through judgments. Martin-Löf's type theory **MLTT** is not described as a theory of first-order logic such as for example the set theory of Zermelo-Fraenkel. **MLTT** is instead described in a primitive way through four main forms of judgments

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

to which a judgment is added to derive the Γ contexts of the form

$$\Gamma\ cont$$

The contexts include the empty list and lists of assumptions in the form

$$x_1 \in A_1, x_2 \in A(x_1), ...x_n \in A_n(x_1, ..., x_{n_1})$$

with a telescopic trend, in the sense that in such lists the first assumption a left consists of a variable x_1 typed with a closed type A_1, while the second assumption consists of a variable x_2 typed with a type $A(x_1)$ which may depend from the assumption x_1 and so on up to the last assumption given by the variable x_n which is typed with a type that may depend on all previous assumptions.

Furthermore the meaning of the four main forms of judgments is the following:

- the judgement A *type* $[\Gamma]$ states that A is a type in the context Γ;
- the judgement $A = B$ *type* $[\Gamma]$ states that the type A is equal by definition to the type B in the context Γ;
- the judgement $a \in A$ $[\Gamma]$ states that the term a is of type A in the context Γ;
- the judgement $a = b \in A$ $[\Gamma]$ states that the term a of type A is equal by definition to the term b also of type A in the context Γ.

The type theory then consists of inference rules to derive Γ and then contexts judgments in the forms listed above. These judgments of Martin-Löf's Type Theory admit at least two interpretations: a computational and a set one. They also allow an interpretation of the connectives and quantifiers of intuitionistic logic.

According to the computational interpretation the types are identified as types of data and their elements as programs dependent on context input.

According to the set interpretation, the types not depending on a context represent sets and context-dependent types represent families of sets indexed by the context while their elements are simply seen as elements of the set or family of sets indexed by the context.

The interpretation of logical connectives and logical quantifications given by Martin-Löf in his type theory is an extension of that given by Curry and Howard. According to this interpretation some types represent propositions dependent on a context, that is, they represent predicates and their elements are proof-term or encodings of their demonstrations depending on the assumptions of the context on which they are defined.

The variety of interpretations of the concept of type and its elements presented is at the base of the connections of the theory of types with logic, computer science and math foundation.

3.1 Set theory in the Martin-Löf's Type Theory.

We will underline all the characteristic aspects of the Martin-Löf type theory as a set theory of constructive mathematics.

Predicativity of the Martin-Löf Type Theory.

All types of **MLTT** in its various versions are predicative in the sense that they are all inductively generated by constructors to whom it is associated an induction principle with a recursive scheme to define functions from the type introduced to values in types already defined previously.

The inductive definition scheme of **MLTT** types and terms follows that of the natural deduction of Gentzen-Prawitz through introduction rules to which they are associated, according to a uniform scheme, rules of elimination and rules of definitive equality of the manufacturers introduced.

Two forms of equality between terms. A novelty peculiar to Martin-Löf's type theory is the distinction of two forms of equality between terms and types. We have already mentioned that, in MLTT, given two terms of type A under a certain context

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

we can express on them a judgment of definitional equality

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

which, we recall, also says that the terms a and b are computationally equal. But given the aforementioned terms within MLTT we can also form a type dependent on propositional equality

$$Id(A, a, b)\ type\ [\Gamma]$$

which is to be considered as a real proposition whose elements denote proofs of the proposition of equality of the term a with b. In particular if we can derive a proof-term p of this type, or if we succeed to derive a judgment of the type in the theory

$$p \in Id(A, a, b)\ [\Gamma]$$

then we can conclude that the term a is propositionally equal to the term b.

Martin-Löf introduced two different versions of his type theory

- an intensional version;
- an extensional version.

that are distinguished precisely by the rules of propositional equality in relation to the fact of being able to make equivalent the derivability of definitional equality $a = b \in A\ [\Gamma]$ between two terms $a \in A\ [\Gamma]$ and $b \in A\ [\Gamma]$ with validity of their propositional equality through a proof-term, for example p, for which derives $p \in Id(A, a, b)\ [\Gamma]$.

The extensional version of the Martin-Löf's Type Theory.
This version is characterized by the fact that definitional equality of two terms is equivalent to the validity of the relative propositional equality in the same terms.

The intensional version of the Martin-Löf's Type Theory.
This version is called intensional in that the definitional equality of two terms only implies the existence of a proof-term of the propositional equality relative to the same terms and

admits examples of terms that are equal from a propositional point of view but are not definitional equal or in other words they are not computationally identical. Another feature of the intensional version is that it is not said that proof-terms of the propositional equality type are unique, that is, the so-called proof-irrelevance holds for the type propositional equality; indeed for this type it is said that is proof-relevance.

Dependence of the equality type on other equality types. A further peculiar characteristic of the propositional equality type introduced by Martin-Löf in all its versions is that the type propositional equality may depend on the proof-term of any proposition, including another propositional equality type. In other words we can consider the propositional equality type of the propositional equality type of two terms

$$Id(Id(A, a, b), p, q)\ type\ [\Gamma]$$

and then the propositional equality type of the propositional equality type of an other equality type of two terms

$$Id(Id(Id(A, a, b), p, q), l, k)\ type\ [\Gamma]$$

and so on by iterating the propositional equality type at will. This dependence gives rise to a sort of infinitive structure of weak groupoid.

Chapter 4

Univalent Foundation and Homotopy Type Theory

Vladimir Voevodsky in 2006 built a model of Martin-Löf's intensional type theory in simplicial sets. This model, called "homotopic", exalts in an original way two peculiar characteristics of Martin-Löf's propositional equality type

- its proof-relevance;
- the possibility of forming the propositional equality type of other propositional equality types thus allowing to associate each type with one infinitary structure of weak groupoid.

According to the homotopic model the types are interpreted as types of homotopy of topological spaces and propositions are homotopic types of their proofs. In particular, this model valids an axiom called univalence that is the main feature of the Univalent Foundation proposed by Voevodsky. The idea of the Univalent Foundation has inspired the introduction of the **Homotopy Type Theory**, in short **HoTT**, as extension of the Martin-Löf type theory with the axiom of univalence which essentially it guarantees that isomorphic structures can be considered equal. This theory appears ideal for developing a synthetic version and constructive theory of classical homotopy since the axiom of univalence essentially asserts that the propositional equality type of two types thought as topological spaces is homotopically equivalent to homotopic equivalences between the two types.

An important advancement in type theory offered by the Homotopy Type Theory is the conception of the so-called higher inductive types built according to an inductive generation scheme that besides generating the elements of the new type it also generates elements of its propositional equality type that represents the homotopic theory. In particular these types include the generation of specific types of quotients.

Chapter 5

The Minimalist Foundation

The Minimalist Foundation, in brief **MF**, is a formal two-tier system conceived by Maria Emilia Maietti and Giovanni Sambin in 2005. It is intended as a foundation basic for constructive mathematics that has the property of being compatible with the most important constructive foundations known in the literature. Both of its levels are Martin-Löf dependent type theories. In the following we set out some reasons motivating the construction of **MF**, his main peculiarities and in what **MF** differs from the Martin-Löf's Type Theory.

Why introduce a minimalist foundation. The idea of building one foundation for constructive mathematics different from those in literature and also equipped with two levels, it is mainly due to two needs:

- the need to introduce a formal language that is as similar as possible to that of informal mathematical practice without giving up interpreting it in natural way in an intensional type theory, like that of Martin-Löf, whose demonstrations have an obvious computational content and can be easily verified at the computer by a proof-assistant;
- the need to provide the constructive mathematics of a minimalist foundation compatible with the most relevant constructive foundations known in the literature, given the absence of a common reference foundation for constructive mathematicians as the set theory of Zermelo-Fraenkel for classic mathematicians and the presence of constructively acceptable principles in some constructive foundations but not in others.

Concept of two-level foundation.
In order to satisfy these needs it was proposed as a notion of a foundation for constructive mathematics a formal two-tier system with:

1. an intensional level consisting of an intensional theory with a computational interpretation that makes evident the extraction of the computational content of his demonstrations; this level, if based on a type theory like the intensional one of Martin-Löf,

can be thought of as the basis of a proof-assistant for interactive computer formalization of his tests;

2. an extensional level consisting of a set theory in a language close to the usual mathematical practice and equipped with an interpretation in the intensional level able to restore at the intensional level the computational information not present at the extensional level, thus proving that the extensional level is obtained by abstraction from intensional one according to the principle of forget-restore. More specifically, as in the example of the Minimalist Foundation, it is required that the sets of the extensional level result interpretable in the intensional level as quotients of sets of the intensional level and in particular the predicates of the extensional level result interpretable as trivial quotients of the corresponding predicates at the intensional level.

This division seems to recall the two versions of Martin-Löf's Type Theory. In fact, both the intensional and the extensional level of the Minimalist Foundation are described as particular theories of the Martin-Löf types based on its intensional and extensional versions respectively.

Main differences between the type theories MF and the Martin-Löf's Type Theory. We briefly mention some distinctive features of MF compared to the Martin-Löf's theory:

- both **MF** levels are compatible with predicative and classical theories, in the sense that their extensions with the principle of excluded middle are still predicative, contrary to the Martin-Löf type theory that with the addition of classical principles becomes impredicative;

- both **MF** levels are independent from choice axioms and choice rules, including the unique choice axiom and the unique choice rule, as a result of the fact that the propositions are defined as primitive types;

- the intensional **MF** type theory can be interpreted in the most relevant theories of known intensional types preserving the intended meaning of sets and propositions;

- the extensional **MF** type theory is interpretable in the most relevant ones set theories known as foundations for constructive mathematics or classical preserving the intended meaning of sets and propositions;

- the intensional level of **MF** is predicative in the sense of S. Feferman about the order sufficient to prove its consistency;

- **MF** is intended as a basic system for developing reverse mathematics for constructive mathematics; therefore **MF** is not intended as a foundation exhaustive to develop all possible constructive mathematics but rather like a minimalist foundation to be extended appropriately with any extra set construction necessary for development desired mathematician (in particular with all the possible definitions of inductive and coinductive sets and propositions useful for a predicative formalization of constructive mathematics).

5.1 The intensional level mTT

The typed calculus mTT is written in the style of Martin-Löf's Type Theory by means of the usual four kinds of judgements, 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 precise rules of mTT are given in [7]. Types include collections, sets, propositions and small propositions and hence the word type is only used as a meta-variable, namely

$$type \in \{col, set, prop, props\}$$

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

$$A\ set\ [\Gamma] \qquad A\ col\ [\Gamma] \qquad A\ prop\ [\Gamma] \qquad A\ prop_s\ [\Gamma]$$

The general idea is to define a many-sorted logic, but now sorts include both sets and collections. The main difference between sets and collections is that sets are those collections that are inductively generated, namely those whose most external constructor is equipped with introduction and elimination rules, and all of their collection components are so. According to this view it is allowed that elimination rules of sets act also towards collections. These sets will be closed under the empty set, the singleton set, strong indexed sums, dependent products, disjoint sums, lists. These constructors are formulated as in Martin-Löf's type theory with the modification that their elimination rules vary on all types. In order to view sets as collections, we add the rule **set-into-col**

$$\frac{A\ set}{A\ col}$$

The logic of the theory is described by means of propositions and small propositions. Small propositions are those propositions closed only under intuitionistic connectives and quantification over sets. To express that a small proposition is also a proposition we add the subtyping rule **props-into-prop**

$$\frac{A\ prop_s}{A\ prop}$$

Since we restrict our consideration only to mathematical propositions, it makes sense to identify a proposition with the collection of its proofs. To this purpose we add the rule **prop-into-col**

$$\frac{A\ prop}{A\ col}$$

However, proofs of small propositions are inductively generated. Hence, small propositions, are though of as sets of their proofs by means of the rule **props-into-set**

$$\frac{A\ prop_s}{A\ set}$$

The rules **props-into-set** and **prop-into-col** allow us to form the strong indexed sum of a small propositional function $\phi(x)\ prop_s\ [x \in A]$, or simply of a propositional function,

$$\sum_{x \in A} \phi(x)$$

both on sets and on collections. Given that we will define a subset as the equivalence class of a small propositional function, then the props-into-set rule is relevant to turn a small propositional function on a set into a set, and hence to represent functions between subsets and to represent families indexed on a subset. The same can be said about subcollections. Moreover, the identification of a proposition with the collection (or set) of its proofs allows also to derive all the induction principles for propositions depending on a set, because set elimination rules can act towards all collections including propositions.

Below we give some examples of mTT entities, recalling that the entire mTT system along with its rules can be found in [7] and for easiness, the piece of context common to all judgements involved in a rule is omitted and typed variables appearing in a context are meant to be added to the implicit context as the last one.

5.2 True proposition

Between the two truth values, only the false proposition is predefined in mTT. Thus, we show that the true proposition can be constructed with the rules of mTT.

We define the true proposition as $\top := \bot \to \bot$ while its formation rule is

$$\text{F-}\top) \quad \top\ prop_s$$

and the introduction rule is

$$\text{I-}\top) \quad \star \in \top$$

5.3 Propositional Equality

Another type of proposition is the propositional equality. There are various ways to describe the propositional equality and one of these is Leibniz propositional equality whose elimination rule is a restricted form of the elimination rule of the Martin-Löf propositional equality:

Leibniz Propositional Equality

$$\text{F-Id)} \quad \frac{A\ col \qquad a \in A \qquad b \in A}{Id(A,a,b)\ prop}$$

$$\text{I-Id)} \quad \frac{a \in A}{id_A(a) \in Id(A,a,a)}$$

$$\text{E-Id)} \quad \frac{C(x,y)\ prop\ [x \in A, y \in A] \qquad a \in A \qquad b \in A \qquad p \in Id(A,a,b) \qquad c(x) \in C(x,x)\ [x \in A]}{El_{Id}(p,c) \in C(a,b)}$$

$$\text{C-Id)} \quad \frac{C(x,y)\ prop\ [x \in A, y \in A] \qquad a \in A \qquad c(x) \in C(x,x)\ [x \in A]}{El_{Id}(id_A(a),c) = c(a) \in C(a,a)}$$

To interpret logic in type theory, and in particular to make theorems valid with equality of intuitionistic predicative logic, it is sufficient to interpret predicative equality with the propositional equality of Leibniz described above, that is the predefined propositional equality. However, the elimination of Leibniz equality is not exactly the associated inductive elimination to its introduction rules. The propositional equality formulated by Martin-Löf has an elimination rule that approximates better the induction associated with the introduction rule of the propositional equality. This equality is described by the following rules:

Martin-Löf's Propositional Equality

$$\text{F-Id}_{\text{ML}}) \quad \frac{A\ col \qquad a \in A \qquad b \in A}{Id_{ML}(A,a,b)\ prop}$$

$$\text{I-Id}_{\text{ML}}) \quad \frac{a \in A}{id_A(a) \in Id_{ML}(A,a,a)}$$

$$\text{E-Id}_{\text{ML}})\quad \frac{C(x,y,z)\ prop\ [x\in A, y\in A, z\in Id_{ML}(A,x,y)] \qquad a\in A \qquad b\in A \qquad p\in Id_{ML}(A,a,b) \qquad c(x)\in C(x,x,id_A(x))\ [x\in A]}{El_{Id_{ML}}(p,c)\in C(a,b,p)}$$

$$\text{C-Id}_{\text{ML}})\quad \frac{C(x,y,z)\ prop\ [x\in A, y\in A, z\in Id_{ML}(A,x,y)] \qquad a\in A \qquad c(x)\in C(x,x,id_A(x))\ [x\in A]}{El_{Id_{ML}}(id_A(a),c)=c(a)\in C(a,a,id_A(a))}$$

There is another description of the propositional equality, called Propositional Equality with Path Induction, whose rules include an elimination rule that turns out to be the rule of elimination associated with the introduction rule:

Propositional Equality with Path Induction

$$\text{F-Id}_{\text{p}})\quad \frac{A\ col \qquad a\in A \qquad b\in A}{Id_p(A,a,b)\ prop}$$

$$\text{I-Id}_{\text{p}})\quad \frac{a\in A}{id_A(a)\in Id_p(A,a,a)}$$

$$\text{E-Id}_{\text{p}})\quad \frac{C(y,z)\ prop\ [y\in A, z\in Id_p(A,a,y)] \qquad a\in A \qquad b\in A \qquad p\in Id_p(A,a,b) \qquad c\in C(a,id_A(a))}{El_{Id_p}(p,c)\in C(b,p)}$$

$$\text{C-Id}_{\text{p}})\quad \frac{C(y,z)\ prop\ [y\in A, z\in Id_p(A,a,y)] \qquad a\in A \qquad c\in C(a,id_A(a))}{El_{Id_p}(id_A(a),c)=c\in C(a,id_A(a))}$$

Definition 5.3.1 (Equivalence of types)**.** *A type A is said to be equivalent to a type B in mTT if we can derive two terms $pf_1\in B\ [x\in A]$ and $pf_2\in A\ [y\in B]$.*

Definition 5.3.2 (Isomorphism of types)**.** *A type A is said to be isomorphic to a type B in mTT if two terms*

$$f(x)\in B\ [x\in A] \qquad g(y)\in A\ [y\in B]$$

can be derived such that there are proof-term pf_1 and pf_2 whereby they are derived:

$$pf_1\in Id(A,x,g(f(x)))\ [x\in A] \qquad pf_2\in Id(B,y,f(g(y)))\ [y\in B]$$

Theorem 5.3.1. *Leibniz Propositional Equality, Martin-Löf's Propositional Equality and Propositional Equality with Path Induction are equivalent.*

Proof. We prove this statement in a circular way, firstly we show that given a term for Leibniz Propositional Equality we have one for Martin-Löf's Propositional Equality

$$\text{E-Id)}\quad \frac{Id_{ML}(A,x,y)\ prop\ [x \in A, y \in A] \qquad a \in A \qquad b \in A \qquad p \in Id(A,a,b) \qquad id_A(x) \in Id_{ML}(A,x,x)\ [x \in A]}{El_{Id}(p,(x).id_A(x)) \in Id_{ML}(A,a,b)}$$

Next we show that given a term for Martin-Löf's Propositional Equality we have one for Propositional Equality with Path Induction

$$\text{E-Id}_{\text{ML}})\quad \frac{Id_p(A,x,y)\ prop\ [x \in A, y \in A, z \in Id_{ML}(A,x,y)] \qquad a \in A \qquad b \in A \qquad q \in Id_{ML}(A,a,b) \qquad id_A(x) \in Id_p(A,x,x)\ [x \in A]}{El_{Id_{ML}}(q,(x).id_A(x)) \in Id_p(A,a,b)}$$

In the end we show that given a term for Propositional Equality with Path Induction we have one for Leibniz Propositional Equality

$$\text{E-Id}_{\text{p}})\quad \frac{Id(A,a,y)\ prop\ [y \in A, z \in Id_p(A,a,y)] \qquad a \in A \qquad b \in A \qquad r \in Id_p(A,a,b) \qquad id_A(a) \in Id(A,a,a)}{El_{Id_p}(r, id_A(a)) \in Id(A,a,b)}$$

□

Theorem 5.3.2. *Martin-Löf's Propositional Equality and Propositional Equality with Path Induction are isomorphic.*

Proof. We call

$$f(r) := El_{Id}(El_{Id_p}(r, id_A(a)), (x).id_A(x)) \in Id_{ML}(A,a,b)\ [r \in Id_p(A,a,b)]$$
$$g(q) := El_{Id_{ML}}(q,(x).id_A(x)) \in Id_p(A,a,b)\ [q \in Id_{ML}(A,a,b)]$$

Firstly, for $r \in Id_p(A,a,b)$ we construct the following prooftree:

$$\text{E-Id}_{\text{p}})\quad \frac{Id(Id_p(A,a,y),z,g(f(z)))\ prop\ [y \in A, z \in Id_p(A,a,y)] \qquad a \in A \qquad b \in A \qquad r \in Id_p(A,a,b) \qquad \pi_1}{El_{Id_p}(r,c) \in Id(Id_p(A,a,b),r,g(f(r)))}$$

where π_1 is the prooftree easily constructed using C-Id), subT) and conv) and knowing that

$$id_{Id_p(A,a,a)}(id_A(a)) \in Id(Id_p(A,a,a), id_A(a), id_A(a))$$

in order to obtain that $id_{Id_p(A,a,a)}(id_A(a))$ is also a term for $Id(Id_p(A,a,a), id_A(a), g(f(id_A(a))))$. Conversely we have:

$$\text{E-Id}_{\text{ML}})\quad \frac{Id(Id_{ML}(A,x,y), z, f(g(z)))\ prop\ [x \in A, y \in A, z \in Id_{ML}(A,x,y)] \qquad a \in A \qquad b \in A \qquad q \in Id_{ML}(A,a,b) \qquad \pi_2}{El_{Id_{ML}}(q, (x).c(x)) \in Id(Id_{ML}(A,a,b), q, f(g(q)))}$$

where π_2 is the prooftree easily constructed using C-Id), subT) and conv) and knowing that

$$id_{Id_{ML}(A,x,x)}(id_A(x)) \in Id(Id_{ML}(A,x,x), id_A(x), id_A(x))\ [x \in A]$$

in order to obtain that $id_{Id_{ML}(A,x,x)}(id_A(x))$ is also a term for $Id(Id_{ML}(A,x,x), id_A(x), f(g(id_A(x)))\ [x \in A]$. □

Theorem 5.3.3. *Equivalence of Martin-Löf's Propositional Equality and Propositional Equality with Path Induction rules*

Proof. First of all we see that formation and introduction rules are exactly the same. Now, one of the hypothesis of E-Id$_{\text{ML}}$) and C-Id$_{\text{ML}}$) is that $C(x,y,z)$ is a proposition for every $x, y \in A$ and $z \in Id_{ML}(A,x,y)$ so clearly we have that also $C(a,y,z)$ with $a \in A$ fixed.

$$\text{E-Id}_{\text{p}})\quad \frac{C(a,y,z)\ prop\ [y \in A, z \in Id_{ML}(A,a,y)] \qquad a \in A \qquad b \in A \qquad p \in Id_p(A,a,b) \qquad c \in C(a,a,id_A(a))}{El_{Id_p}(p,c) \in C(a,b,p)}$$

$$\text{C-Id}_{\text{p}})\quad \frac{C(a,y,z)\ prop\ [y \in A, z \in Id_{ML}(A,a,y)] \qquad a \in A \qquad c \in C(a,a,id_A(a))}{El_{Id_p}(id_A(a),c) = c \in C(a,a,id_A(a))}$$

So we have shown that Propositional Equality with Path Induction rules imply Martin-Löf's Propositional Equality rules. Then, the converse:

$$\text{E-}\to)\quad \frac{c \in C(a, id_A(a)) \qquad \pi_1}{Ap_{\to}(El_{Id_{ML}}(p, \lambda_{\to} w^{C(x,id_A(x))}.w), c) \in C(b,p)}$$

where π_1 is

$$\text{E-Id}_{\text{ML}})\quad \frac{C(x, id_A(x)) \to C(y,z)\ prop\ [x \in A, y \in A, z \in Id_{ML}(A,x,y)] \qquad a \in A \qquad b \in A \qquad p \in Id_{ML}(A,a,b) \qquad \pi_2}{El_{Id_{ML}}(p, \lambda_{\to} w^{C(x,id_A(x))}.w) \in C(a, id_A(a)) \to C(b,p)}$$

and π_2 is

$$\text{I-}\to)\quad \frac{w \in C(x, id_A(x))\ [x \in A, w \in C(x, id_A(x))]}{\lambda_{\to} w^{C(x,id_A(x))}.w \in C(x, id_A(x)) \to C(x, id_A(x))\ [x \in A]}$$

while the conversion rule is

$$\beta\text{C-}\to)\quad \frac{c \in C(a, id_A(a)) \qquad w \in C(a, id_A(a))\ [w \in C(a, id_A(a)] \qquad C(a, id_A(a))\ prop}{Ap_{\to}(\lambda_{\to} w^{C(a,id_A(a))}.w, c) = c \in C(a, id_A(a))}$$

□

We will omitt subscripts ML and p while dealing with Martin-Löf's Propositional Equality and Propositional Equality with Path Induction since they are equivalent to Leibniz Propositional Equality.

5.3.1 Properties

Below we give some useful properties of the propositional equality.

Lemma 5.3.1 (Inverse)**.** *For every collection A, every $a, b \in A$ and $p \in Id(A, a, b)$ there is a term $p^{-1} \in Id(A, b, a)$, such that $(id_A(a))^{-1} = id_A(a) \in Id(A, a, a)$ for each $a \in A$.*

Proof. The proof follows from the following prooftrees:

$$\text{E-Id})\quad \frac{Id(A, y, x)\ prop\ [x \in A, y \in A] \qquad a \in A \qquad b \in A \qquad p \in Id(A, a, b) \qquad id_A(x) \in Id(A, x, x)\ [x \in A]}{El_{Id}(p, (x).id_A(x)) \in Id(A, b, a)}$$

$$\text{C-Id})\quad \frac{Id(A, y, x)\ prop\ [x \in A, y \in A] \qquad a \in A \qquad id_A(x) \in Id(A, x, x)\ [x \in A]}{El_{Id}(id_A(a), (x).id_A(x)) = id_A(a) \in Id(A, a, a)}$$

□

Lemma 5.3.2 (Concatenation)**.** *For every collection A, every $a, b, c \in A$ and every $p \in Id(A, a, b)$ and $q \in Id(A, b, c)$ there is a proof-term $p \cdot q \in Id(A, a, c)$, such that $id_A(a) \cdot id_A(a) = id_A(a) \in Id(A, a, a)$ for any $a \in A$.*

Proof. The proof follows from the following prooftree:

$$\text{E-}\forall)\quad \frac{q \in Id(A,b,c) \qquad \pi_1}{Ap_\forall(Ap_\forall(El_{Id}(p,(x).\lambda_\forall z^A.\lambda_\forall w^{Id(A,x,z)}.w),c),q) \in Id(A,a,c)}$$

where π_1 is

$$\text{E-}\forall)\quad \frac{c \in A \qquad \pi_2}{Ap_\forall(El_{Id}(p,(x).\lambda_\forall z^A.\lambda_\forall w^{Id(A,x,z)}.w),c) \in \forall_{w\in Id(A,b,c)} Id(A,a,c)}$$

π_2 is

$$\text{E-Id})\quad \frac{\forall_{z\in A}\forall_{w\in Id(A,y,z)} Id(A,x,z)\ prop\ [x\in A, y\in A] \qquad a\in A \qquad b\in A \qquad p\in Id(A,a,b) \qquad \pi_2}{El_{Id}(p,(x).\lambda_\forall z^A.\lambda_\forall w^{Id(A,x,z)}.w) \in \forall_{z\in A}\forall_{w\in Id(A,b,z)} Id(A,a,z)}$$

π_3 is

$$\text{I-}\forall)\quad \frac{\pi_4 \qquad \forall_{w\in Id(A,x,z)} Id(A,x,z)\ prop\ [x\in A, z\in A]}{\lambda_\forall z^A.\lambda_\forall w^{Id(A,x,z)}.w \in \forall_{z\in A}\forall_{w\in Id(A,x,z)} Id(A,x,z)\ [x\in A]}$$

and π_4 is

$$\text{I-}\forall)\quad \frac{w\in Id(A,x,z)\ [x\in A, z\in A, w\in Id(A,x,z)] \qquad Id(A,x,z)\ prop\ [x\in A, z\in A, w\in Id(A,x,z)]}{\lambda_\forall w^{Id(A,x,z)}.w \in \forall_{w\in Id(A,x,z)} Id(A,x,z)\ [x\in A, z\in A]}$$

while the conversion rule is

$$\beta\text{C-}\forall)\quad \frac{id_A(a)\in Id(A,a,a) \qquad w\in Id(A,a,a)\ [w\in Id(A,a,a)] \qquad Id(A,a,a)\ prop\ [w\in Id(A,a,a)]}{Ap_\forall(Ap_\forall(El_{Id}(id_A(a),(x).\lambda_\forall z^A.\lambda_\forall w^{Id(A,x,z)}.w),a),id_A(a)) = id_A(a) \in Id(A,a,a)}$$

□

Lemma 5.3.3. *Suppose A is a collection, that $a,b,c,d \in A$ and that $p \in Id(A,a,b)$ and $q\in Id(A,b,c)$ and $r \in Id(A,c,d)$. We have the following terms:*

1. $pf_1 \in Id(Id(A,a,b),p,p\cdot id_A(b))$ *and* $pf_2 \in Id(Id(A,a,b),p,id_A(a)\cdot p)$
2. $pf_2 \in Id(Id(A,b,b),p^{-1}\cdot p, id_A(b))$ *and* $pf_3 \in Id(Id(A,a,a),p\cdot p^{-1}, id_A(a))$
3. $pf_3 \in Id(Id(A,a,b),(p^{-1})^{-1},p)$

4. $pf_5 \in Id(Id(A,a,d), p \cdot (q \cdot r), (p \cdot q) \cdot r)$

Proof. We prove these statements with Path Induction:

1. E-Id) $$\frac{Id(Id(A,a,y), z, z \cdot id_A(y))\ prop\ [y \in A, z \in Id(A,a,y)] \qquad a \in A \qquad b \in A \qquad p \in Id(A,a,b) \qquad \pi_1}{El_{Id}(p, id_{Id(A,a,a)}(id_A(a))) \in Id(Id(A,a,b), p, p \cdot id_A(b))}$$

   where π_1 is obtained from $id_A(a) = id_A(a) \cdot id_A(a) \in Id(A,a,a)$. The other equality is proven similarly.

2. E-Id) $$\frac{Id(Id(A,y,y), z^{-1} \cdot z, id_A(y))\ prop\ [y \in A, z \in Id(A,a,y)] \qquad a \in A \qquad b \in A \qquad p \in Id(A,a,b) \qquad \pi_2}{El_{Id}(p, id_{Id(A,a,a)}(id_A(a))) \in Id(Id(A,b,b), p^{-1} \cdot p, id_A(b))}$$

   where π_2 is obtained from $(id_A(a))^{-1} \cdot id_A(a) = id_A(a) \in Id(A,a,a)$. The other equality is similar.

3. E-Id) $$\frac{Id(Id(A,a,y), (z^{-1})^{-1}, z)\ prop\ [y \in A, z \in Id(A,a,y)] \qquad a \in A \qquad b \in A \qquad p \in Id(A,a,b) \qquad \pi_3}{El_{Id}(p, id_{Id(A,a,a)}(id_A(a))) \in Id(Id(A,a,b), (p^{-1})^{-1}, p)}$$

   where π_3 is obtained from $((id_A(a))^{-1})^{-1} = id_A(a) \in Id(A,a,a)$.

4. E-Id) $$\frac{Id(Id(A,a,y), p \cdot (q \cdot z), (p \cdot q) \cdot z)\ prop\ [y \in A, z \in Id(A,c,y)] \qquad c \in A \qquad d \in A \qquad r \in Id(A,c,d) \qquad \pi_4}{El_{Id}(r, id_{Id(A,a,c)}(p \cdot q)) \in Id(Id(A,a,d), p \cdot (q \cdot r), (p \cdot q) \cdot r)}$$

   where π_4 is the prooftree obtained from $id_{Id(A,a,c)}(p \cdot q) \in Id(Id(A,a,d), p \cdot q, p \cdot q)$ and applying subT) and conv) in order to obtain that $id_{Id(A,a,c)}(p \cdot q) \in Id(Id(A,a,d), p \cdot (q \cdot id_A(c)), (p \cdot q) \cdot id_A(c))$

□

Lemma 5.3.4. *Suppose that $f(x) \in B\ [x \in A]$ is a function. For any $a, b \in A$ and $p \in Id(A,a,b)$ there is a term*

$$pf \in Id(B, f(a), f(b)).$$

Moreover, for each $a \in A$ we have $f(id_A(a)) = id_B(f(a))$.

Proof. We construct the following prooftree:

$$\text{E-Id)}\quad \frac{Id(B,f(x),f(y))\ prop\ [x\in A, y\in A] \qquad a\in A \qquad b\in A \qquad p\in Id(A,a,b) \qquad id_B(f(x))\in Id(B,f(x),f(x))\ [x\in A]}{El_{Id}(p,(x).id_B(f(x)))\in Id(B,f(a),f(b))}$$

$$\text{C-Id)}\quad \frac{Id(B,f(x),f(y))\ prop\ [x\in A, y\in A] \qquad a\in A \qquad id_B(f(x))\in Id(B,f(x),f(x))\ [x\in A]}{El_{Id}(id_A(a),(x).id_B(f(x))) = id_B(f(a))\in Id(B,f(a),f(a))}$$

Set $f(p) := El_{Id}(p,(x).id_B(f(x)))\in Id(B,f(a),f(b))\ [p\in Id(A,a,b)]$ □

Lemma 5.3.5 (Transport)**.** *Suppose that P is a proposition family over a collection A and that $p\in Id(A,a,b)$. Then there is a term $p_*\in P(a)\to P(b)$.*

Proof. Since P is a proposition family over A, $P(a)$ is a proposition for every $a\in A$ and then we have:

$$\text{E-Id)}\quad \frac{\text{F-}\!\to)\ \dfrac{P(a)\ prop\ [x\in A] \qquad P(b)\ prop\ [y\in A]}{P(x)\to P(y)\ prop\ [x\in A, y\in A]} \qquad a\in A \qquad b\in A \qquad p\in Id(A,a,b) \qquad \pi}{El_{Id_p}(p,(x).\lambda_{\to}z^{P(x)}.z)\in P(a)\to P(b)}$$

where π is

$$\text{I-}\!\to)\quad \frac{z\in P(x)\ [x\in A, z\in P(x)] \qquad P(x)\ prop\ [x\in A]}{\lambda_{\to}z^{P(x)}.z\in P(x)\to P(x)\ [x\in A]}$$

Set $p_* := El_{Id_p}(p,(x).\lambda_{\to}z^{P(x)}.z)\in P(a)\to P(b)\ [p\in Id(A,a,b)]$

□

Corollary 5.3.4. *$(id_A(a))_*$ is the identity:*

Proof. We have seen before that $P(a)\to P(a)$ is a proposition.

$$\text{C-Id)}\quad \frac{\text{F-}\!\to)\ \dfrac{P(a)\ prop\ [x\in A] \qquad P(b)\ prop\ [y\in A]}{P(x)\to P(y)\ prop\ [x\in A, y\in A]} \qquad a\in A \qquad \pi}{El_{Id_p}(id_A(a),(x).\lambda_{\to}z^{P(x)}.z) = \lambda_{\to}z^{P(a)}.z\in P(a)\to P(a)}$$

□

5.4 Natural numbers

In our foundation we can construct the set of natural numbers $\mathbb{N}$ with the rules of mTT; in fact, the principal idea is to define $\mathbb{N}$ as $List(N_1)$ and to derive its rules from $List$ ones. We start constructing the Formation rule; for S) we know that N_1 is a set and then we have

$$\text{F-list)} \quad \frac{N_1 \ set}{List(N_1) \ set}$$

We can now define $\mathbb{N} := List(N_1)$.

Next, in order to build elements of $\mathbb{N}$, we use Introduction rules, I_1-list) and I_2, as follows:

$$\mathrm{I}_1\text{-list)} \quad \frac{\mathbb{N} \ set}{\epsilon \in \mathbb{N}}$$

$$\mathrm{I}_2\text{-list)} \quad \frac{n \in \mathbb{N} \qquad \star \in N_1}{cons(n, \star) \in \mathbb{N}}$$

Let $0 := \epsilon$ and $succ(n) := cons(n, \star)$ be the "zero" and the "successor". We obtain the remaining rules of $\mathbb{N}$ eliminating the assumptions that hold in our foundation. Summing up we have:

$$\text{F-}\mathbb{N}) \quad \mathbb{N} \ set$$

$$\mathrm{I}_1\text{-}\mathbb{N}) \quad 0 \in \mathbb{N}$$

$$\mathrm{I}_2\text{-}\mathbb{N}) \quad \frac{n \in \mathbb{N}}{succ(n) \in \mathbb{N}}$$

$$\text{E-}\mathbb{N}) \quad \frac{A(x) \ col \ [x \in \mathbb{N}] \qquad n \in \mathbb{N} \qquad a \in A(0) \qquad b(x,y) \in A(succ(x)) \ [x \in \mathbb{N}, y \in A(x)]}{El_{\mathbb{N}}(n, a, b) \in A(n)}$$

$$\mathrm{C}_1\text{-}\mathbb{N}) \quad \frac{A(x) \ col \ [x \in \mathbb{N}] \qquad a \in A(0) \qquad b(x,y) \in A(succ(x)) \ [x \in \mathbb{N}, y \in A(x)]}{El_{\mathbb{N}}(0, a, b) = a \in A(0)}$$

$$\mathrm{C}_2\text{-}\mathbb{N}) \quad \frac{A(x) \ col \ [x \in \mathbb{N}] \qquad n \in \mathbb{N} \qquad a \in A(0) \qquad b(x,y) \in A(succ(x)) \ [x \in \mathbb{N}, y \in A(x)]}{El_{\mathbb{N}}(succ(n), a, b) = b(n, El_{\mathbb{N}}(n, a, b)) \in A(succ(n))}$$

5.4.1 Properties

Now, we define a propositional family:

$$\text{E-}\mathbb{N})\quad \frac{prop_s\ col \qquad m \in \mathbb{N} \qquad \pi_1 \qquad \pi_2}{El_{\mathbb{N}}(m, El_{\mathbb{N}}(n, \top, \bot), (y).El_{\mathbb{N}}(n, \bot, y)) \in prop_s}$$

where π_1 is

$$\text{E-}\mathbb{N})\quad \frac{prop_s\ col \qquad n \in \mathbb{N} \qquad \top \in prop_s \qquad \bot \in prop_s}{El_{\mathbb{N}}(n, \top, \bot) \in prop_s}$$

and π_2 is

$$\text{E-}\mathbb{N})\quad \frac{prop_s\ col \qquad n \in \mathbb{N} \qquad \bot \in prop_s \qquad y \in prop_s\ [y \in prop_s]}{El_{\mathbb{N}}(n, \bot, y) \in prop_s\ [y \in prop_s]}$$

We call

$$code(m, n) := El_{\mathbb{N}}(m, El_{\mathbb{N}}(n, \top, \bot), (y).El_{\mathbb{N}}(n, \bot, y)) \in prop_s\ [m \in \mathbb{N}, n \in \mathbb{N}]$$

By conversion rule C-$\mathbb{N}$) we have:

$$\begin{aligned} code(0, 0) &= \top \\ code(succ(m), 0) &= \bot \\ code(0, succ(n)) &= \bot \\ code(succ(m), succ(n)) &= code(m, n) \end{aligned}$$

We also define by recursion a dependent function $r(n) \in code(n, n)$, for all $n \in \mathbb{N}$

$$\text{E-}\mathbb{N})\quad \frac{code(x, x)\ col\ [x \in \mathbb{N}] \qquad n \in \mathbb{N} \qquad \star \in code(0, 0) \qquad \pi_3}{El_{\mathbb{N}}(n, \star, (y).y) \in code(n, n)}$$

where π_3 is

$$y \in code(succ(x), succ(x)) = code(x, x)\ [x \in \mathbb{N}, y \in code(x, x)]$$

We call

$$r(n) := El_{\mathbb{N}}(n, \star, (y).y) \in code(n, n)\ [n \in \mathbb{N}]$$

and by C-ℕ) we have:

$$r(0) = \star$$
$$r(succ(n)) = r(n)$$

Theorem 5.4.1. *For all $m, n \in \mathbb{N}$ we have that $Id(\mathbb{N}, m, n)$ and $code(m, n)$ are equivalent propositions.*

Proof. For all $m, n \in \mathbb{N}$ and $p \in Id(\mathbb{N}, m, n)$ we have

$$p_*(r(m)) \in code(m, n)$$

we call

$$encode(m, n, p) := p_*(r(m)) \in code(m, n)\ [m \in \mathbb{N}, n \in \mathbb{N}]$$

Conversely, we define

$$\text{E-}{\to})\quad \frac{p \in code(m, n) \qquad \pi_1}{Ap_{\to}(El_{\mathbb{N}}(m, pf_2, pf_3), p) \in Id(\mathbb{N}, m, n)}$$

where π_1 is

$$\text{E-}\mathbb{N})\quad \frac{code(x, n) \to Id(\mathbb{N}, x, n)\ col\ [x \in \mathbb{N}] \qquad m \in \mathbb{N} \qquad \pi_2 \qquad \pi_3}{El_{\mathbb{N}}(m, pf_2, pf_3) \in code(m, n) \to Id(\mathbb{N}, m, n)}$$

where $pf_2 := El_{\mathbb{N}}(n, \lambda_{\to} z^{\top}.id_{\mathbb{N}}(0), \lambda_{\to} w^{\perp}.r_0(w))$ and π_2 is

$$\text{E-}\mathbb{N})\quad \frac{code(0, y) \to Id(\mathbb{N}, 0, y)\ col\ [y \in \mathbb{N}] \qquad n \in \mathbb{N} \qquad \pi_4 \qquad \pi_5}{El_{\mathbb{N}}(n, \lambda_{\to} z^{\top}.id_{\mathbb{N}}(0), \lambda_{\to} w^{\perp}.r_0(w)) \in code(0, n) \to Id(\mathbb{N}, 0, n)}$$

Recalling that $code(0, 0) = \top$ and $code(0, succ(y)) = \perp$ for every $y \in \mathbb{N}$ we have that π_4 is

$$\text{I-}{\to})\quad \frac{id_{\mathbb{N}}(0) \in Id(\mathbb{N}, 0, 0) \qquad \top\ prop \qquad Id(\mathbb{N}, 0, 0) prop}{\lambda_{\to} z^{\top}.id_{\mathbb{N}}(0) \in \top \to Id(\mathbb{N}, 0, 0)}$$

and π_5 is

$$\text{I-}\to)\ \frac{\text{E-Fs)}\ \dfrac{w \in \bot\ [w \in \bot] \qquad Id(\mathbb{N}, 0, succ(y))\ prop\ [y \in \mathbb{N}]}{r_0(w) \in Id(\mathbb{N}, 0, succ(y))\ [y \in \mathbb{N}, w \in \bot]} \qquad \bot\ prop \qquad Id(\mathbb{N}, 0, succ(y))\ prop\ [y \in \mathbb{N}]}{\lambda_{\to} w^{\bot}.r_0(w) \in \bot \to Id(\mathbb{N}, 0, succ(y))\ [y \in \mathbb{N}]}$$

While we have $pf_3 := El_{\mathbb{N}}(n, \lambda_{\to} v^{\bot}.r_0(v), Ap_{\forall}(\lambda_{\forall} u^{code(x,n) \to Id(\mathbb{N},x,n)}.pf_1)$ and that π_3 is

$$\text{E-}\mathbb{N})\ \frac{code(succ(x), y) \to Id(\mathbb{N}, succ(x), y)\ col\ [x \in \mathbb{N}, y \in \mathbb{N}] \qquad n \in \mathbb{N} \qquad \pi_6 \qquad \pi_7}{El_{\mathbb{N}}(n, \lambda_{\to} v^{\bot}.r_0(v), Ap_{\forall}(\lambda_{\forall} u^{code(x,n) \to Id(\mathbb{N},x,n)}.pf_1) \in code(succ(x), n) \to Id(\mathbb{N}, succ(x), n)\ [x \in \mathbb{N}, u \in code(x, n) \to Id(\mathbb{N}, x, n)]}$$

where π_6 is similar to π_5 and, recalling that $code(succ(x), succ(y)) = code(x, y)$ for every $x, y \in \mathbb{N}$ we have that π_7 is

$$\text{E-}\forall)\ \frac{u \in code(x, n) \to Id(\mathbb{N}, x, n)\ [x \in \mathbb{N}, u \in code(x, n) \to Id(\mathbb{N}, x, n)] \qquad \pi_8}{Ap_{\forall}(\lambda_{\forall} u^{code(x,n) \to Id(\mathbb{N},x,n)}.pf_1, u) \in code(x, y) \to Id(\mathbb{N}, succ(x), succ(y))\ [x \in \mathbb{N}, y \in \mathbb{N}, u \in code(x, n) \to Id(\mathbb{N}, x, n)]}$$

$$\beta\text{C-}\forall)\ \frac{n \in \mathbb{N} \qquad \pi_9 \qquad \forall_{u \in code(x,y) \to Id(\mathbb{N},x,y)} code(x, y) \to Id(\mathbb{N}, succ(x), succ(y))\ prop\ [x \in \mathbb{N}, y \in \mathbb{N}]}{\lambda_{\forall} u^{code(x,n) \to Id(\mathbb{N},x,n)}.pf_1 \in \forall_{u \in code(x,n) \to Id(\mathbb{N},x,n)} code(x, y) \to Id(\mathbb{N}, succ(x), succ(y))\ [x \in \mathbb{N}, y \in \mathbb{N}]}$$

π_9 is

$$\text{I-}\forall)\ \frac{\pi_{10} \qquad code(x, y) \to Id(\mathbb{N}, succ(x), succ(y))\ prop\ [x \in \mathbb{N}, y \in \mathbb{N}]}{\lambda_{\forall} u^{code(x,y) \to Id(\mathbb{N},x,y)}.pf_1 \in \forall_{u \in code(x,y) \to Id(\mathbb{N},x,y)} code(x, y) \to Id(\mathbb{N}, succ(x), succ(y))\ [x \in \mathbb{N}, y \in \mathbb{N}]}$$

where $pf_1 := \lambda_{\to} v^{code(x,y)}.succ(Ap_{\to}(u, v))$ and π_{10} is

$$\text{I-}\to)\ \frac{\pi_{11} \qquad code(x, y)\ prop\ [x \in \mathbb{N}, y \in \mathbb{N}] \qquad Id(\mathbb{N}, succ(x), succ(y))\ prop\ [x \in \mathbb{N}, y \in \mathbb{N}]}{\lambda_{\to} v^{code(x,y)}.succ(Ap_{\to}(u, v)) \in code(x, y) \to Id(\mathbb{N}, succ(x), succ(y))\ [x \in \mathbb{N}, y \in \mathbb{N}, u \in code(x, y) \to Id(\mathbb{N}, x, y)]}$$

where π_{11} is

$$succ(Ap_{\rightarrow}(u,v)) \in Id(\mathbb{N}, succ(x), succ(y))\ [x \in \mathbb{N}, y \in \mathbb{N}, u \in code(x,y) \rightarrow Id(\mathbb{N},x,y), v \in code(x,y)]$$

that is obtained by the following prooftree

$$\text{E-}\rightarrow)\ \frac{v \in code(x,y)\ [x \in \mathbb{N}, y \in \mathbb{N}, v \in code(x,y)] \qquad u \in code(x,y) \rightarrow Id(\mathbb{N},x,y)\ [x \in \mathbb{N}, y \in \mathbb{N}, u \in code(x,y) \rightarrow Id(\mathbb{N},x,y)]}{Ap_{\rightarrow}(u,v) \in Id(\mathbb{N},x,y)\ [x \in \mathbb{N}, y \in \mathbb{N}, u \in code(x,y) \rightarrow Id(\mathbb{N},x,y), v \in code(x,y)]}$$

□

As a result we will use later, we show the following:

Theorem 5.4.2 (Peano Axioms)**.** *Peano Axioms hold in mTT, that is the following statements hold:*

1. 0 *is a term for* $\mathbb{N}$
2. *For every* $n \in \mathbb{N}$, $succ(n)$ *is a term for* $\mathbb{N}$
3. *For every* $m, n \in \mathbb{N}$, *there is a term* $pf_1 \in Id(\mathbb{N}, succ(m), succ(n)) \rightarrow Id(\mathbb{N}, m, n)$;
4. *For every* $n \in \mathbb{N}$, *there is a term* $pf_2 \in \neg Id(\mathbb{N}, 0, succ(n))$.

Proof.

1. It is the I_1-$\mathbb{N}$) rule;
2. It is the I_2-$\mathbb{N}$) rule;
3. We construct:

$$\text{I-}\rightarrow)\ \frac{\pi_1 \qquad Id(\mathbb{N}, succ(m), succ(n))\ prop \qquad Id(\mathbb{N}, m, n)\ prop}{\lambda_{\rightarrow} z^{Id(\mathbb{N}, succ(m), succ(n))}.decode(m, n, encode(m, n, z)) \in Id(\mathbb{N}, succ(m), succ(n)) \rightarrow Id(\mathbb{N}, m, n)}$$

Where π_1 is the term

$$decode(m, n, encode(m, n, z)) \in Id(\mathbb{N}, m, n)\ [z \in Id(\mathbb{N}, succ(m), succ(n))]$$

in fact we have

$$encode(m, n, z) \in code(succ(m), succ(n)) = code(m, n)\ [z \in Id(\mathbb{N}, succ(m), succ(n))]$$

4. Recalling that $code(0, succ(n)) = \bot$ we have:

$$\text{I-}\!\to) \quad \frac{encode(0, succ(n), x) \in \bot\ [x \in Id(\mathbb{N}, 0, succ(n))] \qquad \top\ prop \qquad Id(\mathbb{N}, 0, succ(n))\ prop}{\lambda_{\to} x^{Id(\mathbb{N},0,succ(n))}.encode(0, succ(n), x) \in \neg Id(\mathbb{N}, 0, succ(n))}$$

$\square$

5.5 Uniqueness of identity proofs and Hedberg's Theorem

Definition 5.5.1 (UIP)**.** *The collection A satisfies the property of* uniqueness of identity proofs *if for all $a, b \in A$ and $p, q \in Id(A, a, b)$ we have a term $pf \in Id(Id(A, a, b), p, q)$.*

Here is another equivalent characterization, involving Streicher's "Axiom K" ([11]):

Definition 5.5.2 (Axiom K)**.** *A collection A satisfies Axiom K if for all $a \in A$ and $p \in Id(A, a, a)$ we have a term $pf \in Id(Id(A, a, a), p, id_A(a))$.*

Theorem 5.5.1. *A collection A satisfies UIP if and only if satisfies Axiom K*

Proof. Clearly Axiom K is a special case of UIP:

$$\text{E-}\forall) \quad \frac{\text{I-Id)}\ \dfrac{a \in A}{id_A(a) \in Id(A, a, a)} \qquad \text{E-}\forall)\ \dfrac{a \in A \qquad pf_1 \in \forall_{b \in A} \forall_{q \in Id(A,a,b)} Id(Id(A, a, b), p, q)}{Ap_\forall(pf_1, a) \in \forall_{q \in Id(A,a,a)} Id(Id(A, a, a), p, q)}}{Ap_\forall(Ap_\forall(pf_1, a), id_A(a)) \in Id(Id(A, a, a), p, id_A(a))}$$

Conversely, we have:

$$\text{E-Id)} \quad \frac{\forall_{p \in Id(A,a,y)} Id(Id(A, a, y), p, z)\ prop\ [y \in A, z \in Id(A, a, y)] \qquad a \in A \qquad b \in A \qquad q \in Id_p(A, a, b) \qquad pf_2 \in \forall_{p \in Id(A,a,a)} Id(Id(A, a, a), p, id_A(a))}{El_{Id_p}(q, pf) \in \forall_{p \in Id(A,a,b)} Id(Id(A, a, b), p, q)}$$

$\square$

Lemma 5.5.1. *There is a term $\rho \in \forall_{x \in A} \in \neg\neg Id(A, x, x)$.*

Proof. To obtain a term for $\neg\neg Id(A, x, x)$ we can construct the following prooftree:

$$\text{I-}\forall)\ \dfrac{\text{I-}\!\to)\ \dfrac{\text{E-}\!\to)\ \dfrac{\text{I-Id})\ \dfrac{x \in A\ [x \in A]}{id_A(x) \in Id(A,x,x)\ [x \in A]} \qquad z \in \neg Id(A,x,x)\ [x \in A, z \in \neg Id(A,x,x)]}{Ap_{\to}(z, id_A(x)) \in \bot\ [x \in A, z \in \neg Id(A,x,x)]}}{\lambda_{\to} z^{\neg Id(A,x,x)}.Ap_{\to}(z, id_A(x)) \in \neg\neg Id(A,x,x)\ [x \in A]} \qquad \neg\neg Id(A,x,x)\ prop\ [x \in A]}{\lambda_{\forall} x^{A}.\lambda_{\to} z^{\neg Id(A,x,x)}.Ap_{\to}(z, id_A(x)) \in \forall_{x \in A} \in \neg\neg Id(A,x,x)}$$

We call

$$\rho := \lambda_{\forall} x^{A}.\lambda_{\to} z^{\neg Id(A,x,x)}.Ap_{\to}(z, id_A(x)) \in \forall_{x \in A} \in \neg\neg Id(A,x,x)$$

and

$$r := \lambda_{\to} z^{\neg Id(A,a,a)}.Ap_{\to}(z, id_A(a)) = \rho(a) \in \neg\neg Id(A,a,a)$$

□

Theorem 5.5.2. *If a collection A has the property that $\neg\neg Id(A,a,b) \to Id(A,a,b)$ for any $a,b \in A$, then A satisfies Axiom K.*

Proof. For every $a \in A$ and $p \in Id(A,a,a)$, we take a term $f \in \forall_{a \in Id(A,a,a)}(\neg\neg Id(A,a,a) \to Id(A,a,a))$ and then, recalling that p_* is the transport along p previously defined, we construct the following prooftree:

$$\text{E-Id})\ \dfrac{Id(Id(A,y,y), z_*(f(a,r)), f(y, z_*(r)))\ prop\ [y \in A, z \in Id(A,a,y)] \qquad a \in A \qquad p \in Id(A,a,a) \qquad \pi_1}{El_{Id}(p, id_{Id(A,a,a)}(f(a,r))) \in Id(Id(A,a,a), p_*(f(a,r)), f(a, p_*(r)))}$$

where π_1 is the prooftree constructed by taking $id_{Id(A,a,a)}(f(a,r)) \in Id(Id(A,a,a), f(a,r), f(a,r))$ and using subT) and conv) rules to obtain that $id_{Id(A,a,a)}(f(a,r))$ is also a term for the proposition

$$Id(Id(A,a,a), (id_A(a))_*(f(a,r)), f(a, (id_A(a))_*(r)))$$

Now we construct two other terms that concatenated with

$$pf_1 = El_{Id_p}(p, id_{Id(A,a,a)}(f(a,r))) \in Id(Id(A,a,a), p_*(f(a,r)), f(a, p_*(r)))$$

will result in a term for the proposition $Id(Id(A,a,a), f(a,r) \cdot p, f(a,r))$. The first one is obtained from this prooftree:

$$\text{E-Id})\ \dfrac{Id(Id(A,a,a), f(y, z_*(r)), f(y, \rho(y)))\ prop\ [y \in A, z \in Id(A,a,y)] \qquad a \in A \qquad p \in Id(A,a,a) \qquad \pi_2}{El_{Id}(p, id_{Id(A,a,a)}(f(a,r))) \in Id(Id(A,a,a), f(a, p_*(r)), f(a,r))}$$

where π_2 is the prooftree similar to π_1 constructed by taking $id_{Id(A,a,a)}(f(a,r)) \in Id(Id(A,a,a), f(a,r), f(a,r))$ and using subT) and conv) rules to obtain that $id_{Id(A,a,a)}(f(a,r))$ is also a term for the proposition

$$Id(Id(A,a,a), f(a,(id_A(a))_*(r)), f(a,r))$$

Notice that the terms obtained in π_1 and π_2 are the same so they are those for the propositions

$$Id(Id(A,a,a), p_*(f(a,r)), f(p_*(a,r))) \qquad Id(Id(A,a,a), f(a,(id_A(a))_*(r)), f(a,r)).$$

The second is constructed seeing that

$$(id_A(a))_*(f(a,r)) = f(a,r) = f(a,r) \cdot id_A(a)$$

for C-Id), so there is a

$$pf_4 \in Id(Id(A,a,a), (id_A(a))_*(f(a,r)), f(a,r) \cdot id_A(a))$$

$$\text{E-Id)} \quad \frac{Id(Id(A,y,y), z_*(f(a,r)), f(a,r) \cdot z)\ prop\ [y \in A, z \in Id(A,a,y)] \qquad a \in A \qquad a \in A \qquad p \in Id(A,a,a) \qquad \pi_3}{El_{Id}(p, id_{Id(A,a,a)}(f(a,r))) \in Id(Id(A,a,a), p_*(f(a,r)), f(a,r) \cdot p)}$$

Once again the proofterm is very similar and it can be shown to be $(pf_1)^{-1}$.

So, in order to obtain a term for $Id(Id(A,a,a), f(a,r) \cdot p, f(a,r))$, we just need to concatenate the terms obtained before:

$$(pf_1)^{-1} \cdot pf_1 \cdot pf_1 = pf_1 \in Id(Id(A,a,a), f(a,r) \cdot p, f(a,r))$$

We set $q := f(a,r)$ for brevity.

$$\text{E-}\forall) \quad \frac{q \in Id(A,a,a) \qquad \pi_4}{Ap_\forall((El_{Id}(q,c), q) \in Id(Id(A,a,a), q^{-1} \cdot (q \cdot p), q^{-1} \cdot q)}$$

where π_4 is:

$$\text{E-Id)} \quad \frac{\forall_{z \in Id(A,a,y)} Id(Id(A,y,a), z^{-1} \cdot (q \cdot p), z^{-1} \cdot q)\ prop\ [y \in A, z \in Id(A,a,y)] \qquad a \in A \qquad q \in Id(A,a,a) \qquad \pi_5}{El_{Id_p}(q,c) \in \forall_{w \in Id(A,a,a)} Id(Id(A,a,a), q^{-1} \cdot (q \cdot p), q^{-1} \cdot q)}$$

and π_5 is the prooftree constructed noticing that, for the properties of the inverse and concatenation, there are

$$pf_2 \in Id(Id(A,a,a),(id_A(a))^{-1} \cdot (q \cdot p), q \cdot p) \qquad pf_3 \in Id(Id(A,a,a),(id_A(a))^{-1} \cdot q, q)$$

so we have:

$$pf_4 := pf_2 \cdot pf_1 \cdot (pf_3)^{-1} \in Id(Id(A,a,a),(id_A(a))^{-1} \cdot (q \cdot p),(id_A(a))^{-1} \cdot q)$$

and then the bottom of the prooftree π_4 is:

$$\lambda_\forall id_A(a)^{Id(A,a,a)}.pf_4 \in \forall_{id_A(a) \in Id(A,a,a)} Id(Id(A,a,a),(id_A(a))^{-1} \cdot (q \cdot p),(id_A(a))^{-1} \cdot q)$$

We call $pf_5 := Ap_\forall((El_{Id}(q,c),q) \in Id(Id(A,a,a), q^{-1} \cdot (q \cdot p), q^{-1} \cdot q)$.

In the end, for the properties of the inverse and concatenation, we have terms:

$$pf_6 \in Id(Id(A,a,a),p,id_A(a) \cdot p)$$
$$pf_7 \in Id(Id(A,a,a),id_A(a),q^{-1} \cdot q)$$
$$pf_8 \in Id(Id(A,a,a),(q^{-1} \cdot q) \cdot p, q^{-1} \cdot (q \cdot p))$$

Moreover we have also $pf_9 \in Id(Id(A,a,a),id_A(a) \cdot p,(q^{-1} \cdot q) \cdot p)$ from pf_{12}. Lastly to obtain a proofterm for Axiom K we just need to concatenate:

$$pf_6 \cdot pf_8 \cdot pf_9 \cdot pf_5 \cdot (pf_7)^{-1} \in Id(Id(A,a,a),p,id_A(a))$$

□

Notice that seeing $f(a,r) \in Id(A,a,a)$ as a path from point a of the space A in itself and deriving

$$pf_1 \in Id(Id(A,a,a),f(a,r) \cdot p, f(a,r))$$

means that $f(a,r)$ has a *fixed point* p that is every p is a fixed point of $f(a,r)$.

Lemma 5.5.2. *For any collection α and $p \in \alpha \vee \neg\alpha$, we have a term $pf \in \neg\neg\alpha \to \alpha$.*

Proof.

$$\text{I-}\!\to)\ \dfrac{\text{E-}\vee)\ \dfrac{p \in \alpha \vee \neg\alpha \qquad x \in \alpha\ [x \in \alpha] \qquad \text{E-Fs)}\ \dfrac{\text{E-}\!\to)\ \dfrac{y \in \neg\alpha\ [y \in \neg\alpha] \qquad z \in \neg\neg\alpha\ [z \in \neg\neg\alpha]}{Ap_\to(z,y) \in \bot\ [y \in \neg\alpha, z \in \neg\neg\alpha]}}{r_0(Ap_\to(z,y)) \in \neg\alpha\ [y \in \neg\alpha, z \in \neg\neg\alpha]}}{El_\vee(p,(x).x,(y).r_0(Ap_\to(z,y))) \in \alpha\ [z \in \neg\neg\alpha]}}{\lambda_\to z^{\neg\neg\alpha}.El_\vee(p,(x).x,(y).r_0(Ap_\to(z,y))) \in \neg\neg\alpha \to \alpha}$$

□

Theorem 5.5.3 (Hedberg)**.** *If A has decidable equality, then A satisfies UIP.*

Proof. If A has decidable equality, it follows that $\neg\neg Id(A,a,b) \rightarrow Id(A,a,b)$ for any $a, b \in A$. Since Axiom K is equivalent to UIP, Hedberg's theorem follows from the previous theorem. □

Theorem 5.5.4. *The set $\mathbb{N}$ of natural numbers has decidable equality, and hence satisfies UIP.*

Proof. We proceed by induction on m and case analysis on n:

$$\text{E-}\mathbb{N})\quad \frac{Id(\mathbb{N},x,n) \vee \neg Id(\mathbb{N},x,n)\ col\ [x \in \mathbb{N}] \qquad m \in \mathbb{N} \qquad \pi_1 \qquad \pi_2}{El_{\mathbb{N}}(m,a,b) \in Id(\mathbb{N},m,n) \vee \neg Id(\mathbb{N},m,n)}$$

Case $m = 0$ (π_1)

$$\text{E-}\mathbb{N})\quad \frac{Id(\mathbb{N},0,z) \vee \neg Id(\mathbb{N},0,z)\ col\ [z \in \mathbb{N}] \qquad n \in \mathbb{N} \qquad \pi_3 \qquad \pi_4}{El_{\mathbb{N}}(n, inl_{\vee}(id_{\mathbb{N}}(0)), inr_{\vee}(pf_1)) \in Id(\mathbb{N},0,n) \vee \neg Id(\mathbb{N},0,n)}$$

$a := El_{\mathbb{N}}(n, inl_{\vee}(id_{\mathbb{N}}(0)), inr_{\vee}(pf_1))$

Subcase $m = 0$ and $n = 0$ (π_3)

$$\begin{array}{l}\text{I-Id)}\\ \text{I}_1\text{-}\vee)\end{array}\quad \frac{\dfrac{0 \in \mathbb{N}}{id_{\mathbb{N}}(0) \in Id(\mathbb{N},0,0)} \qquad Id(\mathbb{N},0,0)\ prop \qquad \neg Id(\mathbb{N},0,0)\ prop}{inl_{\vee}(id_{\mathbb{N}}(0)) \in Id(\mathbb{N},0,0) \vee \neg Id(\mathbb{N},0,0)}$$

Subcase $m = 0$ and $n = succ(z)$ (π_4)

$$\text{I}_2\text{-}\vee)\quad \frac{\pi_5 \qquad Id(\mathbb{N},0,succ(z))\ prop\ [z \in \mathbb{N}] \qquad \neg Id(\mathbb{N},0,succ(z))\ prop\ [z \in \mathbb{N}]}{inr_{\vee}(pf_1) \in Id(\mathbb{N},0,succ(z)) \vee \neg Id(\mathbb{N},0,succ(z))\ [z \in \mathbb{N}, y \in Id(\mathbb{N},0,z) \vee \neg Id(\mathbb{N},0,z)]}$$

and π_5 is $pf_1 \in \neg Id(\mathbb{N},0,succ(z))\ [z \in \mathbb{N}]$ which is the validity of the fourth Peano Axiom.

Case $m = succ(x)$ (π_2)

$$\text{E-}\mathbb{N})\quad \frac{Id(\mathbb{N}, succ(x), z) \vee \neg Id(\mathbb{N}, succ(x), z)\ col\ [x \in \mathbb{N}, z \in \mathbb{N}] \qquad n \in \mathbb{N} \qquad \pi_6 \qquad \pi_7}{El_{\mathbb{N}}(n, pf_2, Ap_{\forall}(y, pf_3)) \in Id(\mathbb{N}, succ(x), n) \vee \neg Id(\mathbb{N}, succ(x), n)\ [x \in \mathbb{N}, y \in Id(\mathbb{N}, x, n) \vee \neg Id(\mathbb{N}, x, n)]}$$

$b := El_{\mathbb{N}}(n, pf_2, Ap_{\forall}(y, pf_3))$

Subcase $m = succ(x)$ **and** $n = 0$ **(**π_6**)**

pf_2 in π_6 is obtained similarly to π_4

Subcase $m = succ(x)$ **and** $n = succ(z)$ **(**π_7**)**

$$\text{E-}\forall)\quad \frac{y \in Id(\mathbb{N}, x, n) \vee \neg Id(\mathbb{N}, x, n)\ [x \in \mathbb{N}, y \in Id(\mathbb{N}, x, n) \vee \neg Id(\mathbb{N}, x, n)] \qquad \pi_8}{Ap_{\forall}(y, pf_3) \in Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}, y \in Id(\mathbb{N}, x, n) \vee \neg Id(\mathbb{N}, x, n)]}$$

$pf_3 := \lambda_{\forall} y^{Id(\mathbb{N},x,n) \vee \neg Id(\mathbb{N},x,n)}.El_{\vee}(y, pf_4, pf_5)$ and π_8 is

$$\beta\text{C-}\forall)\quad \frac{n \in \mathbb{N} \qquad \pi_9 \qquad \forall_{y \in Id(\mathbb{N},x,z) \vee \neg Id(\mathbb{N},x,z)} Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ prop\ [x \in \mathbb{N}, z \in \mathbb{N}]}{\lambda_{\forall} y^{Id(\mathbb{N},x,n) \vee \neg Id(\mathbb{N},x,n)}.El_{\vee}(y, pf_4, pf_5) \in \forall_{y \in Id(\mathbb{N},x,n) \vee \neg Id(\mathbb{N},x,n)} Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}]}$$

π_9 is

$$\text{I-}\forall)\quad \frac{\pi_{10} \qquad Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ prop\ [x \in \mathbb{N}, z \in \mathbb{N}, y \in Id(\mathbb{N}, x, z) \vee \neg Id(\mathbb{N}, x, z)]}{\lambda_{\forall} y^{Id(\mathbb{N},x,z) \vee \neg Id(\mathbb{N},x,z)}.El_{\vee}(y, pf_4, pf_5) \in \forall_{y \in Id(\mathbb{N},x,z) \vee \neg Id(\mathbb{N},x,z)} Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}]}$$

π_{10} is

$$\frac{Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ col\ [x \in \mathbb{N}, z \in \mathbb{N}] \qquad \pi_{11} \qquad \pi_{12} \qquad \pi_{13}}{El_{\vee}(y, pf_4, pf_5) \in Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}, y \in Id(\mathbb{N}, x, y) \vee \neg Id(\mathbb{N}, x, y)]}$$

we have

$$\pi_{11} := y \in Id(\mathbb{N}, x, z) \vee \neg Id(\mathbb{N}, x, z)\ [x \in \mathbb{N}, z \in \mathbb{N}, y \in Id(\mathbb{N}, x, y) \vee \neg Id(\mathbb{N}, x, y)]$$

and

$$pf_4 := inl_{\vee}(succ(v)) \in Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}, v \in Id(\mathbb{N}, x, z)]$$

and π_{12} is

$$\text{I}_1\text{-}\vee)\quad \frac{\pi_{14} \qquad Id(\mathbb{N}, succ(x), succ(z))\ prop\ [x \in \mathbb{N}, z \in \mathbb{N}] \qquad \neg Id(\mathbb{N}, succ(x), succ(z))\ prop\ [x \in \mathbb{N}, z \in \mathbb{N}]}{inl_{\vee}(succ(v)) \in Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}, v \in Id(\mathbb{N}, x, z)]}$$

where π_{14} is

$$succ(v) \in Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}, v \in Id(\mathbb{N}, x, z)]$$

while

$$pf_5 := inr_{\vee}(\lambda u^{Id(\mathbb{N}, succ(x), succ(z))}.Ap_{\rightarrow}(w, pf_6(u)))\ [x \in \mathbb{N}, z \in \mathbb{N}, w \in \neg Id(\mathbb{N}, x, z)]$$

and π_{13} is

$$\text{I}_2\text{-}\vee)\quad \frac{\pi_{15} \qquad Id(\mathbb{N}, succ(x), succ(z))\ prop\ [x \in \mathbb{N}, z \in \mathbb{N}] \qquad \neg Id(\mathbb{N}, succ(x), succ(z))\ prop\ [x \in \mathbb{N}, z \in \mathbb{N}]}{inr_{\vee}(\lambda u^{Id(\mathbb{N}, succ(x), succ(z))}.Ap_{\rightarrow}(w, pf_6(u))) \in Id(\mathbb{N}, succ(x), succ(z)) \vee \neg Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}, w \in \neg Id(\mathbb{N}, x, z)]}$$

where π_{15} is

$$\text{I-}\rightarrow)\quad \frac{\pi_{16} \qquad Id(\mathbb{N}, succ(x), succ(z))\ prop\ [x \in \mathbb{N}, z \in \mathbb{N}] \qquad \perp prop}{\lambda u^{Id(\mathbb{N}, succ(x), succ(z))}.Ap_{\rightarrow}(w, pf_6(u)) \in \neg Id(\mathbb{N}, succ(x), succ(z))\ [x \in \mathbb{N}, z \in \mathbb{N}, w \in \neg Id(\mathbb{N}, x, z)]}$$

and π_{16}

$$\text{E-}\rightarrow)\quad \frac{\pi_{17} \qquad w \in \neg Id(\mathbb{N}, x, z)\ [w \in \neg Id(\mathbb{N}, x, z)]}{Ap_{\rightarrow}(w, pf_6(u)) \in \perp\ [x \in \mathbb{N}, z \in \mathbb{N}, u \in Id(\mathbb{N}, succ(x), succ(z)), w \in \neg Id(\mathbb{N}, x, z)]}$$

where π_{17} is

$$pf_6(u) \in Id(\mathbb{N}, x, z)\ [x \in \mathbb{N}, z \in \mathbb{N}, u \in Id(\mathbb{N}, succ(x), succ(z))]$$

that holds for the third Peano Axiom.

□

Chapter 6

Conclusions

From the validity of Hedberg's theorem in mTT shown here we conclude that no choice principle is needed to prove the theorem, since these are not valid in mTT.

It is important to have shown this theorem for mTT because it is expected to imply - as in Homotopy Type Theory - that the classical principle of excluded middle is not valid in the extension of the Minimalist Foundation with the addition of a collection of sets and Voevodsky's Univalent Axiom - stating an equivalence between isomorphisms on a set with the set of identity proofs of the equality of the set with itself (thought of as a code of a universe).

Bibliography

[1] E. Bishop. *Foundations of Constructive Analysis*, McGraw-Hill, 1967.

[2] C. Burrill, *Foundations of real numbers*, McGraw-Hill, 1967.

[3] H. B. Griffiths, P. J. Hilton, *A Comprehensive Textbook of Classical Mathematics: A Contemporary Interpretation*, Springer-Verlag, New York, 1970.

[4] H. Hosni, G. Lolli, C. Toffalori, *Le direzioni della ricerca logica in Italia 2*, ETS, 2018

[5] A.G. Howson, *A Handbook of Terms used in Algebra and Analysis*, Cambridge University Press, 1972.

[6] N. Jacobson, *Lectures in Abstract Algebra: I. Basic Concepts*, Van Nostrand, 1951.

[7] M. E. Maietti, A minimalist two-level foundation for constructive mathematics, *Annals of Pure and Applied Logic*, 2009.

[8] M. E. Maietti and G. Sambin. Toward a minimalist foundation for constructive mathematics. *From Sets and Types to Topology and Analysis: Practicable Foundations for Constructive Mathematics*, number 48 in Oxford Logic Guides. Oxford University Press, 2005

[9] R. Nederpelt, H. Geuvers, *Type Theory and Formal Proof: An Introduction*, Cambridge University Press, 2014

[10] B. Nordstrom, K. Petersson, and J. Smith. *Programming in Martin Lof 's Type Theory.* Clarendon Press, Oxford, 1990.

[11] T. Streicher, *Investigations into intensional type theory*, Habilitationsschrift, Ludwig-Maximilians-Universitat Munchen, 1993.

[12] S. P. Thompson, M. Gardner, *Calculus Made Easy*, St. Martin's Press, 1998.

[13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* Institute for Advanced Study, 2013. see https://homotopytypetheory.org/book.