Separating Markov's Principles

Liron Cohen, Yannick Forster, Dominik Kirst, Bruno da Rocha Paiva, Vincent Rahli

Separating Markov's Principles

Liron Cohen
cliron@cs.bgu.ac.il
Ben-Gurion University of the Negev
Beer-Sheva, Israel

Yannick Forster
yannick.forster@inria.fr
Inria
Paris, France

Dominik Kirst
kirst@cs.bgu.ac.il
Ben-Gurion University of the Negev
Beer-Sheva, Israel

Bruno da Rocha Paiva
bmd202@student.bham.ac.uk
University of Birmingham
Birmingham, UK

Vincent Rahli
v.rahli@bham.ac.uk
University of Birmingham
Birmingham, UK

ABSTRACT

Markov's principle (MP) is an axiom in some varieties of constructive mathematics, stating that Σ^0_1 propositions (i.e. existential quantification over a decidable predicate on $\mathbb{N}$) are stable under double negation. However, there are various non-equivalent definitions of decidable predicates and thus Σ^0_1 in constructive foundations, leading to non-equivalent Markov's principles. While this fact is well-reported in the literature, it is often overlooked, leading to wrong claims in standard references and published papers.

In this paper, we clarify the status of three natural variants of MP in constructive mathematics, by giving respective equivalence proofs to different formulations of Post's theorem, to stability of termination of computations, to completeness of various proof systems w.r.t. some model-theoretic semantics for Σ^0_1-theories, and to finiteness principles for both extended natural numbers and trees. The first definition ($\mathsf{MP}_{\mathbb{P}}$) uses a purely propositional definition of Σ^0_1 for predicates on natural numbers $\mathbb{N}$, while the second one ($\mathsf{MP}_{\mathbb{B}}$) relies on functions $\mathbb{N} \to \mathbb{B}$, and the third one ($\mathsf{MP}_{\mathsf{PR}}$) on a subset of these functions expressible in an explicit model of computation.

We then prove that $\mathsf{MP}_{\mathbb{P}}$ is *strictly* stronger than $\mathsf{MP}_{\mathbb{B}}$, and that $\mathsf{MP}_{\mathbb{B}}$ is *strictly* stronger than $\mathsf{MP}_{\mathsf{PR}}$ for variants of Martin-Löf's constructive type theory (MLTT), leading to separation results for the above theorems. These separations are achieved through a model construction of MLTT in $\mathrm{TT}^{\square}_{C}$, a type theory parameterised by effects, which can be syntactically restricted as needed. We replicate effectful techniques going back to Kreisel twice to refute different logical principles (first $\mathsf{MP}_{\mathbb{B}}$, then $\mathsf{MP}_{\mathbb{P}}$), while simultaneously satisfying variants of those principles (first $\mathsf{MP}_{\mathsf{PR}}$, then $\mathsf{MP}_{\mathbb{B}}$) when effects are restricted.

All our results are checked by a proof assistant.

Author version of paper published at 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '24), July 8–11, 2024, Tallinn, Estonia. https://doi.org/10.1145/3661814.3662104

1 INTRODUCTION

Markov's Principle (MP) is a central principle in constructive mathematics, nowadays most commonly stated as follows [7, 8, 46]:

$$\forall f : \mathbb{N} \to \mathbb{B}.\ \neg\neg(\exists n.\ fn = \mathrm{true}) \to \exists n.\ fn = \mathrm{true}$$

It states that Σ^0_1 propositions, i.e. existential quantifications over decidable predicates, are stable under double negation. While not generally accepted in all flavours of constructive mathematics, it is a principle of the Russian school led by Markov [3, Ch. 3]. It also has a central status in constructive reverse mathematics [8, 20], where it is well-known to be equivalent to a multitude of principles spanning many areas of mathematics and theoretical computer science, going back to Gödel's insight that his completeness proof for first-order logic w.r.t. Tarski semantics requires MP.

However, as is often the case when working in constructive foundations, there are multiple, non-equivalent ways to define MP, explainable by different possible definitions of when a predicate is decidable. The standard reference book in constructive mathematics by Troelstra and van Dalen discusses three versions [57]: The first, to which we refer as $\mathsf{MP}_{\mathbb{P}}$, states that existential quantification over classical predicates A, i.e. predicates for which $\forall n : \mathbb{N}.\ An \vee \neg An$ holds, is stable under double negation. The second, to which we refer as $\mathsf{MP}_{\mathbb{B}}$, uses Boolean-valued functions instead of classical predicates. Troelstra and van Dalen remark (in passing, without a theorem) that $\mathsf{MP}_{\mathbb{P}}$ and $\mathsf{MP}_{\mathbb{B}}$ are equivalent under the axiom of unique choice for natural number relations, and separable by a model using lawless (choice) sequences. The third variant, referred to as $\mathsf{MP}_{\mathsf{PR}}$, uses *primitive recursive* Boolean functions. $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$ are equivalent under the axiom CT ("Church's thesis", stating that any function $f : \mathbb{N} \to \mathbb{N}$ is computable in a model of computation such as Turing machines).

In general, the three variants of MP are *not* equivalent. The first such separation result was shown by Smorynski [52], who proves that $\mathsf{MP}_{\mathsf{PR}}$ is strictly weaker than $\mathsf{MP}_{\mathbb{P}}$ over Heyting arithmetic. The proof is based on Kripke's elaboration [32] of a proof idea of independence of MP due to Kreisel [26].

While unique choice, rendering $\mathsf{MP}_{\mathbb{P}}$ and $\mathsf{MP}_{\mathbb{B}}$ equivalent, is present in many constructive foundations, CT, rendering $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$ equivalent, is only present in CRM and independent in all other common foundations. Given the central status of MP in all of constructive mathematics, it may be surprising that confusion about these strictly different variants still arises frequently, though the confusion can possibly be explained by the strong historic connection of MP to CRM, where the variants coincide.

However, at the time of writing, Wikipedia [63] and the nLab [37] – two frequently consulted online sources maintained by well-known researchers in the community – contain mistakes about the status of MP, and so do recently published papers. We first discuss the rich history of consistency, independence, and separation results before explaining the frequent mistakes and how the present paper aims at clarifying the situation once and for all.

Derivability, Consistency, Independence. Derivability of $\mathsf{MP}_{\mathbb{P}}$ as a rule (only considering closed formulas) in (first-order) Heyting arithmetic goes back to Gödel's Dialectica interpretation [16], and the first consistency proof for a (higher-order) formal system was given for $\mathsf{MP}_{\mathbb{P}}$ by Kleene and Vesley [22]. Such consistency proofs can even be given while maintaining properties integral to intuitionistic logic such as the witness property [17].

There seem to be two main techniques to prove the independence of MP, both going back to Kreisel.

In 1958, Kreisel [25] observed that using modified function realizability, the Independence of Premise axiom (IP) holds, but together with $\mathsf{MP}_{\mathsf{PR}}$ implies that the limited principle of omniscience (LPO) holds in the model – which is contradictory in realizability. This technique was recently adapted by Pédrot and Tabareau [46] to show the independence of $\mathsf{MP}_{\mathbb{B}}$ from constructive type theory – a prominent, modern foundation of constructive mathematics – using exceptions as effects to validate IP and then by showing that adding MP would yield LPO again – which is contradictory since the type theory is still computational.

In another paper published in 1958, Kreisel furthermore mentions the idea that free choice sequences can be used to refute MP [26]. This technique has its roots in Brouwer's "weak counterexamples" [59], that rely on sequences built from undecided predicates to refute results. It was formally worked out by Kripke [32] (for his system FC) as well as Kleene and Vesley [22] (for **I**). Myhill [36] gives a more abstract account of the technique without using a model, by assuming a strong form of Kripke's schema and a negation of a consequence of LPO, which in turn could be proved consistent using free choice sequences. Recently, the technique was used by Coquand and Mannaa [7] to show the independence of $\mathsf{MP}_{\mathbb{B}}$ for constructive type theory, relying on a forcing extension of Martin-Löf Type Theory [40] (MLTT); as well as by Cohen and Rahli [5], who work in a general framework for effectful type theory, where the forcing conditions used in [7] are internalized in the object theory as effectful computations.

Clarifying the status of MP's variants. In constructive reverse mathematics [8, 20], where one proves equivalences between subclassical axioms and theorems, MP plays a central role. For instance, variants of MP are known to be equivalent to Post's theorem from computability theory [57], the statement that a computation that does not run forever terminates (this form is used in CRM), completeness theorems for first-order or propositional logic [8, 13, 18, 50], the statement that if an extended natural number is not infinite it is finite (which seems to be folklore and is noted in the nLab [37]), and the same statement for binary trees [2].

However, one needs to be careful not to confuse the different variants of MP. At the time of writing, the nLab [37] states that $\mathsf{MP}_{\mathbb{B}}$ is equivalent to "If a Turing machine does not run forever, then it halts". However, this is wrong in general, the statement about Turing machines is only equivalent to $\mathsf{MP}_{\mathsf{PR}}$, i.e. the stated equivalence would need CT. Wikipedia [63] states that $\mathsf{MP}_{\mathsf{TM}}$ – a variant of $\mathsf{MP}_{\mathbb{B}}$ which requires computability by Turing machines – is equivalent to $\mathsf{MP}_{\mathsf{PR}}$ assuming CT. However, CT is not necessary for the proof, since Kleene's normal form theorem is provable without axioms. Subsequently, Wikipedia states that $\mathsf{MP}_{\mathbb{B}}$ is equivalent to $\mathsf{MP}_{\mathsf{TM}}$ – *this* equivalence requires CT. Similar problems can also be found in research papers: E.g. Pédrot [44] states that $\mathsf{MP}_{\mathbb{B}}$ is equivalent to "a Turing machine that does not loop necessarily terminates", which again is only equivalent to $\mathsf{MP}_{\mathsf{PR}}$. An earlier version of [49] contained a similar formulation.

In this paper, we aim to clarify the status of MP once and for all. We explain how all three variants of MP are equivalent to respective, but strictly different versions to the principles listed above, including the one concerning termination.

Our proofs isolate the essence of the equivalences by working against double negation elimination over general classes of predicates. We identify the minimal closure properties for these classes, which can then be instantiated to different definitions of Σ^0_1 to yield the usual proofs. Our proofs are valid in any concrete foundation for constructive mathematics.

From Independence to Separation. In the same spirit as the work by Coquand, Mannaa, Pédrot, Tabareau, Cohen, and Rahli [5, 7, 46] adapting Kreisel's ideas to type theory, we proceed to construct models that separate the three variants of MP following Smorynski's idea [52]. Concretely, we build models for variants of MLTT relying on the effectful type theory $\mathrm{TT}^{\Box}_{C}$ [5], which is especially well-suited for this task for two main reasons. (i) It is formalised in Agda, and therefore provides high correctness guarantees, which is especially valuable for such brittle proofs. (ii) It provides constructs to syntactically restrict the effects terms can have, and in particular allows syntactically quantifying over effect-free functions, meaning that many arguments can be conducted in the object theory rather than having to be carried out in the metatheory.

Metatheory. The equivalence proofs between MP and other principles are conducted in informal constructive mathematics, meaning they hold in all foundations giving formal meaning to constructive mathematics. In particular they hold for intuitionistic higher-order arithmetics such as HA^{ω}, intuitionistic set theories such as IZF or CZF, or constructive type theories such as MLTT [40], the Calculus of Inductive Constructions [43] (CIC), or Homotopy Type Theory [58] (HoTT). The equivalence proofs are verified using the Coq proof assistant [54], which implements CIC, because CIC is the system among the ones mentioned above proving the *least* amount of choice principles. The separation arguments are stated for MLTT because MLTT is a sweet spot regarding the succinctness of its definition and is at the heart of most dependent type theories. They are verified using Agda [38, 39], because we are building on the existing Agda formalisations of $\mathrm{TT}^{\Box}_{C}$ [5] and MLTT [1, 41].

We use standard notation ($\forall, \exists, \wedge, \vee, \to, \neg, \top, \bot$); $\mathbb{N}$ for the type of natural numbers; $\mathbb{B}$ for the type of Booleans true and false; $\mathtt{List}(A)$ for the type of lists over A with elements $[]$ and $e :: l$ where $e : A$ and $l : \mathtt{List}(A)$. In addition, $l ::^{r} e$ appends e to l, and $l +\!\!+ k$ appends the list k to l.

Contributions. To summarise, our contributions are as follows:

First, we give an encyclopedic overview of well-known equivalence proofs in their most general form, simplifying several of them. E.g., we observe that the equivalence of MP and Post's theorem can be seen as an instance of the equivalence proof of the law of excluded middle and double negation elimination which makes the logical complexity of the predicates involved explicit. We also observe that completeness of a vast variety of well-known proof systems w.r.t. model semantics is equivalent to MP, by isolating that the system only has to be consistent (i.e. there needs to be an unprovable formula) and compact (i.e. derivations w.r.t. a possibly infinite theory only require finitely many assumptions). Secondly, we give a separation of $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$ in $\mathrm{TT}^{\square}_{C}$, and use the same strategy to separate $\mathsf{MP}_{\mathbb{P}}$ and $\mathsf{MP}_{\mathbb{B}}$. Thirdly, we contribute a model construction of MLTT in $\mathrm{TT}^{\square}_{C}$. This enables transporting the separation results to MLTT, yielding the first separation results for MP in type theory, strengthening previous independence results [5, 7, 46]. Furthermore, it enables new independence and separation results for MLTT via effectful techniques.

2 CONSTRUCTIVE NOTIONS OF EXISTENCE

In systems such as (higher-order) arithmetic or (first- or higher-order) set theory, propositions are defined via a language of logic. In type theory, one instead defines propositions as types via the Curry-Howard isomorphism. This modeling has some degrees of freedom, which entails what exactly is provable about propostions. In this paper, we write $\mathbb{P}$ for the set or type of propositions, and $\|X\|$ for the proposition that the type X is inhabited. We call the operation $\|\cdot\|$ a (weak) propositional truncation operation, following [24].

In (dependent) type theory, propositional truncation can be used to define (anonymous, logical) existence $\exists$ by applying it to the dependent pair type Σ. In MLTT, no native truncation operation is available and *all* types are morally identified with propositions, i.e. one could define truncation as the identity, entailing $\exists := \Sigma$.

In CIC, one has a separate (impredicative) universe of propositions and defines truncation of A as the inductive proposition with elements $\|a\|$ for $a : A$. Case analysis on proofs of inhabitedness is only allowed to produce another proof – never a type or something like a natural number. Consequently, proofs of $\exists$ cannot be turned into elements of the Σ type.

In HoTT, one models propositions semantically as types of which all inhabitants are equal, and obtains that $\|X\|$ isomorphic to $\forall P : \mathbb{P}.\ (X \to P) \to P$, i.e. one can eliminate from truncation into arbitrary propositions. In particular, if Ax ensures that x is unique, one can extract a witness x from $\exists x.\ Ax$.

As a consequence, the axiom of unique choice, which makes a connection between (unique) existence and Σ explicit, becomes provable in HoTT and MLTT:

$$\forall R : A \to B \to \mathbb{P}.\ (\forall a.\exists! b.\ Rab) \to \exists g.\forall a.\ Ra(ga)$$

This is because the proposition is equivalent to

$$\forall R : A \to B \to \mathbb{P}.\ (\forall a.\exists! b.\ Rab) \to \|\forall a.\Sigma b.\ Rab\|$$

The full (type-theoretic) axiom of choice, replacing unique existence $\exists!$ with existence $\exists$, however is only provable in MLTT. Notably, all mentioned systems, including the set-theoretic ones, prove the following variant of the axiom of choice for relations over natural numbers, where the relation is computationally decidable:

$$\forall f : A \to \mathbb{N} \to \mathbb{B}.\ (\forall a.\exists b.\ fab = \mathsf{true}) \to \exists g.\forall a.\ fa(ga) = \mathsf{true}$$

One often calls this (countable) Δ^0_1-choice. The proof of this principle differs by system. In MLTT, it is a simple consequence of the full axiom of choice. In CIC, it can be constructed via explicit recursion on an inductive well-foundedness proof of a certain relation. In HoTT and set theory, it follows from unique choice, because the following equivalence holds in all discussed systems:

$$\begin{aligned}\forall f : \mathbb{N} \to \mathbb{B}.\ (&\exists n.\ fn = \mathsf{true}) \\ &\leftrightarrow (\exists! n.\ fn = \mathsf{true} \wedge \forall m < n.\ fm = \mathsf{false})\end{aligned}$$

Note that in all systems, the Δ^0_1 variant of choice immediately generalises to the following Σ^0_1 version where $f : A \to \mathbb{N} \to \mathbb{N} \to \mathbb{B}$:

$$(\forall x.\exists n.\exists m.\ fxnm = \mathsf{true}) \to \exists g.\ \forall x.\ \exists m.\ fx(gx)m = \mathsf{true}$$

Note that for CIC, HoTT, and set theory, a Π^0_1 variant where $\exists m$ is replaced by $\forall m$ is already not provable anymore. In CIC, even the unique choice variant with $\exists! n.\forall m$ for this fails.

3 MARKOV'S PRINCIPLES

Markov's principle (MP) is a form of double negation elimination for a certain class of propositions, namely Σ^0_1 propositions. We first discuss double negation elimination over arbitrary classes of propositions C, and then introduce three non-equivalent definitions of Σ^0_1, before using them to introduce three different variants of MP.

We call any $C : \forall X.\ (X \to \mathbb{P}) \to \mathbb{P}$ a *class of predicates*. When dealing with such classes we leave the first type argument implicit. We write $C(P)$ given a proposition P as shorthand for $C(\lambda\_ : \mathbb{N}.\ P)$.

We call a class C of predicates *closed under disjunction* if $C(p) \to C(q) \to C(\lambda x.\ px \vee qx)$, *closed under point restriction* if $C(p) \to \forall x : X.\ C(\lambda y : X.\ px)$, and *closed under emptiness* if $C(\lambda x : X.\ \bot)$.

We now introduce double negation elimination over C-propositions, and prove that under certain mild assumptions on C, this is equivalent to double negation elimination over C-predicates.

$$C\text{-}\mathsf{DNE} := \forall P : \mathbb{P}.\ C(P) \to \neg\neg P \to P$$

Theorem 3.1. *For any class C that is closed under point restriction, C-DNE is equivalent to $\forall p : \mathbb{N} \to \mathbb{P}.\ C(p) \to \forall n.\ \neg\neg pn \to pn$.*

Proof. The left-to-right direction requires closure under point restriction, the other direction is straightforward. □

Note that while DNE (for the full class of all propositions) is equivalent to the law of excluded middle $\mathrm{LEM} := \forall P.\ P \vee \neg P$, we do not further discuss C-LEM because the equivalence does not hold for all classes C: Additionally to being closed under disjunction, C would have to be closed under negation – a condition not fulfilled by e.g. Σ^0_1 predicates which we now introduce.

A predicate is Σ^0_1 if it can be characterised by an existential quantification followed by a decidable predicate. Three definitions of decidable predicates arise naturally in sufficiently expressive foundations of constructive mathematics: via classical predicates, i.e. predicates $A : \mathbb{N} \to \mathbb{P}$ such that $\forall n.\ An \vee \neg An$; via Boolean functions $f : \mathbb{N} \to \mathbb{B}$; and via *computable* Boolean functions defined using a

model of computation, e.g. Turing machines (TMs). Formally:

$$\mathbb{P}\text{-}\mathcal{D}(p) := \forall x.\ px \vee \neg px$$
$$\mathbb{B}\text{-}\mathcal{D}(p) := \exists f : \mathbb{N} \to \mathbb{B}.\forall x.\ (px \wedge fx = \text{true}) \vee (\neg px \wedge fx = \text{false})$$
$$\mathsf{TM}\text{-}\mathcal{D}(p) := \exists f : \mathbb{N} \to \mathbb{B}.\forall x.\ (px \wedge fx = \text{true}) \vee (\neg px \wedge fx = \text{false}) \wedge \mathsf{TM}\text{-computable}f$$

We verbalise the forms as *propositionally decidable*, *Boolean decidable*, and TM-*decidable* predicates. Note that TM-decidable predicates are Boolean decidable, and Boolean decidable predicates are propositionally decidable. Conversely, Boolean decidable predicates are TM-decidable under the axiom CT ("Church's thesis"), stating that *every* function is (TM-)computable [28, 29]. Furthermore, propositionally decidable predicates are Boolean decidable under unique choice. In systems with computational sum types $A + B$, one has:

Lemma 3.2. $\mathbb{B}\text{-}\mathcal{D}(p) \leftrightarrow \|\forall x.\ (px) + (\neg px)\|$

Those three definitions of decidability give rise to three natural and non-equivalent definitions of Σ^0_1 predicates. We can give the definition parametrical in a definition of decidability:

$$D\text{-}\Sigma^0_1(p) := \exists A : X \to \mathbb{N} \to \mathbb{P}.\ D(p) \wedge \forall x.\ px \leftrightarrow \exists n.\ Axn$$

We abbreviate $\mathbb{P}\text{-}\mathcal{D}\text{-}\Sigma^0_1$, $\mathbb{B}\text{-}\mathcal{D}\text{-}\Sigma^0_1$, and $\mathsf{TM}\text{-}\mathcal{D}\text{-}\Sigma^0_1$ as $\mathbb{P}\text{-}\Sigma^0_1$, $\mathbb{B}\text{-}\Sigma^0_1$, and $\mathsf{TM}\text{-}\Sigma^0_1$ respectively. Note that all three notions of Σ^0_1 are closed under disjunctions, point restriction, and emptiness.

We now introduce three versions of MP and then prove that they are exactly DNE for the different notions of Σ^0_1 introduced. Technically, we start by introducing *four* versions of MP in the forms they often appear in the literature: using classical propositions, Boolean functions, computable functions, and primitive recursive functions. The last two versions of MP are well-known to be equivalent via the Kleene normal form theorem, we give the proof below.

$$\begin{aligned}
\mathsf{MP}_{\mathbb{P}} &:= \forall A : \mathbb{N} \to \mathbb{P}.\ (\forall n.\ An \vee \neg An) \to \neg\neg(\exists n.\ An) \to (\exists n.\ An)\\
\mathsf{MP}_{\mathbb{B}} &:= \forall f : \mathbb{N} \to \mathbb{B}.\ \neg\neg(\exists n.\ fn = \text{true}) \to (\exists n.\ fn = \text{true})\\
\mathsf{MP}_{\mathsf{TM}} &:= \forall f : \mathbb{N} \to \mathbb{B}.\ \mathsf{TM}\text{-computable}f \to \neg\neg(\exists n.\ fn = \text{true}) \to (\exists n.\ fn = \text{true})\\
\mathsf{MP}_{\mathsf{PR}} &:= \forall f : \mathbb{N} \to \mathbb{B}.\ \text{primitive-recursive } f \to \neg\neg(\exists n.\ fn = \text{true}) \to (\exists n.\ fn = \text{true})
\end{aligned}$$

Theorem 3.3. *We have* $\mathsf{MP}_{\mathbb{P}} \to \mathsf{MP}_{\mathbb{B}}$ *and* $\mathsf{MP}_{\mathbb{B}} \to \mathsf{MP}_{\mathsf{TM}}$ *due to* $\mathsf{MP}_{\mathbb{P}} \leftrightarrow \mathbb{P}\text{-}\Sigma^0_1\text{-DNE}$, $\mathsf{MP}_{\mathbb{B}} \leftrightarrow \mathbb{B}\text{-}\Sigma^0_1\text{-DNE}$, $\mathsf{MP}_{\mathsf{TM}} \leftrightarrow \mathsf{TM}\text{-}\Sigma^0_1\text{-DNE}$

The first implication is an equivalence given the axiom of unique choice and the second under CT. $\mathsf{MP}_{\mathbb{B}}$ is proved independent from type theory by Coquand and Mannaa [7] as well as by Pédrot and Tabareau [46], and $\mathsf{MP}_{\mathsf{TM}}$ by Forster, Kirst, and Wehr [13] (called MP_{L}). In Appx. A we discuss that formulating MP via Σ instead of $\exists$, $\neg\forall\neg$ instead of $\neg\neg\exists$, or $\mathbb{N}$ instead of $\mathbb{B}$ does not matter.

Kleene's normal form theorem [21] states that the termination of any partial computable function can be expressed via a single existential quantification and a primitive recursive predicate:

Theorem 3.4. *Let f be a partial computable function. Then there exists a primitive recursive total function g such that fx terminates with value v if and only if $\exists n.\ g\,x\,v\,n = \text{true}$.*

Since standard pairing function on the natural numbers are also primitive recursive, we have that $\mathsf{MP}_{\mathsf{PR}}$ and $\mathsf{MP}_{\mathsf{TM}}$ are equivalent.

Theorem 3.5. $\mathsf{MP}_{\mathsf{PR}} \leftrightarrow \mathsf{MP}_{\mathsf{TM}}$

Proof. The right-to-left direction is straightforward, since any primitive recursive function is TM-computable.

For the other direction, let f be a TM-computable function. Via the Kleene normal form theorem, we obtain a primitive recursive g such that $\forall x.\ \exists n.\ g\,x\,(f\,x)\,n = \text{true}$. Now, define $hx := g\,(\pi_1 x)\,\text{true}\,(\pi_2 x)$, where $\pi_{1,2}$ are projections of pairing.

We have that $\exists n.\ fn = \text{true} \leftrightarrow \exists nm.\ g\,n\,\text{true}\,m = \text{true} \leftrightarrow \exists x.\ g\,(\pi_1 x)\,\text{true}\,(\pi_2 x) = \text{true} \leftrightarrow \exists x.\ hx = \text{true}$. In particular $\neg\neg(\exists x.\ hx = \text{true}) \to (\exists x.\ hx = \text{true})$ (which follows from $\mathsf{MP}_{\mathsf{PR}}$) implies $\neg\neg(\exists n.\ fn = \text{true}) \to (\exists n.\ fn = \text{true})$. □

4 EQUIVALENCE TO OTHER PRINCIPLES

MP is central in constructive reverse mathematics [8, 20]. It is equivalent to a wide range of principles from computability theory (Post's theorem and $\neg\neg$-stability of termination of computations), logic (completeness theorems), or regarding basic concepts such as the one-point compactification of natural numbers or binary trees.

In this section, we define all these principles in a general form by abstracting out classes of predicates, and then give the equivalence proofs only requiring necessary closure properties of the classes.

The accompanying Coq proofs are in `equivalences.v`.

4.1 Post's Theorem

Post's Theorem (PT) [48, 57] states that Σ^0_1 predicates with complement in Σ^0_1 are decidable:

$$D\text{-PT} := \forall p : \mathbb{N} \to \mathbb{P}.\ D\text{-}\Sigma^0_1(p) \to D\text{-}\Sigma^0_1(\lambda x.\ \neg px) \to D(p)$$

Theorem 4.1. *Let C be a class that is closed under disjunction, point restriction, and emptiness. Then C-DNE if and only if $\forall x.\ qx \vee \neg qx$ for any $q : \mathbb{N} \to \mathbb{P}$ such that q and its complement are C.*

Proof. For the left-to-right direction, to prove $qx \vee \neg qx$, we can apply C-DNE because C is closed under disjunction. It suffices to prove $\neg\neg(qx \vee \neg qx)$ which is an intuitionistic tautology.

For the other direction, to prove px from $\neg\neg px$, we apply the assumption with $qy := px$ (i.e. the point restriction of p to x). Now, q is in C because it is a point restriction of p and p is in C, and the complement of q is in C because it is empty due to the assumption $\neg\neg px$. □

Note that the statement degenerates to the fact that double negation elimination is equivalent to the law of excluded middle if C is chosen to be the class containing all predicates.

Corollary 4.2. MP *is equivalent to PT for all three variants.*

Proof. We prove that the propositional variant of PT from the last theorem is individually equivalent to all three versions of PT. For the backward directions, it suffices to observe that all definitions of $\mathcal{D}(p)$ imply $\forall x.\ px \vee \neg px$.

For the forwards direction for the $\mathbb{P}$ case, there is nothing to prove. For the forwards direction of the $\mathbb{B}$ case, we need that $\Sigma^0_1(p) \to \Sigma^0_1(q) \to \forall x.\ px \vee qx \to (px) + (qx)$, see e.g. [10,

Lem. 4.58]. The forwards direction of the PR case is then a simple computability proof for the latter argument. □

The equivalence for $\mathsf{MP}_{\mathbb{B}}$ is formalised in Coq by Forster, Kirst, and Smolka [11] the equivalence for $\mathsf{MP}_{\mathsf{PR}}$ by Forster and Smolka [14].

4.2 Termination of Computation

The statement of MP in CRM can be given as "a computation halts if it does not run forever".

We introduce here three versions of computability of relations $R : \mathbb{N} \to \mathbb{N} \to \mathbb{P}$: R is $\mathbb{B}$-computable, if there is a partial function $f : \mathbb{N} \rightharpoonup \mathbb{N}$ (see e.g. [10, § 4.5]) such that $\forall xy.\ Rxy \leftrightarrow fx \downarrow y$. R is TM-computable, if this function is furthermore computable in a model of computation. Finally, we define that R is $\mathbb{P}$-computable if its graph $\lambda(x, y).\ Rxy$ is $\mathbb{P}$-Σ^0_1, because we can then prove:

Theorem 4.3. *R is computable if and only if its graph is Σ^0_1.*

Proof. For $\mathbb{P}$, the statement is trivial. For $\mathbb{B}$, it requires using step-indexed evaluation to n such that $fx \downarrow y$ in n steps. For TM, it requires proving that this unbounded search is computable. □

With Thm. 3.1 we immediately have:

Corollary 4.4. *The variants of* MP *are equivalent to the statement that for any computable R, $\forall x.\ \neg\neg(\exists y.\ Rxy) \to \exists y.\ Rxy$.*

Note that in particular, the statement "a Turing machine halts if it does not run forever" is equivalent to $\mathsf{MP}_{\mathsf{PR}}$.

4.3 Extended Natural Numbers

One can model the extension of $\mathbb{N}$ with a point of infinity as monotonous infinite sequences of truth values b_i (if b_i then b_j holds for $j \geq i$). This type is also called the one-point compactification of the natural numbers. MP is equivalent to "an extended natural number which is not infinite is finite". As far as we were able to find out, this result seems to be folklore and is recorded in the nLab [37].

We define an extended natural number principle in C:

$$D\text{-ENNP} := \forall p : \mathbb{N} \to \mathbb{P}.\ D(p) \to (\forall i.\ pi \to \forall j \geq i.\ pj) \to \neg\neg(\exists n.\ pn) \to \exists n.\ pn$$

The mentioned equivalence proof can again be given generically:

Theorem 4.5. *D-Σ^0_1-DNE $\to$ D-ENNP, and the converse holds if D fulfills $D(p) \to D(\lambda n.\ \exists m \leq n.\ pm)$.*

It is a simple corollary then that the statement that an extended natural number which is not infinite is finite is equivalent to $\mathsf{MP}_{\mathbb{P}}$ if sequences are defined as predicates $\mathbb{N} \to \mathbb{P}$, and to $\mathsf{MP}_{\mathbb{B}}$ if defined as functions $\mathbb{N} \to \mathbb{B}$. Defining sequences as *computable* functions $\mathbb{N} \to \mathbb{B}$ is unusual, but is equivalent to $\mathsf{MP}_{\mathsf{PR}}$.

4.4 Binary Trees

Berger, Ishihara, and Schuster [2] prove that MP is equivalent to a principle regarding decidable binary trees, stating that "if a tree is not infinite, then it is finite". We introduce a tree as a predicate on lists of Booleans, which contains the empty list and is closed under taking prefixes. A tree is finite if it has a maximum depth, and infinite if for any depth n there is a path l with length n.

We introduce a tree finiteness principle as follows:

$$D\text{-TFP} := \forall \tau : \mathtt{List}(\mathbb{B}) \to \mathbb{P}.\ D(\tau) \to \tau[] \to (\forall l_1 l_2.\ \tau(l_1 +\!\!+ l_2) \to \tau l_1) \to \neg(\forall n.\exists l.\ |l| = n \wedge \tau l) \to \exists n.\forall l.\ |l| \geq n \to \neg\tau l$$

Theorem 4.6. *D-Σ^0_1-DNE $\leftrightarrow$ D-TFP given that $D(\tau) \to D(\lambda n : \mathbb{N}.\ \forall l : \mathtt{List}(\mathbb{B}).\ |l| = n \to \neg\tau l)$ (for the forwards direction), $D(A) \to D(\lambda l : \mathtt{List}(\mathbb{B}).\ \forall n.\ n < |l| \to \neg An)$ (for the backwards direction), and $D(p) \to \forall x.\ px \vee \neg px$.*

Proof. For the forward direction, assume D-Σ^0_1-DNE, and let an infinite tree τ in D be given. First observe that it suffices to prove $\exists n.\ pn$ for $pn := \forall l.\ |l| = n \to \neg\tau l$. We can then apply D-Σ^0_1-DNE and the assumption that $D(p)$. We have to deduce a contradiction by assuming $\neg\exists n.\forall l.\ |l| = n \to \neg\tau l$, which we do by proving that the tree is not in fact infinite $\forall n.\exists l.\ |l| = n \wedge \tau l$. The relevant observation is that $\exists l.\ |l| = n \wedge \tau l$ is logically decidable as long as $D(\tau)$ ensures that τ is, the rest is tedious but straightforward.

For the backwards direction fix A in D and define the tree $\tau l := \forall n < |l|.\ \neg An$. By assumption, $D(\tau)$. It now suffices to prove that τ is bounded if and only if $\exists n.\ An$, which is again relying on the assumption that if $D(A)$, then A is logically decidable. □

The conditions on D hold for all variants of decidability we introduced, because the first two concern bounded quantification over decidable predicates, and all variants imply logical decidability.

Corollary 4.7. *$\mathsf{MP}_{\mathbb{P}}$ is equivalent to $\mathbb{P}$-$\mathcal{D}$-TFP, $\mathsf{MP}_{\mathbb{B}}$ to $\mathbb{B}$-$\mathcal{D}$-TFP, and $\mathsf{MP}_{\mathsf{PR}}$ to TM-$\mathcal{D}$-TFP.*

4.5 Completeness Theorems in Logic

Completeness theorems are a central topic both in constructive and classical reverse mathematics.

Regarding the constructive line, it was already known to Gödel that completeness of natural deduction w.r.t. Tarski semantics over the $(\forall, \to, \bot)$-fragment of classical first-order logic implies $\mathsf{MP}_{\mathsf{PR}}$ [27]. Regarding the classical line, it is a folklore theorem that completeness theorems already for propositional logic require comprehension principles such as Weak König's Lemma [51].

A recent line of work by Herbelin, Ilik, Kirst, Forster, and Wehr [13, 18, 19] has exposed that the reverse mathematical analysis can be split into two phases: First, different forms of Markov's principle are equivalent to forms of stability of provability in proof systems. Second, proving quasi-completeness, i.e. that certain forms of validity imply the double negation of provability require more logical principles, depending on the connectives of the logic.

We work with respect to a general class C of theories $\mathcal{T} : \mathcal{F} \to \mathbb{P}$. If $C(\mathcal{T})$ we call $\mathcal{T}$ a C-theory. We treat any list $A : \mathtt{List}(\mathcal{F})$ as a theory implicitly.

We explain here the first connection as abstractly as possible, using a type of formulas $\mathcal{F} : \mathbb{T}$, a constant $\bot : \mathcal{F}$, and a proof system $\cdot \vdash \cdot : (\mathcal{F} \to \mathbb{P}) \to \mathcal{F} \to \mathbb{P}$ which supports the assumption rule $\mathcal{T}\varphi \to \mathcal{T} \vdash \varphi$, is consistent, i.e. $[] \nvdash \bot$, and compact, i.e. $\mathcal{T} \vdash \varphi \to \exists A.\ (\forall \psi \in A.\ \mathcal{T}\psi) \wedge A \vdash \varphi$. The proof essentially generalizes [50, Lem. 4.5]. Note that $\bot$ just has to be an underivable formula – we do not require any exfalso quod libet property.

Theorem 4.8. *Stability of C-theory provability implies C*-DNE, *and the converse holds for any C s.t.* $C(\mathcal{T}) \to C(\mathcal{T} \vdash \varphi)$.

Proof. The backwards direction is straightforward. We only prove the forwards direction. Assume that for any $\mathcal{T}$ with $C(\mathcal{T})$ provability is stable, and let a proposition P with $C(P)$ and $\neg\neg P$ be given. Setting $\mathcal{T}\varphi := P$ we first prove as an intermediate result that $\neg\neg(\mathcal{T} \vdash \bot)$: Since the goal is negative, we can turn the assumption of $\neg\neg P$ into P. It suffices to prove $\mathcal{T} \vdash \bot$, which follows since $\bot \vdash \bot$ by the assumption rule and $\mathcal{T}\bot$ because P holds. Now because $C(P)$ holds we have $C(\mathcal{T})$ and thus $\mathcal{T} \vdash \bot$ by assumption of stability. Using compactness, this means that $A \vdash \bot$ for some A with $\forall \psi \in A.\ \mathcal{T}\psi$. If A is empty, we have a contradiction due to $[] \vdash \bot$ and consistency. If A is not empty, we have a formula in $\mathcal{T}$ and thus P holds as needed. □

The equivalences to $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$ for stability of classical first-order provability are already proved in Coq by Forster, Kirst, and Wehr [12]. Our proof can immediately be instantiated to prove the equivalence of $\mathsf{MP}_{\mathbb{P}}$ to the stability of classical first-order provability for $\mathbb{P}$-Σ^0_1 theories. The theorem also applies to other systems:

Corollary 4.9. *Completeness for* Σ^0_1*-theories for classical first-order logic and classical propositional logic w.r.t. Tarski models and quasi-completeness for intuitionistic first-order logic and intuitionistic propositional logic w.r.t. Kripke models are respectively equivalent to the corresponding variant of* MP.

Proof. Since quasi-completeness for all of these systems is constructively provable, completeness is respectively equivalent to stability of provability. It is tedious but straightforward to prove that provability in Σ^0_1 theories is itself Σ^0_1. □

5 $\mathrm{TT}^{\Box}_{C}$ PRIMER

To separate the three variants of MP mentioned above for MLTT, we make use of an intermediate theory called $\mathrm{TT}^{\Box}_{C}$, defining models of MLTT as the composition of a translation of MLTT to $\mathrm{TT}^{\Box}_{C}$ and models of $\mathrm{TT}^{\Box}_{C}$. The $\mathrm{TT}^{\Box}_{C}$ family of type theories was put forward in [5] as a general framework for exploring the validity status of key principles such as the Law of Excluded Middle in various instantiated type theories. Concretely, $\mathrm{TT}^{\Box}_{C}$, which is formalised in Agda, captures type theories (1) parameterised by a choice operator C of stateful computations that can evolve non-deterministically over time, captured by a poset $\mathcal{W}$ of worlds, and that can change the state of the world; and (2) modeled through an abstract modality $\Box$. In this work, we focus on specific instantiations of $\mathrm{TT}^{\Box}_{C}$ (i.e., $\Box$ and C) for satisfying or falsifying the various variants of MP. The results presented below have also been formalised in Agda: https://github.com/vrahli/opentt/blob/lics24/lics24.lagda. From now on we will use the symbol ⚙ to link to the corresponding definition or result in the formalisation. Some of the results presented below are sometimes simplified compared to [5], as well as compared to subsequent presentations [4, 6] and to the formalisation, in part due to our focus on specific instantiations of $\mathrm{TT}^{\Box}_{C}$.

5.1 Syntax

Fig. 1 presents a subset of $\mathrm{TT}^{\Box}_{C}$'s terms $t \in \mathrm{Term}$ and contexts $\Gamma \in \mathrm{Ctxt}$, which is relevant for the purpose of this paper (see [5] for more details), where x belongs to a set of variables Var, n is a metatheoretical number, v is a value, and δ belongs to a set of choice names $\mathcal{N}$ discussed in Sec. 5.3. $\mathrm{TT}^{\Box}_{C}$'s syntax is similar to that of MLTT [40], which is further discussed in Sec. 6, and whose syntax is also presented in Fig. 1, along with an interpretation of MLTT's syntax in $\mathrm{TT}^{\Box}_{C}$. Fig. 1 also presents some $\mathrm{TT}^{\Box}_{C}$ notation that will be useful in the rest of this paper. Because MLTT and $\mathrm{TT}^{\Box}_{C}$ share similar types, we make use of colors to distinguish the two: blue for MLTT expressions and red for $\mathrm{TT}^{\Box}_{C}$ expressions.

Types are syntactic forms that are given semantics in Sec. 5.6 via a forcing interpretation. The type system includes the type of natural numbers $\mathbb{N}$, dependent products $\mathbf{\Pi} x{:}A.B$, dependent sums $\mathbf{\Sigma} x{:}A.B$, disjoint union $A{+}B$, and a hierarchy of universes $\mathbb{U}_i$ indexed by universe levels, which for the purpose of this paper are numbers. It also includes equality types $a = b \in A$ expressing that a and b are equal members of the type A, and a truncation operator $\downarrow A$ that "erases" A's computational content. The type $\mathtt{E}(A)$ carves out an effect-free subtype of A, and is further discussed in Sec. 5.5.

Compared to non-effectful type theories such as MLTT, discussed further in Sec. 6, $\mathrm{TT}^{\Box}_{C}$ additionally includes the following computations (see Sec. 5.4 for further details): in the context of this paper, δ is a choice name that can be thought of as a pointer to a reference cell holding a list of values (Booleans in Sec. 7 and propositions in Sec. 8); $!n(k)$ is used to access the k^{th} choice pointed to by n; $\mathsf{fix}(t)$ is a general fixpoint operator; $\mathtt{let}\ x = t\ \mathtt{in}\ u$ allows evaluating arguments in an otherwise call-by-name semantics.

5.2 Worlds

To capture the *time* notion underlying choice operators, $\mathrm{TT}^{\Box}_{C}$ is parameterized by a Kripke frame [31, 32] defined as follows:

Definition 5.1 (⚙ Kripke frame). A Kripke frame is defined as a set of *worlds* $\mathcal{W}$ equipped with a reflexive and transitive binary relation $\sqsubseteq$.

Let w range over $\mathcal{W}$. We sometimes write $w' \sqsupseteq w$ for $w \sqsubseteq w'$, which is read w' *extends* w. Let $\mathcal{P}_w$ be the collection of predicates on world extensions, i.e., functions in $\forall w' \sqsupseteq w.\mathbb{P}$. Note that due to $\sqsubseteq$'s transitivity, if $P \in \mathcal{P}_w$ then for every $w' \sqsupseteq w$, it naturally restricts to a predicate in $\mathcal{P}_{w'}$. We make use of the following notation, where $P \in \mathcal{P}_w$: $\forall^{\sqsubseteq}_w(P) := \forall w' \sqsupseteq w.P(w')$ and $\exists^{\sqsubseteq}_w(P) := \exists w' \sqsupseteq w.P(w')$. For readability, we sometime write $\forall^{\sqsubseteq}_w(w'.P)$ instead of $\forall^{\sqsubseteq}_w(\lambda w'.P)$ and $\exists^{\sqsubseteq}_w(w'.P)$ instead of $\exists^{\sqsubseteq}_w(\lambda w'.P)$.

5.3 Choices

$\mathrm{TT}^{\Box}_{C}$ includes effectful computations that rely on worlds to record choices and provides operators to manipulate the choices stored in a world, which we now recall. Choices are referred to through their names. A concrete example of such choices are reference cells in programming languages, where a variable name pointing to a reference cell is the name of the corresponding reference cell. Another example, which we rely on in this paper, is the notion of choice sequences [22, 30, 35, 55, 56, 60, 62], which stem from Brouwer's intuitionistic logic, and can be seen (and implemented) as reference cells storing lists of values, e.g., numbers or Booleans. Therefore, $\mathrm{TT}^{\Box}_{C}$'s computation system is parameterized by a set $\mathcal{N}$ of *choice names*, equipped with a decidable equality, and an operator

$$
\begin{array}{rcl}
t & ::= & x \mid \lambda x : t.t \mid t\ t \mid \mathsf{0} \mid \mathsf{S}\ t \mid \mathbb{N}_{\mathsf{ind}}\ t\ t\ t\ t \mid \Pi x{:}t.t \mid \mathbb{N} \mid \mathbb{U} \\
 & \mid & \langle t, t\rangle \mid \mathtt{fst}(t) \mid \mathtt{snd}(t) \mid \Sigma x{:}t.t \mid \star \mid \mathbb{1} \mid \mathbb{0}_{\mathsf{ind}}\ t\ t \mid \mathbb{0} \\
\Gamma & ::= & \bullet \mid \Gamma, x : t
\end{array}
$$

$$
\begin{array}{rcl}
[\![x]\!] & = & x \\
[\![\mathsf{0}]\!] & = & \underline{0} \\
[\![\mathbb{U}]\!] & = & \mathbb{U}_1 \\
[\![\mathbb{N}]\!] & = & \mathbb{N} \\
[\![\star]\!] & = & \star \\
[\![\mathbb{1}]\!] & = & \mathbb{1} \\
[\![\mathbb{0}]\!] & = & \mathbb{0} \\
[\![\mathsf{S}\ t]\!] & = & \mathsf{S}\ [\![t]\!]
\end{array}
\qquad
\begin{array}{rcl}
[\![\lambda x : T.t]\!] & = & \lambda x.[\![t]\!] \\
[\![f\ a]\!] & = & [\![f]\!]\ [\![a]\!] \\
[\![\Pi x{:}A.B]\!] & = & \Pi x{:}[\![A]\!].[\![B]\!] \\
[\![\Sigma x{:}A.B]\!] & = & \Sigma x{:}[\![A]\!].[\![B]\!] \\
[\![\langle t, u\rangle]\!] & = & \langle [\![t]\!], [\![u]\!]\rangle \\
[\![\mathtt{fst}(t)]\!] & = & \mathtt{fst}([\![t]\!]) \\
[\![\mathtt{snd}(t)]\!] & = & \mathtt{snd}([\![t]\!]) \\
[\![\mathbb{0}_{\mathsf{ind}}\ A\ ea]\!] & = & [\![e]\!] \\
[\![\mathbb{N}_{\mathsf{ind}}\ P\ z\ s\ n]\!] & = & \mathbb{N}_{\mathsf{ind}}\ [\![z]\!]\ [\![s]\!]\ [\![n]\!] \\
[\![\Gamma, x : A]\!] & = & [\![\Gamma]\!], x : [\![A]\!]
\end{array}
\qquad
[\![\bullet]\!] = \bullet
$$

$$
\begin{array}{rcl}
v & ::= & \lambda x.t \mid \underline{n} \mid \langle t, t\rangle \mid \star \mid \mathtt{inl}(t) \mid \mathtt{inr}(t) \\
 & \mid & \Pi x{:}t.t \mid \mathbb{N} \mid \mathbb{U}_i \mid \Sigma x{:}t.t \mid t = t \in t \mid t{+}t \mid \downarrow t \\
 & \mid & \delta \mid \mathsf{E}(t) \\
t & ::= & v \mid x \mid t\ t \mid \mathsf{S}\ t \mid \mathbb{N}_{\mathsf{ind}}\ t\ t\ t \mid \mathtt{let}\ \langle x, y\rangle = t\ \mathtt{in}\ t \\
 & \mid & \mathtt{case}\ t\ \mathtt{of}\ \mathtt{inl}(x) \Rightarrow t \mid \mathtt{inr}(y) \Rightarrow t \\
 & \mid & \mathtt{let}\ x = t\ \mathtt{in}\ t \mid !t(t) \mid \mathtt{fix}(t) \\
\Gamma & ::= & \bullet \mid \Gamma, x : t
\end{array}
$$

$$
\begin{array}{rcll}
\mathbb{0} & := & \underline{0} = \underline{1} \in \mathbb{N} & !\delta := \lambda n.!\delta(n) \\
\mathbb{1} & := & \underline{0} = \underline{0} \in \mathbb{N} & t_1 \to t_2 := \Pi x{:}t_1.t_2 \\
2 & := & \mathbb{1}{+}\mathbb{1} & \neg T := T \to \mathbb{0} \\
\mathbb{t} & := & \mathtt{inl}(\star) & \mathtt{fst}(t) := \mathtt{let}\ \langle x, y\rangle = t\ \mathtt{in}\ x \\
\mathbb{f} & := & \mathtt{inr}(\star) & \mathtt{snd}(t) := \mathtt{let}\ \langle x, y\rangle = t\ \mathtt{in}\ y
\end{array}
$$

$$\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 := \mathtt{case}\ t_1\ \mathtt{of}\ \mathtt{inl}(\_) \Rightarrow t_2 \mid \mathtt{inr}(\_) \Rightarrow t_3$$

Figure 1: MLTT syntax (top/left), $\mathrm{TT}^{\Box}_{C}$ syntax (top/right), MLTT interpretation in $\mathrm{TT}^{\Box}_{C}$ (bottom/left), $\mathrm{TT}^{\Box}_{C}$ notation (bottom/right)

that given a list of names, returns a name not in the list. This can be given by, e.g., nominal sets [47]. In what follows we let δ range over $\mathcal{N}$, and take $\mathcal{N}$ to be $\mathbb{N}$ for simplicity. Choices are defined abstractly as follows:

Definition 5.2 (⚙ Choices). A choice operator is a set $C \subseteq \mathrm{Term}$ of *choices*, ranged over by c, that contains two syntactically different values κ and σ (κ is sometimes referred to as the "default" choice), equipped with a partial function $\mathrm{read} \in \mathcal{W} \to \mathcal{N} \to \mathbb{N} \to C$, and total functions $\nu C \in \mathcal{W} \to \mathcal{N}$ and $\mathrm{start}\nu C \in \mathcal{W} \to \mathcal{W}$. Given $w \in \mathcal{W}$, $\delta \in \mathcal{N}$, and $n \in \mathbb{N}$: the choice $\mathrm{read}(w, \delta, n)$, if it exists, is meant to be the n^{th}-choice made for δ w.r.t. w; $\nu C(w)$ is meant to return a new choice name not present in w; and $\mathrm{start}\nu C(w)$ is meant to return an extension of w with the new name $\nu C(w)$. Those functions are required to satisfy the following properties:

- E1 $\forall(w : \mathcal{W}).w \sqsubseteq \mathrm{start}\nu C(w)$
- E2 Initially, the only possible choice is the default choice, i.e.: $\forall(w : \mathcal{W}).\mathrm{def}C(\mathrm{start}\nu C(w), \nu C(w))$, where $\mathrm{def}C(w, \delta) := \forall(n : \mathbb{N})(c : C).\mathrm{read}(w, \delta, n) = c \to c = \kappa$

We require $C \subseteq \mathrm{Term}$, so that a choice read using read can be used in a $\mathrm{TT}^{\Box}_{C}$ computation (see Sec. 5.4).

In particular, choice sequences of Booleans, which are ever increasing finite lists of Booleans, form a choice operator:

Example 5.3 (⚙ Choice Sequences). We define choice sequences of Booleans as follows. Let $\mathcal{N} := \mathbb{N}$ and $C := \{\sigma, \kappa\}$, where $\sigma := \mathbb{t}$ and $\kappa := \mathbb{f}$ (this example is in particular used in Sec. 7, where while the default choice is $\mathbb{f}$, it is always possible to make the syntactically different choice $\mathbb{t}$). Worlds are lists of entries: $\mathcal{W} := \mathtt{List}(\mathcal{N} \times \mathtt{List}(C))$. $\sqsubseteq$ is the reflexive transitive closure of two operations that allow: (i) creating a new choice sequence by extending a world w with a name δ not occurring in w as follows: $(\delta, [\,]) :: w$; and (ii) updating an existing choice sequence by appending a choice to the corresponding list, i.e., if (δ, l) occurs in w, a new δ-choice c is made by replacing (δ, l) with $(\delta, l ::^{r} c)$ in w. We define $\mathrm{read}(w, \delta, n)$ so that it returns l's n^{th} element if (δ, l) occurs in w, and is undefined otherwise. $\nu C(w)$ returns a choice sequence name not occurring in w; and $\mathrm{start}\nu C(w)$ uses the first operation described above to add a new choice sequence entry to w with name $\nu C(w)$. Property E2 in then trivially satisfied because new choice sequence entries are always of the form $(\delta, [\,])$, and so trivially for any c in $[\,]$, c is κ.

As we will see below it is important in Ex. 5.3 that worlds only store a finite number of values for each choice sequence (the lists of choices made so far), which is a key aspect of choice sequences, because in Sec. 7.1 we will need the ability to make the choice $\mathbb{t}$ in any world (using Def. 7.1), and in addition the ability to keep on making the default choice $\mathbb{f}$ (using Def. 7.2). Note that those choices are not made using $\mathrm{TT}^{\Box}_{C}$ computations, but in the metatheory, through the $\sqsubseteq$ relation by extending worlds.

5.4 Operational Semantics

$\mathrm{TT}^{\Box}_{C}$'s call-by-name operational semantics (⚙) is defined using a relation $w \vdash t \mapsto t'$, which captures that t reduces to t' in one step of computation w.r.t. the world w. Its main cases are as follows, where $t[x\backslash u]$ stands for the capture-avoiding substitution of all the free occurrences of x in t by u, and where v is a value:

$$
\begin{array}{ll}
w \vdash \boxed{(\lambda x.t)}\ u \mapsto t[x\backslash u] & w \vdash \mathtt{let}\ x = \boxed{v}\ \mathtt{in}\ t \mapsto t[x\backslash v] \\
w \vdash \mathsf{S}\ \boxed{\underline{n}} \mapsto \underline{1+n} & w \vdash !\boxed{\delta}(\boxed{\underline{n}}) \mapsto \mathrm{read}(w, \delta, n) \\
w \vdash \mathbb{N}_{\mathsf{ind}}\ t_1\ t_2\ \boxed{\underline{0}} \mapsto t_1 & w \vdash \mathbb{N}_{\mathsf{ind}}\ t_1\ t_2\ \boxed{\underline{1+n}} \mapsto t_2\ \underline{n}\ (\mathbb{N}_{\mathsf{ind}}\ t_1\ t_2\ \underline{n}) \\
w \vdash \mathtt{let}\ \langle x, y\rangle = \boxed{\langle t_1, t_2\rangle}\ \mathtt{in}\ t \mapsto t[x\backslash t_1; y\backslash t_2] & \\
w \vdash \mathtt{case}\ \boxed{\mathtt{inl}(t)}\ \mathtt{of}\ \mathtt{inl}(x) \Rightarrow t_1 \mid \mathtt{inr}(y) \Rightarrow t_2 \mapsto t_1[x\backslash t] & \\
w \vdash \mathtt{case}\ \boxed{\mathtt{inr}(t)}\ \mathtt{of}\ \mathtt{inl}(x) \Rightarrow t_1 \mid \mathtt{inr}(y) \Rightarrow t_2 \mapsto t_2[y\backslash t] &
\end{array}
$$

Therefore, through the rule for !, a choice name δ can be used in a computation to access (or "read") choices from a world. This allows getting the n^{th} δ-choice from the current world w.

We indicate with boxes the arguments that have to be reduced before these axioms can be used. We omit the congruence rules that allow computing within terms such as: if $w \vdash t_1 \mapsto t_2$ then $w \vdash t_1(u) \mapsto t_2(u)$. We denote by $\_ \vdash \_ \mapsto^{*} \_$ the reflexive transitive closure of the $\_ \vdash \_ \mapsto \_$ relation, i.e., $w \vdash a \mapsto^{*} b$ states that a computes to b in 0 or more steps, w.r.t. the world w. We further define $w \vdash a \Mapsto b := \forall^{\sqsubseteq}_{w}(w'.w' \vdash a \mapsto^{*} b)$, which captures that a computes to b in all extensions of w.

5.5 Different Levels of Effects

As made precise in Sec. 5.6, $\mathrm{TT}^{\square}_{C}$'s type system allows capturing different levels of effects that we now explain. However, we only present those relevant to the present discussion, introducing a subset of $\mathrm{TT}^{\square}_{C}$ where types are either types of effect-free computations, or are types of expressions that can potentially read choices using the ! operator. Furthermore, as we focus on choice sequences in this paper, while more choices can be made over time, once a choice has been made, it cannot be modified. Therefore, if $!\delta(\underline{k})$ reduces to a number in some world w_1 then it reduces to that same number in any $w_2 \sqsupseteq w_1$, and it follows that if $w \vdash a \mapsto^* b$ then $w \vdash a \Longmapsto b$ holds too. Hence, we enforce that all types are closed under $w \vdash a \Longmapsto b$ in Sec. 5.6.

In addition to the above effects, and in order to implement more general forms of references, $\mathrm{TT}^{\square}_{C}$'s computation system further allows adding and modifying choices as explained in [4], which we omit from the presentation. Furthermore, while types such as $\mathbb{N}$ are defined using $w \vdash a \Longmapsto b$ in Fig. 2, when considering $\mathrm{TT}^{\square}_{C}$ in its full generality, these types are defined using $w \vdash a \mapsto^* b$ instead so as to allow for a wide range of effects by default. Type modalities can then be used to restrict the level of reading (through a $\mathtt{R}$ modality, that enforces that $w \vdash a \Longmapsto b$ whenever $w \vdash a \mapsto^* b$) and writing (through a $\mathtt{W}$ modality) expressions can perform. For example, the $\mathbb{N}$ type contains expressions of the form $\underline{k}$ as well as effectful computations of the form $!\delta(\underline{k})$ when choices are numbers, even when $!\delta(\underline{k})$ returns different numbers in w_1 and $w_2 \sqsupseteq w_1$.

It then becomes critical for example to prove that $w \vDash !\delta \in \mathbb{N} \to 2$ when choices are Booleans (see Thm. 5.8, which is key to prove the separation result presented in Sec. 7), that the $\mathtt{R}$ and $\mathtt{W}$ modalities are applied to $\mathbb{N}$ to restrict the effects indices can have. Consequently, in the $\mathrm{TT}^{\square}_{C}$ interpretation of MLTT discussed in Sec. 6, in order to uniformly interpret MLTT datatypes, the $\mathtt{R}$ and $\mathtt{W}$ modalities need to be applied to all corresponding $\mathrm{TT}^{\square}_{C}$ datatypes. A type of the form $\mathtt{R}(\mathtt{W}(A))$ is then a type of potentially effectful expressions, that are however not observable, in particular in the sense that choices do not change once they have been made. Those modalities are also critical to validate elimination rules, e.g., for numbers, because, as explained in [45], we cannot have both observational effects and dependent elimination, while we can validate an elimination rule for $\mathtt{R}(\mathtt{W}(\mathbb{N}))$ since the effects are not observable in the above sense.

However, as we restrict the presentation of $\mathrm{TT}^{\square}_{C}$ in this paper to a subset of its effects, those modalities become moot and can therefore be omitted altogether. The only modality we require is $\mathtt{E}$. The type $\mathtt{E}(A)$ is a subtype of A, which is inhabited by A's elements that compute to effect-free expressions. Its semantics in Sec. 5.6 is defined in terms of the $\mathsf{effect\text{-}free}(t)$ predicate that expresses that no expressions of the form $!t(u)$ occur in t.

5.6 Forcing Interpretation

$\mathrm{TT}^{\square}_{C}$'s types are interpreted via the forcing interpretation presented in Fig. 2 in which forcing conditions are worlds (we omit universes for readability). This interpretation, adapted from [5, 6], and reminiscent of sheaf models such as [15, 33, 42, 53, 61], is a logical relation defined using induction-recursion [9] as follows: (1) the inductive relation $w \vDash T_1 \equiv T_2$ expresses type equality at the world w, i.e., the two closed terms T_1 and T_2 are equal types at the world w; (2) the recursive function $w \vDash t_1 \equiv t_2 \in T$ expresses equality in a type, i.e., that the two closed terms t_1 and t_2 are equal expressions in the type T at the world w. We further use the following notation:

$$w \vDash t \in T := w \vDash t \equiv t \in T \qquad w \vDash T := \exists(t : \mathsf{Term}).w \vDash t \in T$$
$$\mathsf{Fam}_w(A_1, A_2, B_1, B_2) := w \vDash A_1 \equiv A_2 \wedge \forall^{\sqsubseteq}_w \left(\begin{array}{l} w'.\forall(a_1, a_2 : \mathsf{Term}).\ w' \vDash a_1 \equiv a_2 \in A_1 \\ \qquad\qquad \to w' \vDash B_1[x\backslash a_1] \equiv B_2[x\backslash a_2] \end{array} \right)$$

When $w \vDash T$, we say that T *is inhabited at world* w. This forcing interpretation is parametrized by a modality $\square$:

Definition 5.4 (⚙ Modality). A modality $\square$ is a family of predicates $\square_w \in \mathcal{P}_w \to \mathbb{P}$ over worlds w, which satisfy the following properties, where we write $\square_w(w'.P)$ for $\square_w \lambda w'.P$, and where the free variables $w \in \mathcal{W}$, $P, Q \in \mathcal{P}_w$, and $p \in \mathbb{P}$ are universally quantified:

- $\square_1$ (monotonicity): $\forall^{\sqsubseteq}_w(w'.\square_w P \to \square_{w'} P)$
- $\square_2$ (distribution): $\square_w(w'.P\ w' \to Q\ w') \to \square_w P \to \square_w Q$
- $\square_3$ (density): $\square_w(w'.\square_{w'} P) \to \square_w P$
- $\square_4$ (weakening): $\forall^{\sqsubseteq}_w(P) \to \square_w P$
- $\square_5$ (reflexivity): $\square_w(w'.p) \to p$

The above properties are in particular key in proving that the above interpretation yields a type system in the sense of Thm. 5.6.

Modalities are derived from covering relations [5, Prop. 23], where $w \triangleleft o$ captures that $o \in \mathcal{P}_w$ "covers" the world w, as follows:

$$\square_w P := \exists(o : \mathcal{P}_w).w \triangleleft o \wedge \forall^{\sqsubseteq}_w(w'.o(w') \to P(w'))$$

Coverings are simpler to define than modalities. However modalities are easier to reason about than coverings. Covering relations are required to satisfy suitable properties discussed in [5, Def. 22] to be able to derive modalities from them. We focus in this paper on Beth coverings, which allow capturing aspects of choice sequences (see Thm. 5.8), which we use to falsify some versions of MP. In particular, Beth coverings capture the fact that choices are not always immediately available in a world, but only eventually.

Example 5.5 (⚙ Beth Covering). The Beth covering is defined as follows, i.e., $w \triangleleft_B o$ whenever o contains at least one world from each "branch" (captured by chain below) starting from w:

$$w \triangleleft_B o := \forall c : \mathsf{chain}(w).\exists(n : \mathbb{N})(w' : \mathcal{W}).w' \sqsubseteq (c\ n) \wedge o(w')$$

where $\mathsf{chain}(w)$ is the set of sequences of worlds in $\mathbb{N} \to \mathcal{W}$ such that $c \in \mathsf{chain}(w)$ iff (1) $w \sqsubseteq c\ 0$, (2) for all $i \in \mathbb{N}$, $c\ i \sqsubseteq c\ (i+1)$; and (3) c is *progressing*, which is formally defined in [5, Def. 25], and informally captures that there exists two worlds $w_1 \sqsubseteq w_2$ along c and an n such that $\mathsf{read}(w_1, \delta, n)$ is undefined, while $\mathsf{read}(w_2, \delta, n)$ is defined, and this infinitely often for all choice names δ.

We finally show that the semantics presented in this section allows proving that $\mathrm{TT}^{\square}_{C}$ is a standard type system:

Theorem 5.6 (⚙). *Given a computation system with choices C and a modality $\square$, $\mathrm{TT}^{\square}_{C}$ is a standard type system in the sense that its forcing interpretation induced by $\square$ satisfies the following properties*

Numbers • $w \Vdash \mathbb{N} \equiv \mathbb{N} \iff \top$
• $w \Vdash t \equiv t' \in \mathbb{N} \iff \Box_w(w'.\exists(n:\mathbb{N}).w' \vdash t \mapsto \underline{n} \wedge w' \vdash t' \mapsto \underline{n})$

Equalities • $w \Vdash a_1 = b_1 \in A \equiv a_2 = b_2 \in B \iff$
$w \Vdash A \equiv B \wedge \forall^{\sqsubseteq}_w(w'.w' \Vdash a_1 \equiv a_2 \in A) \wedge \forall^{\sqsubseteq}_w(w'.w' \Vdash b_1 \equiv b_2 \in B)$
• $w \Vdash a_1 \equiv a_2 \in a = b \in A \iff \Box_w(w'.w' \Vdash a \equiv b \in A)$

Truncation • $w \Vdash \downarrow A \equiv \downarrow B \iff w \Vdash A \equiv B$
• $w \Vdash a_1 \equiv a_2 \in \downarrow A \iff \Box_w(w'.w' \Vdash A)$

Modality closure • $w \Vdash T_1 \equiv T_2 \iff$
$\Box_w(w'.\exists(T_1', T_2' : \mathsf{Term}).w' \vdash T_1 \mapsto T_1' \wedge w' \vdash T_2 \mapsto T_2' \wedge w' \vdash T_1' \equiv T_2')$
• $w \Vdash t_1 \equiv t_2 \in T \iff$
$\Box_w(w'.\exists(T' : \mathsf{Term}).w' \vdash T \mapsto T' \wedge w' \vdash t_1 \equiv t_2 \in T')$

Products • $w \Vdash \Pi x{:}A_1.B_1 \equiv \Pi x{:}A_2.B_2 \iff \mathsf{Fam}_w(A_1, A_2, B_1, B_2)$
• $w \Vdash f \equiv g \in \Pi x{:}A.B \iff \Box_w(w'.\forall(a_1, a_2 : \mathsf{Term}).w' \vdash a_1 \equiv a_2 \in A \to w' \vdash f(a_1) \equiv g(a_2) \in B[x \backslash a_1])$

Sums • $w \Vdash \Sigma x{:}A_1.B_1 \equiv \Sigma x{:}A_2.B_2 \iff \mathsf{Fam}_w(A_1, A_2, B_1, B_2)$
• $w \Vdash p_1 \equiv p_2 \in \Sigma x{:}A.B \iff \Box_w(w'.\exists(a_1, a_2, b_1, b_2 : \mathsf{Term}).w' \vdash a_1 \equiv a_2 \in A \wedge w' \vdash b_1 \equiv b_2 \in B[x \backslash a_1] \wedge w' \vdash p_1 \mapsto \langle a_1, b_1 \rangle \wedge w' \vdash p_2 \mapsto \langle a_2, b_2 \rangle)$

Disjoint unions • $w \Vdash A_1 + B_1 \equiv A_2 + B_2 \iff w \Vdash A_1 \equiv A_2 \wedge w \Vdash B_1 \equiv B_2$
• $w \Vdash a_1 \equiv a_2 \in A + B \iff \Box_w(w'.\exists(u, v : \mathsf{Term}).(w' \vdash a_1 \mapsto \mathtt{inl}(u) \wedge w' \vdash a_2 \mapsto \mathtt{inl}(v) \wedge w' \vdash u \equiv v \in A) \vee (w' \vdash a_1 \mapsto \mathtt{inr}(u) \wedge w' \vdash a_2 \mapsto \mathtt{inr}(v) \wedge w' \vdash u \equiv v \in B))$

Effect-free • $w \Vdash \mathsf{E}(A) \equiv \mathsf{E}(B) \iff w \Vdash A \equiv B$
• $w \Vdash a \equiv b \in \mathsf{E}(A) \iff \Box_w(w'.w \Vdash a \equiv b \in A \wedge (\exists(c : \mathsf{Term}).w' \vdash a \mapsto c \wedge \text{effect-free}(c)) \wedge (\exists(d : \mathsf{Term}).w' \vdash b \mapsto d \wedge \text{effect-free}(d)))$

Figure 2: Forcing Interpretation

(where free variables are universally quantified):

TS_t	*(transitivity):*	$w \Vdash T_1 \equiv T_2 \to w \Vdash T_2 \equiv T_3 \to w \Vdash T_1 \equiv T_3$
		$w \Vdash t_1 \equiv t_2 \in T \to w \Vdash t_2 \equiv t_3 \in T \to w \Vdash t_1 \equiv t_3 \in T$
TS_s	*(symmetry):*	$w \Vdash T_1 \equiv T_2 \to w \Vdash T_2 \equiv T_1$
		$w \Vdash t_1 \equiv t_2 \in T \to w \Vdash t_2 \equiv t_1 \in T$
TS_r	*(computation):*	$w \Vdash T_1 \mapsto T_2 \to (w \Vdash T \equiv T_1 \leftrightarrow w \Vdash T \equiv T_2)$
		$w \Vdash t_1 \mapsto t_2 \to (w \Vdash t \equiv t_1 \in T \leftrightarrow w \Vdash t \equiv t_2 \in T)$
TS_m	*(monotonicity):*	$w \Vdash T_1 \equiv T_2 \to \forall^{\sqsubseteq}_w(w'.w' \Vdash T_1 \equiv T_2)$
		$w \Vdash t_1 \equiv t_2 \in T \to \forall^{\sqsubseteq}_w(w'.w' \Vdash t_1 \equiv t_2 \in T)$
TS_l	*(locality):*	$\Box_w(w'.w' \Vdash T_1 \equiv T_2) \to w \Vdash T_1 \equiv T_2$
		$\Box_w(w'.w' \Vdash t_1 \equiv t_2 \in T) \to w \Vdash t_1 \equiv t_2 \in T$
TS_c	*(consistency):*	$\neg w \Vdash \mathbb{0}$

One key property used to falsify $\mathsf{MP}_{\mathbb{B}}$ in Sec. 7.1, that we wish a choice name δ to satisfy, is that $!\delta \in \mathbb{N} \to \mathbb{2}$ when δ is the name of a sequence of Boolean choices, and this in any world w. This is an important property of choices sequences (of Booleans): they are functions that eventually return a Boolean for any input n, once the n^{th} choice has been made. To prove this we further require that $\Box$ satisfies the following retrieving property.

Definition 5.7 (⚙ Retrieving). The modality $\Box$ is called *retrieving* if the following property holds:

$$\forall w\ \delta\ n.\Box_w(w'.\exists(c : C).\forall^{\sqsubseteq}_{w'}(w''.\mathsf{read}(w'', \delta, n) = c))$$

In particular, the modality derived from the Beth covering (Ex. 5.5) satisfies this property (⚙). Let us sketch the proof. Given a world w and a chain $c \in \mathsf{chain}(w)$, since c is progressing, there must be a $w' \sqsupseteq w$ where $\mathsf{read}(w', \delta, n)$ is defined, and since, as explained in Ex. 5.3, existing choices cannot be modified, it must be that $\mathsf{read}(w'', \delta, n)$ for all $w'' \sqsupseteq w'$.

Using this property we now prove that choice names of Boolean choice sequences inhabit the type $\mathbb{N} \to \mathbb{2}$ in any world w, even though w does not have all the corresponding choices, and even though the types $\mathbb{N}$ and $\mathbb{2}$ rule out some effects as explained in Sec. 5.5. Intuitively, $!\delta \in \mathbb{N} \to \mathbb{2}$, even in worlds where there are no δ-choices, such as the empty world [] defined in Ex. 5.3, because thanks to the locality property TS_l, it is enough to prove that $\Box_w(w'.w' \Vdash !\delta(\underline{k}) \in \mathbb{2})$ for any $w \in \mathcal{W}$ and $k \in \mathbb{N}$. Now, if $\Box$ is obtained for example from a Beth covering as defined in Ex. 5.5, it is enough to exhibit a covering $o \in \mathcal{P}_w$ of w, such that $!\delta(\underline{k})$ computes to a Boolean in all the worlds in o (which the retrieving property guarantees to exist). This can be achieved by defining the covering so that it contains only worlds with at least k δ-choices.

Theorem 5.8 (⚙). *For all worlds w and choice names δ of a Boolean choice sequence, $w \Vdash !\delta \in \mathbb{N} \to \mathbb{2}$.*

Proof. According to Π's semantics from Fig. 2 and using both $\Box_4$ and the computation property TS_r (Thm. 5.6), we must prove $w_1 \Vdash !\delta(n) \in \mathbb{2}$ given $w_1 \sqsupseteq w$ and $w_1 \Vdash n \in \mathbb{N}$. According to $\mathbb{N}$'s semantics, we know that $\Box_{w_1}(w'.\exists(k : \mathbb{N}).w' \vdash n \mapsto \underline{k})$, which follows from the clauses in Fig. 2. Using locality (Thm. 5.6) it suffices to prove $\Box_{w_1}(w'.w' \Vdash !\delta(n) \in \mathbb{2})$, and using $\Box_2$ (Def. 5.4), $w_2 \Vdash !\delta(n) \in \mathbb{2}$ given $w_2 \sqsupseteq w_1$ and $w_2 \vdash n \mapsto \underline{k}$, where $k \in \mathbb{N}$. Using TS_r, it is enough to prove $w_2 \Vdash !\delta(\underline{k}) \in \mathbb{2}$. Because $\Box$ is retrieving, we obtain $\Box_{w_2}(w'.\exists(c : C).\forall^{\sqsubseteq}_{w'}(w''.\mathsf{read}(w'', \delta, k) = c))$. Using locality it suffices to prove $\Box_{w_2}(w'.w' \Vdash !\delta(\underline{k}) \in \mathbb{2})$, and using $\Box_2$ we have to prove $w_3 \Vdash !\delta(\underline{k}) \in \mathbb{2}$ given $w_3 \sqsupseteq w_2$ and $\forall^{\sqsubseteq}_{w_3}(w'.\mathsf{read}(w', \delta, k) = c)$ for some choice $c \in \{\mathbb{t}, \mathbb{f}\}$. We derive that $w_3 \vdash !\delta(\underline{k}) \mapsto c$, and using TS_r, it is enough to prove $w_2 \Vdash c \in \mathbb{2}$, which is straightforward. □

5.7 Sequents and Rules

A $\mathrm{TT}^{\Box}_C$ sequent is of the form $\Gamma \vdash t : T$, expressing that t has type T in the context Γ. Sequents are interpreted using the pairwise functionality approach [34, Def. 4.5], further discussed in [23, Sec. 2.2.2]. To define this semantics, we first define some notation. Let Sub be the type of substitutions, i.e., partial maps from variables to closed terms, written as follows: $[x_1 \backslash t_1; \cdots; x_n \backslash t_n]$. Given a substitution s of the above form, and $i \leq n$, we write $s|_i$ for $[x_1 \backslash t_1; \cdots; x_i \backslash t_i]$. We further define the following relations (equality between substitutions and equality between contexts), where the substitutions s_1 and s_2 are of the form $[x_1 \backslash t_1; \cdots; x_n \backslash t_n]$ and $[x_1 \backslash u_1; \cdots; x_n \backslash u_n]$, respectively, and Γ is a context of the form $x_1 : A_1, \cdots, x_n : A_n$:

$$w \Vdash s_1 \equiv s_2 \in \Gamma \quad := \quad \forall(i : [1..n]).w \Vdash t_i \equiv u_i \in A_i[s_1|_{i-1}]$$
$$w \Vdash \Gamma[s_1] \equiv \Gamma[s_2] \quad := \quad \forall(i : [1..n]).w \Vdash A_i[s_1|_{i-1}] \equiv A_i[s_2|_{i-1}]$$

Given a sequent S of the form $\Gamma \vdash t : A$, its semantics, written as $w \Vdash S$, is defined as follows:

$$\forall(s_1, s_2 : \mathrm{Sub}).\ (w \Vdash s_1 \equiv s_2 \in \Gamma \wedge w \Vdash \Gamma[s_1] \equiv \Gamma[s_2]) \\ \to (w \Vdash A[s_1] \equiv A[s_2] \wedge w \Vdash t[s_1] \equiv t[s_2] \in A[s_1])$$

Finally, a $\mathrm{TT}^{\square}_{C}$ inference rule R that allows deriving the sequent S_0 from the sequents $S_1, \ldots, S_n$, is valid w.r.t. the world w, written $w \Vdash R$, if $w \Vdash S_1 \to \cdots \to w \Vdash S_n \to w \Vdash S_0$.

Since our main focus is to prove separation results for MLTT, we only present $\mathrm{TT}^{\square}_{C}$'s semantics, which is used to interpret MLTT, and omit its inference rules.

6 INTERPRETATION OF MLTT IN $\mathrm{TT}^{\square}_{C}$

MLTT [40], whose syntax is recalled in Fig. 1, is a dependent type theory with dependent products and sums, a base type of numbers, and a type of types, $\mathbb{U}$, that, in particular, allows quantifying over types. It is at the heart of many type theories such as CIC [43], Agda's modern version of MLTT [39], HoTT [58], etc.

MLTT comes with five judgments for: well-formed contexts $\vdash \Gamma$, well-formed types $\Gamma \vdash A$, well-typed terms $\Gamma \vdash t : A$, convertible types $\Gamma \vdash A \equiv B$, and convertible terms $\Gamma \vdash t \equiv u : A$. Appx. B recalls MLTT's typing and conversion rules for convenience.

Fig. 1 interprets MLTT's syntax in $\mathrm{TT}^{\square}_{C}$. Crucially, this translation satisfies the following properties, which are used to prove Thm. 6.1 and to satisfy $\mathsf{MP}_{\mathsf{PR}}$ in Sec. 7.3, respectively:

- $[\![t[x\backslash u]]\!] = [\![t]\!][x\backslash [\![u]\!]]$ for any MLTT terms t and u
- $\mathsf{effect\text{-}free}([\![t]\!])$ for any MLTT term t

MLTT judgments are translated to $\mathrm{TT}^{\square}_{C}$ sequents as follows, where the universe level $i > 1$ is a parameter of the translation:

$$\begin{array}{lcl} [\![\vdash \Gamma]\!] & = & \top \\ [\![\Gamma \vdash A]\!] & = & [\![\Gamma]\!] \vdash [\![A]\!] : \mathbb{U}_i \\ [\![\Gamma \vdash t : A]\!] & = & [\![\Gamma]\!] \vdash [\![t]\!] : [\![A]\!] \\ [\![\Gamma \vdash A \equiv B]\!] & = & [\![\Gamma]\!] \vdash \star : [\![A]\!] = [\![B]\!] \in \mathbb{U}_i \\ [\![\Gamma \vdash t \equiv u : A]\!] & = & [\![\Gamma]\!] \vdash \star : [\![t]\!] = [\![u]\!] \in [\![A]\!] \end{array}$$

Using this translation of MLTT to $\mathrm{TT}^{\square}_{C}$, as well as the semantics presented in Sec. 5.6 and Sec. 5.7, we can build models of MLTT:

Theorem 6.1 (⚙). *Given an MLTT inference rule of the following form (left), where J_i are MLTT judgments, and a world w, then the following $\mathrm{TT}^{\square}_{C}$ inference rule (right) is valid w.r.t. w:*

$$\mathit{MLTT}: \ \frac{J_1 \quad \cdots \quad J_n}{J_0} \qquad \mathit{TT}^{\square}_{C}: \ \frac{[\![J_1]\!] \quad \cdots \quad [\![J_n]\!]}{[\![J_0]\!]}$$

Let us illustrate how this proof works by considering MLTT's elimination rule for numbers from Fig. 3 in Appx. B:

$$\frac{\begin{array}{l}\Gamma, x : \mathbb{N} \vdash P \qquad \Gamma \vdash z : P[x\backslash 0] \\ \Gamma \vdash s : \Pi y{:}\mathbb{N}.P[x\backslash y] \to P[x\backslash \mathsf{S}\ y] \qquad \Gamma \vdash n : \mathbb{N}\end{array}}{\Gamma \vdash \mathbb{N}_{\mathsf{ind}}\ P\ z\ s\ n : P[x\backslash n]}$$

We first convert it to a $\mathrm{TT}^{\square}_{C}$ rule using the $[\![\_]\!]$ interpretation:

$$\frac{\begin{array}{l}[\![\Gamma]\!], x : \mathbb{N} \vdash [\![P]\!] : \mathbb{U}_i \qquad [\![\Gamma]\!] \vdash [\![z]\!] : [\![P]\!][x\backslash 0] \\ [\![\Gamma]\!] \vdash [\![s]\!] : \Pi y{:}\mathbb{N}.[\![P]\!][x\backslash y] \to [\![P]\!][x\backslash \mathsf{S}\ y] \qquad [\![\Gamma]\!] \vdash [\![n]\!] : \mathbb{N}\end{array}}{[\![\Gamma]\!] \vdash \mathbb{N}_{\mathsf{ind}}\ [\![z]\!]\ [\![s]\!]\ [\![n]\!] : [\![P]\!][x\backslash [\![n]\!]]}$$

which is an instance of the following rule, which is valid according to the above semantics for $\mathrm{TT}^{\square}_{C}$:

$$\frac{\begin{array}{l}\Gamma, x : \mathbb{N} \vdash P : \mathbb{U}_i \qquad \Gamma \vdash z : P[x\backslash \underline{0}] \\ \Gamma \vdash s : \Pi y{:}\mathbb{N}.P[x\backslash y] \to P[x\backslash \mathsf{S}\ y] \qquad \Gamma \vdash n : \mathbb{N}\end{array}}{\Gamma \vdash \mathbb{N}_{\mathsf{ind}}\ z\ s\ n : P[x\backslash n]}$$

7 SEPARATION OF $\mathsf{MP}_{\mathbb{B}}$ AND $\mathsf{MP}_{\mathsf{PR}}$

We prove that instantiating C with choice sequences and $\square$ with a Beth modality as in [5] yields a model validating constructively $\neg\mathsf{MP}_{\mathbb{B}}$, but still validating $\mathsf{MP}_{\mathsf{PR}}$ assuming Markov's Principle in the meta-theory (while we could use meta-$\mathsf{MP}_{\mathsf{PR}}$, we use here meta-$\mathsf{MP}_{\mathbb{B}}$ for convenience, avoiding having to encode functions using primitive recursive functions). To do so, we express $\mathbb{N}$ and $\mathbb{B}$ as the $\mathrm{TT}^{\square}_{C}$ types $\mathbb{N}$ and 2 of possibly effectful terms. Those effects allow refuting $\mathsf{MP}_{\mathbb{B}}$ because f can be undetermined for all inputs and thus satisfy $\neg\neg(\exists n.\ fn = \mathrm{true})$ but not $\exists n.\ fn = \mathrm{true}$. However, primitive recursive functions can be encoded as natural numbers, and thus behave like pure, effect-free functions. Concretely, Sec. 7.1 proves the following, where we write $b = \mathbb{t}$ for $b = \mathbb{t} \in 2$, and $\exists x{:}A.B$ for either $\Sigma x{:}A.B$ (⚙) or $\downarrow\Sigma x{:}A.B$ (⚙):

$$\forall(w : \mathcal{W}).w \not\Vdash \Pi f{:}\mathbb{N} \to 2.(\neg\neg\exists n{:}\mathbb{N}.f\ n = \mathbb{t}) \to \exists n{:}\mathbb{N}.f\ n = \mathbb{t}$$

Sec. 7.2 shows that assuming $\mathsf{MP}_{\mathbb{B}}$ in the meta-theory, MP for effect-free (or *pure*) functions is valid in *all* models in [5], $\Pi_{\boldsymbol{p}} x{:}A.B$ defined as $\Pi x{:}\mathsf{E}(A).B$, i.e. letting f range over pure terms only (see ⚙ for the $\Sigma x{:}A.B$ version and ⚙ for the $\downarrow\Sigma x{:}A.B$ version):

$$\forall(w : \mathcal{W}).w \Vdash \Pi_{\boldsymbol{p}} f{:}\mathbb{N} \to 2.(\neg\neg\exists n{:}\mathbb{N}.f\ n = \mathbb{t}) \to \exists n{:}\mathbb{N}.f\ n = \mathbb{t}$$

We refer to this version of MP as MP_{pure}. To show that this implies $\mathsf{MP}_{\mathsf{PR}}$, note that $\mathsf{MP}_{\mathsf{PR}}$ can be equivalently stated as

$$\forall(w : \mathcal{W}).w \Vdash \Pi m{:}\mathbb{N}.(\neg\neg\exists n{:}\mathbb{N}.\mathrm{eval}\ m\ n = \mathbb{t}) \to \exists n{:}\mathbb{N}.\mathrm{eval}\ m\ n = \mathbb{t}$$

where $\mathrm{eval} : \mathbb{N} \to \mathbb{N} \to 2$ is a pure function which interprets its first argument as the Gödelisation of a primitive recursive f, such that for any such f there is m with $\forall n.fn = \mathrm{eval}\ m\ n$. Now, whenever an effectful m evaluates to c in a world w, we have that eval c is a pure function for all $w' \sqsupseteq w$, making MP_{pure} applicable (⚙).

We now exhibit one class of models that allow both falsifying $\mathsf{MP}_{\mathbb{B}}$ and satisfying $\mathsf{MP}_{\mathsf{PR}}$ simultaneously. Those models are the ones presented in Sec. 5.6 and Sec. 5.7 such that C (Def. 5.2) is an *immutable* (Def. 7.1) type of Boolean choices $\mathbb{t}$ and $\mathbb{f}$, and $\square$ (Def. 5.4) is *retrieving* (Def. 5.7) and *choice-following* (Def. 7.2), which are used to falsify $\mathsf{MP}_{\mathbb{B}}$ in Sec. 7.1.

7.1 Falsifying $\mathsf{MP}_{\mathbb{B}}$

This section shows that the above models allow proving the following formula, adapting the proof used in [5] to falsify LEM:

$$\forall(w : \mathcal{W}).w \not\Vdash \Pi f{:}\mathbb{N} \to 2.(\neg\neg\exists n{:}\mathbb{N}.f\ n = \mathbb{t}) \to \exists n{:}\mathbb{N}.f\ n = \mathbb{t} \quad (1)$$

To prove this, we will instantiate the Π using a fresh choice name δ that inhabits $\mathbb{N} \to 2$. We will then show that (A) even though $\neg\neg\exists n{:}\mathbb{N}.f\ n = \mathbb{t}$ holds because it is always possible to make the choice $\mathbb{t}$ in a branch starting from w, (B) $\exists n{:}\mathbb{N}.f\ n = \mathbb{t}$ does not hold because there might be a branch from w where the choice $\mathbb{t}$ is never made. To capture this high-level proof-sketch, we assume that $\boldsymbol{\kappa}$ is $\mathbb{f}$ and $\boldsymbol{\sigma}$ is $\mathbb{t}$ as in Ex. 5.3, and we require that C and $\square$ satisfy the

immutability and choice-following properties presented below. Immutability is used to prove (A) by showing that we can always make the immutable choice $\Bbbt$, while retrieving and choice-following are used to prove (B) by showing that there cannot be a number n such that $\mathsf{read}(w', \delta, n)$ is $\Bbbt$ for some extension w' of w, by following a branch from w where only the choice default $\boldsymbol{\kappa}$ is made.

Definition 7.1 (⚙ Immutability). $\mathcal{C}$ is called *immutable* if there exist a function $\mathsf{freeze} \in \mathcal{N} \to \mathcal{C} \to \mathcal{W} \to \mathcal{W}$ ($\mathsf{freeze}(\delta, c, w)$ is intended to return a world w' that extends w with the δ-choice c, such that c can be retrieved in any $w'' \sqsupseteq w'$), satisfying:

- I1 $\forall \delta\ w\ c. w \sqsubseteq \mathsf{freeze}(\delta, c, w)$
- I2 Immutable choices stay immutable:
  $\forall \delta\ w\ c. \exists (n : \mathbb{N}). \forall^{\sqsubseteq}_{\mathsf{freeze}(\delta,c,w)} (w'. \mathsf{read}(w', \delta, n) = c)$

Definition 7.2 (⚙ Choice-following). A modality $\Box$ is called *choice-following* if:

$$\forall (\delta : \mathcal{N})(w : \mathcal{W})(P : \mathcal{P}_w).\\ \mathsf{def}\mathcal{C}(w, \delta) \to \Box_w P \to \exists^{\sqsubseteq}_w (w'. P\ w' \wedge \mathsf{def}\mathcal{C}(w', \delta))$$

To prove Eq. (1), we assume a world w, and that the following formula is inhabited at world w, and we derive a contradiction:

$$\Pi f{:}\mathbb{N} \to 2. (\neg\neg \exists n{:}\mathbb{N}. f\ n = \Bbbt) \to \exists n{:}\mathbb{N}. f\ n = \Bbbt \tag{2}$$

We generate a new choice name $\delta := \nu\mathcal{C}(w)$, such that by property E2, the only δ-choice in $w_1 := \mathsf{start}\nu\mathcal{C}(w)$ is the default choice $\Bbbf$. Thanks to E1 and monotonicity (Thm. 5.6) we obtain that the formula in Eq. (2) is inhabited at w_1. We now instantiate that formula with $!\delta$, and have to prove that $!\delta \in \mathbb{N} \to 2$ (Step 1 below), and that $\neg\neg \exists n{:}\mathbb{N}. !\delta(n) = \Bbbt$ (Step 2) to obtain that $\exists n{:}\mathbb{N}. !\delta(n) = \Bbbt$ is inhabited too, from which we derive a contradiction (Step 3).

Step 1: We first obtain that $w_1 \vDash\ !\delta \in \mathbb{N} \to 2$ from Thm. 5.8.

Step 2: We now prove that $\neg\neg \exists n{:}\mathbb{N}. !\delta(n) = \Bbbt$ is inhabited at world w_1, which is equivalent to proving[1]

$$\forall^{\sqsubseteq}_{w_1} \left(w_2. \neg \forall^{\sqsubseteq}_{w_2} (w_3. \neg \exists (n : \mathsf{Term}). w_3 \vDash n \in \mathbb{N} \wedge w_3 \vDash\ !\delta(n) = \Bbbt)\right)$$

Therefore, we assume $w_2 \sqsupseteq w_1$, and derive a contradiction from:

$$\forall^{\sqsubseteq}_{w_2} (w_3. \neg \exists (n : \mathsf{Term}). w_3 \vDash n \in \mathbb{N} \wedge w_3 \vDash\ !\delta(n) = \Bbbt) \tag{3}$$

We instantiate Eq. (3) with $w_3 := \mathsf{freeze}(\delta, \Bbbt, w_2)$, which is an extension of w_2 by I1. We obtain a contradiction by exhibiting a term n such that $w_3 \vDash n \in \mathbb{N}$ and $w_3 \vDash\ !\delta(n) = \Bbbt$. We obtain this term thanks to I2, which gives us a number k such that $\mathsf{read}(w', \delta, k)$ returns $\Bbbt$ in any $w' \sqsupseteq w_3$. We can easily prove that $\underline{k} \in \mathbb{N}$. Finally, proving $w_3 \vDash\ !\delta(\underline{k}) = \Bbbt$ follows using a proof similar to the one of Thm. 5.8.

Step 3: Thanks to the above steps, we now have a proof that $w_1 \vDash \exists n{:}\mathbb{N}. !\delta(n) = \Bbbt$, from which we derive a contradiction. According to the semantics of this formula, using the fact that $\Box$ is choice-following (Def. 7.2) and retrieving (Def. 5.7), we obtain a world $w_2 \sqsupseteq w_1$ and a number $k \in \mathbb{N}$ such that $w_2 \vdash n \Mapsto \underline{k}$ and $w_2 \vdash !\delta(\underline{k}) \Mapsto \Bbbt$, and such that the only δ-choice in w_2 is $\Bbbf$. Therefore, it must be that $w_2 \vdash !\delta(\underline{k}) \Mapsto \Bbbf$, which contradicts $w_2 \vdash !\delta(\underline{k}) \Mapsto \Bbbt$.

In the above proof of Eq. (1), $\exists x{:}A.B$ could either be $\Sigma x{:}A.B$ or $\downarrow\Sigma x{:}A.B$, because the formula is a negation, and we therefore do not have to construct an inhabitant of a Σ type. In particular, in Step 3, we obtain a proof that $w_1 \vDash \exists n{:}\mathbb{N}. !\delta(n) = \Bbbt$ from Eq. (2). However, the inhabitant of that formula does not matter, i.e., the term t satisfying $w_1 \vDash t \in \exists n{:}\mathbb{N}. !\delta(n) = \Bbbt$ is irrelevant. What matters is that we can derive from the semantics of the formula that

$$\Box_{w_1} (w_2. \exists (k : \mathbb{N}). w_2 \vdash n \Mapsto \underline{k} \wedge w_2 \vdash\ !\delta(\underline{k}) \Mapsto \Bbbt) \tag{4}$$

If $\exists x{:}A.B$ is $\downarrow\Sigma x{:}A.B$ compared to $\Sigma x{:}A.B$, we only have to use $\Box_3$ once more to derive the formula in Eq. (4).

7.2 Satisfying $\mathsf{MP}_{\mathsf{PR}}$

We first start by showing how to satisfy MP_{pure}, i.e., we show how the models mentioned above allow satisfying:

$$\forall (w : \mathcal{W}). w \vDash \mathbf{\Pi}_p f{:}\mathbb{N} \to 2. (\neg\neg \exists n{:}\mathbb{N}. f\ n = \Bbbt) \to \exists n{:}\mathbb{N}. f\ n = \Bbbt$$

We begin with a proof when $\exists x{:}A.B$ is $\downarrow\Sigma x{:}A.B$, and discuss the $\Sigma x{:}A.B$ case below, which is more involved as it requires building a non-trivial proof of $\Sigma n{:}\mathbb{N}. f\ n = \Bbbt$ knowing that $\neg\neg \Sigma n{:}\mathbb{N}. f\ n = \Bbbt$.

Assume $w \in \mathcal{W}$ and $w \vDash f \in \mathbb{N} \to 2$ such that f is effect-free, and such that $w \vDash \neg\neg \downarrow\Sigma n{:}\mathbb{N}. f\ n = \Bbbt$, and let us prove that $w \vDash \downarrow\Sigma n{:}\mathbb{N}. f\ n = \Bbbt$. For this, it is enough to exhibit a $k \in \mathbb{N}$ such that $w \vDash f\ \underline{k} = \Bbbt$. Using Markov's Principle in the metatheory (in the form $\mathsf{MP}_{\mathbb{B}}$ for convenience while $\mathsf{MP}_{\mathsf{PR}}$ would be enough), we derive the above by proving that $w \vDash f\ \underline{k} = \Bbbt$ is decidable for all $k \in \mathbb{N}$ (Step 1), and by proving that $\neg\neg \exists (k : \mathbb{N}). w \vDash f\ \underline{k} = \Bbbt$ (Step 2).

To prove the above, we make use of the following result that worlds are irrelevant for effect-free expressions:

Lemma 7.3 (⚙ Effect-free computations). *For all $t, u \in \mathsf{Term}$ and $w_1, w_2 \in \mathcal{W}$, if $\mathsf{effect\text{-}free}(t)$ and $w_1 \vdash t \Mapsto u$ then $w_2 \vdash t \Mapsto u$.*

Step 1: We start by proving that $w \vDash f\ \underline{k} = \Bbbt$ is decidable for all $k \in \mathbb{N}$. Let us assume that $k \in \mathbb{N}$. Using the fact that $w \vDash f\ \underline{k} \in 2$, and therefore, by the semantics of 2, that

$$\Box_w (w'. w' \vdash f\ \underline{k} \Mapsto \Bbbt \vee w' \vdash f\ \underline{k} \Mapsto \Bbbf)$$

and using the combination of $\Box_2$ and $\Box_5$, we conclude that there exists a $w' \sqsupseteq w$ such that $w' \vdash f\ \underline{k} \Mapsto \Bbbt$ or $w' \vdash f\ \underline{k} \Mapsto \Bbbf$. We analyze the two cases.

If $w' \vdash f\ \underline{k} \Mapsto \Bbbt$, then in general we cannot conclude that $w \vDash f\ \underline{k} = \Bbbt$. However, since $\mathsf{effect\text{-}free}(f\ \underline{k})$, then by Lem. 7.3, $w \vdash f\ \underline{k} \Mapsto \Bbbt$, from which we conclude $w \vDash f\ \underline{k} = \Bbbt$.

If $w' \vdash f\ \underline{k} \Mapsto \Bbbf$, then we prove that $w \nvDash f\ \underline{k} = \Bbbt$, i.e., we assume $w \vDash f\ \underline{k} = \Bbbt$, and prove a contradiction. Using $\Box_2$, $\Box_4$ and $\Box_5$ from Def. 5.4, as $w \vDash f\ \underline{k} = \Bbbt$ holds, then we obtain that $w'' \vdash f\ \underline{k} \Mapsto \Bbbt$ for some $w'' \sqsupseteq w$, potentially different from w'.[2] By Lem. 7.3, we obtain $w' \vdash f\ \underline{k} \Mapsto \Bbbt$, contradicting $w' \vdash f\ \underline{k} \Mapsto \Bbbf$.

Step 2: We now prove $\neg\neg \exists (k : \mathbb{N}). w \vDash f\ \underline{k} = \Bbbt$ using the fact that $w \vDash \neg\neg \downarrow\Sigma n{:}\mathbb{N}. f\ n = \Bbbt$. We assume $\neg \exists (k : \mathbb{N}). w \vDash f\ \underline{k} = \Bbbt$, and derive a contradiction. From $w \vDash \neg\neg \downarrow\Sigma n{:}\mathbb{N}. f\ n = \Bbbt$, we derive that

$$\forall^{\sqsubseteq}_{w} \left(w_1. \neg \forall^{\sqsubseteq}_{w_1} (w_2. \neg \exists (n : \mathsf{Term}). w_2 \vDash n \in \mathbb{N} \wedge w_2 \vDash f\ n = \Bbbt)\right)$$

[1] To prove this equivalence, we rely on the fact that $w \vDash \neg T$ is equivalent to $\forall^{\sqsubseteq}_{w_1} (w. \neg w \vDash T)$, from which it follows that $w \vDash \neg\neg T$ is equivalent to $\forall^{\sqsubseteq}_{w_1} \left(w. \neg \forall^{\sqsubseteq}_{w_2} (w_1. \neg w \vDash T)\right)$.

[2] This is a typical pattern. From $w \vDash f\ \underline{k} = \Bbbt$ we derive $\Box_w (w'. w' \vdash f\ \underline{k} \Mapsto \Bbbt)$. As we are proving $\bot$, using $\Box_5$, it is enough to prove $\Box_w (w'. \bot)$, and applying $\Box_2$ to $\Box_w (w'. w' \vdash f\ \underline{k} \Mapsto \Bbbt)$, and then using $\Box_4$, it is then enough to prove a contradiction assuming $w' \vdash f\ \underline{k} \Mapsto \Bbbt$ for some $w' \sqsupseteq w$.

which we instantiate with w (which extends w since $\sqsubseteq$ is reflexive), and from which we derive a contradiction by proving that

$$\forall^{\sqsubseteq}_{w}\,(w_1.\neg\exists(n : \mathsf{Term}).w_1 \vDash n \in \mathbb{N} \wedge w_1 \vDash f\ n = \mathbb{t})$$

To prove this formula, we assume $w' \sqsupseteq w$ and $n \in \mathsf{Term}$ such that $w' \vDash n \in \mathbb{N}$ and $w' \vDash f\ n = \mathbb{t}$ and prove a contradiction. Using $\Box_2$, $\Box_4$ and $\Box_5$ from Def. 5.4, we obtain that $w'' \vDash f\ \underline{k} \Mapsto \mathbb{t}$ for some $k \in \mathbb{N}$ and $w'' \sqsupseteq w'$ such that $w'' \vDash n \Mapsto \underline{k}$. Using Lem. 7.3, we obtain $w \vDash f\ \underline{k} \Mapsto \mathbb{t}$, which contradicts our assumption $\neg\exists(k : \mathbb{N}).w \vDash f\ \underline{k} = \mathbb{t}$.

Non-truncated version. To prove that MP_{pure} holds when $\exists x{:}A.B$ is $\Sigma x{:}A.B$, we need to show that it is inhabited by a program that computes the witness of the existential. We show here that:

$$w \vDash \mathsf{search} \in \mathsf{MP}_{pure}$$

where search makes use of a general recursive function that recursively searches for the witness of the existential:

$$\mathsf{search} := \lambda f.\lambda c.\langle \mathsf{fix}(\lambda R.\lambda n.\mathsf{if}\ f(n)\ \mathsf{then}\ n\ \mathsf{else}\ R(\mathsf{S}\ n))\ \underline{0}, \star\rangle$$

We can then use the fact that $w \vDash \mathsf{MP}_{pure}$ to prove that search inhabits MP_{pure}. Let $f \in \mathbb{N} \to \mathbb{2}$ such that $\neg\neg\exists n{:}\mathbb{N}.f\ n = \mathbb{t}$. We then have to prove that $F := \mathsf{fix}(\lambda R.\lambda n.\mathsf{if}\ f(n)\ \mathsf{then}\ n\ \mathsf{else}\ R(\mathsf{S}\ n))\ \underline{0}$ computes to a number n such that $f\ n = \mathbb{t}$, assuming $\downarrow\Sigma n{:}\mathbb{N}.f\ n = \mathbb{t}$. Thanks to $\downarrow\Sigma n{:}\mathbb{N}.f\ n = \mathbb{t}$ we know that F terminates, and by its definition that it computes to a term n such that $f\ n$ computes to $\mathbb{t}$.

Gödelised version. We explain here how we can derive $w \vDash \lambda n.\mathtt{let}\ v = n\ \mathtt{in}\ e(\mathsf{eval}\ v) \in \mathsf{MP}_{\mathsf{PR}}$ from a proof of $w \vDash e \in \mathsf{MP}_{pure}$ where $\mathsf{MP}_{\mathsf{PR}}$ is here the Gödelised version:

$$\Pi m{:}\mathbb{N}.(\neg\neg\exists n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}) \to \exists n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}$$

where $\mathsf{eval} : \mathbb{N} \to \mathbb{N} \to \mathbb{2}$ is a pure function (i.e., effect-free(eval)). Let $w \vDash m \in \mathbb{N}$ and let us derive the following formula, where $E := \mathtt{let}\ v = m\ \mathtt{in}\ e(\mathsf{eval}\ v)$:

$$w \vDash E \in (\neg\neg\exists n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}) \to \exists n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}$$

To use MP_{pure}, we need to instantiate it with an effect-free function, which $\mathsf{eval}\ m$ might not be because m might not be effect-free. Because $w \vDash m \in \mathbb{N}$, it follows that $\Box_w(w'.\exists(k : \mathbb{N}).w' \vDash m \Mapsto \underline{k})$. Therefore, using $\Box_2$ and locality (Thm. 5.6), it is enough to prove that assuming $w' \sqsupseteq w$ and $k \in \mathbb{N}$ such that $w' \vDash m \Mapsto \underline{k}$, then:

$$w' \vDash E \in (\neg\neg\exists n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}) \to \exists n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}$$

Then, by computation it is enough to prove

$$w' \vDash e(\mathsf{eval}\ \underline{k}) \in (\neg\neg\exists n{:}\mathbb{N}.\mathsf{eval}\ \underline{k}\ n = \mathbb{t}) \to \exists n{:}\mathbb{N}.\mathsf{eval}\ \underline{k}\ n = \mathbb{t}$$

Since $\mathsf{eval}\ \underline{k}$ is effect-free, we conclude by applying MP_{pure} to $\mathsf{eval}\ \underline{k}$.

7.3 Separation in MLTT

We now show how to use the above results to further separate $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$ in MLTT (our formalisation relies on the Agda formalisation of MLTT from [1, 41]). While the version of MLTT presented in Sec. 6 does not include Booleans, since they can straightforwardly be defined using numbers, we assume here the existence of an MLTT Boolean type $\mathbb{2}$, with constructors $\mathbb{t}$ and $\mathbb{f}$ and eliminator $\mathbb{2}_{\mathsf{ind}}\ A\ b\ t\ u$ (we write $b = \mathbb{t}$ for $\mathbb{2}_{\mathsf{ind}}\ \mathbb{U}\ b\ \mathbb{1}\ \mathbb{0}$). We recall here the two MLTT (top) and $\mathsf{TT}^{\Box}_{C}$ (bottom) versions of MP in consideration, where $\mathsf{eval} : \mathbb{N} \to \mathbb{N} \to \mathbb{2}$ and $\mathsf{eval} := [\![\mathsf{eval}]\!]$:

$$\begin{array}{lcl}
\mathsf{MP}_{\mathbb{B}} & := & \Pi f{:}\mathbb{N} \to \mathbb{2}.(\neg\neg\Sigma n{:}\mathbb{N}.f\ n = \mathbb{t}) \to \Sigma n{:}\mathbb{N}.f\ n = \mathbb{t} \\
\mathsf{MP}_{\mathsf{PR}} & := & \Pi m{:}\mathbb{N}.(\neg\neg\Sigma n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}) \to \Sigma n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t} \\
\mathsf{MP}_{\mathbb{B}} & := & \Pi f{:}\mathbb{N} \to \mathbb{2}.(\neg\neg\Sigma n{:}\mathbb{N}.f\ n = \mathbb{t}) \to \Sigma n{:}\mathbb{N}.f\ n = \mathbb{t} \\
\mathsf{MP}_{\mathsf{PR}} & := & \Pi m{:}\mathbb{N}.(\neg\neg\Sigma n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}) \to \Sigma n{:}\mathbb{N}.\mathsf{eval}\ m\ n = \mathbb{t}
\end{array}$$

Using a similar encoding of $b = \mathbb{t}$ as used for MLTT, it follows that: $[\![\mathsf{MP}_{\mathbb{B}}]\!] = \mathsf{MP}_{\mathbb{B}}$ and $[\![\mathsf{MP}_{\mathsf{PR}}]\!] = \mathsf{MP}_{\mathsf{PR}}$. Given MLTT terms t and A, we write $\mathsf{TT}^{\Box}_{C} \vDash A$ for $\forall(w : \mathcal{W}).w \vDash [\![A]\!]$, and $\mathsf{TT}^{\Box}_{C} \vDash t : A$ for $\forall(w : \mathcal{W}).w \vDash [\![t]\!] \in [\![A]\!]$. Given the above results, we derive the following for the class of models described above:

$$\neg(\mathsf{TT}^{\Box}_{C} \vDash \mathsf{MP}_{\mathbb{B}}) \qquad (\mathsf{TT}^{\Box}_{C} \vDash \mathsf{eval} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}) \to (\mathsf{TT}^{\Box}_{C} \vDash \mathsf{MP}_{\mathsf{PR}})$$

8 SEPARATION OF $\mathsf{MP}_{\mathbb{P}}$ AND $\mathsf{MP}_{\mathbb{B}}$

Using a similar technique to the one used to separate $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$ in Sec. 7, we now use $\mathsf{TT}^{\Box}_{C}$ to also separate $\mathsf{MP}_{\mathbb{P}}$ and $\mathsf{MP}_{\mathbb{B}}$, where $\mathsf{MP}_{\mathbb{P}}$ is defined in $\mathsf{TT}^{\Box}_{C}$ as follows, where i is a universe level, $T \vee U := \downarrow(T + U)$, and $\exists x{:}A.B$ is $\downarrow\Sigma x{:}A.B$ (as discussed in Appx. A):

$$\Pi f{:}\mathbb{N} \to \mathbb{U}_i.(\Pi n{:}\mathbb{N}.f\ n \vee \neg f\ n) \to (\neg\neg\exists n{:}\mathbb{N}.f\ n) \to \exists n{:}\mathbb{N}.f\ n$$

Compared to Sec. 7, to falsify $\mathsf{MP}_{\mathbb{P}}$ in Sec. 8.1 we use the propositional choices $\boldsymbol{\sigma} := \mathbb{1}$ and $\boldsymbol{\kappa} := \mathbb{0}$, while Sec. 7.1 uses the Boolean choices $\mathbb{t}$ and $\mathbb{f}$ to falsify $\mathsf{MP}_{\mathbb{B}}$. Furthermore, as for the proof of satisfaction of $\mathsf{MP}_{\mathsf{PR}}$ in Sec. 7.2, the proof of satisfaction of $\mathsf{MP}_{\mathbb{B}}$ in Sec. 8.2 makes use of the fact that a computation that makes limited use of effects can perform that computation starting from any world. By "limited use" we mean: effect-free computations in Sec. 7.2, and Boolean-valued functions that therefore can make limited use of propositional choices in Sec. 8.2.

We now exhibit a class of models that allow both falsifying $\mathsf{MP}_{\mathbb{P}}$ while simultaneously satisfying $\mathsf{MP}_{\mathbb{B}}$. Those models are the models presented in Sec. 5.6 and Sec. 5.7 such that C is an immutable type of propositional choices $\boldsymbol{\sigma} := \mathbb{1}$ and $\boldsymbol{\kappa} := \mathbb{0}$, and $\Box$ is retrieving and choice-following, which are used to falsify $\mathsf{MP}_{\mathbb{P}}$ in Sec. 8.1.

8.1 Falsifying $\mathsf{MP}_{\mathbb{P}}$

We start by showing that the above models allow falsifying $\mathsf{MP}_{\mathbb{P}}$ by proving $\forall(w : \mathcal{W}).w \nvDash \mathsf{MP}_{\mathbb{P}}$ (⚙).

We assume a world w and derive a contradiction from $w \vDash \mathsf{MP}_{\mathbb{P}}$. We instantiate f with a "fresh" (w.r.t. the current world) choice sequence $\delta := \nu C(w)$, which effectively is not present in w but is present in an extension $w_1 := \mathsf{start}\nu C(w)$ of w (thanks to property E1). Compared to Sec. 7.1, δ is here the choice name of a choice sequence of propositional choices $\mathbb{1}$ and $\mathbb{0}$, with default choice $\boldsymbol{\kappa} := \mathbb{0}$, instead of Boolean choices $\mathbb{t}$ and $\mathbb{f}$. Note that instantiating f with $\lambda n.(!\delta(n) = \mathbb{t})$, when δ is the name of a Boolean choice sequence, which then inhabits $\mathbb{N} \to \mathbb{U}_i$, would also allow falsifying $\mathsf{MP}_{\mathbb{P}}$, but it would not allow satisfying $\mathsf{MP}_{\mathbb{B}}$ as explained in Sec. 7. To falsify $\mathsf{MP}_{\mathbb{P}}$ we then have to prove that at world w_1, $!\delta \in \mathbb{N} \to \mathbb{U}_i$ (Step 1), $\Pi n{:}\mathbb{N}.!\delta(n) \vee \neg!\delta(n)$ (Step 2) and $\neg\neg\exists n{:}\mathbb{N}.!\delta(n)$ (Step 3), and disprove $\exists n{:}\mathbb{N}.!\delta(n)$ (Step 4). Because some of the steps are similar to those in Sec. 7.1, we only sketch them here.

Step 1: Similar to Step 1 in Sec. 7.1, we first obtain that $w_1 \vDash !\delta \in \mathbb{N} \to \mathbb{U}_i$ using a straightforward adaptation of Thm. 5.8.

Step 2: To prove $w_1 \Vdash \Pi n{:}\mathbb{N}.!\delta(n) \vee \neg!\delta(n)$, assume that $w_2 \Vdash n \in \mathbb{N}$ for some $w_2 \sqsupseteq w_1$ and prove that $w_2 \Vdash !\delta(n) \vee \neg!\delta(n)$. As in the proof of Thm. 5.8, using TS_l, $\Box_2$, and the fact that $\Box$ is retrieving, we then have to prove that $w_3 \Vdash !\delta(n) \vee \neg!\delta(n)$ given $w_3 \sqsupseteq w_2$, $w_3 \vdash n \Mapsto \underline{k}$, where $k \in \mathbb{N}$, and $w_3 \vdash !\delta(\underline{k}) \Mapsto b$, where $b \in \{\mathbb{1}, \mathbb{0}\}$. If b is $\mathbb{1}$, we straightforwardly get a proof of $w_3 \Vdash !\delta(n)$ using TS_r because $w_3 \vdash !\delta(n) \Mapsto \mathbb{1}$, which implies $w_3 \Vdash !\delta(n) \vee \neg!\delta(n)$. If b is $\mathbb{0}$, we straightforwardly get a proof of $w_3 \Vdash \neg!\delta(n)$, again using TS_r, which also implies $w_3 \Vdash !\delta(n) \vee \neg!\delta(n)$.

Step 3: Similar to Step 2 in Sec. 7.1, to prove $\neg\neg\exists n{:}\mathbb{N}.!\delta(n)$ we prove that $\neg\exists n{:}\mathbb{N}.!\delta(n)$ does not hold in any extension w_1 of the current world w. For this, we exhibit the world $w_2 := \mathsf{freeze}(\delta, \mathbb{1}, w_1)$ that extends w_1 with the choice $\mathbb{1}$. Given that choice, say at index k, it then suffices to show $w_2 \vdash !\delta(\underline{k}) \Mapsto \mathbb{1}$ to contradict $\neg\exists n{:}\mathbb{N}.!\delta(n)$.

Step 4: Similar to Step 3 in Sec. 7.1, to disprove $\exists n{:}\mathbb{N}.!\delta(n)$ it is enough to exhibit an infinite sequence of worlds extending w where the choice $\mathbb{1}$ is never made. For this we show that such a path from w exists where only the default choice $\mathbb{0}$ is ever made using in particular the fact that $\Box$ is choice-following (Def. 7.2).

8.2 Satisfying $\mathsf{MP}_{\mathbb{B}}$

We finally show that the above models allow satisfying $\mathsf{MP}_{\mathbb{B}}$, i.e., we prove $\forall(w : \mathcal{W}).w \Vdash \mathsf{MP}_{\mathbb{B}}$ (⚙).

For this, assume $f \in \mathbb{N} \to 2$ such that $\neg\neg\exists n{:}\mathbb{N}.f\ n = \mathbb{1}$, and prove $\exists n{:}\mathbb{N}.f\ n = \mathbb{1}$ in some world $w_1 \sqsupseteq w$. The proof is similar to the one for pure functions in Sec. 7.2 since choices are here $\mathbb{1}$ or $\mathbb{0}$, and therefore do not influence the way Boolean-valued functions compute in different worlds. More precisely, we prove that if a term computes to a Boolean in some world w', then it computes to that same Boolean in any world w'' since no $\mathrm{TT}^{\Box}_{C}$ computation acts on $\mathbb{1}$ or $\mathbb{0}$:

Lemma 8.1 (⚙). *For all $t, a, b \in \mathsf{Term}$ and $w_1, w_2 \in \mathcal{W}$, if $w_1 \vdash t \Mapsto \mathtt{inl}(a)$ and $w_2 \vdash t \Mapsto \mathtt{inr}(b)$ then $\bot$.*

The proof proceeds then as in Sec. 7.2 except that we use Lem. 8.1 instead of Lem. 7.3. For example, in Step 1, to prove that $w \Vdash f\ \underline{k} = \mathbb{1}$ is decidable for all $k \in \mathbb{N}$, in the case where $w' \vdash f\ \underline{k} \Mapsto \mathbb{1}$, we once again conclude that $w \vdash f\ \underline{k} \Mapsto \mathbb{1}$, using now Lem. 8.1.

8.3 Separation in MLTT

Note that $\mathsf{MP}_{\mathbb{P}}$ cannot be stated in MLTT, because MLTT does not allow capturing a computationally irrelevant *or*, and therefore does not allow stating $(\forall n.A\ n \vee \neg A\ n)$ because the natural formulation $(\forall n.A\ n + \neg A\ n)$ would be equivalent to the boolean form. Therefore, we introduce $\mathrm{MLTT}_{\downarrow}$ an extension of MLTT with anonymous existence, concretely with a form of weak propositional truncation, following [24]. We first extend MLTT's syntax as follows:

$$t ::= \cdots \mid \downarrow t \mid \downarrow\mathtt{intro}(t) \mid \downarrow\mathtt{elim}(t, t)$$

In addition, we extend its typing rules as follows:

$$\frac{\Gamma \vdash A}{\Gamma \vdash \downarrow A} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash \downarrow\mathtt{intro}(a) : \downarrow A} \qquad \frac{\Gamma \vdash f : A \to \downarrow B \quad \Gamma \vdash t : \downarrow A}{\Gamma \vdash \downarrow\mathtt{elim}(f, t) : \downarrow B}$$

We now extend the $\mathrm{TT}^{\Box}_{C}$ interpretation of MLTT as follows:

$$[\![\downarrow A]\!] = \downarrow[\![A]\!] \qquad [\![\downarrow\mathtt{intro}(a)]\!] = \star \qquad [\![\downarrow\mathtt{elim}(f, t)]\!] = \star$$

We can now express both $\mathsf{MP}_{\mathbb{P}}$ and $\mathsf{MP}_{\mathbb{B}}$ in $\mathrm{MLTT}_{\downarrow}$, and using the same method as in Sec. 7.3, we obtain that Sec. 8.1 and Sec. 8.2 entail that an immutable choice operator C, with $\mathbb{0}$ and $\mathbb{1}$ as choices, and a retrieving, choice-following, provide $\mathrm{MLTT}_{\downarrow}$ models that falsify $\mathsf{MP}_{\mathbb{P}}$ and satisfy $\mathsf{MP}_{\mathbb{B}}$.

9 CONCLUSION

We have presented an overview of proofs of equivalences to MP for three variants for MP, namely $\mathsf{MP}_{\mathbb{P}}$, $\mathsf{MP}_{\mathbb{B}}$, and $\mathsf{MP}_{\mathsf{PR}}$; we provided the first separation of $\mathsf{MP}_{\mathbb{P}}$ and $\mathsf{MP}_{\mathbb{B}}$, as well as $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$, for the $\mathrm{TT}^{\Box}_{C}$ theory; and finally applied those results to separate those three variants of MP for MLTT. We conjecture that our models and effectful techniques could be used to separate variants of other axioms such as LPO in the future.

ACKNOWLEDGMENTS

We have posed the question how to separate $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$ in type theory to many people over the years, and want to thank Thierry Coquand, Hugo Herbelin, Hajime Ishihara, Pierre-Marie Pédrot, and Gert Smolka for discussions and useful hints about directions to chase. Furthermore, Martín Escardó provided useful feedback on earlier versions of this paper and the idea to extend the question to $\mathsf{MP}_{\mathbb{P}}$ came up in a discussion with Dirk Pattinson. Thank you!

Liron Cohen was partially supported by Grant No. 2020145 from the United States-Israel Binational Science Foundation (BSF). Yannick Forster was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101024493. Dominik Kirst was supported by a Minerva Fellowship of the Minerva Stiftung Gesellschaft für die Forschung mbH.

REFERENCES

[1] Andreas Abel, Joakim Öhman, and Andrea Vezzosi. 2018. Decidability of conversion for type theory in type theory. *Proc. ACM Program. Lang.* 2, POPL (2018), 23:1–23:29. https://doi.org/10.1145/3158111

[2] Josef Berger, Hajime Ishihara, and Peter Schuster. 2012. The weak König lemma, Brouwer's fan theorem, De Morgan's law, and dependent choice. *Reports on Mathematical Logic* 47 (2012), 63. https://doi.org/10.4467/20842589RM.12.003.0684

[3] Douglas Bridges and Fred Richman. 1987. *Varieties of constructive mathematics.* Vol. 97. Cambridge University Press. https://doi.org/10.1017/CBO9780511565663

[4] Liron Cohen, Bruno da Rocha Paiva, Vincent Rahli, and Ayberk Tosun. 2023. Inductive Continuity via Brouwer Trees. In *MFCS (LIPIcs, Vol. 272).* Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 37:1–37:16. https://doi.org/10.4230/LIPICS.MFCS.2023.37

[5] Liron Cohen and Vincent Rahli. 2022. Constructing Unprejudiced Extensional Type Theories with Choices via Modalities. In *FSCD (LIPIcs, Vol. 228).* Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 10:1–10:23. https://doi.org/10.4230/LIPIcs.FSCD.2022.10

[6] Liron Cohen and Vincent Rahli. 2023. Realizing Continuity Using Stateful Computations. In *CSL (LIPIcs, Vol. 252)*, Bartek Klin and Elaine Pimentel (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 15:1–15:18. https://doi.org/10.4230/LIPIcs.CSL.2023.15

[7] Thierry Coquand and Bassel Mannaa. 2017. The Independence of Markov's Principle in Type Theory. *Log. Methods Comput. Sci.* 13, 3 (2017). https://doi.org/10.23638/LMCS-13(3:10)2017

[8] Hannes Diener. 2020. Constructive Reverse Mathematics. *arXiv:1804.05495 [math]* (2020). arXiv:1804.05495 [math]

[9] Peter Dybjer and Anton Setzer. 1999. A Finite Axiomatization of Inductive-Recursive Definitions. In *TLCA (LNCS, Vol. 1581).* Springer, 129–146. https://doi.org/10.1007/3-540-48959-2_11

[10] Yannick Forster. 2021. *Computability in Constructive Type Theory.* Ph. D. Dissertation. Saarland University. https://doi.org/10.22028/D291-35758

[11] Yannick Forster, Dominik Kirst, and Gert Smolka. 2019. On synthetic undecidability in Coq, with an application to the Entscheidungsproblem. In *CPP*. ACM Press. https://doi.org/10.1145/3293880.3294091

[12] Yannick Forster, Dominik Kirst, and Dominik Wehr. 2020. Completeness Theorems for First-Order Logic Analysed in Constructive Type Theory. In *International Symposium on Logical Foundations of Computer Science*. Springer, 47–74. https://doi.org/10.1007/978-3-030-36755-8_4

[13] Yannick Forster, Dominik Kirst, and Dominik Wehr. 2021. Completeness theorems for first-order logic analysed in constructive type theory (extended version). *Journal of Logic and Computation* 31, 1 (Jan. 2021), 112–151. https://doi.org/10.1093/logcom/exaa073

[14] Yannick Forster and Gert Smolka. 2017. Weak Call-by-Value Lambda Calculus as a Model of Computation in Coq. In *ITP (Lecture Notes in Computer Science, Vol. 10499)*, Mauricio Ayala-Rincón and César A. Muñoz (Eds.). Springer, 189–206. https://doi.org/10.1007/978-3-319-66107-0_13

[15] Michael P. Fourman. 1982. Notions of Choice Sequence. In *The L. E. J. Brouwer Centenary Symposium*, A.S. Troelstra and D. van Dalen (Eds.). Studies in Logic and the Foundations of Mathematics, Vol. 110. Elsevier, 91–105. https://doi.org/10.1016/S0049-237X(09)70125-9

[16] Von Kurt Gödel. 1958. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica* 12, 3-4 (Dec. 1958), 280–287. https://doi.org/10.1111/j.1746-8361.1958.tb01464.x

[17] Hugo Herbelin. 2010. An Intuitionistic Logic that Proves Markov's Principle. In *2010 25th Annual IEEE Symposium on Logic in Computer Science*. IEEE. https://doi.org/10.1109/lics.2010.49

[18] Hugo Herbelin and Danko Ilik. 2016. An analysis of the constructive content of Henkin's proof of Gödel's completeness theorem. (2016). http://pauillac.inria.fr/~herbelin/articles/godel-completeness-draft16.pdf Draft.

[19] Hugo Herbelin and Dominik Kirst. 2023. New observations on the constructive content of first-order completeness theorems. In *29th International Conference on Types for Proofs and Programs TYPES 2023–Abstracts*.

[20] Hajime Ishihara. 2006. Reverse Mathematics in Bishop's Constructive Mathematics. *Philosophia Scientae* CS 6 (Sept. 2006), 43–59. https://doi.org/10.4000/philosophiascientiae.406

[21] Stephen Cole Kleene. 1936. General recursive functions of natural numbers. *Mathematische annalen* 112, 1 (1936), 727–742.

[22] Stephen C. Kleene and Richard E. Vesley. 1965. *The Foundations of Intuitionistic Mathematics, especially in relation to recursive functions*. North-Holland Publishing Company.

[23] Alexei Kopylov. 2004. *Type Theoretical Foundations for Data Structures, Classes, and Objects*. Ph. D. Dissertation. Cornell University, Ithaca, NY.

[24] Nicolai Kraus, Martín Escardó, Thierry Coquand, and Thorsten Altenkirch. 2017. Notions of Anonymous Existence in Martin-Löf Type Theory. *Logical Methods in Computer Science* 13 (2017).

[25] Georg Kreisel. 1958. The Non-derivability of $\neg(x)A(x) \rightarrow (\exists x)\neg(Ax), A(x)$ primitive recursive, in intuitionistic formal systems (abstract). *Jour. Symb. Logic, 23(4):456–457* (1958).

[26] Georg Kreisel. 1958. A remark on free choice sequences and the topological completeness proofs. *The Journal of Symbolic Logic* 23, 4 (1958), 369–388.

[27] Georg Kreisel. 1962. On Weak Completeness of Intuitionistic Predicate Logic. *J. Symb. Log.* 27, 2 (1962), 139–158. https://doi.org/10.2307/2964110

[28] Georg Kreisel. 1965. Mathematical logic. *Lectures in modern mathematics* 3 (1965), 95–195. https://doi.org/10.2307/2315573

[29] Georg Kreisel. 1970. Church's thesis: a kind of reducibility axiom for constructive mathematics. In *Studies in Logic and the Foundations of Mathematics*. Vol. 60. 121–150. https://doi.org/10.1016/S0049-237X(08)70746-8

[30] Georg Kreisel and Anne S. Troelstra. 1970. Formal systems for some branches of intuitionistic analysis. *Annals of Mathematical Logic* 1, 3 (1970), 229–387. https://doi.org/10.1016/0003-4843(70)90001-X

[31] Saul A. Kripke. 1963. Semantical Analysis of Modal Logic I. Normal Propositional Calculi. *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik* 9, 5-6 (1963), 67–96. https://doi.org/10.1002/malq.19630090502

[32] Saul A. Kripke. 1965. Semantical Analysis of Intuitionistic Logic I. In *Formal Systems and Recursive Functions*, J.N. Crossley and M.A.E. Dummett (Eds.). Studies in Logic and the Foundations of Mathematics, Vol. 40. Elsevier, 92–130. https://doi.org/10.1016/S0049-237X(08)71685-9

[33] Saunders Mac Lane and Ieke Moerdijk. 1992. *Sheaves in Geometry and Logic. A First Introduction to Topos Theory*. Springer.

[34] Paul F. Mendler. 1988. *Inductive Definition in Type Theory*. Ph. D. Dissertation. Cornell University, Ithaca, NY.

[35] Joan R. Moschovakis. 1993. An intuitionistic theory of lawlike, choice and lawless sequences. In *Logic Colloquium'90: ASL Summer Meeting in Helsinki*. Association for Symbolic Logic, 191–209.

[36] John Myhill. 1963. The invalidity of Markoff's schema. *Mathematical Logic Quarterly* 9, 23 (1963).

[37] nLab authors. 2024. Markov's principle. Revision 17.

[38] Ulf Norell. 2007. *Towards a practical programming language based on dependent type theory*. Vol. 32. Chalmers University of Technology.

[39] Ulf Norell, Nils Anders Danielsson, Jesper Cockx, and Andreas Abel. [n. d.]. Agda Wiki. http://wiki.portal.chalmers.se/agda/pmwiki.php.

[40] Per Martin-Löf (notes by Giovanni Sambin). 1984. *Intuitionistic type theory*. Vol. 9. Bibliopolis Naples.

[41] Joakim Ohman, Andrea Vezzosi, and Andreas Abel. 2023. *A Logical Relation for Martin-Löf Type Theory in Agda*. https://github.com/mr-ohman/logrel-mltt

[42] Erik Palmgren. 1997. Constructive Sheaf Semantics. *Math. Log. Q.* 43 (1997), 321–327. https://doi.org/10.1002/MALQ.19970430304

[43] Christine Paulin-Mohring. 2015. Introduction to the Calculus of Inductive Constructions. https://hal.inria.fr/hal-01094195

[44] Pierre-Marie Pédrot. 2020. Russian Constructivism in a Prefascist Theory. In *LICS*, Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller (Eds.). ACM, 782–794. https://doi.org/10.1145/3373718.3394740

[45] Pierre-Marie Pédrot and Nicolas Tabareau. 2020. The fire triangle: how to mix substitution, dependent elimination, and effects. *Proc. ACM Program. Lang.* 4, POPL (2020), 58:1–58:28. https://doi.org/10.1145/3371126

[46] Pierre-Marie Pédrot and Nicolas Tabareau. 2018. Failure is Not an Option. In *European Symposium on Programming*. Springer, 245–271. https://doi.org/10.1007/978-3-319-89884-1_9

[47] Andrew M Pitts. 2013. Nominal Sets: Names and Symmetry in Computer Science, volume 57 of Cambridge Tracts in Theoretical Computer Science.

[48] Emil L. Post. 1944. Recursively enumerable sets of positive integers and their decision problems. *bulletin of the American Mathematical Society* 50, 5 (1944), 284–316. https://doi.org/10.1090/S0002-9904-1944-08111-1

[49] Pierre-Marie Pédrot. 2024. On the logical structure of choice and bar induction principles. In *39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Talinn, Estonia, July 8 - July 11, 2024*. ACM. https://doi.org/10.1145/3661814.3662070

[50] Ian Shillito and Dominik Kirst. 2024. A Mechanised and Constructive Reverse Analysis of Soundness and Completeness of Bi-intuitionistic Logic. In *CPP*, Amin Timany, Dmitriy Traytel, Brigitte Pientka, and Sandrine Blazy (Eds.). ACM, 218–229. https://doi.org/10.1145/3636501.3636957

[51] Stephen George Simpson. 2009. *Subsystems of second order arithmetic*. Vol. 1. Cambridge University Press. https://doi.org/10.1007/978-3-642-59971-2

[52] Craig A Smorynski. 1973. Applications of Kripke models. In *Number 344 in Lecture notes in mathematics*. Springer, 324–391.

[53] Thomas Streicher. 2009. Forcing for IZF in Sheaf Toposes. *Georgian Mathematical Journal* 16, 1 (2009), 203–209. https://doi.org/doi:10.1515/GMJ.2009.203

[54] The Coq Development Team. 2023. *The Coq Proof Assistant*. https://doi.org/10.5281/zenodo.8161141

[55] Anne S. Troelstra. 1977. *Choice sequences: a chapter of intuitionistic mathematics*. Clarendon Press Oxford.

[56] Anne S. Troelstra. 1985. Choice Sequences and Informal Rigour. *Synthese* 62, 2 (1985), 217–227.

[57] Anne S. Troelstra and Dirk van Dalen. 1988. Constructivism in mathematics. Vol. I. *Studies in Logic and the Foundations of Mathematics* 26 (1988).

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

[59] Mark van Atten. 2004. *On Brouwer*. Cengage Learning.

[60] Mark van Atten and Dirk van Dalen. 2002. Arguments for the continuity principle. *Bulletin of Symbolic Logic* 8, 3 (2002), 329–347. http://www.math.ucla.edu/~asl/bsl/0803/0803-001.ps

[61] Gerrit Van Der Hoeven and Ieke Moerdijk. 1984. Sheaf models for choice sequences. *Annals of Pure and Applied Logic* 27, 1 (1984), 63–107. https://doi.org/10.1016/0168-0072(84)90035-6

[62] Wim Veldman. 2001. Understanding and Using Brouwer's Continuity Principle. In *Reuniting the Antipodes — Constructive and Nonstandard Views of the Continuum*. Synthese Library, Vol. 306. Springer Netherlands, 285–302. https://doi.org/10.1007/978-94-015-9757-9_24

[63] Wikipedia contributors. 2023. Markov's principle — Wikipedia, The Free Encyclopedia. [Online; accessed 19-January-2024], https://en.wikipedia.org/w/index.php?title=Markov%27s_principle&oldid=1186833796.

A EQUIVALENT FORMULATIONS OF MP

In Sec. 2 we have already discussed that for foundations both supporting propositional existence $\exists$ and computational dependent pairing Σ, guarded minimisation operators can be implemented, and thus $\exists n.\ fn = \text{true}$ and $\Sigma n.\ fn = \text{true} \wedge \forall m < n.\ fm = \text{false}$ are interchangeable.

Thus, in particular $\exists$ and Σ can be interchanged in the formulations of $\mathsf{MP}_{\mathbb{B}}$ and $\mathsf{MP}_{\mathsf{PR}}$. They can, however, *not* be interchanged in the definition of $\mathsf{MP}_{\mathbb{P}}$, because $\mathsf{MP}_{\mathbb{P}}$ is a consequence of the law

$$\frac{}{\vdash \bullet} \qquad \frac{\vdash \Gamma \quad \Gamma \vdash A}{\vdash \Gamma, x : A} \qquad \frac{\vdash \Gamma}{\Gamma \vdash \mathbb{U}} \qquad \frac{\Gamma \vdash A : \mathbb{U}}{\Gamma \vdash A} \qquad \frac{\vdash \Gamma \quad (x : A) \in \Gamma}{\Gamma \vdash A}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash t : B}{\Gamma, x : A \vdash t : B} \qquad \frac{\Gamma \vdash A \quad \Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A.t : \Pi x{:}A.B} \qquad \frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B}{\Gamma \vdash \Pi x{:}A.B}$$

$$\frac{\Gamma \vdash t : \Pi x{:}A.B \quad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B[x \backslash u]} \qquad \frac{\Gamma \vdash A : \mathbb{U} \quad \Gamma, x : A \vdash B : \mathbb{U}}{\Gamma \vdash \Pi x{:}A.B : \mathbb{U}}$$

$$\frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B \quad \Gamma \vdash t : A \quad \Gamma \vdash u : B[x \backslash t]}{\Gamma \vdash \langle t, u \rangle : \Sigma x{:}A.B}$$

$$\frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B}{\Gamma \vdash \Sigma x{:}A.B} \qquad \frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B \quad \Gamma \vdash t : \Sigma x{:}A.B}{\Gamma \vdash \mathtt{fst}(t) : A}$$

$$\frac{\Gamma \vdash A : \mathbb{U} \quad \Gamma, x : A \vdash B : \mathbb{U}}{\Gamma \vdash \Sigma x{:}A.B : \mathbb{U}} \qquad \frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B \quad \Gamma \vdash t : \Sigma x{:}A.B}{\Gamma \vdash \mathtt{snd}(t) : B[x \backslash \mathtt{fst}(t)]}$$

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathbb{1} : \mathbb{U}} \qquad \frac{\vdash \Gamma}{\Gamma \vdash \mathbb{0} : \mathbb{U}} \qquad \frac{}{\Gamma \vdash \star : \mathbb{1}} \qquad \frac{\Gamma \vdash e : \mathbb{0} \quad \Gamma \vdash A}{\Gamma \vdash \mathbb{0}_{\mathsf{ind}}\ A\ e : A}$$

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathbb{N} : \mathbb{U}} \qquad \frac{\begin{array}{l}\Gamma, x : \mathbb{N} \vdash P \qquad \Gamma \vdash z : P[x \backslash \mathsf{0}] \\ \Gamma \vdash s : \Pi y{:}\mathbb{N}.P[x \backslash y] \to P[x \backslash \mathsf{S}\ y] \qquad \Gamma \vdash n : \mathbb{N}\end{array}}{\Gamma \vdash \mathbb{N}_{\mathsf{ind}}\ P\ z\ s\ n : P[x \backslash n]}$$

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathsf{0} : \mathbb{N}} \qquad \frac{\Gamma \vdash t : \mathbb{N}}{\Gamma \vdash \mathsf{S}\ t : \mathbb{N}} \qquad \frac{\Gamma \vdash t : A \quad \Gamma \vdash B \quad \Gamma \vdash A \equiv B}{\Gamma \vdash t : B}$$

Figure 3: MLTT's Typing rules

of excluded middle, and a Σ version of $\mathsf{MP}_{\mathbb{P}}$ together with excluded middle proves the axiom of countable choice – but countable choice is not a consequence of the law of excluded middle.

Theorem A.1. *The following are all equivalent to* $\mathsf{MP}_{\mathbb{B}}$*:*

$$\forall f : \mathbb{N} \to \mathbb{B}.\ \neg\neg(\Sigma n.\ fn = \mathsf{true}) \to (\Sigma n.\ fn = \mathsf{true})$$
$$\forall f : \mathbb{N} \to \mathbb{B}.\ \neg\neg(\exists n.\ fn = \mathsf{true}) \to (\Sigma n.\ fn = \mathsf{true})$$
$$\forall f : \mathbb{N} \to \mathbb{B}.\ \ \neg(\forall n.\ fn = \mathsf{false}) \to (\exists n.\ fn = \mathsf{true})$$
$$\forall f : \mathbb{N} \to \mathbb{B}.\ \ \neg(\forall n.\ fn = \mathsf{false}) \to (\Sigma n.\ fn = \mathsf{true})$$

and the same for the corresponding variants with f *being primitive-recursive or* TM*-computable.*

Regarding minimality, we can state all versions of Markov's principle with a minimality condition:

Theorem A.2.

$$\mathsf{MP}_{\mathbb{P}} \leftrightarrow \forall A : \mathbb{N} \to \mathbb{P}.\ (\forall n.\ An \vee \neg An) \to \neg\neg(\exists n.\ An) \to \exists n.\ An \wedge \forall m < n.\ \neg Am$$
$$\mathsf{MP}_{\mathbb{B}} \leftrightarrow \forall f : \mathbb{N} \to \mathbb{B}.\neg\neg(\exists n.\ fn = \mathsf{true}) \to \exists n.\ fn = \mathsf{true} \wedge \forall m < n.\ fm = \mathsf{false}$$
$$\mathsf{MP}_{\mathsf{PR}} \leftrightarrow \forall f : \mathbb{N} \to \mathbb{B}.\ \textit{primitive-recursive}\ f \to \neg\neg(\exists n.\ fn = \mathsf{true}) \to \exists n.\ fn = \mathsf{true} \wedge \forall m < n.\ fm = \mathsf{false}$$

Lastly, using $\mathbb{B}$ is not crucial for $\mathsf{MP}_{\mathbb{B}}$, $\mathsf{MP}_{\mathsf{TM}}$, and $\mathsf{MP}_{\mathsf{PR}}$.

Theorem A.3. *We have that*

$$\mathsf{MP}_{\mathbb{B}} \leftrightarrow \exists f : \mathbb{N} \to \mathbb{N}.\ \neg\neg(\exists n.\ fn = 0) \to \exists n.\ fn = 0$$

and the same for f *being* TM*-computable or primitive recursive.*

B MLTT'S TYPING AND CONVERSION RULES

Fig. 3 presents MLTT's typing rules, while Fig. 4 presents its conversion rules. As mentioned in Thm. 6.1, all these rules are satisfied by the semantics presented in Sec. 5.6, Sec. 5.7, and Sec. 6, which consists in a translation of MLTT into $\mathrm{TT}^{\square}_{C}$ composed with an interpretation of $\mathrm{TT}^{\square}_{C}$.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t \equiv t : A} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \equiv A} \qquad \frac{\Gamma \vdash A \equiv B : \mathbb{U}}{\Gamma \vdash A \equiv B} \qquad \frac{\Gamma \vdash b \equiv a : A}{\Gamma \vdash a \equiv b : A} \qquad \frac{\Gamma \vdash B \equiv A}{\Gamma \vdash A \equiv B} \qquad \frac{\Gamma \vdash a \equiv b : A \quad \Gamma \vdash b \equiv c : A}{\Gamma \vdash a \equiv c : A}$$

$$\frac{\Gamma \vdash a \equiv b : A \quad \Gamma \vdash A \equiv B}{\Gamma \vdash a \equiv b : B} \qquad \frac{\Gamma \vdash A \equiv B \quad \Gamma \vdash B \equiv C}{\Gamma \vdash A \equiv C} \qquad \frac{\Gamma \vdash A \quad \Gamma \vdash f : \Pi x{:}A.B \quad \Gamma \vdash g : \Pi x{:}A.B \quad \Gamma, x : A \vdash f\ x \equiv g\ x : B}{\Gamma \vdash f \equiv g : \Pi x{:}A.B}$$

$$\frac{\Gamma \vdash A \equiv C : \mathbb{U} \quad \Gamma, x : A \vdash B \equiv D : \mathbb{U}}{\Gamma \vdash \Pi x{:}A.B \equiv \Pi x{:}C.D : \mathbb{U}} \qquad \frac{\Gamma \vdash A \equiv C \quad \Gamma, x : A \vdash B \equiv D}{\Gamma \vdash \Pi x{:}A.B \equiv \Pi x{:}C.D}$$

$$\frac{\Gamma \vdash A \quad \Gamma, x : A \vdash t : B \quad \Gamma \vdash u : A}{\Gamma \vdash (\lambda x : A.t)\ u \equiv t[x \backslash u] : B[x \backslash u]} \qquad \frac{\Gamma \vdash f \equiv g : \Pi x{:}A.B \quad \Gamma \vdash a \equiv b : A}{\Gamma \vdash f\ a \equiv g\ b : B[x \backslash a]}$$

$$\frac{\Gamma \vdash A \equiv C : \mathbb{U} \quad \Gamma, x : A \vdash B \equiv D : \mathbb{U}}{\Gamma \vdash \Sigma x{:}A.B \equiv \Sigma x{:}C.D : \mathbb{U}} \qquad \frac{\Gamma \vdash A \equiv C \quad \Gamma, x : A \vdash B \equiv D}{\Gamma \vdash \Sigma x{:}A.B \equiv \Sigma x{:}C.D}$$

$$\frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B \quad \Gamma \vdash a : A \quad \Gamma \vdash b : B[x \backslash a]}{\Gamma \vdash \mathtt{fst}(\langle a, b \rangle) \equiv a : A} \qquad \frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B \quad \Gamma \vdash a : A \quad \Gamma \vdash b : B[x \backslash a]}{\Gamma \vdash \mathtt{snd}(\langle a, b \rangle) \equiv b : B[x \backslash a]}$$

$$\frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B \quad \Gamma \vdash a : A \quad \Gamma \vdash a \equiv b : \Sigma x{:}A.B}{\Gamma \vdash \mathtt{fst}(a) \equiv \mathtt{fst}(b) : A} \qquad \frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B \quad \Gamma \vdash a : A \quad \Gamma \vdash a \equiv b : \Sigma x{:}A.B}{\Gamma \vdash \mathtt{snd}(a) \equiv \mathtt{snd}(b) : B[x \backslash \mathtt{fst}(a)]}$$

$$\frac{\Gamma \vdash A \quad \Gamma, x : A \vdash B \quad \Gamma \vdash a : \Sigma x{:}A.B \quad \Gamma \vdash b : \Sigma x{:}A.B \quad \Gamma \vdash \mathtt{fst}(a) \equiv \mathtt{fst}(b) : A \quad \Gamma \vdash \mathtt{snd}(a) \equiv \mathtt{snd}(b) : B[x \backslash \mathtt{fst}(a)]}{\Gamma \vdash a \equiv b : \Sigma x{:}A.B}$$

$$\frac{\Gamma \vdash n \equiv m : \mathbb{N}}{\Gamma \vdash \mathtt{S}\ n \equiv \mathtt{S}\ b : \mathbb{N}} \qquad \frac{\begin{array}{l}\Gamma, x : \mathbb{N} \vdash P_1 \equiv P_2 \qquad \Gamma \vdash z_1 \equiv z_2 : P_1[x \backslash \mathtt{0}] \\ \Gamma \vdash s_1 \equiv s_2 : \Pi y{:}\mathbb{N}.P_1[x \backslash y] \to P_1[x \backslash \mathtt{S}\ y] \\ \Gamma \vdash n_1 \equiv n_2 : \mathbb{N}\end{array}}{\Gamma \vdash \mathbb{N}_{\mathtt{ind}}\ P_1\ z_1\ s_1\ n_1 \equiv \mathbb{N}_{\mathtt{ind}}\ P_2\ z_2\ s_2\ n_2 : P[x \backslash n_1]}$$

$$\frac{\begin{array}{l}\Gamma, x : \mathbb{N} \vdash P \qquad \Gamma \vdash z : P[x \backslash \mathtt{0}] \\ \Gamma \vdash s : \Pi y{:}\mathbb{N}.P[x \backslash y] \to P[x \backslash \mathtt{S}\ y]\end{array}}{\Gamma \vdash \mathbb{N}_{\mathtt{ind}}\ P\ z\ s\ \mathtt{0} \equiv z : P[x \backslash \mathtt{0}]} \qquad \frac{\begin{array}{l}\Gamma, x : \mathbb{N} \vdash P \qquad \Gamma \vdash z : P[x \backslash \mathtt{0}] \\ \Gamma \vdash s : \Pi y{:}\mathbb{N}.P[x \backslash y] \to P[x \backslash \mathtt{S}\ y] \\ \Gamma \vdash n : \mathbb{N}\end{array}}{\Gamma \vdash \mathbb{N}_{\mathtt{ind}}\ P\ z\ s\ (\mathtt{S}\ n) \equiv s\ n\ \mathbb{N}_{\mathtt{ind}}\ P\ z\ s\ n : P[x \backslash \mathtt{S}\ n]}$$

Figure 4: MLTT's conversion rules