Logical Methods in Computer Science
Volume 20, Issue 1, 2024, pp. 19:1–19:24
https://lmcs.episciences.org/

Submitted Mar. 07, 2023
Published Mar. 07, 2024

AN ANALYSIS OF TENNENBAUM'S THEOREM IN CONSTRUCTIVE TYPE THEORY *

MARC HERMES [a,b] AND DOMINIK KIRST [b]

[a] Radboud University Nijmegen, Netherlands
e-mail address: Marc.Hermes@cs.ru.nl

[b] Universität des Saarlandes, Saarbrücken, Germany
e-mail address: kirst@ps.uni-saarland.de

Abstract. Tennenbaum's theorem states that the only countable model of Peano arithmetic (PA) with computable arithmetical operations is the standard model of natural numbers. In this paper, we use constructive type theory as a framework to revisit, analyze and generalize this result.

The chosen framework allows for a synthetic approach to computability theory, exploiting that, externally, all functions definable in constructive type theory can be shown computable. We then build on this viewpoint, and furthermore internalize it by assuming a version of Church's thesis, which expresses that any function on natural numbers is representable by a formula in PA. This assumption provides for a conveniently abstract setup to carry out rigorous computability arguments, even in the theorem's mechanization.

Concretely, we constructivize several classical proofs and present one inherently constructive rendering of Tennenbaum's theorem, all following arguments from the literature. Concerning the classical proofs in particular, the constructive setting allows us to highlight differences in their assumptions and conclusions which are not visible classically. All versions are accompanied by a unified mechanization in the Coq proof assistant.

1. Introduction

There are well-known proofs in classical logic showing that the first-order theory of Peano arithmetic (PA) has non-standard models, meaning models which are not isomorphic to the standard model $\mathbb{N}$. One way to do so (*cf.* [BBJ02]), starts by adding a new constant symbol c to the language of PA, together with the enumerable list of new axioms $c \neq 0$, $c \neq 1$, $c \neq 2$ etc. This theory has the property that every finite subset of its axioms is satisfied by the standard model $\mathbb{N}$, since we can always give a large enough interpretation of the constant c in $\mathbb{N}$. Hence, by the compactness theorem, the full theory has a model $\mathcal{M}$, which must then be non-standard since the interpretation of c in $\mathcal{M}$ corresponds to an element which is larger than any number n in $\mathbb{N}$.

Key words and phrases: first-order logic, Peano arithmetic, Tennenbaum's theorem, constructive type theory, Church's thesis, synthetic computability, Coq.

* This is an extended version of a paper published at FSCD 2022 [HK22].

DOI:10.46298/LMCS-20(1:19)2024
© M. Hermes and D. Kirst
Creative Commons

The presence of non-standard elements like this has interesting consequences. PA can prove that for every bound n, sums of the form $\sum_{k \leq n} a_k$ exist, so in particular for example the Gaussian sum $\sum_{k \leq n} k$. The presence of the non-standard element c in $\mathcal{M}$ allows for the creation of infinite sums like $\sum_{k \leq c} k$. Remarkably, while this means that it includes a summation over all natural numbers, the model specifies a single element as the result of this infinite sum. A general PA model $\mathcal{M}$ therefore exhibits behaviors which disagree with the common intuition that computations in PA are finitary, which are—in the end—largely based on the familiarity with the standard model $\mathbb{N}$.

These intuitions are still not too far off the mark, as was demonstrated by Stanley Tennenbaum [Ten59] in a remarkable theorem. By being a little more restrictive on the models under consideration, $\mathbb{N}$ regains a unique position:

> **Tennenbaum's Theorem:** Apart from the standard model $\mathbb{N}$, there is no countable *computable* model of first-order PA.

A model is considered *computable* if its elements can be coded by numbers in $\mathbb{N}$, and the arithmetic operations on its elements can be realized by computable functions on these codes. Usually, Tennenbaum's theorem is formulated in a classical framework such as ZF set theory, and the precise meaning of *computable* is given by making reference to a concrete model of computation like Turing machines, μ-recursive functions, or the λ-calculus [Kay11, Smi14]. In a case like this, where computability theory is applied rather than developed, the computability of a function is rarely proven by exhibiting an explicit construction in the specific model, but rather by invoking the informal *Church-Turing thesis*, stating that every function intuitively computable is computable in the chosen model. Proving the computability of a function is then reduced to giving an intuitive justification.

The focus of this paper lies on revisiting Tennenbaum's theorem and several of its proofs in a constructive type theory (CTT). In contrast to classical treatments, the usage of a constructive meta-theory enables us to formally assume *Church's thesis* [Kre70, TvD88, For21a] in the form of an axiom, stating that every total function is computable. By its usage, the elegant and succinct paper-style computability proofs can be reproduced, but in a fully formal manner, also allowing for straightforward mechanized proofs.

In the constructive type theory that we will specify in Section 2, the addition of this axiom becomes possible since we can adapt the approach of *synthetic computability* [Ric83, Bau06, FKS19]: Any function term that is definable in CTT by the virtue of its syntactic rules, can externally be observed to be a computable function. Following through on this external observation, it can be taken as a justification to also internally treat functions as if they were computable. For example, we will make use of this perspective when defining a predicate on a type X to be *decidable* if there exists a function $f : X \to \mathbb{B}$ computing booleans which reflect the truth values of p (Definition 2.4).

This approach leads to a simplification when it comes to the statement of Tennenbaum's theorem itself: By interpreting arithmetic operations as type-theoretic functions in CTT, all models are automatically computable in the synthetic sense. Consequently, "computable model" no longer needs to be part of the theorem statement. Moreover, the given synthetic proofs unveil the computational essence of Tennenbaum's theorem with neither the technical overhead of constructions in a formal model of computability nor the informal practice to simply disregard computability arguments. The paper therefore contributes to an active line of research on mechanized metamathematics [Kir22] with the two-fold goal to obtain a uniform and constructive development of the foundations of mathematics, complemented by a principled mechanization for guaranteed correctness and collaborative use.

In the above sketched framework, we follow the classical presentations of Tennenbaum's theorem [Kay11, Smi14] and develop constructive versions that only assume a type-theoretic version of *Markov's principle* [MC17]. This is then complemented by the adaption of an inherently more constructive variant given by McCarty [McC87, McC88].

Concretely, our contributions can be summarized as follows:

- We review several existing proofs of Tennenbaum's theorem drawn from the literature, and carry them out in a constructive meta-theory. We work out subtle differences in the strengths of their conclusions, which are left invisible in any classical treatment, but come to light once they are viewed through a constructive lens.
- By considering models with a decidable divisibility relation (Theorem 7.10), we extend the theorem to models which do not have to be discrete or enumerable.
- We provide a Coq mechanization covering all results presented in this paper.[1]

The present paper is an extended version of [HK22] and adds the following contributions:

- In [HK22], we only gave a reference to a possible proof strategy for showing the existence of HA-inseparable formulas (Definition 7.11). A comment in [Pet22] pointed to a more straightforward approach, which we were able to mechanize, and have added to Section 7.3.
- We added a short discussion in Section 7.4, in which we aim to abstractly identify the main ingredients that are used in Section 7.3 to derive Tennenbaum's theorem.
- Our Coq mechanization has been re-based and now relies on a collaborative Coq library for first-order logic [KHD$^+$22], to which we have contributed by integrating the mechanization developed for this publication. This integration not only enabled the utilization of pre-existing definitions for Δ_1 and Σ_1–formulas but also provides additional validation for our application of Church's thesis for Robinson arithmetic ($\mathsf{CT_Q}$), given the library includes its derivation from a more conventional formulation of Church's thesis [KP23].
- The presentation of several proofs, definitions and theorem statements in Section 5 and Section 7 has been revised. Additionally, a mistake in [HK22] has been corrected; the original version of HA-coding (*cf.* [HK22] Hypothesis 2) was not constructively provable, the new one (Hypothesis 7.20) is.

To conclude this introduction, we give a brief overview on the structure of the paper, in the order that we consider most suitable for a first reading:

The main results of the analysis are summarized in Section 8, where we give a tabulated overview on the different variants of Tennenbaum's theorem that result from the various proofs. It clarifies which assumptions are made for each version, and we give a brief discussion of what to take from these differences. The complete proofs are covered in Section 7, ending with Section 7.4 in which we abstractly capture the essence of Tennenbaum's theorem.

In Section 6 we motivate and introduce our chosen formulation of Church's thesis which is utilized as an axiom. Basic results about PA's standard and non-standard models are shown in Section 4 and then used in Section 5 to establish results that allow the encoding of predicates on $\mathbb{N}$, which are essential in the proof of Tennenbaum's theorem.

To make the paper self-contained, we also give an introduction to the essential features of constructive type theory, synthetic computability, and the type-theoretic specification of first-order logic in Section 2. This is continued in Section 3 by the presentation of the first-order axiomatization of PA as given in previous work [KH21, KH23].

[1]The full mechanization is available on GitHub [Her23] and can conveniently be viewed on a webpage [HK23]. We make use of a fact for which we gave no mechanized proof, namely Hypothesis 7.20 in Section 7.3.

2. Preliminaries

2.1. **Constructive Type Theory.** The chosen framework for this paper is a constructive type theory (CTT). More specifically, it will be the calculus of inductive constructions (CIC) [CH88, PM93] which is implemented in the Coq proof assistant [Coq23]. It provides a predicative hierarchy of *type universes* above a single impredicative universe $\mathbb{P}$ of *propositions* and the capability of inductive type definitions. On the type level, we have the unit type $\mathbb{1}$ with a single element, the void type $\mathbb{0}$, function spaces $X \to Y$, products $X \times Y$, sums $X + Y$, dependent products[2] $\forall (x : X).\, A\, x$, and dependent sums $\Sigma (x : X).\, A\, x$. On the propositional level, analogous notions to the one listed for types in the above are present, but denoted by their usual logical notation ($\top$, $\bot$, $\to$, $\wedge$, $\vee$, $\forall$, $\exists$).[3] It is important to note that the so-called *large eliminations* from the impredicative $\mathbb{P}$ into higher types of the hierarchy are restricted. In particular, it is generally not possible to show $(\exists x.\, p\, x) \to \Sigma x.\, p\, x$.[4] The restriction does however allow for large elimination of the equality predicate $=$ of type $\forall X.\, X \to X \to \mathbb{P}$, as well as function definitions by well-founded recursion.

We will also use the basic inductive types of *Booleans* ($\mathbb{B} := \mathsf{tt} \mid \mathsf{ff}$), *Peano natural numbers* ($n : \mathbb{N} := 0 \mid n{+}1$), the *option type* ($\mathcal{O}(X) := {}^{\circ}x \mid \emptyset$) and *lists* ($l : \mathsf{List}(X) := [\,] \mid x{::}l$). Furthermore, by X^n we denote the type of *vectors* $\vec{v}$ of length $n : \mathbb{N}$ over X.

Given predicates $P, Q : X \to \mathbb{P}$ on a type X, we will occasionally use the set notation $P \subseteq Q$ for expressing $\forall x : X.\, P\, x \to Q\, x$.

Definition 2.1. *A proposition $P : \mathbb{P}$ is called* definite *if $P \vee \neg P$ holds and* stable *if $\neg\neg P \to P$. The same terminology is used for predicates $p : X \to \mathbb{P}$ given they are* pointwise definite *or* stable*. We furthermore want to recall the following logical principles:*

$$\mathsf{LEM} := \forall P : \mathbb{P}.\, \mathsf{definite}\, P \qquad \textit{(Law of Excluded Middle)}$$
$$\mathsf{DNE} := \forall P : \mathbb{P}.\, \mathsf{stable}\, P \qquad \textit{(Double-Negation Elimination)}$$
$$\mathsf{MP} := \forall f : \mathbb{N} \to \mathbb{N}.\, \mathsf{stable}\, (\exists n.\, f n = 0) \qquad \textit{(Markov's Principle)}$$

Note that LEM and DNE are equivalent and not provable in CIC, while MP is much weaker and has a constructive interpretation [MC17]. For convenience, and as used by Bauer [Bau08], we adapt the reading of double negated statements like $\neg\neg P$ as "*potentially P*".[5]

Remark 2.2 (Handling $\neg\neg$). *Given any propositions A, B we constructively have the equivalence $(A \to \neg B) \leftrightarrow (\neg\neg A \to \neg B)$, meaning that when trying to prove a negated goal, we can remove double negations in front of any assumption. More generally, any statement of the form $\neg\neg A_1 \to \ldots \to \neg\neg A_n \to \neg\neg C$ is equivalent to $A_1 \to \ldots \to A_n \to \neg\neg C$ and since $C \to \neg\neg C$ holds, it furthermore suffices to show $A_1 \to \ldots \to A_n \to C$ in this case. In the following, we will make use of these facts without further notice.*

[2] As is custom in Coq, we write $\forall$ in place of the symbol Π for dependent products.

[3] Negation $\neg A$ is used as an abbreviation for both $A \to \bot$ and $A \to \mathbb{0}$.

[4] The direction $(\Sigma x.\, p\, x) \to \exists x.\, p\, x$ is however always provable. Intuitively, one can think of $\exists x.\, p\, x$ as stating the mere existence of a value satisfying p, while $\Sigma x.\, p\, x$ is a type that also carries a value satisfying p.

[5] A referee pointed out to us that $\neg\neg P$ is closely related to $\Box\Diamond P$ in modal logic, where $\Box$ stands for *necessity* and $\Diamond$ for *possibility*. The statement can therefore also be read as "by necessity, P must be possible".

2.2. **Synthetic Computability.** As already expressed in Section 1, constructive type theory permits us to take a viewpoint that considers all functions to be computable functions, yielding simple definitions [FKS19] of many textbook notions of computability theory:

Definition 2.3 (Enumerability). *Let $p : X \to \mathbb{P}$ be some predicate. We say that p is* enumerable *if there is an* enumerator *$f : \mathbb{N} \to \mathcal{O}(X)$ such that $\forall x : X.\, p\, x \leftrightarrow \exists n.\, f n = {}^\circ x$.*

Definition 2.4 (Decidability). *Let $p : X \to \mathbb{P}$ be some predicate. We call $f : X \to \mathbb{B}$ a* decider *for p and write $\mathsf{decider}\, p\, f$ iff $\forall x : X.\, p\, x \leftrightarrow f x = \mathsf{tt}$. We then define the following notions of decidability:*

- $\mathsf{Dec}\, p := \exists f : X \to \mathbb{B}.\, \mathsf{decider}\, p\, f$
- $\mathsf{dec}(P : \mathbb{P}) := P + \neg P$.

In both cases we will often refer to the predicate or proposition simply as being decidable.

We also expand the synthetic vocabulary with notions for types. In the textbook setting, many of them can only be defined for sets which are in bijection with $\mathbb{N}$, but synthetically they can be handled in a very uniform way.

Definition 2.5. *We call a type X*

- enumerable *if $\lambda x : X.\top$ is enumerable,*
- discrete *if there exists a decider for equality $=$ on X,*
- separated *if there exists a decider for apartness $\neq$ on X,*
- witnessing *if $\forall f : X \to \mathbb{B}.\, (\exists x.\, f x = \mathsf{tt}) \to \Sigma x.\, f x = \mathsf{tt}$.*

Fact 2.6. *In* CIC*, the types $\mathbb{N}$ and $\mathbb{N}^2$ are witnessing.* □

2.3. **First-Order Logic.** In order to study Tennenbaum's theorem, we need to give a description of the first-order theory of PA and the associated intuitionistic theory of *Heyting arithmetic* (HA), which has the same axiomatization, but uses intuitionistic first-order logic. We follow prior work in [FKS19, FKW21, KH21] and describe first-order logic as embedded inside the constructive type theory, by inductively defining formulas, terms, and the deduction system. We then define a semantics for this logic, which uses Tarski models and interprets formulas over the respective domain of the model. The type of natural numbers $\mathbb{N}$ will then naturally be a model of HA.

Before specializing to one particular theory, we keep the definition of first-order logic general and fix some arbitrary signature $\Sigma = (\mathcal{F}; \mathcal{P})$ for function and predicate symbols.

Definition 2.7 (Terms and Formulas). *We define terms $t : \mathsf{tm}$ and formulas $\varphi : \mathsf{fm}$ inductively:*

$$s, t : \mathsf{tm} ::= x_n \mid f\, \vec{v} \qquad (n : \mathbb{N},\ f : \mathcal{F},\ \vec{v} : \mathsf{tm}^{|f|})$$

$$\alpha, \beta : \mathsf{fm} ::= \bot \mid P\, \vec{v} \mid \alpha \to \beta \mid \alpha \wedge \beta \mid \alpha \vee \beta \mid \forall\, \alpha \mid \exists\, \beta \qquad (P : \mathcal{P},\ \vec{v} : \mathsf{tm}^{|P|}),$$

where $|f|$ and $|P|$ are the arities of the function symbol f and predicate symbol P, respectively.

We use de Bruijn indexing to formalize the binding of variables to quantifiers. This means that the variable x_n at some position in a formula is *bound* to the n-th quantifier preceding this variable in the syntax tree of the formula. If there is no quantifier binding the variable, it is said to be *free*.

Definition 2.8 (Substitution). *Given a variable assignment $\sigma : \mathbb{N} \to \mathsf{tm}$ we recursively define* substitution *on terms by $x_k[\sigma] := \sigma\, k$ and $f\,\vec{v} := f(\vec{v}[\sigma])$, and extended to formulas by*

$$\bot[\sigma]{:=}\bot \qquad (P\,\vec{v})[\sigma]{:=}P\,(\vec{v}[\sigma]) \qquad (\alpha \,\dot{\square}\, \beta)[\sigma]{:=}\alpha[\sigma] \,\dot{\square}\, \beta[\sigma] \qquad (\dot{\nabla}\,\varphi)[\sigma]{:=}\dot{\nabla}(\varphi[x_0; \lambda x.(\sigma x)[\uparrow]])$$

where $\dot{\square}$ is any logical connective and $\dot{\nabla}$ any quantifier. The variable assignment $x;\sigma$ is defined by $(x;\sigma)\,0{:=}x$ as well as $(x;\sigma)(n{+}1){:=}\sigma\, n$ and simply appends x as the first element to $\sigma : \mathbb{N} \to \mathsf{tm}$. By $\uparrow$ we designate the assignment $\lambda n.\, x_{n+1}$ shifting all variable indices by one.

Definition 2.9 (Natural Deduction). *Natural deduction $\vdash : \mathsf{List}(\mathsf{fm}) \to \mathsf{fm} \to \mathbb{P}$ is characterized inductively by the usual rules (see Appendix A). We write $\vdash$ for intuitionistic natural deduction and $\vdash_c$ for the classical variant, which extends $\vdash$ by adding every instance of Peirce's law $((\varphi \to \psi) \to \varphi) \to \varphi$.*[6]

Definition 2.10 (Tarski Semantics). *A* model *$\mathcal{M}$ consists of a type D designating its domain together with functions $f^{\mathcal{M}} : D^{|f|} \to D$ and $P^{\mathcal{M}} : D^{|P|} \to \mathbb{P}$ for all symbols f in $\mathcal{F}$ and P in $\mathcal{P}$. We will also use $\mathcal{M}$ to refer to the domain. Functions $\rho : \mathbb{N} \to \mathcal{M}$ are called environments and are used as variable assignments to recursively give evaluations to terms:*

$$\hat{\rho}\, x_k := \rho\, k \qquad \hat{\rho}\,(f\,\vec{v}) := f^{\mathcal{M}}(\hat{\rho}\,\vec{v}) \qquad (v : \mathsf{tm}^n)$$

This interpretation is then extended to formulas via the satisfaction relation:

$$\begin{aligned} \mathcal{M} \vDash_\rho P\,\vec{v} &:= P^{\mathcal{M}}(\hat{\rho}\,\vec{v}) & \mathcal{M} \vDash_\rho \alpha \to \beta &:= \mathcal{M} \vDash_\rho \alpha \to \mathcal{M} \vDash_\rho \beta \\ \mathcal{M} \vDash_\rho \alpha \wedge \beta &:= \mathcal{M} \vDash_\rho \alpha \wedge \mathcal{M} \vDash_\rho \beta & \mathcal{M} \vDash_\rho \alpha \vee \beta &:= \mathcal{M} \vDash_\rho \alpha \vee \mathcal{M} \vDash_\rho \beta \\ \mathcal{M} \vDash_\rho \forall\, \alpha &:= \forall x : D.\ \mathcal{M} \vDash_{x;\rho} \alpha & \mathcal{M} \vDash_\rho \exists\, \alpha &:= \exists x : D.\ \mathcal{M} \vDash_{x;\rho} \alpha \end{aligned}$$

We say that a formula φ holds in the model *$\mathcal{M}$ and write $\mathcal{M} \vDash \varphi$ if for every ρ we have $\mathcal{M} \vDash_\rho \varphi$. We extend this notation to theories $\mathcal{T} : \mathsf{fm} \to \mathbb{P}$ by writing $\mathcal{M} \vDash \mathcal{T}$ iff $\forall \varphi.\, \mathcal{T}\,\varphi \to \mathcal{M} \vDash \varphi$, and we write $\mathcal{T} \vDash \varphi$ if $\mathcal{M} \vDash \varphi$ for all models $\mathcal{M}$ with $\mathcal{M} \vDash \mathcal{T}$. Given a $e : \mathcal{M}$, we also use the notation $\mathcal{M} \vDash \varphi(e)$ for all ρ we have $\mathcal{M} \vDash_{e;\rho} \varphi$.*

Fact 2.11 (Soundness). *For any formula φ and theory $\mathcal{T}$, if $\mathcal{T} \vdash \varphi$ then $\mathcal{T} \vDash \varphi$.* □

From the next section on, we will use conventional notation with named variables instead of explicitly writing formulas with de Bruijn indices.

3. Axiomatization of Peano Arithmetic and Heyting Arithmetic

We present PA following [KH21], as a first-order theory with a signature consisting of symbols for the constant zero, the successor function, addition, multiplication and equality:

$$\Sigma_{\mathsf{PA}} := (\mathcal{F}_{\mathsf{PA}}; \mathcal{P}_{\mathsf{PA}}) = (0,\, S, +, \times; =)$$

The finite core of PA axioms consists of statements characterizing the successor function, as well as addition and multiplication:

$$\begin{aligned} \text{Disjointness: } & \forall x.\, Sx = 0 \to \bot & \text{Injectivity: } & \forall xy.\, Sx = Sy \to x = y \\ +\text{-base: } & \forall x.\, 0 + x = x & +\text{-recursion: } & \forall xy.\, (Sx) + y = S(x + y) \\ \times\text{-base: } & \forall x.\, 0 \times x = 0 & \times\text{-recursion: } & \forall xy.\, (Sx) \times y = y + x \times y \end{aligned}$$

[6] Another way to treat the distinction between classical and intuitionistic theories would be to add all instances of Peirce's law to the axioms of a theory, instead of building them into the deduction system.

We then get the full (and infinite) axiomatization of PA with the axiom scheme of induction for unary formulas. In our meta-theory the schema is a type-theoretic function on formulas:

$$\lambda\varphi.\,\varphi[0] \to (\forall\, x.\,\varphi[x] \to \varphi[Sx]) \to \forall\, x.\,\varphi[x]$$

If instead of the induction scheme we add the axiom $\forall\, x.\, x = 0 \vee \exists\, y.\, x = Sy$, we get the theory Q known as *Robinson arithmetic*. Both PA and Q also contain axioms for equality:

$$\begin{aligned}
\text{Reflexivity:}\quad & \forall\, x.\, x = x \\
\text{Symmetry:}\quad & \forall\, xy.\, x = y \to y = x \\
\text{Transitivity:}\quad & \forall\, xyz.\, x = y \to y = z \to x = z \\
S\text{-equality:}\quad & \forall\, xy.\, x = y \to Sx = Sy \\
+\text{-equality:}\quad & \forall\, xyuv.\, x = u \to y = v \to x + y = u + v \\
\times\text{-equality:}\quad & \forall\, xyuv.\, x = u \to y = v \to x \times y = u \times v
\end{aligned}$$

Using classical derivability $\vdash_c$ we get the classical first-order theory of Peano arithmetic $\mathsf{PA} \vdash_c$. Its intuitionistic counterpart $\mathsf{PA} \vdash$ uses intuitionistic derivability $\vdash$ and is called *Heyting arithmetic*. Given that the constructive type theory selected for this work only provides a model for the intuitionistic theory, we will restrict our focus on Heyting arithmetic. To emphasize this, we will from now on write HA instead of PA.

For simplicity, we only consider models that interpret the equality symbol with the actual equality relation of its domain, so-called *extensional* models. In the Coq development we make the equality symbol a syntactic primitive, therefore enabling the convenient behavior that the interpreted equality reduces to actual equality.

Definition 3.1. *We recursively define a function* $\overline{\cdot} : \mathbb{N} \to \mathsf{tm}$ *by* $\overline{0} := 0$ *and* $\overline{n+1} := S\overline{n}$*, giving every natural number a representation as a term. Any term* t *which is of the form* $\overline{n}$ *will be called* numeral.

We furthermore use notations for expressing *less than* $x < y := \exists\, k.\, S(x + k) = y$, *less than or equal* $x \leq y := \exists\, k.\, x + k = y$ and for *divisibility* $x \mid y := \exists\, k.\, x \times k = y$.

The formulas of HA can be classified in a hierarchy based on their computational properties. We will only consider two levels of this hierarchy:

Definition 3.2 (Δ_1 and Σ_1–formulas (*cf.* [KP23])). *A formula* φ *is* Δ_1 *if for every assignment* σ *that only substitutes closed terms, we have* $\mathsf{Q} \vdash \varphi[\sigma]$ *or* $\mathsf{Q} \vdash \neg\varphi[\sigma]$*. A formula is* Σ_1 *if it is of the form* $\exists x_1 \ldots \exists x_n.\, \varphi_0$*, where* φ_0 *is a* Δ_1*–formula.*

Given a Σ_1–formula $\exists x_1 \ldots \exists x_n.\varphi$ where φ is Δ_1, we can prove it equivalent to the formula $\exists x\, \exists x_1 < x \ldots \exists x_n < x.\, \varphi$. Since Δ_1–formulas stay Δ_1 if they are closed by a bounded quantifier, this shows that the initial Σ_1–formula can be written as a Δ_1–formula, which is preceded by exactly one existential quantifier. We will occasionally make use of this fact and refer to it as Σ_1*–compression*. A more syntactic definition of Δ_1 would characterize them as the formulas which are equivalent to both a Π_1 and Σ_1–formula. For our purposes the definition which only stipulates the necessary decidability properties is sufficient, as it implies the absoluteness and completeness properties we will need [KP23]:

Fact 3.3 (Σ_1–Completeness). *For any* Σ_1*–formula* φ *we have* $\mathbb{N} \vDash \varphi$ *iff* $\mathsf{HA} \vdash \varphi$*.* □

Fact 3.4 (Δ_1–Absoluteness). *Let* $\mathcal{M} \vDash \mathsf{HA}$ *and* φ *be any closed* Δ_1*–formula, then we have* $\mathbb{N} \vDash \varphi \to \mathcal{M} \vDash \varphi$*.* □

4. Standard and Non-standard Models of HA

From now on $\mathcal{M}$ will always designate a HA model. Any model like this has an interpretation $0^{\mathcal{M}}$ of the zero symbol, as well as an interpretation $S^{\mathcal{M}} : \mathcal{M} \to \mathcal{M}$ of the symbol for the successor. By repeated application of $S^{\mathcal{M}}$ we can therefore get the sequence of elements $0^{\mathcal{M}}, S^{\mathcal{M}}0^{\mathcal{M}}, S^{\mathcal{M}}S^{\mathcal{M}}0^{\mathcal{M}}, \ldots$, essentially giving us a copy of the standard numbers inside $\mathcal{M}$. We will now put this intuition more formally.

Fact 4.1. *We recursively define a function $\nu : \mathbb{N} \to \mathcal{M}$ by $\nu\, 0 := 0^{\mathcal{M}}$ and $\nu\,(n+1) := S^{\mathcal{M}}(\nu\ n)$. Furthermore, we define the predicate $\mathsf{std} := \lambda e.\, \exists n.\, \nu\, n = e$ and refer to e as a* standard number *if $\mathsf{std}\, e$ and* non-standard *if $\neg\, \mathsf{std}\, e$. We then have*

(1) *$\hat{\rho}\, \overline{n} = \nu\, n$ for any $n : \mathbb{N}$ and environment $\rho : \mathbb{N} \to \mathcal{M}$.*
(2) *ν is an injective homomorphism and therefore an* embedding *of $\mathbb{N}$ into $\mathcal{M}$.*

Both facts are taken as justification to abuse notation and also write $\overline{n}$ for $\nu\, n$. □

Usually we would have to write $0^{\mathcal{M}}, S^{\mathcal{M}}, +^{\mathcal{M}}, \times^{\mathcal{M}}, =^{\mathcal{M}}$ for the interpretations of the respective symbols in a model $\mathcal{M}$. For better readability we will however take the freedom to overload the symbols $0, S, +, \cdot, =$ to also refer to these interpretations.

Definition 4.2. *$\mathcal{M}$ is called a* standard model *if there is a bijective homomorphism $\varphi : \mathbb{N} \to \mathcal{M}$. We will accordingly write $\mathcal{M} \cong \mathbb{N}$ if this is the case.*

We can show that ν is essentially the only homomorphism from $\mathbb{N}$ to $\mathcal{M}$ we need to worry about, since it is unique up to extensional equality of functions:

Lemma 4.3. *Let $\varphi : \mathbb{N} \to \mathcal{M}$ be a homomorphism, then $\forall x : \mathbb{N}.\, \varphi\, x = \nu\, x$.*

Proof. By induction on x and using the fact that both are homomorphisms. □

We now have two equivalent ways to express standardness of a model.

Lemma 4.4. *$\mathcal{M} \cong \mathbb{N}$ iff $\forall e : \mathcal{M}.\, \mathsf{std}\, e$.*

Proof. Given $\mathcal{M} \cong \mathbb{N}$, there is an isomorphism $\varphi : \mathbb{N} \to \mathcal{M}$. Since φ is surjective, Lemma 4.3 implies that ν must also be surjective. For the converse: if ν is surjective, it is an isomorphism since it is injective by Fact 4.1. □

Having seen that every model contains a unique embedding of $\mathbb{N}$, one may wonder whether there is a formula φ which could define and pick out precisely the standard numbers in $\mathcal{M}$. Lemma 4.5 gives a negative answer to this question:

Lemma 4.5. *There is a unary formula $\varphi(x)$ with $\forall e : \mathcal{M}.\, \big(\mathsf{std}\, e \leftrightarrow \mathcal{M} \vDash \varphi(e)\big)$ if and only if $\mathcal{M} \cong \mathbb{N}$.*

Proof. Given a formula φ with the stated property, we certainly have $\mathcal{M} \vDash \varphi(\overline{0})$ since $\overline{0}$ is a standard number, and clearly $\mathcal{M} \vDash \varphi(x) \implies \mathsf{std}\, x \implies \mathsf{std}\,(Sx) \implies \mathcal{M} \vDash \varphi(Sx)$. Thus, by induction in the model, we have $\mathcal{M} \vDash \forall x.\, \varphi(x)$, which is equivalent to $\forall e : \mathcal{M}.\, \mathsf{std}\, e$. The converse implication holds by choosing the formula $x = x$. □

We now turn our attention to models which are not isomorphic to $\mathbb{N}$.

Fact 4.6. *For any $e : \mathcal{M}$, we have $\neg\, \mathsf{std}\, e$ iff $\forall n : \mathbb{N}.\ e > \overline{n}$.* □

Definition 4.7. *Founded on the result of Fact 4.6 we write $e > \mathbb{N}$ iff $\neg\, \mathsf{std}\, e$ and call $\mathcal{M}$*

- non-standard *(written $\mathcal{M} > \mathbb{N}$) iff there is $e : \mathcal{M}$ such that $e > \mathbb{N}$,*

- not standard *(written $\mathcal{M} \ncong \mathbb{N}$) iff* $\neg\mathcal{M} \cong \mathbb{N}$.

We will also write $e{:}\mathcal{M} > \mathbb{N}$ to express that e is a non-standard element in $\mathcal{M}$.

Of course, we have $\mathcal{M} > \mathbb{N} \to \mathcal{M} \ncong \mathbb{N}$, but the converse implication does not hold constructively in general, so the distinction of both notions becomes meaningful.

Lemma 4.8 (Overspill). *If $\mathcal{M} \ncong \mathbb{N}$ and $\varphi(x)$ is unary with $\mathcal{M} \vDash \varphi(\overline{n})$ for every $n{:}\mathbb{N}$, then*

(1) $\neg\big(\forall e{:}\mathcal{M}.\, \mathcal{M} \vDash \varphi(e) \to \mathsf{std}\, e\big)$
(2) $\mathsf{stable}\ \mathsf{std} \to \neg\neg\,\exists\, e > \mathbb{N}.\, \mathcal{M} \vDash \varphi(e)$
(3) $\mathsf{DNE} \to \exists\, e > \mathbb{N}.\, \mathcal{M} \vDash \varphi(e)$.

Proof. (1) Assuming $\forall e{:}\mathcal{M}.\ \mathcal{M} \vDash \varphi(e) \to \mathsf{std}\, e$ and combining it with our assumption that φ holds on all numerals, Lemma 4.5 implies $\mathcal{M} \cong \mathbb{N}$, giving us a contradiction. For (2) note that we constructively have that $\neg\exists e{:}\mathcal{M}.\, \neg\mathsf{std}\, e \wedge \mathcal{M} \vDash \varphi(e)$ implies $\forall e{:}\mathcal{M}.\, \mathcal{M} \vDash \varphi(e) \to \neg\neg\,\mathsf{std}\, e$, and by using the stability of std we therefore get a contradiction in the same way as in (1). Statement (3) immediately follows from (2). □

From Lemma 4.8 we learn that under certain conditions, whenever a formula is satisfied on all standard numbers $\overline{n}$, this satisfaction "spills over" into the non-standard part of the model, meaning there is a non-standard element which also satisfies the formula. In the next section, we will encounter our first application of this principle.

5. Coding Finite and Infinite Predicates

There is a standard way in which finite sets of natural numbers can be encoded by a single natural number. Assuming we have some injective function $\pi : \mathbb{N} \to \mathbb{N}$ whose image consists only of prime numbers, and given a finite set of numbers like $S := \{4, 13, 21, 33\}$, we can encode this set by the single number $c := \pi_4 \cdot \pi_{13} \cdot \pi_{21} \cdot \pi_{33}$. It then satisfies $n \in S \leftrightarrow \pi_n \mid c$, allowing us to reconstruct S by checking which primes divide c.

Instead of applying this to sets, we can also use it to encode bounded portions of predicates on $\mathbb{N}$.

Lemma 5.1. *Given $n{:}\mathbb{N}$ and any predicate $p : \mathbb{N} \to \mathbb{P}$ with $\forall x < n.\, p\,x \vee \neg\, p\,x$, we have*

$$\exists c{:}\mathbb{N}\ \forall u{:}\mathbb{N}.\big(u < n \to (p\,u \leftrightarrow \pi_u \mid c)\big) \wedge \big(\pi_u \mid c \to u < n\big)$$

The right part of the conjunction assures that no primes above π_n end up in the code c.

Proof. We do a proof by induction on n. For $n = 0$ we can choose $c{:=}1$. In the induction step, the induction hypothesis gives us a code $c{:}\mathbb{N}$ which codes p up to n. Since by assumption, p is definite below Sn, we know that $p\,n \vee \neg p\,n$, allowing us to consider two cases: If $p\,n$, we set the new code to be $c' := c \cdot \pi_n$, if $\neg p\,n$ we simply set $c' := c$. In both cases one can now verify that c' will correctly code p up to Sn. □

Corollary 5.2 (Finite Coding in $\mathbb{N}$). *Given any $p : \mathbb{N} \to \mathbb{P}$ and bound $n{:}\mathbb{N}$, we have*

$$\neg\neg\,\exists c{:}\mathbb{N}\ \forall u{:}\mathbb{N}.\big(u < n \to (p\,u \leftrightarrow \pi_u \mid c)\big) \wedge \big(\pi_u \mid c \to u < n\big)$$

Note that if p is definite, we can drop the $\neg\neg$.

Proof. If p is definite, we trivially have $\forall x < n.\, p\,x \vee \neg\, p\,x$, so Lemma 5.1 gives us the $\neg\neg$–free existence as claimed. Without assuming definiteness, we can still constructively show $\neg\neg(\forall x < n.\, p\,x \vee \neg\, p\,x)$ by induction on n, which combined with Lemma 5.1 gives us the existence, but behind a double negation. □

With a proof of the encoding in $\mathbb{N}$ we can give a straightforward proof that this is possible in any model of HA.

Remark 5.3. *To formulate the above result in a generic model $\mathcal{M} \vDash \mathsf{HA}$, we require an object-level representation of the prime function π. For now, we will simply assume that we have such a binary formula $\Pi(x, y)$ and defer the justification to Section 6.*

The statement "π_u divides c" can now be expressed by $\exists\, p.\ \Pi(u, p) \wedge p \mid c$, for which we will abuse notation and simply write $\Pi(u) \mid c$.

Lemma 5.4 (Finite Coding in $\mathcal{M}$). *For any binary formula $\alpha(x, y)$ and $n : \mathbb{N}$ we have*

$$\mathcal{M} \vDash \forall\, e\, \neg\neg\, \exists\, c\, \forall\, u < \overline{n}.\ \alpha(u, e) \leftrightarrow \Pi(u) \mid c.$$

Proof. Let $e : \mathcal{M}$, and define the predicate $p := \lambda u : \mathbb{N}.\, \mathcal{M} \vDash \alpha(\overline{u}, e)$. Then Corollary 5.2 potentially gives us a code $a : \mathbb{N}$ for p up to the bound n. It now suffices to show that the actual existence of $a : \mathbb{N}$ already implies

$$\mathcal{M} \vDash \exists\, c\, \forall\, u < \overline{n}.\ \alpha(u, e) \leftrightarrow \Pi(u) \mid c.$$

And indeed, we can verify that $c = \overline{a}$ shows the existential claim: given $u : \mathcal{M}$ with $\mathcal{M} \vDash u < \overline{n}$ we can conclude that u must be a standard number $\overline{u}$. We then have the equivalences

$$\mathcal{M} \vDash \alpha(\overline{u}, e) \iff p\, u \iff \pi_u \mid a \iff \mathcal{M} \vDash \Pi(\overline{u}) \mid \overline{a}$$

since a codes p and Π represents π. □

Overspill now has interesting consequences when it comes to encoding, as for models that are not standard, it allows for the potential encoding of a complete predicate $p : \mathbb{N} \to \mathbb{P}$, and therefore also of infinite subsets.

Lemma 5.5 (Infinite Coding in $\mathcal{M}$). *If* std *is stable, $\mathcal{M} \not\cong \mathbb{N}$ and $\alpha(x)$ unary, we have*

$$\neg\neg\, \exists c : \mathcal{M}\ \forall u : \mathbb{N}.\ \mathcal{M} \vDash \alpha(\overline{u}) \leftrightarrow \Pi(\overline{u}) \mid c.$$

Proof. Using Lemma 5.4 for the present case where α is unary, we get

$$\mathcal{M} \vDash \neg\neg\, \exists\, c\, \forall\, u < \overline{n}.\ \alpha(u) \leftrightarrow \Pi(u) \mid c$$

for every $n : \mathbb{N}$, so by Lemma 4.8 (Overspill) we get

$$\begin{aligned} &\neg\neg\, \exists\, e > \mathbb{N}.\ \mathcal{M} \vDash \neg\neg\, \exists\, c\, \forall\, u < e.\ \alpha(u) \leftrightarrow \Pi(u) \mid c \\ \implies &\neg\neg\, \exists c : \mathcal{M}\, \forall u : \mathbb{N}.\ \mathcal{M} \vDash \alpha(\overline{u}) \leftrightarrow \Pi(\overline{u}) \mid c, \end{aligned}$$

where we used that since the equivalence holds for all $u < e$ with e non-standard, it will in particular hold for all $u : \mathbb{N}$. □

Lemma 5.6. *If* std *is stable and $\mathcal{M} \not\cong \mathbb{N}$, then for binary $\alpha(x, y)$ and $e : \mathcal{M}$ we have*

$$\neg\neg\, \exists c : \mathcal{M}\ \forall u : \mathbb{N}.\ \mathcal{M} \vDash \alpha(\overline{u}, e) \leftrightarrow \Pi(\overline{u}) \mid c.$$

Proof. Analogous to the proof of Lemma 5.5. □

These coding results allow us to connect a unary formula α to an element $c : \mathcal{M}$ of the model, in such a way that the decidability of the divisibility for c will entail the decidability of $\mathcal{M} \vDash \alpha(\overline{\cdot})$.

6. Church's Thesis for First-Order Arithmetic

Church's thesis (CT) is an axiom of constructive mathematics which states that every total function is computable. We will assume a version of it in this paper, since by its addition to the ambient type theory, we merely need to show that a function can be defined at all, to prove its computability. This makes it possible to stay completely formal, yet achieve a textbook-style conciseness for proofs involving computability, even in their mechanization.

This addition is not possible in every meta-theory. If we were to add it to ZF, it would immediately imply the computability of the function that solves the halting problem, leading to an inconsistent theory. In general however, theories that tend to the constructive side do allow for the consistent addition of this axiom. In the type theory we use in this paper, this is achieved by strictly distinguishing between functional relations and total functions. The aforementioned function that solves the halting problem in ZF can only be shown to be a functional relation, which means we can still safely assume total functions to be computable. Currently, there is no consistency proof for CT (*cf.* Section 7.1 in [For21b]) and the exact type theory we are using, but there are proofs showing that it can be consistently added to very similar systems [Yam20, SU19].

Since CT makes reference to *computability*, its exact form as an axiom does not only depend on the theory in which it is assumed, but also on the model of computation it makes reference to. Robinson's Q, a finitely axiomatized arithmetical system, is expressive enough to serve as a computational model, and is a particularly well-suited choice in our case, leading us to the following formulation of CT which we assume for the remainder of the paper:

Axiom 6.1 (CT_Q). *For every function $f : \mathbb{N} \to \mathbb{N}$ there exists a binary Σ_1–formula $\varphi_f(x, y)$ such that for every $n : \mathbb{N}$ we have $\mathsf{Q} \vdash \forall y.\, \varphi_f(\overline{n}, y) \leftrightarrow \overline{fn} = y$.*

Note that CT_Q can be derived from the more conventional version of Church's thesis for μ-recursive functions [KP23].[7] Using CT_Q we can get an internal representation φ_f of any computable function f, allowing us to reason about the function inside of first-order arithmetic. We also have an immediate use-case for CT_Q, since applying it to the injective prime function π lets us settle the earlier Remark 5.3:

Fact 6.2. *There is a binary Σ_1–formula representing the injective prime function π in Q.* □

In Section 2.1 we defined decidable and enumerable predicates in a synthetic way, but using CT_Q we can now give characterizations and representations of such predicates by formulas in Q (*cf.* [Raa21]).

Definition 6.3. *We call $p : \mathbb{N} \to \mathbb{P}$* weakly representable *if there is a Σ_1–formula $\varphi_p(x)$ such that $\forall n : \mathbb{N}.\, p\, n \leftrightarrow \mathsf{Q} \vdash \varphi_p(\overline{n})$, and* strongly representable *if instead, $p\, n \to \mathsf{Q} \vdash \varphi_p(\overline{n})$ and $\neg p\, n \to \mathsf{Q} \vdash \neg\varphi_p(\overline{n})$ hold for every $n : \mathbb{N}$.*

Lemma 6.4 (Representability Theorem). *Assume CT_Q, and let $p : \mathbb{N} \to \mathbb{P}$ be given.*

(1) *If p is decidable, it is strongly representable.*
(2) *If p is enumerable, it is weakly representable.*

[7] In [KP23], the abbreviation CT_Q was used to refer to a version of Church's thesis which applies to partial functions. From the partial version, the version for total functions can be derived, and in this paper we use CT_Q to refer to the latter, total version.

Proof. If p is decidable, then there is a decider $\mathbb{N} \to \mathbb{B}$ which can be used to define a function $f : \mathbb{N} \to \mathbb{N}$ such that $\forall x : \mathbb{N}.\, p\, x \leftrightarrow f x = 0$, and by $\mathsf{CT_Q}$ there is a binary Σ_1–formula $\varphi_f(x, y)$ representing f. We then define $\varphi_p(x) := \varphi_f(x, \overline{0})$ and deduce

$$\begin{aligned} p\, n &\implies f n = 0 \implies \mathsf{Q} \vdash \overline{f n} = \overline{0} \implies \mathsf{Q} \vdash \varphi_f(\overline{n}, \overline{0}) \implies \mathsf{Q} \vdash \varphi_p(\overline{n}) \\ \neg p\, n &\implies f n \neq 0 \implies \mathsf{Q} \vdash \neg(\overline{f n} = \overline{0}) \implies \mathsf{Q} \vdash \neg\varphi_f(\overline{n}, \overline{0}) \implies \mathsf{Q} \vdash \neg\varphi_p(\overline{n}) \end{aligned}$$

which shows that p is strongly representable.

If p is enumerable, then there is an enumerator $\mathbb{N} \to \mathcal{O}(\mathbb{N})$ which can be used to define a function $f : \mathbb{N} \to \mathbb{N}$ such that $\forall x : \mathbb{N}.\, p\, x \leftrightarrow \exists n.\, f n = x + 1$. By $\mathsf{CT_Q}$ there is a binary Σ_1–formula $\varphi_f(x, y)$ representing f. We then define $\varphi_p(x) := \exists\, n.\, \varphi_f(n, S x)$ giving us

$$\begin{aligned} \mathsf{Q} \vdash \varphi_p(\overline{x}) &\iff \mathsf{Q} \vdash \exists\, n.\, \varphi_f(n, S\overline{x}) \iff \exists n : \mathbb{N}.\, \mathsf{Q} \vdash \varphi_f(\overline{n}, S\overline{x}) \\ &\iff \exists n : \mathbb{N}.\, \mathsf{Q} \vdash \overline{f n} = S\overline{x} \iff \exists n : \mathbb{N}.\, f n = x + 1 \iff p\, x \end{aligned}$$

which shows that p is weakly representable by a Σ_1–formula. □

7. Tennenbaum's Theorem

With our choice of $\mathsf{CTT} + \mathsf{CT_Q}$ for the meta-theory in place, we now begin with our analysis of Tennenbaum's theorem. We will present several proofs of the theorem from the literature. In a classical meta-theory all of these proofs would yield the same result, but in our constructive setting, they turn out to differ in the strength of their assumptions and conclusions. Almost all the proofs will make use of some coding results for non-standard models from Section 5, enabling us to use a single model element to fully encode the standard part of any predicate $p : \mathcal{M} \to \mathbb{P}$.

For the proof in Section 7.1 we will assume enumerability of the model, enabling a very direct diagonal argument [BBJ02]. In Section 7.2 we look at the proof approach that is most prominently found in the literature [Smi14, Kay11] and uses the existence of recursively inseparable sets.

Another refinement of this proof was proposed in a post by Makholm [Mak14] and comes with the advantage that it circumvents the usage of Overspill. Strikingly, it turns out that in the constructive setting, this eliminates the necessity for MP, which is required for the standard proof using inseparable sets. Additionally, we look at the consequences of Tennenbaum's theorem once the underlying semantics is made explicitly constructive. The latter two variations are discussed in Section 7.3.

7.1. Via a Diagonal Argument.

We start by noting that every HA model can prove the most basic fact about division and remainders.

Lemma 7.1 (Euclid's Lemma). *We have*

$$\mathcal{M} \vDash \forall\, e\, d\ \exists\, r\, q.\ e = q \cdot d + r \ \wedge\ (0 < d \to r < d)$$

and the uniqueness property telling us that if $r_1, r_2 < d$ then $q_1 \cdot d + r_1 = q_2 \cdot d + r_2$ implies $q_1 = q_2$ and $r_1 = r_2$.

Proof. For Euclid's lemma, there is a standard proof by induction on e. The uniqueness claim requires some basic results about the strict order. □

Lemma 7.2. *If $\mathcal{M}$ is enumerable and discrete, then $\lambda(n{:}\mathbb{N})(d{:}\mathcal{M}).\,\mathcal{M} \vDash \overline{n} \,|\, d$ has a decider.*

Proof. Let $n{:}\mathbb{N}$ and $d{:}\mathcal{M}$ be given. By Lemma 7.1 we have $\exists q', r'{:}\mathcal{M}.\, d = q' \cdot \overline{n} + r'$. This existence is propositional, so presently we cannot use it to give a decision for $\overline{n} \mid d$. Since $\mathcal{M}$ is enumerable, there is a surjective function $g : \mathbb{N} \to \mathcal{M}$ and the above existence therefore shows $\exists q, r{:}\mathbb{N}.\, d = (g\,q) \cdot \overline{n} + (g\,r)$. Since equality is decidable in $\mathcal{M}$ and $\mathbb{N}^2$ is witnessing, we get $\Sigma q, r{:}\mathbb{N}.\, d = (g\,q) \cdot \overline{n} + (g\,r)$, giving us computational access to r, now allowing us to construct the decision. By the uniqueness part of Lemma 7.1 we have $g\,r = 0 \leftrightarrow \overline{n} \mid d$, so the decidability of $\overline{n} \mid d$ is entailed by the decidability of $g\,r = 0$. □

Lemma 7.3. (1) *If* std *is stable, then $\mathcal{M} \cong \mathbb{N}$ is stable.*
(2) *If* std *is stable, then $\neg\mathcal{M} > \mathbb{N}$ is equivalent to $\mathcal{M} \cong \mathbb{N}$.*
(3) *Assuming* MP*, if $\mathcal{M}$ is discrete, then* std *is stable.*

Proof. (1) is trivial by Lemma 4.4. (2) The implication $\forall e.\, \mathsf{std}\, e \to \neg\exists e.\, \neg\mathsf{std}\, e$ holds constructively, but the converse needs the stability of std. For (3), recall that $\mathsf{std}\, e$ stands for $\exists n{:}\mathbb{N}.\, \overline{n} = e$. Since $\overline{n} = e$ in $\mathcal{M}$ is decidable, stability follows from Fact 2.6. □

Lemma 7.4. *If* std *is stable, $\mathcal{M} \ncong \mathbb{N}$, and $p{:}\mathbb{N} \to \mathbb{P}$ decidable, then potentially there is a code $c{:}\mathcal{M}$ such that $\forall n{:}\mathbb{N}.\; p\,n \leftrightarrow \mathcal{M} \vDash \overline{\pi_n} \mid c$.*

Proof. By Lemma 6.4, there is a formula φ_p strongly representing p. Under the given assumptions, we can use the coding Lemma 5.5, yielding a code $c{:}\mathcal{M}$ for the formula φ_p, such that $\forall u{:}\mathbb{N}.\, \mathcal{M} \vDash \varphi_p(\overline{u}) \leftrightarrow \Pi(\overline{u}) \mid c$. Overall this shows:

$$\begin{aligned} p\,n &\implies \mathsf{Q} \vdash \varphi_p(\overline{n}) \implies \mathcal{M} \vDash \varphi_p(\overline{n}) \implies \mathcal{M} \vDash \Pi(\overline{n}) \mid c \\ \neg p\,n &\implies \mathsf{Q} \vdash \neg\,\varphi_p(\overline{n}) \implies \neg\,\mathcal{M} \vDash \varphi_p(\overline{n}) \implies \neg\,\mathcal{M} \vDash \Pi(\overline{n}) \mid c. \end{aligned}$$

Since p is decidable, the latter implication entails $\mathcal{M} \vDash \Pi(\overline{n}) \mid c \implies p\,n$, which overall shows the desired equivalence. □

This gives us the following version of Tennenbaum's theorem:

Theorem 7.5. *Assuming* MP *and discrete $\mathcal{M}$, enumerability of $\mathcal{M}$ implies $\mathcal{M} \cong \mathbb{N}$.*

Proof. By Lemma 7.3 it suffices to show $\neg\neg\mathcal{M} \cong \mathbb{N}$. So assume $\mathcal{M} \ncong \mathbb{N}$ and try to derive $\bot$. Given the enumerability, there is a surjective function $g{:}\mathbb{N} \to \mathcal{M}$, allowing us to define the predicate $p := \lambda n{:}\mathbb{N}.\, \neg\,\mathcal{M} \vDash \overline{\pi_n} \mid g\,n$, which is decidable by Lemma 7.2. By the coding result in Lemma 7.4 there is an $e{:}\mathcal{M}$ which codes p, and by the surjectivity of g, there is some $c{:}\mathbb{N}$ with $g\,c = e$. Combined, these facts give us

$$\neg\,\mathcal{M} \vDash \overline{\pi_c} \mid g\,c \overset{\text{def.}}{\iff} p\,c \overset{\text{coding}}{\iff} \mathcal{M} \vDash \overline{\pi_c} \mid g\,c$$

leading to the desired contradiction. □

7.2. **Via Inseparable Predicates.** The most frequently reproduced proof of Tennenbaum's theorem [Kay11, Smi14] uses the existence of recursively inseparable sets and non-standard coding to establish the existence of a non-recursive set.

Definition 7.6. *A pair $A, B : \mathbb{N} \to \mathbb{P}$ of predicates is called* inseparable *if they are disjoint and $A \subseteq D$, $B \subseteq \neg D$ implies the undecidability of D.*

Lemma 7.7. *There are inseparable enumerable predicates $A, B : \mathbb{N} \to \mathbb{P}$.*

Proof. We use an enumeration $\Phi_n : \mathsf{fm}$ of formulas to define predicates $A{:=}\lambda n{:}\mathbb{N}.\, \mathsf{Q} \vdash \neg\, \Phi_n(\overline{n})$ and $B := \lambda n{:}\mathbb{N}.\, \mathsf{Q} \vdash \Phi_n(\overline{n})$, which are disjoint since $\mathsf{Q} \nvdash \bot$, and enumerable since proofs over Q can be enumerated. Given a predicate D with $A \subseteq D$, $B \subseteq \neg D$ and assuming it were decidable, Lemma 6.4 gives us a formula strongly representing D, and by the enumeration there is $d{:}\mathbb{N}$ such that Φ_d is said formula. Everything together gives us the following chain of implications:

$$D\, d \overset{\text{s.repr.}}{\Longrightarrow} \mathsf{Q} \vdash \Phi_d(\overline{d}) \overset{\text{def.}}{\Longrightarrow} B\, d \overset{\subseteq}{\Longrightarrow} \neg D\, d \overset{\text{s.repr.}}{\Longrightarrow} \mathsf{Q} \vdash \neg\Phi_d(\overline{d}) \overset{\text{def.}}{\Longrightarrow} A\, d \overset{\subseteq}{\Longrightarrow} D\, d$$

Since this shows $D\, d \Longleftrightarrow \neg D\, d$, we must conclude that D is undecidable. □

Corollary 7.8. *There is a pair α, β of unary Σ_1–formulas such that $A := \lambda n{:}\mathbb{N}.\, \mathsf{Q} \vdash \alpha(\overline{n})$ and $B := \lambda n{:}\mathbb{N}.\, \mathsf{Q} \vdash \beta(\overline{n})$ are inseparable and enumerable.*

Proof. Use weak representability (Lemma 6.4) on the predicates given by Lemma 7.7. □

Contrary to the proof delineated in Section 7.1, the alternative proof via inseparable sets eliminates the need for enumerability of the model. Furthermore, we will now factor Tennenbaum's theorem into two parts, mirroring what Kaye does in [Kay11]. He first shows that for any non-standard model $\mathcal{M}$, a non-recursive coded set exists within $\mathcal{M}$, and separately to this, he establishes that computability of the model would entail the recursiveness of all coded sets. Combining both of these results then yields Tennenbaum's theorem. In the following we will focus on the first part, and establish the existence of an undecidable predicate, which is coded by an element of $\mathcal{M}$. Our coding here concretely uses prime numbers, but in Section 7.4, we will revisit a more abstract perspective on the coding.

Definition 7.9. *For $d{:}\mathcal{M}$ define the predicate $\overline{\cdot} \mid d := \lambda n{:}\mathbb{N}.\, \mathcal{M} \vDash \overline{n} \mid d$.*

Theorem 7.10. *Assuming stability of* std*, then $\mathcal{M} > \mathbb{N}$ implies $\neg\neg\exists d{:}\mathcal{M}.\, \neg\mathsf{Dec}(\overline{\cdot} \mid d)$.*

Proof. We will assume $\neg\exists d.\, \neg\mathsf{Dec}(\overline{\cdot} \mid d)$ and try to reach a contradiction. By Corollary 7.8 there is a pair α', β' of inseparable unary Σ_1–formulas. By Σ_1–compression, they can be written in the form $\exists w.\, \alpha(w, x)$ and $\exists w.\, \beta(w, x)$, where α, β are Δ_1. Since they are disjoint, we have:

$$\mathbb{N} \vDash \forall\, x\, w\, v < \overline{n}.\ \alpha(w, x) \to \beta(v, x) \to \bot$$

for every bound $n : \mathbb{N}$. Due to its bounded quantification, the above formula is also Δ_1, allowing us to use Δ_1–absoluteness (Fact 3.4) to get

$$\mathcal{M} \vDash \forall\, x\, w\, v < \overline{n}.\ \alpha(w, x) \to \beta(v, x) \to \bot$$

Since we are assuming $\mathcal{M} > \mathbb{N}$, we have $\mathcal{M} \ncong \mathbb{N}$, which together with the stability of std allows us to use both Overspill (Lemma 4.8) and coding for predicates (Lemma 5.6). Overspill gives us the potential existence of an element $e{:}\mathcal{M} > \mathbb{N}$ with

$$\mathcal{M} \vDash \forall\, x\, w\, v < e.\ \alpha(w, x) \to \beta(v, x) \to \bot$$

which shows the disjointness of α, β when everything is bounded by e. By the coding result, we can get a code $c{:}\mathcal{M}$ which satisfies

$$\forall u{:}\mathbb{N}.\ \mathcal{M} \vDash (\exists\, w < e.\, \alpha(w, \overline{u})) \leftrightarrow \Pi(\overline{u}) \mid c$$

Given the above equivalence between $D := \lambda n : \mathbb{N}.\, \mathcal{M} \vDash \exists\, w < e.\, \alpha(w, \overline{n})$ and $\Pi(\overline{\cdot}) \mid c$, our initial assumption $\neg\exists d.\, \neg\mathsf{Dec}(\overline{\cdot} \mid d)$ entails that D cannot be undecidable. However, we will now see that D separates the given inseparable formulas:

$$\begin{aligned} &(1)\ \mathsf{Q} \vdash \exists\, w.\, \alpha(w, \overline{\cdot}) \ \subseteq\ D \\ &(2)\ \mathsf{Q} \vdash \exists\, w.\, \beta(w, \overline{\cdot}) \ \subseteq\ \neg D \end{aligned}$$

which will establish its undecidability.

(1) If $\mathsf{Q} \vdash \exists\, w.\, \alpha(w, \overline{n})$ there is $w : \mathbb{N}$ with $\mathbb{N} \vDash \alpha(\overline{w}, \overline{n})$ and $\mathcal{M} \vDash \alpha(\overline{w}, \overline{n})$ by Fact 3.4. Since $\overline{w} < e$ we can therefore show $D\, n$.
(2) If $\mathsf{Q} \vdash \exists\, w.\, \beta(w, \overline{n})$ there is $w : \mathbb{N}$ with $\mathbb{N} \vDash \beta(\overline{w}, \overline{n})$ and $\mathcal{M} \vDash \beta(\overline{w}, \overline{n})$ by Fact 3.4. Assume we had $D\, n$, then there is $v < e$ with $\mathcal{M} \vDash \alpha(v, \overline{n})$, which leads to a contradiction since α, β were shown disjoint below e.

It leaves us with the contradiction that D is both potentially decidable and undecidable. □

7.3. **Variants of the Theorem.** We now investigate two further variants of the theorem, going back to McCarty [McC87, McC88] and Makholm [Mak14] respectively. They make use of a stronger notion of inseparable formulas, which requires the formulas to be provably disjoint on the object level.

Definition 7.11 (HA-inseparable). *A pair of unary formulas $\alpha(x), \beta(x)$ is called* HA-inseparable *if they are disjoint in the sense of $\mathsf{HA} \vdash \neg\exists\, x.\, \alpha(x) \wedge \beta(x)$ and if any D with $\mathsf{Q} \vdash \alpha(\overline{\cdot}) \subseteq D$, $\mathsf{Q} \vdash \beta(\overline{\cdot}) \subseteq \neg D$ is undecidable.*

According to McCarty [McC88], the existence of HA-inseparable formulas can be established by taking the construction of inseparable formulas as seen in Lemma 7.7, and internalizing the given proof within HA. However, as pointed out in [Pet22] (Fact 6.1), *Rosser's trick* [Ros36] can be used to construct the desired HA-inseparable formulas from the inseparable formulas given by Corollary 7.8.

Definition 7.12 (Rosser Formula). *Given any binary formulas $\alpha(t, x)$ and $\beta(t, x)$ we define*

$$(\alpha < \beta)(x) := \exists\, t.\, \alpha(t, x) \wedge \forall\, v.\, \beta(v, x) \to t < v$$

Intuitively, if we interpret $\exists t.\, \alpha(t, x)$ as "*There is some time t at which $\alpha(t, x)$ can be verified*", then $(\alpha < \beta)(x)$ expresses that $\alpha(\,\cdot\,, x)$ will be verified before $\beta(\,\cdot\,, x)$.

Lemma 7.13 (Disjointness of Rosser Formulas). $\mathsf{HA} \vdash \forall x.\, (\alpha < \beta)(x) \to (\beta < \alpha)(x) \to \bot$

Proof. Our goal is to prove $(\alpha < \beta)(x)$, $(\beta < \alpha)(x)$, $\mathsf{HA} \vdash \bot$. From the formulas in the context of the derivation, we can conclude that there are terms t, t' such that

$$\begin{aligned} \alpha(t, x) \wedge \forall\, v.\, \beta(v, x) &\to t < v \\ \beta(t', x) \wedge \forall\, v.\, \alpha(v, x) &\to t' < v \end{aligned}$$

From this we get $t' < t \wedge t < t'$ and then clearly $\mathsf{HA} \vdash t < t' \to t' < t \to \bot$. □

Lemma 7.14 (HA-inseparable formulas). *There are unary* HA*-inseparable formulas.*

Proof. By Corollary 7.8, there are inseparable Σ_1–formulas, and due to Σ_1–compression, we can assume they have the form $\alpha'(x) = \exists\, w.\, \alpha(w, x)$, $\beta'(x) = \exists\, w.\beta(w, x)$, where α, β are both Δ_1. Since we have the equivalence

$$\forall v.\, \beta(v, x) \to w < v \;\leftrightarrow\; \forall v \leq w.\neg\beta(v, x)$$

where the latter formula is Δ_1, we can conclude that the Rosser formula $(\alpha < \beta)(x)$ is Σ_1.

We will now show that $\mathsf{Q} \vdash \alpha'(\overline{n})$ implies $\mathsf{Q} \vdash (\alpha < \beta)(\overline{n})$. From the assumption and soundness we get $w : \mathbb{N}$ such that $\mathbb{N} \vDash \alpha(\overline{w}, \overline{n})$. Since α', β' are inseparable and therefore disjoint, we must have $\mathbb{N} \vDash \neg\beta(\overline{v}, \overline{n})$ for every $v : \mathbb{N}$ and therefore in particular $\mathbb{N} \vDash \forall v.\, \beta(v, \overline{n}) \to \overline{w} < v$. Overall this shows $\mathbb{N} \vDash (\alpha < \beta)(\overline{n})$, which by Σ_1–completeness gives $\mathsf{Q} \vdash (\alpha < \beta)(\overline{n})$. With the exact same reasoning we can get the analogous result for $\beta < \alpha$.

We can now move on to prove that $\alpha < \beta$ and $\beta < \alpha$ constitute a pair of HA-inseparable formulas. The disjointness property follows from the previous Lemma 7.13. Abbreviating any predicate $\lambda n{:}\mathbb{N}.\, \mathsf{Q} \vdash \varphi(\overline{n})$ by $\mathsf{Q} \vdash \varphi$, it remains to show that any D satisfying

$$\begin{aligned} \mathsf{Q} \vdash (\alpha < \beta) \;&\subseteq\; D \\ \mathsf{Q} \vdash (\beta < \alpha) \;&\subseteq\; \neg D \end{aligned}$$

is undecidable. By the implications we have just shown above, we get the inclusions

$$\begin{aligned} \mathsf{Q} \vdash \alpha' \;\subseteq\; \mathsf{Q} \vdash (\alpha < \beta) \;&\subseteq\; D \\ \mathsf{Q} \vdash \beta' \;\subseteq\; \mathsf{Q} \vdash (\beta < \alpha) \;&\subseteq\; \neg D \end{aligned}$$

The undecidability of D therefore follows from the inseparability of the pair α', β'. □

This stronger disjointness property allows us to easily establish that satisfiability of HA-inseparable formulas is undecidable in any model.

Lemma 7.15. *If α, β are* HA*-inseparable, then $\mathcal{M} \vDash \alpha(\overline{\cdot})$ and $\mathcal{M} \vDash \beta(\overline{\cdot})$ are undecidable.*

Proof. Using soundness and the HA-disjointness of α and β, we get

$$\begin{aligned} &\mathsf{Q} \vdash \alpha(\overline{n}) \overset{\text{sound}}{\Longrightarrow} \mathcal{M} \vDash \alpha(\overline{n}) \\ &\mathsf{Q} \vdash \beta(\overline{n}) \overset{\text{sound}}{\Longrightarrow} \mathcal{M} \vDash \beta(\overline{n}) \overset{\text{HA-disj.}}{\Longrightarrow} \neg\mathcal{M} \vDash \alpha(\overline{n}) \end{aligned}$$

Undecidability of $\mathcal{M} \vDash \alpha(\overline{\cdot})$ then follows from inseparability of the given formulas (Corollary 7.8), and the same argument also shows the undecidability of $\mathcal{M} \vDash \beta(\overline{\cdot})$. □

McCarty [McC87, McC88] considers Tennenbaum's theorem with constructive semantics. Instead of models placed in classical set theory, he works in an intuitionistic theory (e.g. IZF), making the interpretation of the object-level disjunction much stronger. By furthermore assuming MP, he is then able to show that all models of HA in this constructive setting are standard. To achieve the constructive rendering of disjunctions, we will locally make use of the following choice principle:

Definition 7.16. *By* AUC *we denote the principle of unique choice:*

$$\forall X\, Y\, (R{:}X \to Y \to \mathbb{P}).\, (\forall x\, \exists! y.\, Rxy) \to \exists f{:}X \to Y.\, \forall x.\, Rx(fx)$$

Note that generally, Church's thesis and unique choice principles combined prove the negation of LEM (*cf.* [For21a]), which will make the results that use AUC (deliberately) anti-classical.

Lemma 7.17. *If* AUC *and* $\mathcal{M} > \mathbb{N}$*, then there is no unary formula* α *such that* $\mathcal{M} \vDash \alpha(\dot{-})$ *is undecidable.*

Proof. Since single instances of the law of excluded middle are provable under double negation, induction on n can be used to prove $\mathcal{M} \vDash \forall\, n.\, \neg\neg\forall\, x < n.\, \alpha(x) \vee \neg\alpha(x)$. Choosing $e : \mathcal{M} > \mathbb{N}$ for the bound n above, we get $\mathcal{M} \vDash \neg\neg\forall\, x < e.\, \alpha(x) \vee \neg\alpha(x)$ and therefore in particular $\neg\neg\,\forall n : \mathbb{N}.\, \mathcal{M} \vDash \alpha(\overline{n}) \vee \neg\alpha(\overline{n})$, meaning $\mathcal{M} \vDash \alpha(\dot{-})$ is potentially definite. Since AUC entails that definite predicates on $\mathbb{N}$ are decidable, we get $\neg\neg\mathsf{Dec}(\mathcal{M} \vDash \alpha(\dot{-}))$ for all formulas α, which implies the non-existence claim. □

Corollary 7.18. *If* AUC*, then* $\neg\mathcal{M} > \mathbb{N}$*.*

Proof. By Lemma 7.14 there exist HA-inseparable formulas, and by Lemma 7.15 their satisfiability is undecidable. This directly contradicts Lemma 7.17. □

Theorem 7.19 (McCarty). *Given* AUC *and* MP*, every model of* HA *is standard.*[8]

Proof. By Corollary 7.18 we have $\neg\mathcal{M} > \mathbb{N}$. Since $\mathsf{HA} \vdash \forall xy.\, x = y \vee \neg\, x = y$, it follows from AUC that every model $\mathcal{M} \vDash \mathsf{HA}$ is discrete. Since we also assume MP, Lemma 7.3 (2) now entails $\mathcal{M} \cong \mathbb{N}$. □

Now turning to Makholm [Mak14], we will no longer require AUC, but will instead make use of the fact that the coding lemma can be established in HA. While we have proven this result on the level of Coq in Corollary 5.2, we did not mechanize its derivation in the first-order theory HA. To make this transparent, we make its assumption explicit here.

Hypothesis 7.20 (HA-Coding). *For any unary formula* $\alpha(x)$*,* HA *can internally prove the coding lemma:* $\mathsf{HA} \vdash \forall\, n\, \neg\neg\exists\, c\, \forall\, u < n.\ \alpha(u) \leftrightarrow \Pi(u) \mid c$.[9]

Theorem 7.21 (Makholm). *If* $\mathcal{M} > \mathbb{N}$ *then* $\neg\neg\exists d : \mathcal{M}.\, \neg\mathsf{Dec}(\dot{-} \mid d)$*.*

Proof. By Lemma 7.14 there are HA-inseparable unary formulas α and β. Using soundness and HA-coding we get that α can be coded up to any bound $n : \mathcal{M}$

$$\mathcal{M} \vDash \forall n\, \neg\neg\exists\, c\, \forall\, u < n.\, \alpha(u) \leftrightarrow \Pi(u) \mid c.$$

By assumption, there is a non-standard element $e : \mathcal{M} > \mathbb{N}$, and picking this for the bound n, we potentially get a code $c : \mathcal{M}$ satisfying

$$\mathcal{M} \vDash \forall\, u < e.\, \alpha(u) \leftrightarrow \Pi(u) \mid c.$$

Since the above equivalence holds for all standard numbers $u : \mathcal{M}$, the undecidability of $\mathcal{M} \vDash \alpha(\dot{-})$ (Lemma 7.15) entails $\neg\mathsf{Dec}(\dot{-} \mid c)$, which shows the existence claim. □

Note the quite remarkable fact that in contrast to Theorem 7.10, we do not need to assume MP or discreteness of the model in order to establish Theorem 7.21.

[8] Or another way to put it: HA is *categorical*.

[9] In the conference paper [HK22], Hypothesis 2 stated that $\mathsf{HA} \vdash \forall\, n\, \exists\, c\, \forall\, u < n.\ \alpha(u) \leftrightarrow \Pi(u) \mid c$. Contrary to what was claimed, this is not provable for arbitrary formulas α. The new formulation of HA-coding solves this mistake through the addition of the double negation.

7.4. **Unearthing the Roots of Tennenbaum's Theorem.** In this section, we want to record briefly and mostly informally some general ingredients that seem to go into a proof of Tennenbaum's theorem.

Taking a look at the proofs of both McCarty and Makholm, we can see a very clear structure. In both of them:

- HA-inseparable formulas are used to derive the existence of an undecidable $\mathcal{M} \vDash \alpha(\overline{\cdot})$.
- Assuming the model has a non-standard element, it is shown that there potentially exists an element with undecidable divisibility on numerals.

In Makholm's proof the latter point was achieved by usage of the coding result Hypothesis 7.20, which establishes a connection between satisfaction of a formula and divisibility with respect to its code number. Close inspection of the proof of Corollary 5.2 reveals that only two properties of divisibility by primes are needed for it:

$$\forall x.\, \neg\, \pi_x \mid 1, \qquad \forall n\, c\, \exists c'\, \forall x.\ \pi_x \mid c' \leftrightarrow x = n \vee \pi_x \mid c.$$

We can abstract away from divisibility and formulate the result as:

Lemma 7.22. *Let $\in : \mathbb{N} \to \mathbb{N} \to \mathbb{P}$ be a binary predicate satisfying the conditions*

(1) $\exists e\, \forall x.\, \neg\, x \in e$ *(Empty set)*
(2) $\forall n\, c\, \exists c'\, \forall x.\, x \in c' \leftrightarrow x = n \vee x \in c$ *(Extension),*

which we can interpret as axiomatizing a weak notion of sets. For any predicate $p : \mathbb{N} \to \mathbb{P}$ we then have $\forall n\, \neg\neg\, \exists c\, \forall u < n.\, p\, u \leftrightarrow u \in c$.

Proof. Following what we did for Corollary 5.2, it is easier to prove the strengthened statement

$$\neg\neg\, \exists c{:}\mathbb{N}\ \forall u{:}\mathbb{N}.\big(u < n \to (p\, u \leftrightarrow u \in c)\big) \wedge \big(u \in c \to u < n\big)$$

by induction on n. The proof then follows the proof of Corollary 5.2. In the base case we make use of the empty set, and in the induction step we do a case distinction on $\neg\neg(p\, n \vee \neg p\, n)$ and utilize the extension property. The details for these steps were checked with Coq. □

If we have a binary formula $\varphi_\in(x, y)$ satisfying the same conditions inside of HA, we can give a derivation of $\mathsf{HA} \vdash \forall n\, \neg\neg\, \exists c\, \forall u < n.\, \alpha(u) \leftrightarrow \varphi_\in(u, c)$ and verbatim run Makholm's proof. Using $\overline{\cdot} \in d$ as a shorthand for $\lambda n.\, \mathcal{M} \vDash \varphi_\in(\overline{n}, d)$ we would then get:

Theorem. *If $\mathcal{M} > \mathbb{N}$ then $\neg\neg\exists d{:}\mathcal{M}.\, \neg\mathsf{Dec}(\overline{\cdot} \in d)$.*

This highlights that the statement of Theorem 7.21 is not inherently tied to divisibility, which is also remarked by Kaye in [Kay11]. It rather seems generally tied to relations that allow the implementation of finite sets or sequences inside of HA. Visser [Vis08] analyzes several axiomatizations of pairs, sets and sequences in first-order theories, and the conditions we have listed above appear as the axioms of the weak set theory WS, which can interpret Q and is therefore essentially undecidable. This latter point raises an interesting question: One can now wonder if it is decidability of $\overline{\cdot} \in d$ or $\overline{\cdot} \mid d$ respectively, together with the essential undecidability of WS and Q, which combine to rule out that the model is non-standard.

8. Discussion

8.1. **General Remarks.** In Section 7, we presented several proofs of Tennenbaum's theorem which we summarize in the table below, listing their assumptions[10] on the left and the conclusion on the right.

MP	AUC	$\mathcal{M}$ discrete	Theorem	Idea/Technique
•		•	$\mathcal{M}$ enumerable $\to$ $\mathcal{M} \cong \mathbb{N}$	Diagonalization
•		•	$\mathcal{M} > \mathbb{N} \to \neg\neg\exists d{:}\mathcal{M}.\neg\mathsf{Dec}(\ulcorner\cdot\urcorner \mid d)$	Inseparability
			$\mathcal{M} > \mathbb{N} \to \neg\neg\exists d{:}\mathcal{M}.\neg\mathsf{Dec}(\ulcorner\cdot\urcorner \mid d)$	Makholm
•	•		$\mathcal{M} \cong \mathbb{N}$	McCarty

First note that in the first row we can clearly show the reverse implication ($\mathcal{M} \cong \mathbb{N}$ implies enumerability). If we additionally assume MP and discreteness of $\mathcal{M}$, the first two results combined give the equivalences

$$\mathcal{M} \text{ enumerable} \iff \mathcal{M} \cong \mathbb{N} \overset{7.3}{\iff} \neg\,\mathcal{M} > \mathbb{N} \iff \neg\exists d.\neg\mathsf{Dec}(\ulcorner\cdot\urcorner \mid d).$$

Furthermore, we can see that Makholm's result is a strengthening of the second one, in the sense that it no longer requires MP nor the discreteness assumption. This becomes possible by the usage of HA-inseparable formulas, overcoming the need for Overspill, which—in the proof of Theorem 7.10—can be identified to be the root for these additional assumptions. In general, we can observe that the results become progressively stronger and less reliant on further assumptions as more and more intermediary results are proven on the object level of HA. For example, instead of using Overspill to establish infinite coding, we later use that the coding lemma holds internally (Hypothesis 7.20). Likewise, using HA-inseparable formulas, instead of inseparable formulas, contributed to another strengthening. This might not be the end of the strengthenings that can be done, as it is remains open whether the constructively stronger $\mathcal{M} > \mathbb{N} \to \exists d.\,\neg\mathsf{Dec}(\ulcorner\cdot\urcorner \mid d)$ can also be shown.

As was pointed out by McCarty in [McC88], a weaker version WCT of CT suffices for his proof, where the code representing a given function is hidden behind a double negation. He mentions in [McC91] that WCT is still consistent with the Fan theorem, while CT is not. Analogously, the following weakening of $\mathsf{CT_Q}$ suffices for all of the proofs that we have presented:

Definition 8.1 ($\mathsf{WCT_Q}$). *For every function $f : \mathbb{N} \to \mathbb{N}$ there* potentially *is a binary Σ_1–formula $\varphi_f(x, y)$ such that for every $n{:}\mathbb{N}$ we have $\mathsf{Q} \vdash \forall y.\,\varphi_f(\overline{n}, y) \leftrightarrow \overline{fn} = y$.*

This only needs few changes of the presented proofs.[11] An advantage of $\mathsf{WCT_Q}$ over $\mathsf{CT_Q}$ is that the former follows from the double negation of the latter and is therefore negative, ensuring that its assumption does not block the reduction of proof-terms found in the mechanization [CDE+13].

Depending on the fragment of first-order logic, one can give constructive proofs of the model existence theorem [FKW21], producing a countable syntactic model with computable functions for every consistent theory. By the argument given in the introduction, model

[10] We do not list the global assumption $\mathsf{CT_Q}$. HA-coding (Hypothesis 7.20) was not mechanized in Coq but is provable, which is why we leave it out of the table.

[11] We could have presented all results with respect to $\mathsf{WCT_Q}$. We opted against this in favor of $\mathsf{CT_Q}$, to avoid additional handling of double negations and to keep the proofs more readable.

existence would yield a countable and computable non-standard model of PA, which at first glance seems to contradict the statement of Tennenbaum's theorem. For any countable non-standard model of PA however, Theorem 7.21 and Lemma 7.2 entail that neither equality nor apartness can be decidable. This is similar in spirit to the results in [GH17], showing that even if the functions of the model are computable, non-computable behavior still emerges, but in relation to equality.

8.2. **Coq Mechanization.** The Coq development is axiom-free and the usage of crucial but constructively justified assumptions $\mathsf{CT_Q}$, MP and AUC is localized in the relevant sections. Apart from these, there is Hypothesis 7.20 which is taken as an additional assumption in the relevant sections. We have given details as to how this hypothesis can be proven, but since we did not yet mechanize the proof, we wanted to make its assumption on the level of the mechanization very explicit, by labeling it as a hypothesis in the accompanying text.

The development depends on and contributes to a growing collaborative Coq library for first-order logic [KHD+22]. Restricting to the files for this project, the line count is roughly 3600 lines of code. From those, 1500 lines on basic results about $\mathbb{N}$ and PA models were reused from earlier work [KH21]. Notably, the formalization of the various coding lemmas from Section 5 took 470 lines and all variants of Tennenbaum's theorem together amount to a total of only 800 lines.

8.3. **Related Work.** Classical proofs of Tennenbaum's theorem can be found in [BBJ02, Smi14, Kay11]. There are also refinements of the theorem which show that computability of either operation suffices [McA82] or which reduce the argument to a weaker induction scheme [Wil85, CMW82]. Constructive accounts were given by McCarty [McC87, McC88] and Plisko [Pli90], and a relatively recent investigation into Tennenbaum phenomena was conducted by Godziszewski and Hamkins [GH17].

For an account of CT as an axiom in constructive mathematics we refer to Kreisel [Kre70] and Troelstra [Tro73]. Investigations into CT and its connections to other axioms of synthetic computability based on constructive type theory were done by Forster [For21a, For22]. While there is no proof for the consistency of CT in CIC, there are consistency proofs for very similar systems [Yam20, SU19, For22].

Compared to the previous conference paper [HK22], this extended journal version relies on the slightly different definition of Σ_1 and Δ_1–formulas used by Kirst and Peters [KP23]. Moreover, they give a derivation of our formulation of $\mathsf{CT_Q}$ from a more conventional formulation of Church's thesis, illustrating that $\mathsf{CT_Q}$ is a convenient axiom for sidestepping much of the first-order encoding overhead, while the work needed to formally capture computation within Q is feasible in its mechanization.

Presentations of first-order logic in the context of proof-checking have already been discussed and used, among others, by Shankar [Sha86], Paulson [**?**], O'Connor [O'C05], as well as Han and van Doorn [HvD20]. We make use of the Coq library for first-order logic [KHD+22, HKK21], which has evolved from several previous projects [FKS19, FKW21, KLW20, KH21] and depends on the Coq library of undecidability proofs [FLWD+20].

Synthetic computability theory was introduced by Richman and Bauer [Ric83, Bau06] and initially applied to constructive type theory by Forster, Kirst, and Smolka [FKS19]. Their synthetic approach to undecidability results has been used in several other projects, all merged into the Coq library of undecidability proofs [FLWD+20].

8.4. **Future Work.** By relying on the synthetic approach, our treatment of Tennenbaum's theorem does not explicitly mention the computability of addition or multiplication of the model. To make these assumptions explicit again, and to also free our development from the necessity of adapting this viewpoint, we could assume an abstract version of CT which makes reference to a T predicate [Kle43, For21a] and is then used to axiomatize T-computable functions. Combining this with a version of CT that stipulates the computability of every T-computable function, would then allow us to specifically assume T-computability for either addition or multiplication, and to formalize the result that T-computability of either operation leads to the model being standard [McA82].

In this work, we have mechanized one of the two hypotheses that remained unmechanized in [HK22], leaving solely the object level coding lemma (Hypothesis 7.20) yet to be mechanized. Achieving this will necessitate the transformation of the proof provided for Corollary 5.2 into a derivation using the axioms of HA. This task will require the establishment of standard results about prime numbers, akin to what we established on the level of Coq's logic in one of the files. Similar to the mechanization of Lemma 7.14, these proofs will significantly benefit from the proof mode developed in [HKK21]. However, even with the proof mode, the translation of these Coq proofs into derivations entails a substantial amount of engineering work. Our preliminary estimates suggest that this will be a relatively independent project on its own, hence we leave it for future exploration.

A more satisfying rendering of McCarty's result would be achieved by changing the semantics (Definition 2.10), and putting the interpretations of formulas on the (proof-relevant) type level instead of the propositional level, therefore removing the need to assume AUC to break the barrier from the propositional to the type level.

Following usual practice in textbooks, we consider the first-order equality symbol as a syntactic primitive and only regard models interpreting it as actual equality in Coq. When treated as axiomatized relation instead, we could consider the (slightly harder to work with) setoid models and obtain the more general result that no computable non-standard setoid model exists.

There are interesting parallels when comparing the proofs of Tennenbaum's theorem and proofs of the first incompleteness theorem. In particular, we saw that the usage of HA-inseparable sets, and therefore the usage of Rosser's trick, leads to an improvement of the constructive Tennenbaum result. Connections between the two theorems are well-known [Kay91, Kay11], but it should be interesting to combine the presented work with work like [KP23], to study their connection in a constructive framework. We hope that this work has illustrated the value of such projects, as they can shed new light on old proofs, bringing forward their constructive content.

Acknowledgments

The authors want to thank Yannick Forster, Johannes Hostert, Benjamin Peters, and the anonymous reviewers for their helpful comments and suggestions.

References

[Bau06] Andrej Bauer. First steps in synthetic computability theory. *Electronic Notes in Theoretical Computer Science*, 155:5–31, 2006.

[Bau08] Andrej Bauer. Intuitionistic mathematics for physics, 2008. URL: `http://math.andrej.com/2008/08/13/intuitionistic-mathematics-for-physics/`.

[BBJ02] George S. Boolos, John P. Burgess, and Richard C. Jeffrey. *Computability and logic*. Cambridge University Press, 2002.

[CDE+13] Thierry Coquand, Nils A. Danielsson, Martın H. Escardó, Ulf Norell, and Chuangjie Xu. Negative consistent axioms can be postulated without loss of canonicity. *Unpublished note*, 2013. URL: https://www.cs.bham.ac.uk/~mhe/papers/negative-axioms.pdf.

[CH88] Thierry Coquand and Gérard Huet. The calculus of constructions. *Information and Computation*, 76(2):95–120, 1988. doi:10.1016/0890-5401(88)90005-3.

[CMW82] Patrick Cegielski, Kenneth McAloon, and George Wilmers. Modèles récursivement saturés de l'addition et de la multiplication des entiers naturels. In *Studies in Logic and the Foundations of Mathematics*, volume 108, pages 57–68. Elsevier, 1982.

[Coq23] The Coq Development Team. The Coq Proof Assistant, June 2023. doi:10.5281/zenodo.8161141.

[FKS19] Yannick Forster, Dominik Kirst, and Gert Smolka. On synthetic undecidability in Coq, with an application to the Entscheidungsproblem. In *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 38–51, 2019.

[FKW21] Yannick Forster, Dominik Kirst, and Dominik Wehr. Completeness theorems for first-order logic analysed in constructive type theory: Extended version. *Journal of Logic and Computation*, 31(1):112–151, 2021.

[FLWD+20] Yannick Forster, Dominique Larchey-Wendling, Andrej Dudenhefner, Edith Heiter, Dominik Kirst, Fabian Kunze, Gert Smolka, Simon Spies, Dominik Wehr, and Maximilian Wuttke. A Coq library of undecidable problems. In *CoqPL 2020 The Sixth International Workshop on Coq for Programming Languages*, 2020.

[For21a] Yannick Forster. Church's Thesis and Related Axioms in Coq's Type Theory. In Christel Baier and Jean Goubault-Larrecq, editors, *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*, volume 183 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2021/13455, doi:10.4230/LIPIcs.CSL.2021.21.

[For21b] Yannick Forster. *Computability in Constructive Type Theory*. PhD thesis, Saarland University, Saarbrücken, November 2021. URL: https://ps.uni-saarland.de/~forster/thesis/phd-thesis-yforster-screen.pdf.

[For22] Yannick Forster. Parametric Church's Thesis: Synthetic Computability Without Choice. In Sergei Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science*, volume 13137 of *Lecture Notes in Computer Science*, pages 70–89, Cham, 2022. Springer International Publishing.

[GH17] Michał Tomasz Godziszewski and Joel David Hamkins. Computable quotient presentations of models of arithmetic and set theory. In Juliette Kennedy and Ruy J.G.B. de Queiroz, editors, *Logic, Language, Information, and Computation*, pages 140–152, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[Her23] Marc Hermes. Github repository, 2023. URL: https://github.com/uds-psl/coq-library-fol/tree/tennenbaum.

[HK22] Marc Hermes and Dominik Kirst. An Analysis of Tennenbaum's Theorem in Constructive Type Theory. In Amy P. Felty, editor, *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, volume 228 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2022/16290, doi:10.4230/LIPIcs.FSCD.2022.9.

[HK23] Marc Hermes and Dominik Kirst. Website, 2023. URL: https://uds-psl.github.io/coq-library-fol/tennenbaum/toc.html.

[HKK21] Johannes Hostert, Mark Koch, and Dominik Kirst. A toolbox for mechanised first-order logic. In *The Coq Workshop 2021*, 2021.

[HvD20] Jesse M. Han and Floris van Doorn. A formal proof of the independence of the continuum hypothesis. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 353–366, 2020.

[Kay91] Richard Kaye. *Models of Peano Arithmetic*. Clarendon Press, 1991.

[Kay11] Richard Kaye. Tennenbaum's theorem for models of arithmetic. *Set Theory, Arithmetic, and Foundations of Mathematics. Ed. by J. Kennedy and R. Kossak. Lecture Notes in Logic. Cambridge*, pages 66–79, 2011.

[KH21] Dominik Kirst and Marc Hermes. Synthetic undecidability and incompleteness of first-order axiom systems in Coq. In Liron Cohen and Cezary Kaliszyk, editors, *12th International Conference on Interactive Theorem Proving (ITP 2021)*, volume 193 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/opus/volltexte/2021/13918`, `doi:10.4230/LIPIcs.ITP.2021.23`.

[KH23] Dominik Kirst and Marc Hermes. Synthetic undecidability and incompleteness of first-order axiom systems in Coq: Extended version. *Journal of Automated Reasoning*, 67(1):13, 2023.

[KHD+22] Dominik Kirst, Johannes Hostert, Andrej Dudenhefner, Yannick Forster, Marc Hermes, Mark Koch, Dominique Larchey-Wendling, Niklas Mück, Benjamin Peters, Gert Smolka, and Dominik Wehr. A Coq library for mechanised first-order logic. In *Coq Workshop*, 2022.

[Kir22] Dominik Kirst. *Mechanised Metamathematics: An Investigation of First-Order Logic and Set Theory in Constructive Type Theory*. PhD thesis, Saarland University, 2022. `https://www.ps.uni-saarland.de/~kirst/thesis/`.

[Kle43] Stephen C. Kleene. Recursive predicates and quantifiers. *Transactions of the American Mathematical Society*, 53(1):41–73, 1943.

[KLW20] Dominik Kirst and Dominique Larchey-Wendling. Trakhtenbrot's theorem in Coq: A Constructive Approach to Finite Model Theory. In *International Joint Conference on Automated Reasoning*, pages 79–96. Springer, 2020.

[KP23] Dominik Kirst and Benjamin Peters. Gödel's Theorem Without Tears - Essential Incompleteness in Synthetic Computability. In *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[Kre70] Georg Kreisel. Church's thesis: a kind of reducibility axiom for constructive mathematics. In *Studies in Logic and the Foundations of Mathematics*, volume 60, pages 121–150. Elsevier, 1970.

[Mak14] Henning Makholm. Tennenbaum's theorem without overspill. Mathematics Stack Exchange, 2014. (version: 2014-01-24). URL: `https://math.stackexchange.com/q/649457`.

[MC17] Bassel Mannaa and Thierry Coquand. The independence of Markov's principle in type theory. *Logical Methods in Computer Science*, 13, 2017.

[McA82] Kenneth McAloon. On the complexity of models of arithmetic. *The Journal of Symbolic Logic*, 47(2):403–415, 1982.

[McC87] David C. McCarty. Variations on a thesis: intuitionism and computability. *Notre Dame Journal of Formal Logic*, 28(4):536–580, 1987.

[McC88] David C. McCarty. Constructive validity is nonarithmetic. *The Journal of Symbolic Logic*, 53(4):1036–1041, 1988. URL: `http://www.jstor.org/stable/2274603`.

[McC91] David C. McCarty. Incompleteness in intuitionistic Metamathematics. *Notre Dame journal of formal logic*, 32(3):323–358, 1991.

[O'C05] Russell O'Connor. Essential incompleteness of arithmetic verified by Coq. In *International Conference on Theorem Proving in Higher Order Logics*, pages 245–260. Springer, 2005.

[Pet22] Benjamin Peters. Thesis: Gödel's Theorem Without Tears - Essential Incompleteness in Synthetic Computability. Bachelor's thesis, Saarland University, 2022.

[Pli90] Valerii E. Plisko. Constructive formalization of the Tennenbaum theorem and its applications. *Mathematical notes of the Academy of Sciences of the USSR*, 48(3):950–957, 1990.

[PM93] Christine Paulin-Mohring. Inductive definitions in the system Coq: rules and properties. In *International Conference on Typed Lambda Calculi and Applications*, pages 328–345. Springer, 1993.

[Raa21] Panu Raatikainen. Gödel's Incompleteness Theorems. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2021 edition, 2021.

[Ric83] Fred Richman. Church's thesis without tears. *The Journal of symbolic logic*, 48(3):797–803, 1983.

[Ros36] Barkley Rosser. Extensions of some theorems of Gödel and Church. *The Journal of Symbolic Logic*, 1:87–91, 1936.

[Sha86] Natarajan Shankar. *Proof-checking metamathematics (theorem-proving)*. PhD thesis, The University of Texas at Austin, 1986.

[Smi14] Peter Smith. Tennenbaum's theorem. Technical report, 2014. URL: `https://www.logicmatters.net/resources/pdfs/tennenbaum_new.pdf`.

[SU19] Andrew Swan and Taichi Uemura. On Church's Thesis in Cubical Assemblies, 2019. `arXiv:1905.03014`.

[Ten59] Stanley Tennenbaum. Non-archimedean models for arithmetic. *Notices of the American Mathematical Society*, 6(270):44, 1959.

[Tro73] Anne S. Troelstra. *Metamathematical investigation of intuitionistic arithmetic and analysis*, volume 344. Springer Science & Business Media, 1973.

[TvD88] Anne S. Troelstra and Dirk van Dalen. *Constructivism in Mathematics, Vol. 1. Studies in Logic and the Foundations of Mathematics, Vol. 121*. North-Holland Press, Amsterdam, 1988.

[Vis08] Albert Visser. Pairs, sets and sequences in first-order theories. *Archive for Mathematical Logic*, 47:299–326, 2008.

[Wil85] George Wilmers. Bounded existential induction. *The Journal of Symbolic Logic*, 50(1):72–90, 1985.

[Yam20] Norihiro Yamada. Game semantics of Martin-Löf type theory, part III: its consistency with Church's thesis, 2020. `arXiv:2007.08094`.

Appendix A. Deduction Systems

Intuitionistic natural deduction $\vdash : \mathsf{List}(\mathsf{fm}) \to \mathsf{fm} \to \mathbb{P}$ is defined inductively by the rules

$$\frac{\varphi \in \Gamma}{\Gamma \vdash \varphi} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi} \qquad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi} \qquad \frac{\Gamma \vdash \varphi \to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi} \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi} \qquad \frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \theta \quad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta}$$

$$\frac{\Gamma[\uparrow] \vdash \varphi}{\Gamma \vdash \forall \varphi} \qquad \frac{\Gamma \vdash \forall \varphi}{\Gamma \vdash \varphi[t]} \qquad \frac{\Gamma \vdash \varphi[t]}{\Gamma \vdash \exists \varphi} \qquad \frac{\Gamma \vdash \exists \varphi \quad \Gamma[\uparrow], \varphi \vdash \psi[\uparrow]}{\Gamma \vdash \psi}$$

where $\varphi[t]$ for a term t is a short for $\varphi[\, t; \lambda n.\, x_n]$. We get the classical variant $\vdash_c$ by adding Peirce's rule as an axiom:

$$\overline{\Gamma \vdash_c ((\varphi \to \psi) \to \varphi) \to \varphi}$$

The deduction systems lift to possibly infinite contexts $\mathcal{T} : \mathsf{fm} \to \mathbb{P}$ by writing $\mathcal{T} \vdash \varphi$ if there is a finite $\Gamma \subseteq \mathcal{T}$ with $\Gamma \vdash \varphi$.

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit `https://creativecommons.org/licenses/by/4.0/` or send a letter to Creative Commons, 171 Second St, Suite 300, San Francisco, CA 94105, USA, or Eisenacher Strasse 2, 10777 Berlin, Germany