Synthetic Undecidability and Incompleteness of First-Order Axiom Systems in Coq

Dominik Kirst
Universität des Saarlandes, Saarland Informatics Campus, Saarbrücken, Germany

Marc Hermes
Universität des Saarlandes, Department of Mathematics, Saarbrücken, Germany

Abstract

We mechanise the undecidability of various first-order axiom systems in Coq, employing the synthetic approach to computability underlying the growing Coq Library of Undecidability Proofs. Concretely, we cover both semantic and deductive entailment in fragments of Peano arithmetic (PA) and Zermelo-Fraenkel set theory (ZF), with their undecidability established by many-one reductions from solvability of Diophantine equations, i.e. Hilbert's tenth problem (H10), and the Post correspondence problem (PCP), respectively. In the synthetic setting based on the computability of all functions definable in a constructive foundation, such as Coq's type theory, it suffices to define these reductions as meta-level functions with no need for further encoding in a formalised model of computation.

The concrete cases of PA and ZF are prepared by a general synthetic theory of undecidable axiomatisations, focusing on well-known connections to consistency and incompleteness. Specifically, our reductions rely on the existence of standard models, necessitating additional assumptions in the case of full ZF, and all axiomatic extensions still justified by such standard models are shown incomplete. As a by-product of the undecidability of ZF formulated using only membership and no equality symbol, we obtain the undecidability of first-order logic with a single binary relation.

2012 ACM Subject Classification Theory of computation → Constructive mathematics; Theory of computation → Type theory; Theory of computation → Logic and verification

Keywords and phrases undecidability, synthetic computability, first-order logic, incompleteness, Peano arithmetic, ZF set theory, constructive type theory, Coq

Digital Object Identifier 10.4230/LIPIcs.ITP.2021.23

Supplementary Material *Sofware*: `https://www.ps.uni-saarland.de/extras/axiomatisations/`

Acknowledgements The authors thank Andrej Dudenhefner, Yannick Forster, Lennard Gäher, Julian Rosemann, Gert Smolka, and the anonymous reviewers for helpful comments and suggestions.

1 Introduction

Being among the mainstream formalisms to underpin mathematics, first-order logic has been subject to investigation from many different perspectives since its concretisation in the late 19th century. One of them is concerned with algorithmic properties, prominently pushed by Hilbert and Ackermann with the formulation of the *Entscheidungsproblem* [16], namely the search for a decision procedure determining the formulas φ that are valid in all interpretations, usually written $\vDash \varphi$. With their groundbreaking work in the 1930s, Turing [41] and Church [6] established that such a general decision procedure cannot exist. However, this outcome can change if one considers validity of φ restricted to interpretations satisfying a given collection $\mathcal{A}$ of axioms, written $\mathcal{A} \vDash \varphi$. Already in 1929, Presburger presented a decision procedure for an axiomatisation of linear arithmetic [28] and Tarski contributed further instances with his work on Boolean algebras, real-closed ordered fields, and Euclidean geometry in the 1940s [8].

On the other hand, as soon as an axiomatisation $\mathcal{A}$ is strong enough to express computation, the undecidability proof for the Entscheidungsproblem can be replayed within $\mathcal{A}$, turning its entailed theory undecidable. Used as standard foundations for large branches of mathematics exactly due to their expressiveness, Peano arithmetic (PA) and Zermelo-Fraenkel

set theory (ZF) are prime examples of such axiomatisations. In this paper, we use the Coq proof assistant [38] to mechanise the undecidability of PA and ZF based on the synthetic approach to computability results available in Coq's constructive type theory.

As is common in constructive foundations, all functions definable in Coq's type theory are effectively computable. So for instance any Boolean function on natural numbers $f : \mathbb{N} \to \mathbb{B}$ coinciding with a predicate $P \subseteq \mathbb{N}$ may be understood as a *decider* for P, even without explicitly relating f to some encoding as a Turing machine, μ-recursive function, or untyped λ-term. In this fashion, many positive notions of computability theory can be rendered *synthetically*, disposing of the need for an intermediate formal model of computation [3, 11]. Moreover, negative notions like *undecidability* are mostly established by transport along *reductions*, i.e. computable functions encoding instances of one problem in terms of another problem. Synthetically, the requirement that reductions are computable is again satisfied by construction. In fact, all problems included in the growing Coq Library of Undecidability Proofs [14] are shown undecidable in the sense that their decidability would entail the decidability of Turing machine halting by synthetic reduction from the latter.

Therefore, revisiting the undecidability of first-order axiom systems using a proof assistant like Coq is worthwhile for several reasons. First, using the synthetic approach to undecidability makes a mechanisation of these fundamental results of metamathematics pleasently feasible [11, 17]. Our mechanisations follow the informal (and instructive) practice to just define and verify reduction functions while leaving their computability implicit, with the key difference that in our constructive setting this relaxation is formally justified.

Secondly, it is well-known that undecidable axiomatisations $\mathcal{A}$ are negation-incomplete, i.e. admit φ with neither $\mathcal{A} \vDash \varphi$ nor $\mathcal{A} \vDash \neg\varphi$. By characterising $\mathcal{A} \vDash \varphi$ with an enumerable deduction system $\mathcal{A} \vdash \varphi$, this is a consequence of Post's theorem [27] stating that bi-enumerable predicates are decidable. Indeed, assuming negation-completeness, also the complement $\mathcal{A} \nvDash \varphi$ would be enumerable via $\mathcal{A} \vdash \neg\varphi$. Based on a synthetic proof of Post's theorem [3, 11], all axiomatisations shown synthetically undecidable in the present paper are incomplete in the sense that their completeness would imply the decidability of Turing machine halting. These algorithmic observations complement the otherwise notoriously hard to mechanise incompleteness proofs based on Gödel sentences [24, 25].

Lastly, undecidability of a first-order axiomatisation $\mathcal{A}$ like PA or ZF can only be established in a stronger system, since a reduction from a non-trivial problem yields the consistency of $\mathcal{A}$. Coq exhibits standard models for PA and ZF (the latter relying on mild assumptions [18]), enabling proofs of their undecidability. In fact, we sharpen the results for weak fragments Q′ and Z′ even strictly below Robinson arithmetic Q and Zermelo set theory Z, respectively, with the latter now also admitting a fully constructive standard model.

In summary, the contributions of this paper can be listed as follows:

- We extend the Coq Library of Undecidability Proofs with verified reductions to Q′, Q, PA, Z′, Z, and ZF(-regularity), regarding both Tarski semantics and natural deduction.
- We verify a translation of set theory over a convenient signature with function symbols for set operations to smaller signatures just containing one or two binary relation symbols.
- By composition, we obtain the undecidability of the Entscheidungsproblem for a single binary relation, improving on a previous mechanisation with additional symbols [11].
- By isolating a generic theorem, we obtain synthetic undecidability and incompleteness for all axiomatisations extending the fragments Q′ and Z′ w.r.t. standard models.

After a preliminary discussion of constructive type theory, synthetic undecidability, and first-order logic in Section 2, we proceed with the general results relating undecidabilitity, incompleteness, and consistency of first-order axiom systems in Section 3. This is followed by the case studies concerning arithmetical axiomatisations (Section 4) as well as set theory with Skolem functions (Section 5) and without (Section 6). We conclude in Section 7.

2 Preliminaries

In order to make this paper self-contained and accessible, we briefly outline the synthetic approach to undecidability proofs and the representation of first-order logic in constructive type theory used in previous papers.

2.1 Constructive Type Theory

We work in the framework of a constructive type theory such as the one implemented in Coq, providing a predicative hierarchy of *type universes* above a single impredicative universe $\mathbb{P}$ of *propositions*. On type level, we have the unit type $\mathbb{1}$ with a single element $* : \mathbb{1}$, the void type $\mathbb{0}$, function spaces $X \to Y$, products $X \times Y$, sums $X + Y$, dependent products $\forall(x : X).\, F\, x$, and dependent sums $\Sigma(x : X).\, F\, x$. On propositional level, these types are denoted by the usual logical notation ($\top$, $\bot$, $\to$, $\wedge$, $\vee$, $\forall$, and $\exists$). So-called *large elimination* from $\mathbb{P}$ into computational types is restricted, in particular case distinction on proofs of $\vee$ and $\exists$ to form computational values is disallowed. On the other hand, this restriction is permeable enough to allow large elimination of the equality predicate $= :\ \forall X.\, X \to X \to \mathbb{P}$ specified by the constructor $\forall(x : X).\, x = x$, as well as function definitions by well-founded recursion.

We employ 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* ($\mathbb{O}(X) := \ulcorner x \urcorner \mid \emptyset$), and *lists* ($l : \mathbb{L}(X) := [\,] \mid x :: l$). We write $|l|$ for the length of a list, $l +\!\!+ l'$ for the concatenation of l and l', $x \in l$ for membership, and just $f\,[x_1; \ldots; x_n] := [f\, x_1; \ldots; f\, x_n]$ for the map function. We denote by X^n the type of *vectors* $\vec{v}$ of length $n : \mathbb{N}$ over X and reuse the definitions and notations introduced for lists.

2.2 Synthetic Undecidability

The base of the synthetic approach to computability theory [30, 3] is the fact that all functions definable in a constructive foundation are computable. This fact applies to many variants of constructive type theory and we let the assumed variant sketched in the previous section be one of those. Of course, we are confident that in particular the polymorphic calculus of cumulative inductive constructions (pCuIC) [36] currently implemented in Coq satisfies this condition although there is no formal proof yet.

Now beginning with positive notions, we can introduce decidability and enumerability of decision problems synthetically, i.e. without reference to a formal model of computation:

Definition 1. *Let $P : X \to \mathbb{P}$ be a predicate over a type X.*

- *P is* decidable *if there exists $f : X \to \mathbb{B}$ s.t. $P\, x$ iff $f\, x = \mathsf{tt}$,*
- *P is* enumerable *if there exists $f : \mathbb{N} \to \mathbb{O}(X)$ s.t. $P\, x$ iff $f\, n = \ulcorner x \urcorner$ for some $n : \mathbb{N}$.*

Note that it is commonly accepted practice to mechanise decidability results in this synthetic sense (e.g. [4, 22, 31]). In the present paper, however, we mostly consider negative results in the form of undecidability of decision problems regarding first-order axiomatisations. Such negative results cannot be established in form of the actual negation of positive results, since constructive type theory is consistent with strong classical axioms turning every problem (synthetically) decidable (as witnessed by fully classical set-theoretic models, cf. [42]).

The approximation chosen in the Coq Library of Undecidability Proofs [14] is to call P (synthetically) undecidable if the decidability of P would imply the decidability of a seed problem known to be undecidable, specifically the halting problem for Turing machines. Therefore the negative notion can be turned into a positive notion, namely the existence of a computable reduction function, that again admits a synthetic rendering:

Definition 2. *Given predicates $P : X \to \mathbb{P}$ and $Q : Y \to \mathbb{P}$, we call a function $f : X \to Y$ a* (many-one) reduction *if $P\,x$ iff $Q\,(f\,x)$ for all x. We write $P \preceq Q$ if such a function exists.*

Then interpreting reductions from the halting problem for Turing machines as undecidability results is backed by the following fact:

Fact 3. *If $P \preceq Q$ and Q is decidable, then so is P.*

Such reductions have already been verified for Hilbert's tenth problem (H_{10}) [20] and the Post correspondence problem (PCP) [10] that we employ in the present paper, so by transitivity it is enough to verify continuing reductions to the axiom systems considered.

2.3 Syntax, Semantics, and Deduction Systems of First-Order Logic

We now review the representation of first-order syntax, semantics, and natural deduction systems developed in previous papers [11, 12, 17]. Beginning with the syntax, we describe terms $t : \mathbb{T}$ and formulas $\varphi : \mathbb{F}$ as inductive types over a fixed signature $\Sigma = (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$ of function symbols $f : \mathcal{F}_\Sigma$ and relation symbols $P : \mathcal{P}_\Sigma$ with arities $|f|$ and $|P|$:

$$t ::= \mathsf{x}_n \mid f\,\vec{t} \quad (n : \mathbb{N}, \vec{t} : \mathbb{T}^{|f|}) \qquad \varphi ::= P\,\vec{t} \mid \bot \mid \varphi \to \psi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \forall\varphi \mid \exists\varphi \quad (\vec{t} : \mathbb{T}^{|P|})$$

Negation $\neg\varphi$ and equivalence $\varphi \leftrightarrow \psi$ are then obtained by the standard abbreviations.

In the chosen de Bruijn representation [7], a bound variable is encoded as the number of quantifiers shadowing its binder, e.g. $\forall x.\, \exists y.\, P\,x\,u \to P\,y\,v$ may be represented by $\forall\,\exists\,P\,\mathsf{x}_1\,\mathsf{x}_4 \to P\,\mathsf{x}_0\,\mathsf{x}_5$. For the sake of legibility, we write concrete formulas with named binders where instructive and defer de Bruijn representations to the Coq development. A formula with all occurring variables bound by some quantifier is called *closed*.

Next, we define the usual Tarski semantics providing an interpretation of formulas:

Definition 4. *A* model $\mathcal{M}$ *consists of a domain type D as well as functions $f^{\mathcal{M}} : D^{|f|} \to D$ and $P^{\mathcal{M}} : D^{|P|} \to \mathbb{P}$ interpreting the symbols in the signature Σ. Given a variable assignment $\rho : \mathbb{N} \to D$ we define* term evaluation $\hat{\rho} : \mathbb{T} \to D$ *and* formula satisfiability $\rho \vDash \varphi$ *by*

$$\hat{\rho}\,\mathsf{x}_n := \rho\,n \qquad \hat{\rho}\,(f\,\vec{t}) := f^{\mathcal{M}}\,(\hat{\rho}\,\vec{t}) \qquad\qquad \rho \vDash P\,\vec{t} \;:=\; P^{\mathcal{M}}(\hat{\rho}\,\vec{t})$$

where the remaining cases of $\rho \vDash \varphi$ map each logical connective to its meta-level counterpart.

If a model $\mathcal{M}$ satisfies a formula φ for all variable assignments ρ, we write $\mathcal{M} \vDash \varphi$. Moreover, given a *theory* $\mathcal{T} : \mathbb{F} \to \mathbb{P}$, we write $\mathcal{M} \vDash \mathcal{T}$ if $\mathcal{M} \vDash \psi$ for all ψ with $\mathcal{T}\,\psi$ and $\mathcal{T} \vDash \varphi$ if $\mathcal{M} \vDash \mathcal{T}$ implies $\mathcal{M} \vDash \varphi$ for all $\mathcal{M}$. The same notations apply to *(finite) contexts* $\Gamma : \mathbb{L}(\mathbb{F})$.

Finally, we represent deduction systems as inductive predicates of type $\mathbb{L}(\mathbb{F}) \to \mathbb{F} \to \mathbb{P}$. In this paper, we consider intuitionistic and classical natural deduction $\Gamma \vdash_i \varphi$ and $\Gamma \vdash_c \varphi$, respectively, and write $\Gamma \vdash \varphi$ if a statement applies to both variants. The rules characterising the two systems are standard and listed in Appendix A, here we only highlight the quantifier rules depending on the de Bruijn encoding of bound variables

$$\frac{\Gamma[\uparrow] \vdash \varphi}{\Gamma \vdash \forall\varphi}\ \text{AI} \qquad \frac{\Gamma \vdash \forall\varphi}{\Gamma \vdash \varphi[t]}\ \text{AE} \qquad \frac{\Gamma \vdash \varphi[t]}{\Gamma \vdash \exists\varphi}\ \text{EI} \qquad \frac{\Gamma \vdash \exists\varphi \quad \Gamma[\uparrow], \varphi \vdash \psi[\uparrow]}{\Gamma \vdash \psi}\ \text{EE}$$

where $\varphi[\sigma]$ denotes the *capture-avoiding instantiation* of a formula φ with a *parallel substitution* $\sigma : \mathbb{N} \to \mathbb{T}$, where the substitution $\uparrow$ maps n to x_{n+1}, where the substitution $(t; \sigma)$ maps 0 to t and $n+1$ to $\sigma\,n$, and where $\varphi[t]$ is short for $\varphi[t; (\lambda n.\, \mathsf{x}_n)]$. Extending the deduction systems to theories $\mathcal{T} : \mathbb{F} \to \mathbb{P}$, we write $\mathcal{T} \vdash \varphi$ if there is $\Gamma \subseteq \mathcal{T}$ with $\Gamma \vdash \varphi$.

Constructively, only soundness of the intuitionistic system ($\mathcal{T} \vdash_i \varphi$ implies $\mathcal{T} \vDash \varphi$) is provable without imposing a restriction on the admitted models (as done in [12]). However, it is easy to verify the usual weakening ($\Gamma \vdash \varphi$ implies $\Delta \vdash \varphi$ for $\Gamma \subseteq \Delta$) and substitution ($\Gamma \vdash \varphi$ implies $\Gamma[\sigma] \vdash \varphi[\sigma]$) properties of both variants by induction on the given derivations. The latter gives rise to named reformulations of (AI) and (EE) helpful in concrete derivations

$$\frac{\Gamma \vdash \varphi[\mathsf{x}_n]}{\Gamma \vdash \forall \varphi}\ \mathsf{x}_n \notin \Gamma, \varphi \qquad \frac{\Gamma \vdash \exists \varphi \quad \Gamma, \varphi[\mathsf{x}_n] \vdash \psi}{\Gamma \vdash \psi}\ \mathsf{x}_n \notin \Gamma, \varphi, \psi$$

where $\mathsf{x}_n \notin \Gamma$ denotes that x_n is *fresh*, i.e. does no occur unbound in any formula of Γ.

The concrete signatures used in this paper all contain a reserved binary relation symbol $\equiv$ for equality. Instead of making equality primitive in the syntax, semantics, and deduction systems, we implicitly restrict $\mathcal{M} \vDash \varphi$ to extensional models $\mathcal{M}$ interpreting $\equiv$ as actual equality $=$ and understand $\mathcal{T} \vdash \varphi$ as derivability from $\mathcal{T}$ augmented with the standard axioms characterising $\equiv$ as an equivalence relation congruent for the symbols in Σ.

3 Undecidable and Incomplete First-Order Axiom Systems

In this section, we record some general algorithmic facts concerning first-order axiomatisations and outline the common scheme underlying the undecidability proofs presented in the subsequent two sections. We fix an enumerable and discrete signature Σ for the remainder of this section and begin by introducing the central notion of axiom systems formally.

Definition 5. *We call a theory $\mathcal{A} : \mathbb{F} \to \mathbb{P}$ an* axiomatisation *if $\mathcal{A}$ is enumerable.*

Any given axiomatisation induces two related decision problems, namely semantic entailment $\mathcal{A}^{\vDash} := \lambda\varphi.\, \mathcal{A} \vDash \varphi$ and deductive entailment $\mathcal{A}^{\vdash} := \lambda\varphi.\, \mathcal{A} \vdash \varphi$. Since in our constructive setting we can show the classical deduction system $\vdash_c$ neither sound nor complete (cf. [12]), we mostly consider a combined notion of classical semantics and intuitionistic deduction:

Definition 6. *We say that a predicate $P : X \to \mathbb{P}$* reduces to *$\mathcal{A}$, written $P \preceq \mathcal{A}$, if there is a function $f : X \to \mathbb{F}$ witnessing both $P \preceq \mathcal{A}^{\vDash}$ and $P \preceq \mathcal{A}^{\vdash_i}$.*

Assuming the law of excluded middle $\mathsf{LEM} := \forall p : \mathbb{P}.\, p \vee \neg p$ would be sufficient to obtain $P \preceq \mathcal{A}^{\vdash_c}$ from $P \preceq \mathcal{A}^{\vDash}$, since then $\mathcal{A} \vdash_c \varphi$ and $\mathcal{A} \vDash \varphi$ coincide. In fact, already the soundness direction is enough for our case studies on PA and ZF, since for them it is still feasible to verify $\mathcal{A} \vdash f\, x$ given $P\, x$ by hand without appealing to completeness.

We now formulate two facts stating the well-known connections of undecidability with consistency and incompleteness for our synthetic setting. The first observation is that verifying a reduction from a non-trivial problem is at least as hard as a consistency proof.

Fact 7. *If $P \preceq \mathcal{A}^{\vdash}$ and there is x with $\neg P\, x$, then $\mathcal{A} \nvdash \bot$.*

Proof. If $f : X \to \mathbb{F}$ witnesses $P \preceq \mathcal{A}^{\vdash}$, then by $\neg P\, x$ we obtain $\mathcal{A} \nvdash f\, x$. This prohibits a derivation $\mathcal{A} \vdash \bot$ by the explosion rule (E). ◀

The second observation is a synthetic version of incompleteness for all axiomatisations strong enough to express an undecidable problem. We follow the common practice to focus on incompleteness of the classical deduction system, see Section 7.1 for a discussion.

Definition 8. *We call $\mathcal{A}$* (negation-)complete *if for all closed φ either $\mathcal{A} \vdash_c \varphi$ or $\mathcal{A} \vdash_c \neg\varphi$.*

Fact 9. *If $\mathcal{A}$ is complete with $\mathcal{A} \nvdash_c \bot$, then $\lambda\varphi.\, \mathcal{A} \vdash_c \varphi$ is decidable for closed φ. Consequently, if f witnesses $P \preceq \mathcal{A}^{\vdash_c}$ such that all $f\, x$ are closed, then P is decidable.*

Proof. By a synthetic version of Post's theorem ([11, Lemma 2.15]) it suffices to show that $\mathcal{A}^{\vdash_c}$ is bi-enumerable, i.e. both $\lambda\varphi.\, \mathcal{A} \vdash_c \varphi$ and $\lambda\varphi.\, \mathcal{A} \nvdash_c \varphi$ are enumerable, and logically decidable, i.e. $\mathcal{A} \vdash_c \varphi$ or $\mathcal{A} \nvdash_c \varphi$ for all φ. This follows by enumerability of $\vdash_c$ and since by consistency and completeness $\mathcal{A} \nvdash_c \varphi$ iff $\mathcal{A} \vdash_c \neg\varphi$. The consequence is by Fact 3. ◀

Note that this fact is an approximation of the usual incompleteness theorem in two ways. First, similar to the synthetic rendering of undecidability, axiomatisations $\mathcal{A}$ subject to a reduction $P \preceq \mathcal{A}^{\vdash_c}$ for P known to be undecidable are only shown incomplete in the sense that their completeness would imply decidability of P. Deriving an actual contradiction would rely on computability axioms (e.g. Church's thesis [19, 9] or an undecidability assumption [11]) or extraction to a concrete model (e.g. a weak call-by-value λ-calculus [13]). Secondly, the fact does not produce a witness of an independent formula the way a more informative proof based on Gödel sentences does. Also note that inconsistent axiomatisations are trivially decidable, so the requirement $\mathcal{A} \nvdash_c \bot$ is inessential (especially given Fact 7).

Next, we outline the general pattern underlying the reductions verified in this paper:

1. We choose an undecidable seed problem $P : X \to \mathbb{P}$ easy to encode in the domain of the target axiomatisations. This will be H_{10} for PA and PCP for ZF.
2. We define the translation function $X \to \mathbb{F}$ mapping instances $x : X$ to formulas φ_x in a way compact enough to be stated without developing much of the internal theory of $\mathcal{A}$.
3. We isolate a finite fragment $A \subseteq \mathcal{A}$ of axioms that suffices to implement the main argument. This yields a reusable factorisation and is easier to mechanise.
4. We verify the semantic part locally by showing for every $\mathcal{M}$ with $\mathcal{M} \vDash A$ that $P\,x$ iff $\mathcal{M} \vDash \varphi_x$. For the backwards direction, we in fact need to restrict $\mathcal{M}$ to satisfy a suitable property of standardness allowing us to reconstruct an actual solution of P.
5. We construct standard models for A and $\mathcal{A}$, possibly relying on additional assumptions.
6. We verify the deductive part by establishing that $P\,x$ implies $A \vdash \varphi_x$, closely following the semantic proof from before. The backwards direction follows from soundness.
7. We conclude the undecidability of A, $\mathcal{A}$, and any $\mathcal{B} \supseteq A$ by virtue of the following:

Theorem 10. *Let a problem $P : X \to \mathbb{P}$, an axiomatisation $\mathcal{A}$, a notion of standardness on models $\mathcal{M} \vDash \mathcal{A}$, and a function $\varphi_{\_} : X \to \mathbb{F}$ be given with the following properties:*

- **(i)** *$P\,x$ implies $\mathcal{A} \vDash \varphi_x$.*
- **(ii)** *Every standard model $\mathcal{M} \vDash \mathcal{A}$ with $\mathcal{M} \vDash \varphi_x$ yields $P\,x$.*
- **(iii)** *$P\,x$ implies $\mathcal{A} \vdash \varphi_x$.*

Then $P \preceq \mathcal{B}$ for all $\mathcal{B} \supseteq \mathcal{A}$ admitting a standard model. Assuming LEM*, then also $P \preceq \mathcal{B}^{\vdash_c}$.*

Proof. We begin with $P \preceq \mathcal{B}^{\vDash}$. That $P\,x$ implies $\mathcal{B} \vDash \varphi_x$ is direct by (i) since every model of $\mathcal{B}$ is a model of $\mathcal{A}$. Conversely, if $\mathcal{B} \vDash \varphi_x$ then in particular the assumed standard model $\mathcal{M} \vDash \mathcal{B}$ satisfies φ_x. Thus we obtain $P\,x$ by (ii).

Turning to $P \preceq \mathcal{B}^{\vdash_i}$, the first direction is again trivial, this time by (iii) and weakening. For the converse, we assume that $\mathcal{B} \vdash_i \varphi_x$ and hence $\mathcal{B} \vDash \varphi_x$ by soundness. Thus we conclude $P\,x$ with the previous argument relying on (ii).

Finally assuming LEM, we obtain $P \preceq \mathcal{B}^{\vdash_c}$ since then already $\mathcal{B} \vdash_c \varphi_x$ implies $\mathcal{B} \vDash \varphi_x$. ◀

Of course (i) follows from (iii) via soundness, so the initial semantic verification could be eliminated from Theorem 10 and the informal strategy outlined before. However, we deem it more instructive to first present a self-contained semantic verification without the overhead introduced by working in a syntactic deduction system, mostly apparent in the Coq mechanisation. Also note that the necessity of a standard model will be no burden in the treatment of PA but in the case of ZF this will require a careful analysis of preconditions.

We end this section with the unsurprising but still important fact that we can reduce the decision problem for finite axiomatisations A to the classical Entscheidungsproblem of first-order logic concerning validity and provability in the empty context [16].

Fact 11. *For $A : \mathbb{L}(\mathbb{F})$ we have $A^{\vDash} \preceq (\lambda\varphi.\ \vDash \varphi)$ and $A^{\vdash} \preceq (\lambda\varphi.\ \vdash \varphi)$.*

Proof. It is straightforward to verify that the function $\lambda\varphi.\ \bigwedge A \to \varphi$ prefixing φ with the conjunction of all formulas in A establishes both reductions. ◀

So the reductions to finite fragments of PA and ZF presented in the next sections in particular complement the direct reductions to the Entscheidungsproblem given in [11].

4 Peano Arithmetic

We begin with a rather simple case study to illustrate our general approach to undecidability and incompleteness. For the theory of Peano arithmetic (PA) we use a signature containing symbols for the constant zero, the successor function, addition, multiplication and equality:

$$(O,\ S\_,\ \_ \oplus \_,\ \_ \otimes \_;\ \_ \equiv \_)$$

The core of PA consists of axioms characterising addition and multiplication:

$$\oplus\text{-base: } \forall x.\, O \oplus x \equiv x \qquad \oplus\text{-recursion: } \forall xy.\, (Sx) \oplus y \equiv S(x \oplus y)$$
$$\otimes\text{-base: } \forall x.\, O \otimes x \equiv O \qquad \otimes\text{-recursion: } \forall xy.\, (Sx) \otimes y \equiv y \oplus x \otimes y$$

The list Q' consisting of these four axioms is strong enough to be undecidable. Undecidability (and incompleteness) then transport in particular to the (infinite) axiomatisation PA adding

$$\text{Disjointness: } \forall x.\, Sx \equiv O \to \bot \qquad \text{Injectivity: } \forall xy.\, Sx \equiv Sy \to x \equiv y$$

and the axiom scheme of induction, which we define as a type-theoretic function on formulas:

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

Another typical reference point in the context of incompleteness is Robinson arithmetic Q, obtained by replacing the induction scheme by the single axiom $\forall x.\, x \equiv O \vee \exists y.\, x \equiv Sy$.

Hilbert's 10th problem (H_{10}) is concerned with the solvability of Diophantine equations and comes as a natural seed problem for showing the undecidability of PA, since the equations are a syntactic fragment of PA formulas. To be more precise, H_{10} consists of deciding whether a Diophantine equation $p = q$ has a solution in the natural numbers $\mathbb{N}$, where p, q are polynomials constructed by parameters, variables, addition, and multiplication:

$$p, q ::= \mathsf{a}_n \mid \mathsf{var}\ k \mid \mathsf{add}\ p\ q \mid \mathsf{mult}\ p\ q \qquad (n, k : \mathbb{N})$$

The evaluation $[\![p]\!]_\alpha$ of a polynomial p for a variable assignment $\alpha : \mathbb{N} \to \mathbb{N}$ is defined by

$$[\![\mathsf{a}_n]\!]_\alpha := n \qquad [\![\mathsf{var}\ k]\!]_\alpha := \alpha\, k \qquad [\![\mathsf{add}\ p\ q]\!]_\alpha := [\![p]\!]_\alpha + [\![q]\!]_\alpha \qquad [\![\mathsf{mult}\ p\ q]\!]_\alpha := [\![p]\!]_\alpha \times [\![q]\!]_\alpha$$

and a Diophantine equation $p = q$ then has a solution, if there is α such that $[\![p]\!]_\alpha = [\![q]\!]_\alpha$. Given their syntactic similarity, it is easy to encode H_{10} into PA, beginning with numerals:

Definition 12. *We define $\nu : \mathbb{N} \to \mathbb{T}$ recursively by $\nu(0) := O$ and $\nu(n+1) := S(\nu(n))$.*

We now translate polynomials into PA terms by defining $p^* : \mathbb{T}$ recursively:

$$\mathsf{a}_n{}^* := \nu(n) \qquad (\mathsf{var}\ k)^* := \mathsf{x}_k \qquad (\mathsf{add}\ p\ q)^* := p^* \oplus q^* \qquad (\mathsf{mult}\ p\ q)^* := p^* \otimes q^*$$

A Diophantine equation with greatest free variable N can now be encoded as the formula $\varphi_{p,q} := \exists^N p^* \equiv q^*$ where we use N leading existential quantifiers to internalise the solvability condition. The formula $\varphi_{p,q}$ thus asserts the existence of a solution for $p = q$ which gives us a natural encoding from Diophantine equations into PA.

We prepare the verification of the three requirements (Facts 19, 21, and 24) necessary to apply Theorem 10 with the following lemma about closed existential formulas:

Lemma 13. *If $\exists^N \varphi$ is closed, then*
(i) *$\mathcal{M} \vDash \exists^N \varphi$ iff there is $\rho : \mathbb{N} \to \mathcal{M}$ such that $\rho \vDash \varphi$,*
(ii) *$\Gamma \vdash \exists^N \varphi$ if there is $\sigma : \mathbb{N} \to \mathbb{T}$ such that $\Gamma \vdash \varphi[\sigma]$.*

Proof. We only provide some intuition for (i). For the implication from left to right, the assumption $\mathcal{M} \vDash \exists^N \varphi$ gives us $x_1, \dots, x_N : \mathcal{M}$ such that $\forall \rho.\, x_1; \dots; x_N; \rho \vDash \varphi$, so in particular we have $\rho' \vDash \varphi$ for $\rho' := x_1; \dots; x_N; (\lambda x.\, O^{\mathcal{M}})$, showing the claim. For the other implication, we get ρ with $\rho \vDash \varphi$. By setting $\rho' := \lambda x.\, \rho(x+N)$ we have $\rho = \rho(0); \dots; \rho(N); \rho'$ and hence there are $x_1, \dots, x_N : \mathcal{M}$ such that $x_1; \dots; x_N; \rho' \vDash \varphi$. Since φ has at most N free variables, ρ' can be exchanged with any other $\tau : \mathbb{N} \to \mathcal{M}$. ◀

By Lemma 13, showing $\varphi_{p,q}$ is equivalent to finding a satisfying environment $\rho : \mathbb{N} \to \mathcal{M}$ for $p^* \equiv q^*$ in a model $\mathcal{M}$ or deductively showing that a substitution $\sigma : \mathbb{N} \to \mathbb{T}$ solves it. This enables us to transport a solution for $p = q$ to both the model and the deduction system.

We now verify the semantic part of the reduction for the axiomatic fragment Q'. To this end, we fix a model $\mathcal{M} \vDash \mathsf{Q}'$ for the next definitions and lemmas.

Definition 14. *We define $\mu : \mathbb{N} \to \mathcal{M}$ by $\mu(0) := O^{\mathcal{M}}$ and $\mu(n+1) := S^{\mathcal{M}}(\mu(n))$.*

The axioms in Q' are sufficient to prove that μ is a homomorphism.

Lemma 15. *For $n, m : \mathbb{N}$, $\mu(n+m) = \mu(n) \oplus^{\mathcal{M}} \mu(m)$ and $\mu(n+m) = \mu(n) \otimes^{\mathcal{M}} \mu(m)$.*

Proof. The proof for addition is done by induction on $n : \mathbb{N}$ and using the axioms for addition in Q'. The proof for multiplication is done in the same fashion, using the axioms for multiplication and the previous result for addition. ◀

Lemma 16. *For any $\rho : \mathbb{N} \to \mathcal{M}$ and $n : \mathbb{N}$ we have $\hat{\rho}\,(\nu(n)) = \mu(n)$.*

Given an assignment $\alpha : \mathbb{N} \to \mathbb{N}$, we can transport the evaluation of a polynomial $[\![p]\!]_\alpha$ to any Q' model by applying μ. The homomorphism property of μ now makes it easy to verify that we get the same result by evaluating the encoded version p^* with the composition $\mu \circ \alpha$.

Lemma 17. *For any polynomial p and $\alpha : \mathbb{N} \to \mathbb{N}$ we have $\widehat{(\mu \circ \alpha)}(p^*) = \mu([\![p]\!]_\alpha)$.*

Proof. By induction on p, using Lemmas 16 and 17. ◀

Corollary 18. *If $p = q$ has a solution α, then in any Q' model $(\mu \circ \alpha) \vDash p^* \equiv q^*$.*

Proof. We have $\mu([\![p]\!]_\alpha) = \mu([\![q]\!]_\alpha) \overset{L.17}{\implies} \widehat{(\mu \circ \alpha)}(p^*) = \widehat{(\mu \circ \alpha)}(q^*) \implies (\mu \circ \alpha) \vDash p^* \equiv q^*$. ◀

Fact 19. *If $p = q$ has a solution, then $\mathsf{Q}' \vDash \varphi_{p,q}$.*

Proof. Let α be the solution of $p = q$, then $(\mu \circ \alpha) \vDash p^* \equiv q^*$ holds by Corollary 18 and since $\exists^N p^* \equiv q^*$ is closed by construction, the goal follows by Lemma 13. ◀

Turning to the converse direction, the natural choice for a standard model is the type $\mathbb{N}$.

Lemma 20. *$\mathbb{N}$ is a model of Q', Q, and PA.*

It is straightforward to extract a solution of $p = q$ if $\mathbb{N} \vDash \varphi_{p,q}$ using the previous lemmas.

Fact 21. *If $\mathbb{N} \vDash \varphi_{p,q}$ then $p = q$ has a solution.*

Proof. By assumption we have $\mathbb{N} \vDash \varphi_{p,q}$ which by Lemma 13 gives us $\alpha : \mathbb{N} \to \mathbb{N}$ with

$$\alpha \vDash p^* \equiv q^* \implies \widehat{(\mu \circ \alpha)}(p^*) = \widehat{(\mu \circ \alpha)}(q^*) \overset{L.17}{\implies} \mu([\![p]\!]_\alpha) = \mu([\![q]\!]_\alpha).$$

Since over $\mathbb{N}$ the function μ is simply the identity, we conclude $[\![p]\!]_\alpha = [\![q]\!]_\alpha$. ◀

The deductive part of the reduction can be shown analogously to Fact 19, encoding the proofs of all intermediate results as ND derivations. We just list the relevant statements and refer to the Coq code for more detail.

Lemma 22. *For $n, m : \mathbb{N}$, $\mathsf{Q}' \vdash \nu(n + m) \equiv \nu(n) \oplus \nu(m)$ and $\mathsf{Q}' \vdash \nu(n \times m) \equiv \nu(n) \otimes \nu(m)$.*

Lemma 23. *If $p = q$ has a solution α, then we can deduce $\mathsf{Q}' \vdash (p^* \equiv q^*)[\nu \circ \alpha]$.*

Fact 24. *If $p = q$ has a solution then $\mathsf{Q}' \vdash \varphi_{p,q}$.*

Now we have all facts in place to verify the reductions with Theorem 10.

Theorem 25. *$\mathsf{H}_{10} \preceq \mathsf{Q}'$, $\mathsf{H}_{10} \preceq Q$, and $\mathsf{H}_{10} \preceq \mathsf{PA}$.*

Proof. Since $\mathbb{N}$ is a standard model for Q', Q, and PA, the claims follow by Theorem 10 since we have shown the three necessary conditions in Facts 19, 21, and 24. ◀

As a consequence of these reductions, we can conclude incompleteness as follows:

Theorem 26. *Assuming LEM, completeness of any extension $\mathcal{A} \supseteq \mathsf{Q}'$ satisfied by the standard model $\mathbb{N}$ would imply the decidability of the halting problem of Turing machines.*

Proof. By Theorems 10 and 25, Fact 9, and the reductions verified in [20]. ◀

We close this section with a remark on separating models of Q', Q, and PA. For any $n : \mathbb{N}$, the quotient $\mathbb{Z}/n\mathbb{Z}$ is a model of Q'. So in particular Q' admits the trivial model and can hence be completed with $\forall xy.\, x \equiv y$, separating it from both Q and PA since they only admit infinite models and are essentially incomplete. A well-known model separating Q and PA is obtained by extending $\mathbb{N}$ to $\mathbb{N}^\infty$ with a maximal number ∞.

5 ZF Set Theory with Skolem Functions

Turning to set theory, we first work in a rich signature providing function symbols for the axiomatic operations of ZF. Concretely, for the rest of this section we fix the signature

$$\Sigma := (\emptyset,\ \{\_,\_\},\ \bigcup\_,\ \mathcal{P}(\_),\ \omega\ ;\ \_ \equiv \_,\ \_ \in \_)$$

with function symbols denoting the empty set, pairing, union, power set, the set of natural numbers, next to the usual relation symbols for equality and membership. Using such Skolem functions for axiomatic and other definable operations is common practice in set-theoretic literature and eases the definition and verification of the undecidability reduction in our case.

That the undecidability result can be transported to minimal signatures just containing equality and membership, or even just the latter, is subject of the next section.

We do not list all axioms in detail but refer the reader to Appendix B, the Coq code, and standard literature (eg. [35]). The only point worth mentioning again is the representation of axiom schemes as functions $\mathbb{F} \to \mathbb{F}$, for instance by the separation scheme expressed as

$$\lambda\varphi.\, \forall x.\, \exists y.\, \forall z.\, z \in y \;\leftrightarrow\; z \in x \wedge \varphi[x].$$

We then distinguish the following axiomatisations:

- Z' is the list containing extensionality and the specifications of the five function symbols.
- Z is the (infinite) theory obtained by adding all instances of the separation scheme.
- ZF is the theory obtained by further adding all instances of the replacement scheme.

Note that in ZF we do not include the axiom of regularity since this would force the theory classical and would require to extend Coq's type theory even further to obtain a model [23]. Alternatively, one could add the more constructive axiom for ϵ-induction, but instead we opt for staying more general and just leave the well-foundedness of sets unspecified.

Following the general outline for the undecidability proofs in this paper, we first focus on verifying a reduction to the base theory Z' and then extend to the stronger axiomatisations by use of Theorem 10. As a seed problem for this reduction, we could naturally pick just any decision problem since set theory is a general purpose foundation expressive enough for most standard mathematics. However, the concrete choice has an impact on the mechanisation overhead, where formalising Turing machine halting directly is tricky enough in Coq's type theory itself, and even a simple problem like H_{10} used in the previous section would presuppose a modest development of number theory and recursion in the axiomatic framework. We therefore base our reduction to Z' on the Post correspondence problem (PCP) which has a simple inductive characterisation expressing a matching problem given a finite stack S of pairs (s, t) of Boolean strings:

$$\frac{(s,t) \in S}{S \rhd (s,t)} \qquad \frac{S \rhd (u,v) \quad (s,t) \in S}{S \rhd (su, tv)} \qquad \frac{S \rhd (s,s)}{\mathsf{PCP}\, S}$$

Informally, S is used to derive pairs (s, t), written $S \rhd (s, t)$ by repeatedly appending the pairs from the stack componentwise in any order or multitude. The instance S admits a solution, written $\mathsf{PCP}\, S$, if a matching pair (s, s) can be derived by this procedure.

Encoding data like numbers and Booleans in set-theoretic terms is standard, using the usual derived notations for binary union $x \cup y$, singletons $\{x\}$, and ordered pairs (x, y):

- Numbers: $\overline{0} := \emptyset$ and $\overline{n+1} := \overline{n} \cup \{\overline{n}\}$
- Booleans: $\overline{\mathsf{tt}} := \{\emptyset\}$ and $\overline{\mathsf{ff}} := \emptyset$
- Strings: $\overline{b_1, \ldots, b_n} := (\overline{b_1}, (\ldots (\overline{b_n}, \emptyset) \ldots))$
- Stacks: $\overline{S} := \{(\overline{s_1}, \overline{t_1}), \ldots, (\overline{s_m}, \overline{t_m})\}$

Starting with an informal idea, the solvability condition of PCP can be directly expressed in set theory by just asserting the existence of a set encoding a match for S:

$$\exists x.\, (x,x) \in \bigcup_{k \in \omega} \overline{S}^k \quad \text{where} \quad \overline{S}^0 = \overline{S} \quad \text{and} \quad \overline{S}^{k+1} = S \boxtimes \overline{S}^k = \bigcup_{s/t \in S} \{(\overline{s}x, \overline{t}y) \mid (x,y) \in \overline{S}^k\}$$

Unfortunately, formalizing this idea is not straightforward, since the iteration operation $\overline{S}^k$ is described by recursion on set-theoretic numbers $k \in \omega$ missing a native recursion principle akin to the one for type-theoretic numbers $n : \mathbb{N}$. Such a recursion principle can of course be derived but in our case it is simpler to inline the main construction.

The main construction used in the recursion theorem for ω is a sequence of finite approximations f accumulating the first k steps of the recursive equations. Since in our case

we do not need to form the limit of this sequence requiring the approximations to agree, it suffices to ensure that at least the first k steps are contained without cutting off, namely

$$f \gg k := (\emptyset, \overline{S}) \in f \land \forall (l, B) \in f.\, l \in k \to (l \cup \{l\}, S \boxtimes B) \in f$$

where we reuse the operation $S \boxtimes B$ appending the encoded elements of the list S componentwise to the elements of the set B as specified above. Note that this operation is not really definable as a function $\mathbb{L}(\mathbb{B}) \to \mathbb{T} \to \mathbb{T}$ and needs to be circumvented by quantifying over candidate sets satisfying the specification. However, for the sake of a more accessible explanation, we leave this subtlety to the Coq code and continue using $S \boxtimes B$ as a function.

Now solvability of S can be expressed formally as the existence of a functional approximation f of length k containing a match (x, x):

$$\varphi_S := \exists k, f, B, x.\, k \in \omega \land (\forall (l, B), (l, B') \in f.\, B = B') \land f \gg k \land (k, B) \in f \land (x, x) \in B$$

We proceed with the formal verification of the reduction function $\lambda S.\, \varphi_S$ by proving the three facts necessary to apply Theorem 10. Again beginning with the semantic part for clarity, we fix a model $\mathcal{M} \vDash \mathsf{Z}'$ for the next lemmas in preparation of the facts connecting $\mathsf{PCP}\, S$ with $\mathcal{M} \vDash \varphi_S$. We skip the development of basic set theory in $\mathcal{M}$ reviewable in the Coq code and only state lemmas concerned with encodings and the reduction function:

Lemma 27. *Let $n, m : \mathbb{N}$ and $s, t : \mathbb{L}(\mathbb{B})$ be given, then the following hold:*

(i) $\mathcal{M} \vDash \overline{n} \in \omega$

(ii) $\mathcal{M} \vDash \overline{n} \notin \overline{n}$

(iii) $\mathcal{M} \vDash \overline{n} \equiv \overline{m}$ *implies* $n = m$

(iv) $\mathcal{M} \vDash \overline{s} \equiv \overline{t}$ *implies* $s = t$

Proof.

(i) By induction on n, employing the infinity axiom characterising ω.

(ii) Again by induction on n, using the fact that numerals $\overline{n}$ are transitive sets.

(iii) By trichotomy we have $n < m$, $m < n$, or $n = m$ as desired. If w.l.o.g. it were $n < m$, then $\mathcal{M} \vDash \overline{n} \in \overline{m}$ would follow by structural induction on the derivation of $n < m$. But then the assumption $\mathcal{M} \vDash \overline{n} \equiv \overline{m}$ would yield $\mathcal{M} \vDash \overline{n} \in \overline{n}$ in conflict with (ii).

(iv) By induction on the given strings, employing injectivity of the encoding of Booleans. ◀

In order to match the structure of iterated derivations encoded in φ_S, we reformulate $S \triangleright (s, t)$ by referring to the composed derivations S^n of length n, now definable by recursion on $n : \mathbb{N}$ via $S^0 := S$ and $S^{n+1} := S \boxtimes S^n$ reusing the operation $\boxtimes$ for lists as expected.

Lemma 28. *$S \triangleright (s, t)$ iff there is $n : \mathbb{N}$ with $(s, t) \in S^n$.*

Then the iterations S^n can be encoded as set-level functions $f_S^n := \{(\emptyset, \overline{S}), \ldots, (\overline{n}, \overline{S^n})\}$ that are indeed recognised by the model $\mathcal{M}$ as correct approximations:

Lemma 29. *For every $n : \mathbb{N}$ we have $\mathcal{M} \vDash f_S^n \gg \overline{n}$.*

Proof. In this proof we work inside of $\mathcal{M}$ to simplify intermediate statements. For the first conjunct, we need to show that $(\emptyset, \overline{S}) \in f_S^n$ which is straightforward since $(\emptyset, \overline{S}) \in f_S^0$ and $f_S^m \subseteq f_S^n$ whenever $m \leq n$. Regarding the second conjunct, we assume $(k, B) \in f_S^n$ with $k \in \overline{n}$ and need to show $(k \cup \{k\}, S \boxtimes B) \in f_S^n$. From $(k, B) \in f_S^n$ we obtain that there is m with $k = \overline{m}$ and $B = \overline{S^m}$. Then from $\overline{m} \in \overline{n}$ and hence $m < n$ we deduce that also $(\overline{m+1}, \overline{S^{m+1}}) \in f_S^n$. The claim follows since $\overline{m+1} = k \cup \{k\}$ and

$$\overline{S^{m+1}} = \overline{S \boxtimes S^n} = S \boxtimes \overline{S^n} = S \boxtimes B$$

using that the $\boxtimes$ operation on lists respecitively sets interacts well with string encodings. ◀

With these lemmas in place, we can now conclude the first part of the semantic verification.

Fact 30. *If* $\mathsf{PCP}\,S$ *then* $\mathsf{Z}' \vDash \varphi_S$.

Proof. Assuming $\mathsf{PCP}\,S$, there are $s : \mathbb{L}(\mathbb{B})$ and $n : \mathbb{N}$ with $(s,s) \in S^n$ using Lemma 28. Now to prove $\mathsf{Z}' \vDash \varphi_S$ we assume $\mathcal{M} \vDash \mathsf{Z}'$ and need to show $\mathsf{Z}' \vDash \varphi_S$. Instantiating the leading existential quantifiers of φ_S with $\overline{n}$, f_S^n, $\overline{S^n}$, and $\overline{s}$ leaves the following facts to verify:

- $\mathcal{M} \vDash \overline{n} \in \omega$, immediate by (i) of Lemma 27.
- Functionality of f_S^n, straightforward by construction of f_S^n.
- $\mathcal{M} \vDash f_S^n \gg \overline{n}$, immediate by Lemma 29.
- $\mathcal{M} \vDash (\overline{n}, \overline{S^n}) \in f_S^n$, again by construction of f_S^n.
- $\mathcal{M} \vDash (\overline{s}, \overline{s}) \in \overline{S^n}$, by the assumption $(s,s) \in S^n$. ◀

For the converse direction, we again need to restrict to models $\mathcal{M}$ only containing standard natural numbers, i.e. satisfying that any $k \in \omega$ is the numeral $k = \overline{n}$ for some $n : \mathbb{N}$. Then the internally recognised solutions correspond to actual external solutions of PCP.

Lemma 31. *If in a standard model* $\mathcal{M}$ *there is a functional approximation* $f \gg k$ *for* $k \in \omega$ *with* $(k, B) \in f$*, then for all* $p \in B$ *there are* $s, t : \mathbb{L}(\mathbb{B})$ *with* $p = (\overline{s}, \overline{t})$ *and* $S \triangleright (s,t)$.

Proof. Since $\mathcal{M}$ is standard, there is $n : \mathbb{N}$ with $k = \overline{n}$, so we have $f \gg \overline{n}$ and $(\overline{n}, B) \in f$. In any model with $f \gg \overline{n}$ we can show that $(\overline{k}, \overline{S^k}) \in f$ by induction on k, so in particular $(\overline{n}, \overline{S^n}) \in f$ in $\mathcal{M}$. But then by functionality of f it must be $B = \overline{S^n}$, so for any $p \in B$ we actually have $p \in \overline{S^n}$ for which it is easy to extract $s, t : \mathbb{L}(\mathbb{B})$ with $p = (\overline{s}, \overline{t})$ and $(s,t) \in S^n$. We then conclude $S \triangleright (s,t)$ with Lemma 28. ◀

Fact 32. *Every standard model* $\mathcal{M} \vDash \mathsf{Z}'$ *with* $\mathcal{M} \vDash \varphi_S$ *yields* $\mathsf{PCP}\,S$.

Proof. A standard model of Z' with $\mathcal{M} \vDash \varphi_S$ yields a functional approximation $f \gg k$ for $k \in \omega$ with some $(k, B) \in f$ and $(x,x) \in B$. Then by Lemma 31 there are $s, t : \mathbb{L}(\mathbb{B})$ with $(x,x) = (\overline{s}, \overline{t})$ and $S \triangleright (s,t)$. By the injectivity of ordered pairs and string encodings ((iv) of Lemma 27) we obtain $s = t$ and thus $S \triangleright (s,s)$. ◀

Finally, we just record the fact that the semantic argument in Fact 32 can be repeated deductively with an analogous intermediate structure.

Fact 33. *If* $\mathsf{PCP}\,S$ *then* $\mathsf{Z}' \vdash \varphi_S$.

With the three facts verifying φ_S in place, we conclude reductions as follows:

Theorem 34. *We have the following reductions.*

- $\mathsf{PCP} \preceq \mathsf{Z}'$*, provided a standard model of* Z' *exists.*
- $\mathsf{PCP} \preceq \mathsf{Z}$*, provided a standard model of* Z *exists.*
- $\mathsf{PCP} \preceq \mathsf{ZF}$*, provided a standard model of* ZF *exists.*

Proof. By Facts 30, 32, and 33 as well as Theorem 10. ◀

In a previous paper [18] based on Aczel's sets-as-trees interpretation [1, 42, 2], we analyse assumptions necessary to obtain models of higher-order set theories in Coq's type theory. The two relevant axioms concerning the type $\mathcal{T}$ of well-founded trees can be formulated as the extensionality of classes, i.e. unary predicates, on trees (CE), and the existence of a description operator for isomorphism classes $[t]_\approx$ of trees (TD):

$$\mathsf{CE} := \forall (P, P' : \mathcal{T} \to \mathbb{P}).\, (\forall t.\, P\,t \leftrightarrow P'\,t) \to P = P'$$

$$\mathsf{TD} := \exists (\delta : (\mathcal{T} \to \mathbb{P}) \to \mathcal{T}).\, \forall P.\, (\exists t.\, P = [t]_\approx) \to P\,(\delta\,P)$$

Then Theorem 34 can be reformulated as follows.

Corollary 35. CE *implies both* PCP $\preceq$ Z′ *and* PCP $\preceq$ Z*, and* CE $\wedge$ TD *implies* PCP $\preceq$ ZF*.*

Proof. By Fact 5.4 and Theorem 5.9 of [18] CE and CE $\wedge$ TD yield models of higher-order Z and ZF set theory, respectively. It is easy to show that they are standard models and satisfy the first-order axiomatisations Z and ZF. ◀

Note that assuming CE to obtain a model of higher-order Z is unnecessary if we allow the interpretation of equality by any equivalence relation congruent for membership, backed by the fully constructive model given in Theorem 4.6 of [18]. This variant is included in the Coq development but we focus on the simpler case of extensional models in this text.

As a consequence of these reductions, we can conclude the incompleteness of ZF.

Theorem 36. *Assuming* LEM*, completeness of any extension* $\mathcal{A} \supseteq$ Z′ *satisfied by a standard model would imply the decidability of the halting problem of Turing machines.*

Proof. By Corollary 35, Theorem 10, Fact 9, and the reductions verified in [10]. ◀

6 ZF Set Theory without Skolem Functions

We now work in the signature $\tilde{\Sigma} := (\_ \equiv \_, \_ \in \_)$ only containing equality and membership. To express set theory in this syntax, we reformulate the axioms specifying the Skolem symbols used in the previous signature Σ to just assert the existence of respective sets, for instance:

$$\emptyset : \quad \forall x.\, x \notin \emptyset \quad \rightsquigarrow \quad \exists u.\, \forall x.\, x \notin u$$

$$\mathcal{P}(x) : \quad \forall xy.\, y \in \mathcal{P}(x) \leftrightarrow y \subseteq x \quad \rightsquigarrow \quad \forall x.\, \exists u.\, \forall y.\, y \in u \leftrightarrow y \subseteq x$$

In this way we obtain axiomatisations $\tilde{\mathsf{Z}}'$, $\tilde{\mathsf{Z}}$, and $\widetilde{\mathsf{ZF}}$ as the respective counterparts of Z′, Z, and ZF. In this section, we show that these symbol-free axiomatisations admit the same reduction from PCP.

Instead of reformulating the reduction given in the previous section to the smaller signature, which would require us to replace the natural encoding of numbers and strings as terms by a more obscure construction, we define a general translation $\tilde{\varphi} : \mathbb{F}_{\tilde{\Sigma}}$ of formulas $\varphi : \mathbb{F}_{\Sigma}$. We then show that $\tilde{\mathsf{Z}}' \vDash \tilde{\varphi}$ implies $\mathsf{Z}' \vDash \varphi$ (Fact 40) and that $\mathsf{Z}' \vdash \varphi$ implies $\tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ (Fact 43), which is enough to deduce the undecidability of $\tilde{\mathsf{Z}}'$, $\tilde{\mathsf{Z}}$, and $\widetilde{\mathsf{ZF}}$ (Theorem 44).

The informal idea of the translation function is to replace terms $t : \mathbb{T}_{\Sigma}$ by formulas $\varphi_t : \mathbb{F}_{\tilde{\Sigma}}$ characterising the index x_0 to behave like t, for instance:

$$\mathsf{x}_n \rightsquigarrow \mathsf{x}_0 \equiv \mathsf{x}_{n+1} \qquad \emptyset \rightsquigarrow \forall \mathsf{x}_0 \notin \mathsf{x}_1 \qquad \mathcal{P}(t) \rightsquigarrow \exists\, \varphi_t[\mathsf{x}_0; \uparrow^2] \wedge \forall \mathsf{x}_0 \in \mathsf{x}_2 \leftrightarrow \mathsf{x}_0 \subseteq \mathsf{x}_1$$

The formula expressing $\mathcal{P}(t)$ first asserts that there is a set satisfying φ_t (where the substitution $\uparrow^n$ shifts all indices by n) and then characterises x_0 (appearing as x_2 given the two quantifiers) as its power set. Similarly, formulas are translated by descending recursively to the atoms, which are replaced by formulas asserting the existence of characterised sets being in the expected relation, for instance:

$$t \in t' \quad \rightsquigarrow \quad \exists\, \varphi_t[\mathsf{x}_0; \uparrow^2] \wedge \exists\, \varphi_{t'}[\mathsf{x}_0; \uparrow^3] \wedge \mathsf{x}_1 \in \mathsf{x}_0$$

We now verify that the translation $\tilde{\varphi}$ satisfies the two desired facts, starting with the easier semantic implication. To this end, we denote by $\tilde{\mathcal{M}}$ the $\tilde{\Sigma}$-model obtained from a Σ-model $\mathcal{M}$ by forgetting the interpretation of the function symbols not present in $\tilde{\Sigma}$. Then for a model $\mathcal{M} \vDash \mathsf{Z}'$, satisfiability is preserved for translated formulas, given that the term characterisations are uniquely satisfied over the axioms of Z′:

Lemma 37. *Given $\mathcal{M} \vDash \mathsf{Z}'$, $t : \mathbb{T}$, $\rho : \mathbb{N} \to \mathcal{M}$, and $x : \mathcal{M}$ we have $x = \hat{\rho}\, t$ iff $(x; \rho) \vDash_{\tilde{\mathcal{M}}} \varphi_t$.*

Proof. By induction on t with x generalised. We only consider the cases x_n and $\emptyset$:

- We need to show $x = \hat{\rho}\, \mathsf{x}_n$ iff $(x; \rho) \vDash_{\tilde{\mathcal{M}}} \mathsf{x}_0 \equiv \mathsf{x}_{n+1}$ which is immediate by definition.
- First assuming $x = \emptyset$, we need to show that $\forall y.\, y \notin x$, which is immediate since $\mathcal{M}$ satisfies the empty set axiom. Conversely assuming $\forall y.\, y \notin x$ yields $x = \emptyset$ by using the extensionality axiom also satisfied by $\mathcal{M}$. ◀

Lemma 38. *Given $\mathcal{M} \vDash \mathsf{Z}'$, $\varphi : \mathbb{F}$, and $\rho : \mathbb{N} \to \mathcal{M}$ we have $\rho \vDash_{\mathcal{M}} \varphi$ iff $\rho \vDash_{\tilde{\mathcal{M}}} \tilde{\varphi}$.*

Proof. By induction on φ with ρ generalised, all cases but atoms are directly inductive. Considering the case $t \in t'$, we first need to show that if $\hat{\rho}\, t \in \hat{\rho}\, t'$, then there are x and x' with $x \in x'$ satisfying φ_t and $\varphi_{t'}$, respectively. By Lemma 37 the choice $x := \hat{\rho}\, t$ and $x' := \hat{\rho}\, t'$ is enough. Now conversely, if there are such x and x', by Lemma 37 we know that $x = \hat{\rho}\, t$ and $x' = \hat{\rho}\, t'$ and thus conclude $\hat{\rho}\, t \in \hat{\rho}\, t'$. The case of $t \equiv t'$ is analogous. ◀

Then the desired semantic implication follows since pruned models $\tilde{\mathcal{M}}$ satisfy $\tilde{\mathsf{Z}}'$:

Lemma 39. *If $\mathcal{M} \vDash \mathsf{Z}'$ then $\tilde{\mathcal{M}} \vDash \tilde{\mathsf{Z}}'$.*

Proof. We only need to consider the axioms concerned with set operations, where we instantiate the existential quantifiers introduced in $\tilde{\mathsf{Z}}'$ with the respective operations available in $\mathcal{M}$. For instance, to show $\tilde{\mathcal{M}} \vDash \exists u. \forall x.\, x \notin u$ it suffices to show that $\forall x.\, x \notin \emptyset$ in $\tilde{\mathcal{M}}$, which is exactly the empty set axiom satisfied by $\mathcal{M}$. ◀

Fact 40. *$\tilde{\mathsf{Z}}' \vDash \tilde{\varphi}$ implies $\mathsf{Z}' \vDash \varphi$.*

Proof. Straightforward by Lemmas 38 and 39. ◀

We now turn to the more involved deductive verification of the translation, beginning with the fact that $\tilde{\mathsf{Z}}'$ proves the unique existence of sets satisfying the term characterisations:

Lemma 41. *For all $t : \mathbb{T}$ we have $\tilde{\mathsf{Z}}' \vdash \exists\, \varphi_t$ and $\tilde{\mathsf{Z}}' \vdash \varphi_t[x] \to \varphi_t[x'] \to x \equiv x'$.*

Proof. Both claims are by induction on t, the latter with x and x' generalised. The former is immediate for variables and $\emptyset$, we discuss the case of $\mathcal{P}(t)$. By induction we know $\tilde{\mathsf{Z}}' \vdash \exists\, \varphi_t$ yielding a set x simulating t and need to show $\tilde{\mathsf{Z}}' \vdash \exists \exists\, \varphi_t[\mathsf{x}_0; \uparrow^2] \wedge \forall \mathsf{x}_0 \in \mathsf{x}_2 \leftrightarrow \mathsf{x}_0 \subseteq \mathsf{x}_1$. After instantiating the first quantifier with the set u guaranteed by the existential power set axiom for the set x and the second quantifier with x itself, it remains to show $\varphi_t[x]$ and $\forall \mathsf{x}_0 \in u \leftrightarrow \mathsf{x}_0 \subseteq x$ which are both straightforward by the choice of x and u.

The second claim follows from extensionality given that the characterisation φ_t specifies its satisfying sets exactly by their elements. So in fact the axioms concerning the set operations are not even used in the proof of uniqueness. ◀

During translation, substitution of terms can be simulated by substitution of variables:

Lemma 42. *Forall $\varphi : \mathbb{F}$ and $t : \mathbb{T}$ we have $\tilde{\mathsf{Z}}' \vdash \varphi_t[x] \to (\tilde{\varphi}[x] \leftrightarrow \widetilde{\varphi[t]})$.*

Proof. By induction on φ, all cases but the atoms are straightforward, relying on the fact that the syntax translation interacts well with variable renamings in the quantifier cases. The proof for atoms relies on a similar lemma for terms stating that $\varphi_s[y; x]$ and $\varphi_{s[t]}[y]$ are interchangeable whenever $\varphi_t[x]$, then the rest is routine. ◀

The previous lemma is the main ingredient to verify the desired proof transformation:

Fact 43. $\mathsf{Z}' \vdash \varphi$ *implies* $\tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$.

Proof. We prove the more general claim that $\Gamma +\!\!+ \mathsf{Z}' \vdash \varphi$ implies $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ by induction on the first derivation. All rules but the assumption rule (A), $\forall$-elimination (AE), and $\exists$-elimination (EE) are straightforward, we explain the former two.

- If $\varphi \in \Gamma +\!\!+ \mathsf{Z}'$, then either $\varphi \in \Gamma$ or $\varphi \in \mathsf{Z}'$. In the former case we have $\tilde{\varphi} \in \tilde{\Gamma}$, so $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ by (A). Regarding the latter case, we can verify $\tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ for all $\varphi \in \mathsf{Z}'$ by rather tedious derivations given the sheer size of some axiom translations.
- If $\Gamma +\!\!+ \mathsf{Z}' \vdash \varphi[t]$ was derived from $\Gamma +\!\!+ \mathsf{Z}' \vdash \forall\, \varphi$, then by the inductive hypothesis we know $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \forall\, \tilde{\varphi}$. Given Lemma 41 we may assume $\varphi_t[x]$ for a fresh variable x. Then by instantiating the inductive hypothesis to x via (AE) we obtain $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \tilde{\varphi}[x]$ and conclude the claim $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \widetilde{\varphi[t]}$ with Lemma 42. ◀

Now the undecidability of the symbol-free axiomatisations can be established.

Theorem 44. CE *implies both* $\mathsf{PCP} \preceq \tilde{\mathsf{Z}}'$ *and* $\mathsf{PCP} \preceq \tilde{\mathsf{Z}}$, *and* $\mathsf{CE} \wedge \mathsf{TD}$ *implies* $\mathsf{PCP} \preceq \widetilde{\mathsf{ZF}}$.

Proof. Similar to Theorem 10 based on Facts 40 and 43 and the reduction from Section 5. ◀

We conclude this section with a brief observation concerning the further reduced signature $\check{\Sigma} := (\_ \in \_)$, full detail can be found in the Coq development. Since equality is expressible in terms of membership by $x \equiv y := \forall z.\, x \in z \leftrightarrow y \in z$, we can rephrase the above translation to yield formulas $\check{\varphi} : \mathbb{F}_{\check{\Sigma}}$ satisfying the same properties as stated in Facts 40 and 43 for a corresponding axiomatisation $\check{\mathsf{Z}}'$. Moreover, since $\check{\mathsf{Z}}'$ does not refer to primitive equality, we can freely interpret it with the fully constructive model given in Theorem 4.6 of [18] and therefore obtain $\mathsf{PCP} \preceq \check{\mathsf{Z}}'$ without assumptions. This allows us to deduce the undecidability of the Entscheidungsproblem in its sharpest possible form:

Theorem 45. *First-order logic with a single binary relation symbol is undecidable.*

Proof. By Fact 11 and the reduction $\mathsf{PCP} \preceq \check{\mathsf{Z}}'$. ◀

7 Discussion

7.1 General Remarks

In this paper, we have described a synthetic approach to the formalisation and mechanisation of undecidability and incompleteness results in first-order logic. The general approach was then instantiated in two case-studies, one concerned with arithmetic theories in the family of PA as the typical systems considered in the investigation of incompleteness, and another one regarding fragments of ZF set theory as one of the standard foundations of mathematics. The chosen strategy complements the considerably harder to mechanise proofs relying on Gödel sentences, and for ZF the choice of PCP as seed problem instead of H_{10} or PA itself is a slight simplification since only a single recursion needs to be simulated. We use this section for some additional remarks based on the helpful feedback by the anonymous reviewers.

As formally stated in Definition 8, we only consider incompleteness as a property of the *classical* deduction system. This is simply owing to the fact that much of the literature on incompleteness seems focused on classical logic, with a notable exception of the more agnostic treatment in [26]. Although likely weaker in general, incompleteness of the *intuitionistic* deduction system can also be considered a meaningful property and follows in an analogous way. Concretely, a corresponding version of Fact 9 holds for the intuitionistic notion, yielding variants of Theorems 26 and 36 provable without LEM.

In alignment with [11] but in contrast to [12], we define semantic entailment $\mathcal{T} \vDash \varphi$ without restricting to *classical models*, i.e. models that satisfy all first-order instances of LEM. In our constructive meta-theory this relaxation is necessary to be able to use the standard models of PA and ZF, which would only be classical in a classical meta-theory. Leaving $\mathcal{T} \vDash \varphi$ in this sense constructively underspecified seems like a reasonable trade for a more economical usage of LEM.

Similarly, we leave it underspecified whether PA and ZF are seen as classical theories or their intuitionistic counterparts, namely Heyting arithmetic and a variant of intuitionistic set theory, respectively. By the choice not to distinguish these explicitly by LEM as a first-order axiom scheme, we leave it to the deduction system to discriminate between both views while the Tarski-style semantics emphasises the classical interpretation (especially in the presence of LEM). For simplicity, we decided to only speak of PA and ZF in the main body of the text, especially since a discussion of intuitionistic set theories would involve choosing a particular system. While IZF is an extension of Z' close to ZF with collection instead of replacement, the more predicative CZF does not have power sets as included in Z'.

7.2 Coq Mechanisation

Our axiom-free mechanisation contributes 5300loc to the Coq Library of Undecidability Proofs [14], on top of about 1300loc that could be reused from previous developments [12, 18]. Remarkably, the reduction from H_{10} to PA consists of only 700loc while already the initial reduction from PCP to ZF in the skolemised signature is above 1600loc. The remaining 3000loc mostly concern the technically more challenging translations to the sparse signatures of $\tilde{\mathsf{Z}}'$ and $\check{\mathsf{Z}}'$ as well as the use of intensional setoid models for the elimination of CE. By the latter, the given reductions can be verified constructively up to Z while the local assumption of TD remains necessary for full ZF. The development is available on our project page (see link in header) and all statements and some highlighted notations in the PDF version of this paper are systematically hyperlinked with HTML documentation of the code.

Our mechanisation of first-order logic unifies ideas from previous versions [11, 12, 17] and is general enough to be reused in other use cases. Notably, we refrained from including equality as a syntactic primitive to treat both intensional and extensional interpretations without changing the underlying signature. On the other hand, with primitive equality, the extensionality of models would hold definitionally and the deduction system could be extended with the Leibniz rule, making the additional axiomatisation of equality obsolete.

Furthermore, manipulating deductive goals of the form $\Gamma \vdash \varphi$ benefitted a lot from custom tactics, mostly to handle substitution and the quantifier rules. The former tactics approximate the automation provided by the Autosubst 2 framework unfortunately relying on functional extensionality [37] and the latter are based on the named reformulations of (AI) and (EE) given in Section 2.3. We are currently working on a more scalable proof mode for deductive goals including a HOAS input language hiding de Bruijn encodings, implementing a two-level approach in comparison to the one-level compromise proposed by Laurent [21].

7.3 Related Work

We report on other mechanisations concerned with incompleteness and undecidability results in first-order logic. Regarding the former, a fully mechanised proof of Gödel's first incompleteness theorem was first given by Shankar [32] using the Nqthm prover. O'Connor [24] implements the same result fully constructively in Coq, and Paulson [25] provides an Isa-

belle/HOL mechanisation of both incompleteness theorems using the theory of hereditarily finite sets instead of a fragment of PA. Moreover, there are several partial mechanisations [29, 5, 33], and Popescu and Traytel [26] investigate the abstract preconditions of the incompleteness theorems using Isabelle/HOL. With the independence of the continuum hypothesis, Han and van Doorn [15] mechanise a specific instance of incompleteness for ZF in Lean. None of these mechanisations approach incompleteness via undecidability.

Turning to undecidability results, Forster, Kirst, and Smolka [11] mechanise the undecidability of the Entscheidungsproblem in Coq, using a convenient signature to encode PCP, and Kirst and Larchey-Wendling [17] give a Coq mechanisation of Trakhtenbrot's theorem [40], stating the undecidability of finite satisfiability. They also begin with a custom signature for the encoding of PCP but provide the transformations necessary to obtain the undecidability result for the minimal signature containing a single binary relation symbol. We are not aware of any previous mechanisations of the undecidability of PA or ZF.

7.4 Future Work

There are two ways how our incompleteness results (Theorems 26 and 36) could be strengthened. First, the assumption of LEM is only due to the fact that we need soundness, for instance to deduce $\mathsf{Q}' \vDash \varphi_{p,q}$ from $\mathsf{Q}' \vdash_c \varphi_{p,q}$. As done previously [11], it should be possible to employ a Friedman translation to extract $\mathsf{Q}' \vdash_i \varphi_{p,q}$ from $\mathsf{Q}' \vdash_c \varphi_{p,q}$ and hence to obtain $\mathsf{Q}' \vDash \varphi_{p,q}$ constructively. Secondly, that supposed negation-completeness only implies synthetic decidability of a halting problem instead of a provable contradiction could be sharpened by extracting all reduction functions to a concrete model of computation like the weak call-by-value λ-calculus L [13]. Then the actual contradiction of an L-decider for L-halting could be derived.

We plan to continue the work on PA with a constructive analysis of Tennenbaum's theorem [39], stating that no computable non-standard model of PA exists. Translated to the synthetic setting where all functions are computable by construction, this would mean that no non-standard model of PA can be defined in Coq's type theory as long as function symbols are interpreted with type-theoretic functions. It will be interesting to investigate which assumptions are necessary to derive this as a theorem in Coq.

Regarding the reductions to ZF, it should be possible to eliminate the infinite set ω used to simplify the accumulation of partial solutions. Then the fully constructive and extensional standard model of hereditarily finite sets [34] would be available. Further eliminating the power set axiom, segments of this model could be used to obtain a more direct mechanisation of Trakhtenbrot's theorem than the previous one using signature transformations [17].

In general, it would be interesting to find a more elementary characterisation of an undecidable binary relation usable for the sharp formulations of the Entscheidungsproblem and Trakhtenbrot's theorem. This might well work without an intermediate axiomatisation of set theory and express an undecidable decision problem more primitively.

Moreover, by a straightforward extension of the translation in Section 6, one could deduce the conservativity of ZF over $\widetilde{\mathsf{ZF}}$, i.e. that if $\mathsf{ZF} \vdash \varphi$ for φ free of function symbols, then already $\widetilde{\mathsf{ZF}} \vdash \varphi$. This is an instance of the more general fact that first-order logic with definable symbols is conservative, which would be a worthwhile addition to our development.

Finally, we plan to mechanise similar undecidability and incompleteness results for second-order logic. Since second-order PA is categorical, in particular the incompleteness of any sound and enumerable deduction system for second-order logic would then follow easily.

References

1 Peter Aczel. The type theoretic interpretation of constructive set theory. In *Studies in Logic and the Foundations of Mathematics*, volume 96, pages 55–66. Elsevier, 1978.

2 Bruno Barras. Sets in Coq, Coq in sets. *Journal of Formalized Reasoning*, 3(1):29–48, 2010.

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

4 Thomas Braibant and Damien Pous. An efficient Coq tactic for deciding Kleene algebras. In *International Conference on Interactive Theorem Proving*, pages 163–178. Springer, 2010.

5 Alan Bundy, Fausto Giunchiglia, Adolfo Villafiorita, and Toby Walsh. An incompleteness theorem via abstraction, 1996. Technical report.

6 Alonzo Church et al. A note on the Entscheidungsproblem. *J. Symb. Log.*, 1(1):40–41, 1936.

7 Nicolaas G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae (Proceedings)*, 75(5):381–392, 1972.

8 John Doner and Wilfrid Hodges. Alfred Tarski and decidable theories. *The Journal of symbolic logic*, 53(1):20–35, 1988.

9 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 *LIPIcs*, pages 21:1–21:19, Dagstuhl, Germany, 2021.

10 Yannick Forster, Edith Heiter, and Gert Smolka. Verification of PCP-related computational reductions in Coq. In *International Conference on Interactive Theorem Proving*, pages 253–269. Springer, 2018.

11 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.

12 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.

13 Yannick Forster and Fabian Kunze. A certifying extraction with time bounds from Coq to call-by-value lambda calculus. In John Harrison, John O'Leary, and Andrew Tolmach, editors, *10th International Conference on Interactive Theorem Proving (ITP 2019)*, volume 141 of *LIPIcs*, pages 17:1–17:19, Dagstuhl, Germany, 2019.

14 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*, New Orleans, LA, United States, 2020. URL: `https://github.com/uds-psl/coq-library-undecidability`.

15 Jesse 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.

16 David Hilbert and Wilhelm Ackermann. *Grundzüge der theoretischen Logik*. Springer, 1928.

17 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 (IJCAR 2020), Paris, France*, Paris, France, 2020. Springer.

18 Dominik Kirst and Gert Smolka. Large model constructions for second-order ZF in dependent type theory. *Certified Programs and Proofs - 7th International Conference, CPP 2018, Los Angeles, USA, 2018*, January 2018.

19 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.

20 Dominique Larchey-Wendling and Yannick Forster. Hilbert's tenth problem in Coq. In *4th International Conference on Formal Structures for Computation and Deduction*, volume 131 of *LIPIcs*, pages 27:1–27:20, February 2019.

21 Olivier Laurent. An anti-locally-nameless approach to formalizing quantifiers. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 300–312, 2021.

22 Petar Maksimović and Alan Schmitt. HOCore in Coq. In *International Conference on Interactive Theorem Proving*, pages 278–293. Springer, 2015.

23 John Myhill. Some properties of intuitionistic Zermelo-Frankel set theory. In *Cambridge Summer School in Mathematical Logic*, pages 206–231. Springer, 1973.

24 Russell O'Connor. Essential incompleteness of arithmetic verified by Coq. In Joe Hurd and Tom Melham, editors, *Theorem Proving in Higher Order Logics*, pages 245–260, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

25 Lawrence C. Paulson. A mechanised proof of Gödel's incompleteness theorems using Nominal Isabelle. *Journal of Automated Reasoning*, 55(1):1–37, 2015.

26 Andrei Popescu and Dmitriy Traytel. A formally verified abstract account of Gödel's incompleteness theorems. In *International Conference on Automated Deduction*, pages 442–461. Springer, 2019.

27 Emil L. Post. Recursively enumerable sets of positive integers and their decision problems. *bulletin of the American Mathematical Society*, 50(5):284–316, 1944.

28 Mojżesz Presburger and Dale Jabcquette. On the completeness of a certain system of arithmetic of whole numbers in which addition occurs as the only operation. *History and Philosophy of Logic*, 12(2):225–233, 1991.

29 Art Quaife. Automated proofs of Löb's theorem and Gödel's two incompleteness theorems. *Journal of Automated Reasoning*, 4(2):219–231, 1988.

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

31 Steven Schäfer, Gert Smolka, and Tobias Tebbi. Completeness and decidability of de Bruijn substitution algebra in Coq. In *Proceedings of the 2015 Conference on Certified Programs and Proofs*, pages 67–73. ACM, 2015.

32 Natarajan Shankar. *Proof-checking metamathematics*. The University of Texas at Austin, 1986. PhD Thesis.

33 Wilfried Sieg and Clinton Field. Automated search for Gödel's proofs. In *Deduction, Computation, Experiment*, pages 117–140. Springer, 2008.

34 Gert Smolka and Kathrin Stark. Hereditarily finite sets in constructive type theory. In *Interactive Theorem Proving - 7th International Conference, ITP 2016, Nancy, France, August 22-27, 2016*, volume 9807 of *LNCS*, pages 374–390. Springer, 2016.

35 Raymond M. Smullyan and Melvin Fitting. *Set theory and the continuum problem*. Dover Publications, 2010.

36 Matthieu Sozeau, Abhishek Anand, Simon Boulier, Cyril Cohen, Yannick Forster, Fabian Kunze, Gregory Malecha, Nicolas Tabareau, and Théo Winterhalter. The MetaCoq Project. *Journal of Automated Reasoning*, 2020.

37 Kathrin Stark, Steven Schäfer, and Jonas Kaiser. Autosubst 2: reasoning with multi-sorted de Bruijn terms and vector substitutions. In *International Conference on Certified Programs and Proofs*, pages 166–180. ACM, 2019.

38 The Coq Development Team. The Coq Proof Assistant, version 8.12.0, 2020. `doi:10.5281/zenodo.4021912`.

39 Stanley Tennenbaum. Non-Archimedean models for arithmetic. *Notices of the American Mathematical Society*, 6(270):44, 1959.

40 Boris A. Trakhtenbrot. The impossibility of an algorithm for the decidability problem on finite classes. *Dokl. Akad. Nok. SSSR*, 70(4):569–572, 1950.

41 Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.

42 Benjamin Werner. Sets in types, types in sets. In *Theoretical Aspects of Computer Software*, pages 530–546. Springer, Berlin, Heidelberg, 1997.

A Deduction Systems

Intuitionistic natural deduction $\Gamma \vdash_i \varphi$ is defined inductively by the following rules:

$$\frac{\varphi \in \Gamma}{\Gamma \vdash \varphi}\ \mathrm{C} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi}\ \mathrm{E} \qquad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ \mathrm{II} \qquad \frac{\Gamma \vdash \varphi \to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \varphi}\ \mathrm{IE}$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}\ \mathrm{CI} \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi}\ \mathrm{CE_1} \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}\ \mathrm{CE_2}$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi}\ \mathrm{DI_1} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}\ \mathrm{DI_2} \qquad \frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \theta \quad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta}\ \mathrm{DE}$$

$$\frac{\Gamma[\uparrow] \vdash \varphi}{\Gamma \vdash \forall \varphi}\ \mathrm{AI} \qquad \frac{\Gamma \vdash \forall \varphi}{\Gamma \vdash \varphi[t]}\ \mathrm{AE} \qquad \frac{\Gamma \vdash \varphi[t]}{\Gamma \vdash \exists \varphi}\ \mathrm{EI} \qquad \frac{\Gamma \vdash \exists \varphi \quad \Gamma[\uparrow], \varphi \vdash \psi[\uparrow]}{\Gamma \vdash \psi}\ \mathrm{EE}$$

The classical variant $\Gamma \vdash_c \varphi$ adds all instances of the Peirce rule $((\varphi \to \psi) \to \varphi) \to \varphi$.

B Axioms of Set Theory

We list the ZF axioms over the signature $\Sigma := (\emptyset, \{\_,\_\}, \bigcup \_, \mathcal{P}(\_), \omega\ ;\ \_ \equiv \_, \_ \in \_)$:

Structural axioms

Extensionality: $\forall xy.\, x \subseteq y \to y \subseteq x \to x \equiv y$

Set operations

Empty set: $\forall x.\, x \notin \emptyset$

Unordered pair: $\forall xyz.\, z \in \{x, y\} \leftrightarrow x \equiv y \vee x \equiv z$

Union: $\forall xy.\, y \in \bigcup x \leftrightarrow \exists z \in x.\, y \in z$

Power set: $\forall xy.\, y \in \mathcal{P}(x) \leftrightarrow y \subseteq x$

Infinity: $(\emptyset \in \omega \wedge \forall x.\, x \in \omega \to x \cup \{x\} \in \omega)$
$\wedge\, (\forall y.\, (\emptyset \in y \wedge \forall x.\, x \in y \to x \cup \{x\} \in y) \to \omega \subseteq y)$

Axiom schemes

Separation: $\lambda\varphi.\, \forall x.\, \exists y.\, \forall z.\, z \in y \leftrightarrow z \in x\ \wedge \varphi[x]$

Replacement $\lambda\varphi.\, (\forall xyy'.\, \varphi[x, y] \to \varphi[x, y'] \to y \equiv y')$
$\to \forall x.\, \exists y.\, \forall z.\, z \in y \leftrightarrow \exists u \in x.\, \varphi[u, z]$

Equality axioms

Reflexivity: $\forall x.\, x \equiv x$

Symmetry: $\forall xy.\, x \equiv y \to y \equiv x$

Transitivity: $\forall xyz.\, x \equiv y \to y \equiv z \to x \equiv z$

Congruence: $\forall xx'yy'.\, x \equiv x' \to y \equiv y' \to x \in y \to x' \in y'$

The core axiomatisation Z′ contains extensionality and the set operation axioms, Z adds the separation scheme, and ZF also adds the replacement scheme. The equality axioms are added when working with the deduction system or in an intensional model.