Peano Arithmetic in Martin-Löf Type Theory

Seminar: Foundations of Mathematics

Philipp Schepper

March 26, 2019

Saarland University – Department of Computer Science

Abstract

To find a foundation of mathematics, one has to compare the different foundational systems. While ZF set theory (ZF) is the most known system, we focus on Martin-Löf type theory (MLTT) in this paper. We use MLTT since it is not based on sets but uses a different approach. In this summary, we give a case study of the well-studied Peano arithmetic for the natural numbers. After the definition and an introduction to the basic properties, we study the Peano axioms. We define the basic mathematical operations addition and multiplication. For subtraction, we give an insight into the problems evolving from the restriction to natural numbers and how one could resolve them. In all cases we draw a comparison between the statement in MLTT and the equivalent statements in ZF. Finally, we compare the two systems and conclude that both systems have different strengths and weaknesses.

1 Introduction

> Hic difficultas maxime ex sermonis ambiguitate oritur.
> Quare summi interest verba ipsa, quibus utimur attente perpendere.
>
> The biggest problems arise from the ambiguities in the speech.
> Therefore, it is most important to check the words used very carefully.
>
> Guiseppe Peano 1889 in [5]

1.1 Motivation

ZF The first answer for a foundation of mathematics would be in most cases Zermelo-Fraenkel set theory (ZF).[1] The concept of sets is well understood in mathematics and a bunch of axioms state very fundamental statements. Not only logical reasoning is possible, but also computation and more applied mathematical topics as functions, for example.

MLTT A newer idea for a foundation of mathematics is the Martin-Löf type theory (MLTT) [4].[2] In the very core, the system consists of types and elements which are introduced by constructors. For working with the elements, eliminators and the corresponding computation rules are introduced to distinguish between the different elements constructed. In this theory, mathematical statements can be translated into special types and their proofs correspond to elements of these types. By this, the checking of a proof changes to type checking.

Natural Numbers A very basic concept of mathematics are numbers and more precise the natural numbers. Since they are used in everyday life, one has a good feeling which properties they (should) satisfy. The most common way to characterize the natural numbers is by the well known Peano axioms. In ZF the natural numbers are defined by interpreting the empty set $\emptyset$ as 0 and the set $x \cup \{x\}$ as the successor of x. By this definition, the use of the natural induction, and the recursion theorem, one can, for example, define addition and multiplication and show that they distribute. While this definition in ZF is an interpretation of already existing objects, the definition of the natural numbers in MLTT has to be different because there is not much freedom for interpretations of already existing types. Since MLTT comes with the possibility to extend the system by other types, one can introduce the natural numbers as a new type. This is the standard way to use them in MLTT. Based on this, one can state the Peano axioms in MLTT and prove that they are valid in the theory. Even computations and the definitions of mathematical operations are possible such that the MLTT version offers the same functionalities as the version in ZF.

1.2 Structure

In Chapter 2 we introduce the natural numbers in MLTT, we show which equality we use, and prove the first Peano axioms. The concept of induction is introduced in Chapter 3. To state the second set of Peano axioms in Chapter 4 we first define the ordering relation on the numbers in the same chapter. In Chapter 5, we discuss the well-foundedness of the natural numbers in MLTT while we show how to work with them in Chapter 6. We conclude in Chapter 7 by giving a comparison with the definition in ZF and a personal opinion about both systems.

2 Definition and Equality

One of the first formalizations of the natural numbers has been written down by Guiseppe Peano in [5]. He states axioms from which he deduces properties of the natural numbers. We follow this approach and define the natural numbers in MLTT in a similar way. But while we define the numbers and their properties, we see that we have different decisions to make. We discuss some of them and state alternative approaches that are also valid and have their right to exist.

[1] Named after Ernst Friedrich Ferdinand Zermelo (1871–1953) and Adolf Abraham Halevi Fraenkel (1891–1965).
[2] By Per Erik Rutger Martin-Löf (1942–today).

A First Decision For the definition of the natural numbers, we have a first decision to make. While Peano starts his original definition with 1 as the smallest number, current approaches start from 0. As it turns out, things become much easier and more elegant to use when we include 0. In a later version of Peano's foundation he also includes 0 as the base to start from.

We define the natural numbers, from now on denoted by $\mathbb{N}$, in MLTT by an inductive approach, meaning the numbers are based on each other:

Definition 2.1 $\dfrac{}{0 : \mathbb{N}}\ (\text{nat0}) \qquad \dfrac{n : \mathbb{N}}{\mathrm{S}\, n : \mathbb{N}}\ (\text{natS})$

As for each type in MLTT, the definition of $\mathbb{N}$ introduces the elimination rule $R_{\mathbb{N}}$ and the corresponding computation rules:

Definition 2.2 $R_{\mathbb{N}} : (\forall P : \mathbb{N} \to U) P\ 0 \to ((\forall n : \mathbb{N}) P(\mathrm{S}\, n)) \to (\forall n : \mathbb{N}) P\ n$

$$\frac{H : P\ 0 \qquad f : (\forall m : \mathbb{N}) P(\mathrm{S}\, m)}{R_{\mathbb{N}}\ P\ H\ f : (\forall n : \mathbb{N}) P\ n}\ P : \mathbb{N} \to U$$

And the computation rules $R_{\mathbb{N}}\ P\ H\ f\ 0 \succ H$ *and* $R_{\mathbb{N}}\ P\ H\ f\ (\mathrm{S}\, n) \succ f\ n$.

While the notation might seem a bit peculiar for the mathematical mature reader, we remark that we stick to the notation used by Martin-Löf, see [4] for example.

One can easily see that Axiom 1 follows directly from our definition of $\mathbb{N}$. Strictly speaking, we cannot state this axiom in MLTT since there is no *type of* operator. But nevertheless the statements are obviously satisfied by our definition of $\mathbb{N}$.

Axiom 1 (The natural numbers are well defined) *Zero is a natural number.*
The successor of each natural number is a natural number.

Equality There are different types of equality in MLTT. At least the definitional, propositional, and inductive equality can be taken into account ([4]). In the following we use the inductive equality. The reason is because the equality is parametric in a type argument, i.e. it depends on a type and has therefore the same properties for all types, and allows only trivial statements as axioms.

Definition 2.3 *We define the equality on elements of types as follows:*

$$\frac{}{Q\ a : a = a}\ (A : U), (a : A)$$

The corresponding eliminator $R_=$ *has type* $(\forall A : U)(\forall P : A \to U)(\forall a : A)(P\ a) \to (\forall b : A)(a = b) \to P\ b$.

$$\frac{H_1 : P\ a \qquad H_2 : a = b}{R_=\ A\ P\ a\ H_1\ b\ H_2 : P\ b}\ (A : U), (P : A \to U), (a, b : A)$$

To simplify the proof terms, we omit A, a, and b in the following quite often. Using this definition of equality we can state a second axiom:

Axiom 2 (Disjointness) *Zero is not the successor of any number:* $(\forall n : \mathbb{N})\, \mathrm{S}\, n \neq 0$

To prove that our definition of $\mathbb{N}$ indeed satisfies this axiom, we use the types $\top$ for truth that comes with the canonical element I and $\bot$, the falsity, which does not come with any element.

Proof. We use the rewriting rules for the equality to change a proof of truth into a proof of falsity assuming we have $\mathrm{S}\, n = 0$. For this distinction we use the eliminator $R_{\mathbb{N}}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\top}\, I}{(\lambda\ m.\ R_{\mathbb{N}}\ (\lambda\ \_.\ U)\ \bot\ (\lambda\ \_.\ \top)\ m)\ (\mathrm{S}\, n)} \qquad \dfrac{}{\mathrm{S}\, n = 0}\ \text{Assumption } H}{(\lambda\ m.\ R_{\mathbb{N}}\ (\lambda\ \_.\ U)\ \bot\ (\lambda\ \_.\ \top)\ m)\ 0}\ R_=}{\bot}$$

This results in the following proof term: $\lambda\, n\, H.\, R_=\, \mathbb{N}\, (\lambda\, m.\, R_{\mathbb{N}}\ (\lambda\ \_.\ U)\ \bot\ (\lambda\ \_.\ \top)\ m)\ (\mathrm{S}\, n)\ I\ 0\ H$ □

Axiom 3 (Injectivity) *Two numbers are equal if their successors are equal:*
$(\forall n, m : \mathbb{N})\, \mathrm{S}\, n = \mathrm{S}\, m \to n = m$

To prove the axiom, we define an auxiliary function, called the predecessor π. π returns the next smaller number or zero if no such number exists.

Definition 2.4 (The predecessor) $\pi \coloneqq R_{\mathbb{N}}\ (\lambda\ \_.\ \mathbb{N})\ 0\ (\lambda\ m.\ m)\ n$. *This gives us the computations:* $\pi\,0 \succ R_{\mathbb{N}}\ (\lambda\ \_.\ \mathbb{N})\ 0\ (\lambda\ m.\ m)\ 0 \succ 0$ *and* $\pi(\mathrm{S}\,n) \succ R_{\mathbb{N}}\ (\lambda\ \_.\ \mathbb{N})\ 0\ (\lambda\ m.\ m)\ (\mathrm{S}\,n) \succ (\lambda\ m.\ m)\ n \succ n$

Proof of Injectivity.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\pi(\mathrm{S}\,n)=\pi(\mathrm{S}\,n)}\,Q \quad \dfrac{}{\mathrm{S}\,n=\mathrm{S}\,m}\,\text{Assumption } H}{\pi(\mathrm{S}\,n)=\pi(\mathrm{S}\,m)}\,R_{=} \qquad \dfrac{}{\pi(\mathrm{S}\,m)=m}\,Q}{\pi(\mathrm{S}\,n)=m}\,R_{=} \quad \dfrac{}{\pi(\mathrm{S}\,n)=n}\,Q}{n=m}\,R_{=}$$

Which is equivalent to the following proof term:

$$\begin{aligned}
&\lambda\ n\ m\ H.\ R_{=}\ (\lambda\ k.\ k=m)\ (\pi(\mathrm{S}\,n))\\
&\qquad (R_{=}\ (\lambda\ k.\ \pi(\mathrm{S}\,n)=k)\ (\pi(\mathrm{S}\,m))\\
&\qquad\quad (R_{=}\ (\lambda\ k.\ \pi(\mathrm{S}\,n)=\pi\,k)\ (\mathrm{S}\,n)\ Q\ (\mathrm{S}\,m)\ H)\\
&\qquad m\ Q)\\
&\quad n\ Q
\end{aligned}$$

□

From now on we switch between proof terms and deduction trees. Both can be transformed into each other. For more complex proofs we only give the ideas behind the proof from which one can construct the proof terms or trees.

3 Induction

When we defined the natural numbers in the previous chapter, we said they are defined as an inductive type. But what does "inductive" and "induction" exactly mean? Closely related to this is the notation of an inductive set that is used more often in ZF. A set J is called inductive if $0 \in J$ and $(\forall n) n \in J \Rightarrow \mathrm{S}\,n = n \cup \{n\} \in J$. We observe, if J is inductive, then $\mathbb{N} \subseteq J$. Therefore, it suffices to show that a set or a property is inductive to prove that all natural numbers are in the set or satisfy the property, respectively. Thus, induction follows rather easily in ZF. But there is another concept closely related to induction: recursion. For recursion we work with the structure of the element and derive properties and decisions from the way the element was created.

In MLTT this concept of induction comes for free and is directly obtained from the definition of the natural numbers. While we had to prove the recursion theorem in ZF separately, we get this concept in MLTT also for free since induction and recursion coincide.

For stating the induction rule we first recap the elimination rule $R_{\mathbb{N}}$ for the natural numbers. This had type $(\forall P : \mathbb{N} \to U)P\ 0 \to ((\forall n : \mathbb{N})P(\mathrm{S}\,n)) \to (\forall n : \mathbb{N})P\ n$. For the induction principle, we extend the type of $R_{\mathbb{N}}$ by the assumption that we have already shown $P\ n$ when we want to prove $P(\mathrm{S}\,n)$.

Definition 3.1 (Induction Rule) $I_{\mathbb{N}} : (\forall P : \mathbb{N} \to U)P\ 0 \to ((\forall n : \mathbb{N})P\ n \to P(\mathrm{S}\,n)) \to (\forall n : \mathbb{N})P\ n$

$$\dfrac{H : P\ 0 \qquad f : (\forall m : \mathbb{N})P\ m \to P(\mathrm{S}\,m)}{I_{\mathbb{N}}\ P\ H\ f : (\forall n : \mathbb{N})P\ n}\ P : \mathbb{N} \to U$$

While the first computation rule is the same as for $R_{\mathbb{N}}$, the second one is changed to the following: $I_{\mathbb{N}}\ P\ H\ f\ (\mathrm{S}\,n) \succ f\ n\ (I_{\mathbb{N}}\ P\ H\ f\ n)$. Using this it is quite easy to see that

$$\lambda\ (P : \mathbb{N} \to U)\ (H_0 : P\ 0)\ (f : (\forall n : \mathbb{N})P(\mathrm{S}\,n))\ (n : \mathbb{N}).\ I_{\mathbb{N}}\ P\ H_0\ (\lambda\ (m : \mathbb{N})\ \_.\ f\ m)\ n$$

has the same type as $R_{\mathbb{N}}$. This makes the elimination rule $R_{\mathbb{N}}$ redundant and can be omitted from now on.

Axiom 4 (Induction) $(\forall P : \mathbb{N} \to U)P\ 0 \to ((\forall n : \mathbb{N})P\ n \to P(\mathrm{S}\,n)) \to (\forall n : \mathbb{N})P\ n$

For stating the more general *complete induction* we first need an ordering of the natural numbers. Both will be introduced in Chapter 4.

4 The Ordering of the Numbers

4.1 Less or Equal?!

In the last chapters we have seen the first Peano axioms and how they are modeled in MLTT. Besides working with the natural numbers, which we do in Chapter 6, we are missing the concept of comparing numbers. In Chapter 2 we have seen the inductive equality. For several types (e.g. booleans) equality is sufficient to compare their elements since we are only interested whether two entities are equal or not. However, this does not work for the natural numbers.

ZF We recall that in ZF the successor of a natural number n was defined by $n \cup \{n\}$. This gives us a very easy way to define the ordering by using the $\in$ operator. This ordering corresponds to the less than ($<$). Same as the standard math interpretation, this operator is transitive but not reflexive. To extend the ordering to a reflexive ordering satisfying the totality and thus the trichotomy, we can use a slightly more complex construct such that $n \leq m := n \in m \vee n = m$.[3] But we could also define $n \leq m := n \subseteq m$. Though, it is not obvious that both definitions are equivalent.

MLTT If we now want to model an ordering of the natural numbers in MLTT, we have to decide which way to go: Should the operator represent the *less than ($<$)* as in ZF or the *less or equal ($\leq$)*? In the following we use the less or equal. This results from the fact that once we defined $\leq$ we can use it to define the less than as follows, without using additional predicates such as equality:

Definition 4.1 (Less than) *Define for all $n, m : \mathbb{N}$: $n < m := \mathrm{S}\, n \leq m$*

Note We could have started by introducing $<$ as an inductive type as well. Then we can define $n \leq m := n < \mathrm{S}\, m$. But we stick to the definition above since this allows to prove reflexivity very easily.

But even with this decision made, we still have to define the $\leq$. With the definition of the natural numbers in mind, it should be clear that we use an inductive type for the $\leq$. We start from some trivial statements and use them to derive more complex relations. For this, we can use at least two ways to go, depending on the ground truth we start from: (i) $(\forall n : \mathbb{N}) 0 \leq n$ with the step rule $n \leq m \Rightarrow \mathrm{S}\, n \leq \mathrm{S}\, m$, or (ii) $(\forall n : \mathbb{N}) n \leq n$ with the step rule $n \leq m \Rightarrow n \leq \mathrm{S}\, m$. Note, that we could also use the step rules for the other ground truths, but by this we could not prove more statements than before! In Section 6.1 we show a third way to define the ordering based on addition.

In the following we use the second version. One of the reasons for this decision is that this definition corresponds to the definition of $\leq$ in the proof assistant Coq[4]. There, this definition has some benefits arising from the structure of type theory, which we do not carry out here (cf. [6] for further information). Another reason is shown below.

Definition 4.2 *Define $\leq$ as an inductive type:*

$$\frac{}{n \leq n}\ (\mathrm{le1}) \qquad \frac{m : \mathbb{N} \qquad n \leq m}{n \leq \mathrm{S}\, m}\ (\mathrm{le2}) \qquad \textit{In both rules } n : \mathbb{N}$$

One can show that this definition of $\leq$ is complete and sound, i.e. we can prove all statements that can be shown in usual mathematics and we can only prove these statements.

Induction on $\leq$ Even if it seems counterintuitive at the first glance to define induction for a predicate, we can define induction on $\leq$. This arises from the fact that in MLTT everything is a type or an element of a specific type. Therefore, induction on the predicate $\leq$ is no different than induction on $\mathbb{N}$ and has a quite similar induction rule. Note, by our design of $\leq$, the n in the inductive type always stays the same. Thus, we can quantify over it at first and call it therefore a parameter. This is the second reason why we have chosen this version of $\leq$.

Definition 4.3 (Induction Rule for $\leq$)
$I_{\leq} : (\forall n : \mathbb{N})(\forall P : \mathbb{N} \to U) P\ n \to ((\forall l : \mathbb{N})(n \leq l) \to P\ l \to P(\mathrm{S}\, l)) \to (\forall m : \mathbb{N})(n \leq m) \to P\ m$

$$\frac{H_0 : P\ n \qquad f : (\forall l : \mathbb{N})(n \leq l) \to P\ l \to P(\mathrm{S}\, l) \qquad H_1 : n \leq m}{I_{\leq}\ n\ P\ H_0\ f\ m\ H_1 : P\ m}\ (P : \mathbb{N} \to U), (n, m : \mathbb{N})$$

[3] This is the standard way to get a total order from a strict order.

[4] Coq ([2]) is an interactive proof assistant based on type theory that has many properties in common with MLTT.

This induction rule captures the idea that assuming (a) the property holds for a natural number n and (b) if the property holds for any number m greater or equal n it holds for the successor of m, we can show that it holds for all numbers greater or equal n.

A somewhat similar property is captured by complete induction. But instead of showing the property for all numbers greater than some other number, we do something reverse. Assuming, that we have shown the property for all smaller numbers then we can derive the property for this number, then the property holds for all numbers.

Lemma 4.1 (Complete Induction)
$(\forall P : \mathbb{N} \to U)((\forall n : \mathbb{N})((\forall m : \mathbb{N}) m < n \to P\ m) \to P\ n) \to (\forall n : \mathbb{N}) P\ n$

Proof. This statement can be shown by induction on n using several of the lemmata from the next section. The proof is actually the same as the one of Lemma 5.1. □

Note The crucial part why this type of induction works, is because there is no infinitely decreasing chain when ordering by $<$. A more detailed view on this is given in Chapter 5 where we have a look at the well-foundedness of the ordering.

4.2 More Peano Axioms

In Chapter 2 we have seen a first set of Peano axioms. Now we can use the results about induction and ordering to state more axioms.

Axiom 5 (Reflexivity) *The less or equal relation is reflexive:* $(\forall n : \mathbb{N}) n \leq n$

Obviously the first rule for $\leq$ (i.e. le1) proves this Axiom.

Axiom 6 (Transitivity) *The ordering $\leq$ is transitive:* $(\forall n, m, k : \mathbb{N}) n \leq m \to m \leq k \to n \leq k$

Proof. The main idea to prove the statement is an induction on the predicate $m \leq k$. If the inequality was constructed by (le1) then $m = k$ and the result follows directly. In the other case, when (le2) was used, we have the induction hypothesis $n \leq k'$ and we know that $m \leq k'$. From this we also get the claim directly.

This outline is formalized in the following proof tree and the corresponding proof term:

$$\dfrac{(H_0 : n \leq m) \vdash n \leq m \qquad \dfrac{(H_0 : n \leq m)(H_1^1 : m \leq k')(H_1^2 : n \leq k') \vdash n \leq k'}{(H_0 : n \leq m)(H_1^1 : m \leq k')(H_1^2 : n \leq k') \vdash n \leq \mathrm{S}\, k'}\,(\text{le2})}{\dfrac{(H_0 : n \leq m)(H_1 : m \leq k) \vdash n \leq k}{\vdash (\forall n, m, k : \mathbb{N}) n \leq m \to m \leq k \to n \leq k}}\,(I_\leq),\ P = \lambda\, l.\ n \leq l$$

$\lambda\ n\ m\ k\ H_0\ H_1.\ I_\leq\ m\ (\lambda\ l.\ n \leq l)\ H_0\ (\lambda\ k'\ H_1^1\ H_1^2.\ \text{le2}\ n\ k'\ H_1^2)\ k\ H_1$ □

Axiom 7 (Antisymmetry) *The ordering $\leq$ is antisymmetric:* $(\forall n, m : \mathbb{N}) n \leq m \to m \leq n \to n = m$

For proving this axiom we use the following two lemmata:

Lemma 4.2 *All successor of numbers are larger than zero:* $(\forall n : \mathbb{N})\, \mathrm{S}\, n \leq 0 \to \bot$

Lemma 4.3 (Remove the S) $(\forall n, m : \mathbb{N})\, \mathrm{S}\, n \leq \mathrm{S}\, m \to n \leq m$

Since both proofs are rather technical and gain not much insight into the work with natural numbers, we omit them here.

Proof of Axiom 7. We do an induction on n while we are still quantifying over m. Then we do a case analysis of m:

- $n = 0, m = 0$: It remains the trivial statement: $0 \leq 0 \to 0 \leq 0 \to 0 = 0$.
- $n = 0, m = \mathrm{S}\, m'$: It remains to show: $0 \leq \mathrm{S}\, m' \to \mathrm{S}\, m' \leq 0 \to 0 = \mathrm{S}\, m'$. For this we use Lemma 4.2 on the second inequality and the exfalso rule[5].
- $n = \mathrm{S}\, n', m = 0$: The remaining statement $\mathrm{S}\, n' \leq 0 \to 0 \leq \mathrm{S}\, n' \to \mathrm{S}\, n' = 0$ can be shown analogously to the previous case.
- $n = \mathrm{S}\, n', m = \mathrm{S}\, m'$: In addition we get the inductive hypothesis $H_{\text{ind}} : (\forall m : \mathbb{N}) n' \leq m \to m \leq n' \to n' = m$. It remains to show: $\mathrm{S}\, n' \leq \mathrm{S}\, m' \to \mathrm{S}\, m' \leq \mathrm{S}\, n' \to \mathrm{S}\, n' = \mathrm{S}\, m'$. By Lemma 4.3 we can remove the S in the assumptions and therefore the claim follows by H_{ind}. □

[5]The *ex falso quodlibet* rule (exfalso for short) has the form: $(\forall \phi : U) \bot \to \phi$, i.e. false implies everything.

Axiom 8 (Totality) *The ordering of the natural numbers is a total order:* $(\forall n, m : \mathbb{N}) n \leq m \vee m \leq n$

Proof. We first do an induction on n and a case distinction of m. Then we analyse the inductive hypothesis and use the claims $0 \leq n$ and $n \leq m \to \mathrm{S}\, n \leq \mathrm{S}\, m$. The proof of these claims is left as an easy exercise to the reader. □

Corollary 4.4 (Trichotomy) $(\forall n, m : \mathbb{N}) n < m \vee n = m \vee m < n$

Proof. The corollary follows by a case analysis of the totality of the two numbers and a distinction how the inequality was constructed. □

5 Well-Foundedness

In this chapter we focus on the well-foundedness of the ordering we defined in the last chapter. The usual definition of this is as follows: For all nonempty sets of natural numbers, there is a smallest element in the set. Expressed in formulas this means: $(\forall P \subseteq \mathbb{N}) P \neq \emptyset \to (\exists n \in P)(\forall m \in P) n \leq m$. This is quite natural since the trichotomy and transitivity show that there is a strictly decreasing chain of all elements in the set. Using this and the fact that for all natural numbers n there are exactly n natural numbers smaller than n, all decreasing chains have to terminate at some point (at the latest with 0). By this argument, it is clear that the statement also holds for undecidable sets such as $P = \{n \mid n \text{ is the Gödel number of some Turing machine that halts on itself}\}$, for example. In this case we cannot construct the element easily but we know it exists. So why do we have an own chapter for this topic?

Well-Foundedness in ZF We recall the idea behind the proof of the well-foundedness in ZF (cf. [3]): There the first step was that we could assume an element n_0 to be in P, this was a proof for the non-emptiness of P: i.e. $((\exists n_0 : \mathbb{N}) n_0 \in P) \equiv (P \neq \emptyset)$. Based on this, we made a case distinction whether this element n_0 is already the smallest element or not. While we are done in the positive case, we receive another element $n < n_0$ in the negative case such that we can apply the inductive hypothesis to obtain the final claim. If we now want to adopt this proof such that it works in MLTT, we have to deal with (at least) the two following problems: How do we get a witness that $P \neq \emptyset$? How to do the case distinction whether this element is already the smallest element or not? In general the problem is that we have to construct the smallest element. Since MLTT is a constructive theory, we cannot only show that such an element exists but we actually have to construct the element.

Modifications for MLTT We can give a workaround for the first problem, i.e. we modify the claim such that it provides us with an element that is in the set. But nevertheless, we still have to deal with the second problem. Therefore, we either have to change the definition or put some constraints on the set such that we can find an element. Later in this chapter we show some modification or rather a new definition of well-foundedness such that we can prove it in MLTT. So, we now focus on the conditions to put on the set. Given some element in the set, it is quite clear how to find the minimal one: we check for all numbers smaller than this number whether they are in the set or not. If we now choose a set for which we can decide membership, i.e. a decidable set, we are done. We say that a set is decidable if there is a computable function $f : \mathbb{N} \to \mathbb{B}$ such that $f(x) = \mathsf{true} \iff x \in P$. For this procedure we have assumed that we have access to some witness that P is non-empty. Surprisingly we do not need this element. It is sufficient if we know that there is an element. Based on this, we can design a new procedure to get the smallest element: Check whether $x \in P$: If $x \notin P$, then continue with $x = x + 1$, otherwise return x. When starting with $x = 0$ we can be sure to find the smallest element, and terminate because there is an element in the set. To actually encode this procedure in MLTT, we need a somewhat strange predicate (we call it acc in the following) that has the following introduction rule:

$$\frac{(n \notin P) \to \mathrm{acc}(\mathrm{S}\, n)}{\mathrm{acc}(n)}$$

Then it suffices to show that $(\exists n : \mathbb{N}) n \in P \to \mathrm{acc}\, 0$. How this can be done and the details of the above idea can be found in Chapter 11 of [6].

Well-foundedness in MLTT As elaborated in the previous part of the chapter we cannot show the standard well-foundedness without any assumption in MLTT. We can even go one step further and prove that the well-foundedness implies the principle of excluded middle (XM). XM captures the idea that all statements are either true or their negation is true (i.e. they are false).

Let p be a predicate p we want to decide. We define $P \coloneqq \lambda\, n.\ (n = 1) \vee (n = 0 \wedge p)$. P induces a set containing all numbers n such that $P\ n$ is true. By the well-foundedness we could get the smallest element m satisfying P. Doing a case distinction of m we can distinguish between both cases and therefore we could decide whether p holds or not.

Therefore, we first constrained the set for which we want to find the smallest element. In this part, we define a new condition for the well-foundedness and call if WF . This condition captures the idea that there is no infinitely decreasing chain of natural numbers. Or framed otherwise, every strictly decreasing chain of natural numbers eventually stops.

Definition 5.1 (Well-foundedness) *We define WF as follows:* $\dfrac{(\forall m : \mathbb{N}) m < n \to WF\ m}{WF\ n}$ (WFI), $n : \mathbb{N}$

Lemma 5.1 (Well-foundedness) *Every decreasing chain* $n_1 > n_2 > \ldots$ *is finite:* $(\forall n : \mathbb{N})\, WF\ n$

Proof. We show the claim by induction on n.

- $n = 0$: It remains to show $(\forall m : \mathbb{N}) m < 0 \to \mathrm{WF}\ m$ since we can analyse the statement WF 0 and there is only one constructor for elements of type WF . But from Lemma 4.2 we know $m < 0 \to \bot$, thus the exfalso rule suffices.
- $n \to \mathrm{S}\, n$: We are given the inductive hypothesis $H_{\mathrm{ind}} : \mathrm{WF}\ n$ and need to show $(\forall m : \mathbb{N}) m < \mathrm{S}\, n \to \mathrm{WF}\ m$ by the same reason as above. We assume $m < \mathrm{S}\, n$, so $\mathrm{S}\, m \leq \mathrm{S}\, n$. But by Lemma 4.3 we get $H : m \leq n$. Now we do a case distinction of H:
  - $m = n$: H_{ind} is a proof of the remaining statement WF m.
  - $m \leq n'$: n was replaced by $\mathrm{S}\, n'$ and thus $H_{\mathrm{ind}} : \mathrm{WF}\ (\mathrm{S}\, n')$. It still remains to show WF m. If we can show $m < \mathrm{S}\, n'$, then we can use H_{ind} to prove the remaining claim because it implies $(\forall m : \mathbb{N}) m < \mathrm{S}\, n' \to \mathrm{WF}\ m$. But the required claim reduces to $\mathrm{S}\, m \leq \mathrm{S}\, n'$ and this holds by the unproven claim used in the proof of Axiom 8 since $m \leq n'$ by assumption.

Another way to prove this lemma is by using complete induction as introduced in Lemma 4.1. Then the claim follows directly using WF as predicate. □

6 Working with Natural Numbers

In the last chapters we have seen a lot of statements and properties about the natural numbers. But one of the most important concepts is still missing: computation. We cannot use the properties proven to far to do actual computation, i.e. we cannot do real mathematics where numbers do not occur in a primitive way. We first introduce addition and multiplication in this chapter. Later we have a look at subtraction and the problems arising from it.

6.1 Addition

First, we define addition of natural numbers. For this we use the induction rule $I_{\mathbb{N}}$ from Chapter 3:

Definition 6.1 (Addition) $(\forall n, m : \mathbb{N})\, \mathrm{add}\ \ m\ n \coloneqq I_{\mathbb{N}}(\lambda\_ .\mathbb{N}) m (\lambda\_\ k.\, \mathrm{S}\, k) n$

For notational convenience we write $n + m$ in the following instead of add $\ n\ m$.

Example 6.2 $1 + 2 \succ^* \mathrm{add}\ \ 1\ (\mathrm{S}(\mathrm{S}\, 0)) \succ \mathrm{S}(\mathrm{add}\ \ 1\ (\mathrm{S}\, 0)) \succ \mathrm{S}(\mathrm{S}(\mathrm{add}\ \ 1\ \ 0)) \succ \mathrm{S}(\mathrm{S}(1)) \succ^* \mathrm{S}(\mathrm{S}(\mathrm{S}\, 0)) \succ^* 3$

Axiom 9 (Associativity) *Addition is associative:* $(\forall n, m, k : \mathbb{N})\ (n + m) + k = n + (m + k)$

Proof. This follows straight forward by an induction on k and by unfolding the definition of addition. □

Axiom 10 (Peano axioms for Addition) $(\forall n : \mathbb{N}) n + 0 = n$ *and* $(\forall n, m : \mathbb{N}) n + \mathrm{S}\, m = \mathrm{S}(n + m)$

While the axiom follows directly from our definition of addition, the proof of the following lemma is more involved even though both statements are quite similar.

Lemma 6.1 (Converse of Axiom 10) $(\forall n : \mathbb{N}) 0 + n = n$ *and* $(\forall n, m : \mathbb{N})\, \mathrm{S}\, n + m = \mathrm{S}(n + m)$

Using both lemmata we can show by an induction that addition of natural numbers is commutative.

Axiom 11 (Commutativity) *Addition is commutative:* $(\forall n, m : \mathbb{N}) n + m = m + n$

Proof of Lemma 6.1.
- We do an induction on n. While for $n = 0$ the statement is trivial, we have to prove $0 + \mathrm{S}\, n = \mathrm{S}\, n$ in the induction step. By applying the definition of addition, this reduces to $\mathrm{S}(0 + n) = \mathrm{S}\, n$ which follows by the inductive hypothesis.
- We do an induction on m. For $m = 0$ the claim is trivial. For the induction step we have to prove $\mathrm{S}\, n + \mathrm{S}\, m = \mathrm{S}(n + \mathrm{S}\, m)$ which reduces to $\mathrm{S}(\mathrm{S}\, n + m) = \mathrm{S}(\mathrm{S}(n + m))$ by the definition of addition. But this claim follows from the inductive hypothesis.

The two proof terms are: $\lambda\, n.\, I_{\mathbb{N}}\ (\lambda\, n.\, 0{+}n = n)\ Q\ (\lambda\, n\ H_{\mathrm{ind}}.\ R'_{=}\ (\lambda\, m.\ \mathrm{S}\, m = \mathrm{S}\, n)\ n\ Q\ (0{+}n)\ H_{\mathrm{ind}})\ n$ and $\lambda\ n\ m.\ I_{\mathbb{N}}\ (\lambda\ l.\ \mathrm{S}\, n + l = \mathrm{S}(n + l))\ Q\ (\lambda\ l\ H_{\mathrm{ind}}.\ R'_{=}\ (\lambda\ k.\ \mathrm{S}\, k = \mathrm{S}(\mathrm{S}(n + l)))\ (\mathrm{S}(n + l))\ Q\ (\mathrm{S}\, n + l)\ H_{\mathrm{ind}})\ m$ □

A new ordering relation As already mentioned in Chapter 4 we can now define the ordering of the natural numbers in a new way. This definition is based on a natural understanding of the ordering. A number m is greater or equal than n if there is some number k such that $n + k = m$. This k corresponds to some (non-negative) gap between the numbers n and m. In the definition of $\leq$ we have used so far, this k is also present but only implicitly: It corresponds to the number of applications of the (le2) rule to derive the final inequality.

Lemma 6.2 (Correctness of $\leq$) $(\forall n, m : \mathbb{N}) n \leq m \leftrightarrow (\exists k : \mathbb{N}) n + k = m$

Proof. The $\rightarrow$ direction follows from an induction on $n \leq m$. The other direction $\leftarrow$ follows by an induction on k and a repeated application of the (le2) rule. □

Note This lemma shows that we can actually prove all statements that can be shown in usual mathematics and we can only show those, i.e. $\leq$ is complete and sound with respect to usual mathematics.

6.2 Multiplication

In the last section we have seen the definition of addition in MLTT. As in other systems like the ZF and usual mathematics we use addition to define multiplication. Therefore, it inherits all the well-known properties.

Definition 6.3 (Multiplication) $(\forall n, m : \mathbb{N})\, \mathrm{mult}\ \ m\ n \coloneqq I_{\mathbb{N}}\ (\lambda\ \_.\ \mathbb{N})\ 0\ (\lambda\ \_\ \ k.\ k + m)\ n$

We write $n \cdot m$ instead of $\mathrm{mult}\ \ n\ m$ in the following. Similar as for addition, the following axiom is precisely captured by our definition of multiplication:

Axiom 12 (Peano axioms for Multiplication) $(\forall m : \mathbb{N}) m \cdot 0 = 0$ *and* $(\forall n, m : \mathbb{N}) m \cdot \mathrm{S}\, n = (m \cdot n) + m$

Doing some more elaborate proofs we can show the following lemmata:

Lemma 6.3 (Distributivity) *Multiplication is distributive over addition:*
$(\forall n, m, k : \mathbb{N})(m + k) \cdot n = m \cdot n + k \cdot n$

Lemma 6.4 (Associativity and Commutativity) *Multiplication is associative and commutative:*
$(\forall n, m, k : \mathbb{N})\ (n \cdot m) \cdot k = n \cdot (m \cdot k)$ *and* $(\forall n, m : \mathbb{N}) n \cdot m = m \cdot n.$

6.3 Subtraction

In the last two section we have seen the two basic arithmetic operations addition and multiplication. Since the remaining operations subtraction and division are closely related to the other two, one could think of introducing the missing ones as well. We first start with subtraction. But when working with natural numbers and subtraction, we run pretty fast into a problem. While statements as $5 - 2$ or $2 - 2$ return positive results, the term $1 - 2$ results in a negative number in usual mathematics. But since MLTT forces everything to have a type, we have to find the type of -1. With our definition from Chapter 2, this is not possible since the smallest number is 0. Therefore, we have to think of another solution of this problem. To avoid these problems with negative numbers, we use a special minus, the truncating minus, which simply ignores values smaller than 0. While this seems to be an ugly hack, it turns out that this system still satisfies some properties. A discussion how to implement negative numbers appropriately is postponed to the end of the section.

Definition 6.4 (Definition of truncating minus) $(\forall m, n : \mathbb{N})\, \mathrm{minus}\ \ m\ n \coloneqq I_{\mathbb{N}}\ (\lambda.\ \mathbb{N})\ m\ (\lambda\ \_\ \ k.\ \pi\, k)\ n$

For simplicity we write $m - n$ instead of minus m n. With this definition we can already compute the result of our examples above:

Example 6.5 $2 - 2 \succ^*$ minus 2 $(\mathrm{S}(\mathrm{S}\,0)) \succ \pi(\text{minus } 2\ (\mathrm{S}\,0)) \succ \pi(\pi(\text{minus } 2\ 0)) \succ^* \pi(\pi(\mathrm{S}(\mathrm{S}\,0))) \succ^* 0$
and $1 - 2 \succ^*$ minus 1 $(\mathrm{S}(\mathrm{S}\,0)) \succ \pi(\text{minus } 1\ (\mathrm{S}\,0)) \succ \pi(\pi(\text{minus } 1\ 0)) \succ^* \pi(\pi(\mathrm{S}\,0)) \succ \pi(0) \succ 0$

But this definition of minus entails various problems. While in usual mathematics addition and subtraction are associative and commutative, this does not hold for this operator. This can easily be seen by the following two examples:

Example 6.6

Associativity:	*Commutativity:*
$1 + (1 - 2) \succ^* 1 + 0 \succ^* 1$	$(1 + 2) - 2 \succ^* 3 - 2 \succ^* 1$
$(1 + 1) - 2 \succ^* 2 - 2 \succ^* 0$	$(1 - 2) + 2 \succ^* 0 + 2 \succ^* 2$

But nevertheless this definition still satisfies some properties and is not too wrong.

Lemma 6.5 *Truncating minus is sound:*
1. $(\forall m : \mathbb{N}) m - 0 = m$
2. $(\forall n, m : \mathbb{N})(m + n) - n = m$
3. $(\forall n : \mathbb{N}) n - n = 0$

And therefore $(\forall n, m : \mathbb{N})(m + n) - n = m + (n - n)$.

Proof.
1. Follows directly from applying the computation rule for $I_{\mathbb{N}}$.
2. Follows by induction on n.
3. Even if the statement seems highly trivial, the proof requires two additional lemmas. After proving $(\forall n, m : \mathbb{N})\, \mathrm{S}\, n - \mathrm{S}\, m = n - m$ and $(\forall n, m : \mathbb{N})\, \pi(\mathrm{S}\, m - n) = m - n$, we can show the claim by induction on n. □

As we have seen above, the problem with subtraction arises if the result of the computation should be negative. When we condition that these events do not occur, we can show more interesting results. The following result is closely related to Statement 2 in Lemma 6.5:

Lemma 6.6 (Lemma 6 in § 2 of [5]) $(\forall n, m : \mathbb{N}) n < m \to (m - n) + n = m$.[6]

One can clearly see that the constraint $n < m$ is required since the claim is false for $n = 1$ and $m = 0$.

A new approach to Integers As we have shown above, it is not possible to do real mathematics only with natural numbers. To have access to subtraction, we need a model/type that has full support for all (positive and negative) integers. For this we define a new type $\mathbb{Z}$ that is based on $\mathbb{N}$. The integers are modeled as tuples $(a, b) \in \mathbb{Z} = \mathbb{N} \times \mathbb{N}$ such that they are interpreted as the number $a - b$, i.e. a is the positive part and b the negative part. Due to this intuitive idea behind the concept, this way to design the integers is also used in some undergrad math classes (e.g. [1]).

With this definition it is easy to define addition and subtraction for $\mathbb{Z}$: $(a, b) + (c, d) \coloneqq (a + c, b + d)$ and $(a, b) - (c, d) \coloneqq (a + d, b + c)$. Furthermore, we can also define multiplication: $(a, b) \cdot (c, d) \coloneqq ((a \cdot c) + (b \cdot d), (a \cdot d) + (b \cdot c))$. With these definitions we can reuse some of the previous results such as commutativity since $\mathbb{Z}$ is based on $\mathbb{N}$.

While this definition is very intuitive to use, it has one major problem: We loose the uniqueness of the representation. For $\mathbb{N}$ it was quite clear that for all numbers there is only one element of type $\mathbb{N}$ that represents this number. But for the above definition of $\mathbb{Z}$ this does not hold anymore: For example, $(3, 2)$ and $(4, 3)$ represent the number 1. To handle this problem, we can define a canonical version (a, b) of each number/tuple for which we require that $a = 0 \vee b = 0$. By identifying tuples with their canonical version, the representation becomes unique. This canonical version for the above example would be $(1, 0)$. Since one value is zero, the other one corresponds to the absolute value of the integer and the position encodes the sign.

Since we have infinitely many representations of each integer, we could think of another encoding: Based on the above observation, we can define a new type $\hat{\mathbb{Z}}$ that consists of tuples with a boolean and a natural number. With $\hat{\mathbb{Z}} = \mathbb{B} \times \mathbb{N}$ the boolean value is a flag for the sign and the natural number is the absolute value. But even here not all numbers are represented in a unique way: For 0 we have the representations $(\mathsf{true}, 0)$ and $(\mathsf{false}, 0)$. We leave it to the reader to find new encodings of the integers besides the shown ones. But in general, all of these possible encodings have different advantages and disadvantages when compared to each other.

[6] Of course the statement also hold if $n = m$, but we stick to Peano's definition.

Division While defining subtraction and negative numbers, we have seen that there are many possible ways to go. All of them have different strengths and weaknesses. And of course the above list does not capture all possible ways to encode the integers in MLTT. As one knows from usual mathematics, subtraction is the "inverse" operation of addition and likewise is division the inverse of multiplication. Therefore, one can think of extending $\mathbb{Z}$ with the division operator. But analogously to subtraction we have to deal with rational numbers that are not integer. This is somewhat similar to the negative numbers in the subtraction setting. For this reason one could think of constructing a new type $\mathbb{Q} = \mathbb{Z} \times \mathbb{Z}$ where the first component represents the numerator and the second one the denominator. Unfortunately we then have the same non-uniqueness as for $\mathbb{Z}$. But since a discussion of rationals if far away from Peano arithmetics, we stop here and leave it to the reader to carry out the details of the definition and to find the problems and how to work around them.

7 Conclusion

After defining the natural numbers, we have seen several properties and ways how to work with them. When recalling the definition of the natural numbers, we can think of an augmented version of the rules: the step rule can be augmented by some label, i.e. we label the steps by elements of an arbitrary but fixed type. By this, we can extend the concept of natural numbers to lists. In this definition of lists, every list is either an empty list or a label in combination with a subsequent list. But when starting from lists, we can also define the natural numbers as lists with labels of type $\top$.

In the following, we first give an overview of the concepts we have seen so far and how they are represented in MLTT and in ZF. We also take a look at the comparison to usual mathematics and show, if appropriate, which system models this behavior better. The second part deals with general problems of both systems. In the last part, we give a personal opinion which proof system seems more appropriate to model the natural numbers to formalize mathematics.

7.1 Comparison to ZF

Definition The most important fact about (natural) numbers is their definition. Since in ZF everything is a set, we have to use sets to model the natural numbers. Starting from interpreting the empty set as zero, we extend this in a clever way to the other numbers. Thus, the feeling for the numbers is not very intuitive since we first have to "translate" by this interpretation for the sets. In contrast, in MLTT we have a new type for the natural numbers. By this we do not have to interpret other elements and describe their meaning. Therefore, this idea is more intuitive since the encoding corresponds to the unary encoding of a natural number. By forcing that everything has a type in MLTT, we do not have to care about functions applied to element whose type is not $\mathbb{N}$; this is not even possible. This distinction cannot be done in ZF because there everything is a set and we cannot distinguish a priori between sets that represent numbers and other sets without a meaning.

Induction As we have seen during the proofs in the last chapters, induction is one of the most important techniques for proving facts about natural numbers. In ZF the induction principle follows directly from the definition of inductive sets (cf. Chapter 3). For MLTT we get this result from the induction rule that results from the definition of the natural numbers; it is an inherent part of MLTT. While induction follows in both systems, a related feature is computation or recursion and shows a difference between both systems. In general, recursion is used to define and evaluate functions. As we have seen above, we used this concept to define addition and multiplication in MLTT. But in ZF we needed the recursion theorem ([3]). This theorem gives us the possibility to combine a base value $b \in \mathbb{N}$ with a step function $\sigma : \mathbb{N} \to \mathbb{N}$ to obtain a unique function $f : \mathbb{N} \to \mathbb{N}$ such that $f(0) = b$ and $f(\mathrm{S}\, n) = \sigma(f(n))$. The corresponding function in MLTT can be written down straight forward as $f \coloneqq \lambda\, n.\ I_{\mathbb{N}}\ b\ (\lambda\ \_.\ \sigma)\ n$.

Ordering One of the most important concepts of natural numbers apart from induction is the ordering of the numbers. As mentioned in Chapter 4, the relation in ZF corresponds to less than. This is because we use the build-in relation $\in$ on the sets to obtain the ordering of the numbers. The less or equal, in contrast, requires the additional use of equality or the use of $\subseteq$ as the less or equal relation. Since MLTT does not come with any non-trivial relations,[7] we have to define a new relation. While this first seems like a problem, it gives us the freedom of designing the relation. This freedom of choice occurred here

[7] The inductive equality only allows trivial statements to prove.

for the second time after we have seen it for the first time while defining the natural numbers. We saw it for the third time when defining the integers. Later we come back to this when giving a final conclusion.

Proofs Proving properties of the natural numbers is a first benchmark for a foundation of mathematics. Besides defining the numbers and stating their properties, the formalization focuses on the preciseness of the proofs. The proofs should be checkable by checking small steps and concluding that the whole proof is correct. In an optimal setting, these proofs are machine checkable such that the user can focus on the main ideas of the proof and let the checker do the bookkeeping. Since ZF is based on first order, we can prove via constructing proof trees for the statements. For MLTT we can also give proof trees by using the typing rules and their representation as proof rules. But this does not correspond to the main idea behind MLTT; everything is an element of a specific type. While the statements are, strictly speaking, special types (mostly function types), a proof is an element of this type. Therefore, the MLTT-manner of proving a statement is giving an element of this type, i.e. a proof term. But no matter which way we choose to go and whether it is ZF or MLTT, the proof terms and the proof trees grow very fast and are not easy to understand already after a few steps.

Well-foundedness One special property of the natural numbers is well-foundedness. It states that every non-empty set of natural numbers has a smallest element. While all other properties we have seen can be proved in ZF and in MLTT, we find a difference for this predicate. In ZF this statement is easily provable, but in MLTT we encountered some problems. The first way to deal with it was to constraint the set such that we could decide the membership. In this case we showed that one can prove the well-foundedness given a witness for the non-emptiness of the set and even without this witness. In the second part we defined a new predicate that captured the idea that every decreasing chain is finite. With this we could phrase the usual well-foundedness in a new way. The well-foundedness was the first property of the natural numbers that could not be proven in both systems. If one continues to prove, one will encounter more statements that are not provable in MLTT. But this is a general problem and is based on the "weaknesses" of the proof system or more precisely on the difference between classic and constructive proof systems. In the next section we come back to this difference.

Working with natural numbers We need to be able to work with natural numbers to be able to deal with non-primitive numbers. After we defined the recursion theorem in ZF, we used it to define addition and multiplication. The same can be done in MLTT using the recursion rules. Both systems are able to express the Peano axioms and are from this point of view equivalent. But the computation in MLTT follows a more intuitive way by using the built-in rules and no additional structures. A big benefit of MLTT is the possibility that we are not only restricted to natural numbers and one type, but we can also define new types with new elements. One application of this was the definition of the integers in Section 6.3. The product construction of the integers can also be done in ZF using tuples. But in this case we have to give an additional interpretation since we are reusing sets again. While this might work out in this case, it is a very elaborate task to do this for other definitions of the integers. Here we can see the third time the freedom to design new types in MLTT. We are not restricted to the existing types, but we can extend the system by arbitrary types representing structures we need.

7.2 General Problems

Power of the Proof System Another important measurement is the power of the proof system, i.e. which statements can be proven or not. The most famous pattern used is XM. In usual mathematics this is not a big problem and most people do not care about it because "why should XM not be true". But when formalizing proofs we cannot work with such vague statements. Thus, one can decide whether to accept XM as an axiom or not. In MLTT, XM is not included and therefore one cannot prove all statements that can be shown in usual mathematics. Likewise as above, this can be seen as a problem or as the motivation to try proving statements without using XM and only use it if it seems that there is no way to succeed without it.

Automation Arising from huge proof trees and terms one could think of some automation of proving. Such a proof assistant is not required to carry out the whole proof by itself, but it should assist the user and do the bookkeeping of all the variables and small details. Because in usual mathematics these details are usually not carried out precisely and formally and from this quite often errors arise since some small details have been overlooked and are not taken into account. Even though such systems exists for ZF

and MLTT, it seems that they are more convenient for MLTT because of its fixed structure with types and elements.

Another foundation The main problem of both approaches is that there are now at least two systems of about same proving strength. Everyone wanting to use a formal system to prove statements has to decide which to choose and results from one system cannot be translated in a trivial way to results in the other system. Thus, two separate systems are developed that have no common interface. But even when a system is chosen, there are still many possible ways to go. When only considering MLTT, we also have seen several ways to model structures. Depending on these definitions, some statements are trivial to prove while others are harder. For example, when we change our definition of addition to be recursive on the first element, then Lemma 6.1 becomes trivial and Axiom 10 needs an as elaborate proof as the lemma currently does. This leads to the question how to compare results even within a formal system. As one can see, this freedom of choice makes proving quite often easier but comparisons are rather complex and more difficult to do.

7.3 Final Conclusion

In the last sections we have shown many similarities and differences between ZF and MLTT. Because of this, it is not possible to declare one system to be the winner. Which system is better, is more a question of style and personal feeling.

For us MLTT is the better tool to use. Mainly because of the reason that we can come up with new things and do not have to rely on already existing objects and constructs. The interpretation of already existing structures is not very intuitive. It is somewhat similar to searching for a treasure: Knowing what to search and how to search for it, one can make progress and understand what is going on, but without any clue one is a bit lost. But the introduction of new types is a very intuitive way to have access to something new. We can define the new types the way we need them and with the properties they need to prove properties in a nice way. But if we are given an existing inductive type that has the same syntactical structure as $\mathbb{N}$, we also have the "problem" of interpretation in MLTT.

But with all the discussions from the last chapters in mind we conclude with the very first sentence of Peano's first paper about formalizing the natural numbers ([5]):

> Quaestiones, quae ad mathematicae fundamenta pertinent,
> etsi hisce temporibus a multis tractatae,
> satisfacienti solutione et adhuc carent.
>
> Questions with regard to the foundation of mathematics,
> even after being considered by many during these times,
> still miss a satisfying answer.

References

[1] Janko Böhm. *Mathematik für Informatiker: Kombinatorik und Analysis*. `http://www.mathematik.uni-kl.de/~boehm/lehre/15_MfI/`. Summer 2015.

[2] *Coq Proof Assistant*. 2019. URL: `http://coq.inria.fr`.

[3] Dominik Kirst. "Foundations of Mathematics: A Discussion of Sets and Types". Bachelor's Thesis. Saarland University, 2018.

[4] Per Martin-Löf. *Intuitionistic Type Theory: Notes by Giovanni Sambin of a Series of Lectures Given in Padua, June 1980*. Prometheus Books, Napoli, June 1985.

[5] Guiseppe Peano. *Arithmetices Principia: Nova Methodo Exposita*. `https://archive.org/details/arithmeticespri00peangoog`. Rome/Florence, 1889.

[6] Gert Smolka. *Introduction to Computational Logic - Lecture Notes*. Summer 2018.