Journal of Automated Reasoning (2023) 67:13
https://doi.org/10.1007/s10817-022-09647-x

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

## Extended Version

**Dominik Kirst**[1] · **Marc Hermes**[2]

Received: 10 September 2021 / Accepted: 28 June 2022 / Published online: 12 March 2023


**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) as well as ZF and related finitary set theories, 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 the considered set theories are supplemented 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 set theories formulated using only membership and no equality symbol, we obtain the undecidability of first-order logic with a single binary relation.

**Keywords** Undecidability · Synthetic computability · First-order logic · Incompleteness · Peano arithmetic · ZF set theory · Constructive type theory · Coq

## 1 Introduction

Being among the mainstream formalisms to underpin mathematics, first-order logic (FOL) 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* [18],

✉ Dominik Kirst
kirst@cs.uni-saarland.de

[1] Saarland Informatics Campus, Saarland University, Saarbrücken, Germany

[2] Department of Mathematics, Saarland University, Saarbrücken, Germany

namely the search for a decision procedure determining the formulas $\varphi$ that are valid in all interpretations, usually written $\vDash \varphi$. With their groundbreaking work in the 1930s, Turing [47] and Church [7] established that such a general decision procedure cannot exist. However, this outcome can change if one considers validity of $\varphi$ 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 [33] and Tarski contributed further instances with his work on Boolean algebras, real-closed ordered fields, and Euclidean geometry in the 1940s [9].

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 [44] to mechanise the undecidability of PA and ZF and related finitary set theories, 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 axiom-free 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, $\mu$-recursive function, or untyped $\lambda$-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 [4, 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 Coq Library of Undecidability Proofs [13] 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 pleasantly feasible [11, 22]. 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 $\varphi$ 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 [32] 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 [4, 11], all axiomatisations shown synthetically undecidable in the present paper are incomplete in the sense that their completeness would imply the decidability of the halting problem (for Turing machines). These algorithmic observations complement the otherwise notoriously hard to mechanise incompleteness proofs based on Gödel sentences [29, 30].

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 classical assumptions [23]), enabling proofs of their undecidability. In fact, we sharpen the results

for fragments $\mathsf{Q}'$ and $\mathsf{Z}'$ even strictly below Robinson arithmetic $\mathsf{Q}$ and Zermelo set theory $\mathsf{Z}$, 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 $\mathsf{Q}'$, $\mathsf{Q}$, $\mathsf{PA}$, $\mathsf{Z}'$, $\mathsf{Z}$, and $\mathsf{ZF}$(-regularity), regarding both Tarski semantics and natural deduction.[1]
- 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 (Strategy 10), we obtain synthetic undecidability and incompleteness for all axiomatisations extending the fragments $\mathsf{Q}'$ and $\mathsf{Z}'$ with respect to standard models.

This extended version of [21] adds the following contributions:

- We eliminate the assumption of excluded middle in the treatment of $\mathsf{PA}$ by means of a general Gödel-Gentzen-Friedman translation (Sect. 5).
- We mechanise direct and indirect reductions to various finitary set theories not requiring or actively refuting infinite sets (Sect. 8).
- We extend on the signature transformation employed for set theory without function symbols to obtain conservativity results (Lemma 53 - Fact 56).
- We analyse the abstract preconditions necessary for the synthetic approach to undecidability and incompleteness of arbitrary formalisms (Sect. 9).

After a preliminary discussion of constructive type theory, synthetic undecidability, and first-order logic in Sect. 2, we proceed with the general results relating undecidabilitity, incompleteness, and consistency of first-order axiom systems in Sect. 3. This is followed by the case studies concerning arithmetical axiomatisations (Sects. 4 and 5), set theory with (Sect. 6) and without (Sect. 7) Skolem functions, as well as finitary set theories (Sect. 8). We conclude with the abstract analysis of undecidability and incompleteness of arbitrary formalisms (Sect. 9) and with a discussion of the Coq mechanisation as well as related and future work Sect. 10.

## 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.

[1] The Coq development is available at www.ps.uni-saarland.de/extras/axiomatisations-ext and systematically hyperlinked with every definition and fact in the PDF version of this document.

### 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* ($\mathbb{B} := \mathtt{tt} \mid \mathtt{ff}$), *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 $f\, l$ for application of the pointwise 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 [4, 35] 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) [41] 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 $f : X \to \mathbb{B}$ with $P\, x$ iff $f\, x = \mathtt{tt}$,
- $P$ is *enumerable* if there exists $f : \mathbb{N} \to \mathbb{O}\,(X)$ with $P\, x$ iff $\exists n.\, f\, n = \ulcorner x \urcorner$.

Note that it is commonly accepted practice to mechanise decidability results in this synthetic sense (e.g. [5, 27, 36]). 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 classical models, cf. [48]).

The approximation chosen in the Coq Library of Undecidability Proofs [13] 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 ($\mathsf{H}_{10}$) [25] 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 FOL

We now review the representation of first-order syntax, semantics, and natural deduction systems developed in previous papers [11, 15, 22]. 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 $|P|$:

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

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

In the chosen de Bruijn representation [8], 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 encodings to the Coq development. A formula with all occurring variables bound by a quantifier is called *closed*.

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

**Definition 4** A *model* $\mathcal{M}$ consists of a type $D$ with functions $f^{\mathcal{M}} : D^{|f|} \to D$ and $P^{\mathcal{M}} : D^{|P|} \to \mathbb{P}$ interpreting the symbols in $\Sigma$. We often use $\mathcal{M}$ itself to refer to its domain $D$. Given an assignment $\rho : \mathbb{N} \to \mathcal{M}$ we define *term evaluation* $\hat{\rho} : \mathbb{T} \to \mathcal{M}$ and *formula satisfaction* $\rho \vDash \varphi$ by

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

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

If a model $\mathcal{M}$ satisfies a formula $\varphi$ for all variable assignments $\rho$, 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 $\psi$ 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 $\mathbb{L}(\mathbb{F}) \to \mathbb{F} \to \mathbb{P}$. 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 of the two systems are standard and listed in Appendix A, here we only highlight the quantifier rules depending on the de Bruijn encoding

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

where $\varphi[\sigma]$ denotes the *capture-avoiding instantiation* of a formula $\varphi$ with a *parallel substitution* $\sigma : \mathbb{N} \to \mathbb{T}$, where the substitution $\uparrow$ maps $n$ to $\mathsf{x}_{n+1}$, 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 [15]). 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 $\mathsf{x}_n$ is *fresh*, i.e. does not occur in any formula of $\Gamma$.

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 define $\mathcal{T} \vdash \varphi$ as derivability from $\mathcal{T}$ augmented with the standard axioms characterising $\equiv$ as an equivalence relation congruent for the symbols in $\Sigma$.

## 3 Undecidable and Incomplete 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 $\Sigma$ for the remainder of this section and begin by introducing the central notion of axiom systems formally.

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

Any 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. [15]), we mostly consider a combined notion of Tarski semantics and intuitionistic deduction (reusing the $\preceq$-notation):

**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 $\mathsf{PA}$ and $\mathsf{ZF}$, since for them it is still feasible to verify $\mathcal{A} \vdash f\, x$ given $P\, x$ by hand without appealing to completeness and the easier verification of $\mathcal{A} \vDash f\, x$.

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 (see Appendix A). □

The second observation is a synthetic version of (negation-)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 Sect. 10.1 for a discussion.

**Definition 8** We call $\mathcal{A}$ *complete* if for all closed $\varphi$, $\mathcal{A} \vdash_c \varphi$ or $\mathcal{A} \vdash_c \neg\varphi$.

**Fact 9** *If $\mathcal{A}$ is complete and $\mathcal{A} \nvdash_c \bot$, then $\mathcal{A}^{\vdash_c}$ is decidable on closed formulas. Hence, 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 $\varphi$. This follows by enumerability of $\vdash_c$ and since by 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 [14, 24] or an undecidability assumption [11]) or extraction to a concrete model (e.g. a weak call-by-value $\lambda$-calculus [12]). 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 of the reductions verified in this paper:

1. We choose an undecidable seed problem $P : X \to \mathbb{P}$ easy to encode in the target axiomatisation. This will be $\mathsf{H}_{10}$ for PA and PCP for ZF.
2. We define the translation function $X \to \mathbb{F}$ mapping instances $x : X$ to formulas $\varphi_x$ 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 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 that $A$, and any sound $\mathcal{B} \supseteq A$ are undecidable and incomplete:

**Strategy 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:

(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. If we additionally assume 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 $\varphi_x$. 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, with LEM we obtain $P \preceq \mathcal{B}^{\vdash_c}$ since then $\mathcal{B} \vdash_c \varphi_x$ implies $\mathcal{B} \vDash \varphi_x$. □

Of course (i) follows from (iii) via soundness, so the initial semantic verification could be eliminated from Strategy 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 instructive fact that the decision problem for finite axiomatisations $A$ reduces to the general Entscheidungsproblem of first-order logic concerning validity and provability in the empty context [18].

**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 $\varphi$ 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]. More general variants of this insight can be formulated as follows:

**Fact 12** *Let $A$ be finite and $\mathcal{B}$ be an arbitrary axiomatisation.*

1. If $A \vdash \mathcal{B}$, then $A \preceq \mathcal{B}$.
2. If $\mathcal{B} \subseteq A$, then $A \preceq \mathcal{B}$.
3. $\mathcal{B} \cup A \preceq \mathcal{B}$.

***Proof*** All witnessed by the reduction $\lambda\varphi.\ \bigwedge A \to \varphi$, (2) is a special case of (1). □

## 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 finite list $\mathsf{Q}'$ 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 function on formulas:

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

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

Turning to undecidability, Hilbert's 10th problem ($\mathsf{H}_{10}$) 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, $\mathsf{H}_{10}$ 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:

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

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

$$[[\mathrm{a}_n]]_\alpha := n \quad [[\mathrm{var}\ k]]_\alpha := \alpha\ k \quad [[\mathrm{add}\ p\ q]]_\alpha := [[p]]_\alpha + [[q]]_\alpha \quad [[\mathrm{mult}\ p\ q]]_\alpha := [[p]]_\alpha \times [[q]]_\alpha$$

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

**Definition 13** We define $\nu(n) : \mathbb{T}$ by $\nu(0) := O$ and $\nu(n+1) := S(\nu(n))$.

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

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

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 20, 22 and 25) necessary for Strategy 10 with the following lemma about existential formulas:

**Lemma 14** *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 $x_1; \dots; x_N; \rho \vDash \varphi$ for all $\rho$, 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 $\rho$ 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 $\varphi$ has at most $N$ free variables, $\rho'$ can be exchanged with any other $\tau : \mathbb{N} \to \mathcal{M}$. □

By Lemma 14, 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 $\mathsf{Q}'$. To this end, we fix a model $\mathcal{M} \vDash \mathsf{Q}'$ for the next definitions and lemmas.

**Definition 15** We define $\mu(n) : D$ by $\mu(0) := O^{\mathcal{M}}$ and $\mu(n+1) := S^{\mathcal{M}}(\mu(n))$.

The axioms in $\mathsf{Q}'$ are sufficient to prove that $\mu$ is a homomorphism.

**Lemma 16** *$\mu(n + m) = \mu(n) \oplus^{\mathcal{M}} \mu(m)$ and $\mu(n \times m) = \mu(n) \otimes^{\mathcal{M}} \mu(m)$.*

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

**Lemma 17** *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 $\mathsf{Q}'$ model by applying $\mu$. The homomorphism property of $\mu$ 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 18** *For any $p$ 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 19** *If $p = q$ has solution $\alpha$, then in any $\mathsf{Q}'$ model $(\mu \circ \alpha) \vDash p^* \equiv q^*$.*

***Proof*** $\mu([\![p]\!]_\alpha) = \mu([\![q]\!]_\alpha) \overset{L.18}{\Longrightarrow} \widehat{(\mu \circ \alpha)}(p^*) = \widehat{(\mu \circ \alpha)}(q^*) \overset{\text{def.}}{\Longrightarrow} (\mu \circ \alpha) \vDash p^* \equiv q^*$ □

**Fact 20** *If $p = q$ has a solution, then $\mathsf{Q}' \vDash \varphi_{p,q}$.*

***Proof*** Let $\alpha$ be the solution of $p = q$, then $(\mu \circ \alpha) \vDash p^* \equiv q^*$ holds by Corollary 19 and since $\exists^N p^* \equiv q^*$ is closed by construction, the goal follows by Lemma 14. □

For the converse direction, we employ the type $\mathbb{N}$ as standard model.

**Lemma 21** *$\mathbb{N}$ is a model of $\mathsf{Q}'$, $\mathsf{Q}$, and $\mathsf{PA}$.*

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

**Fact 22** *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 14 gives us $\alpha : \mathbb{N} \to \mathbb{N}$ with

$$\alpha \vDash p^* \equiv q^* \overset{\text{def.}}{\Longrightarrow} \widehat{(\mu \circ \alpha)}(p^*) = \widehat{(\mu \circ \alpha)}(q^*) \overset{L.18}{\Longrightarrow} \mu([\![p]\!]_\alpha) = \mu([\![q]\!]_\alpha).$$

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

The deductive part of the reduction can be shown analogously to Fact 20, encoding the proofs of all intermediate results as syntactic derivations. We just list the relevant statements and refer to the Coq code for more detail.

**Lemma 23** *$\mathsf{Q}' \vdash \nu(n + m) \equiv \nu(n) \oplus \nu(m)$ and $\mathsf{Q}' \vdash \nu(n \times m) \equiv \nu(n) \otimes \nu(m)$.*

**Lemma 24** *If $p = q$ has a solution $\alpha$, then we can deduce $\mathsf{Q}' \vdash (p^* \equiv q^*)[\nu \circ \alpha]$.*

**Fact 25** *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 Strategy 10.

**Theorem 26** *$\mathsf{H}_{10} \preceq \mathsf{Q}'$, $\mathsf{H}_{10} \preceq Q$, and $\mathsf{H}_{10} \preceq \mathsf{PA}$.*

***Proof*** Since $\mathbb{N}$ is a standard model for $\mathsf{Q}'$, $\mathsf{Q}$, and $\mathsf{PA}$, the claims follow by Strategy 10 since we have shown the three necessary conditions in Facts 20, 22 and 25. □

As a consequence of the reductions, we can directly conclude incompleteness appealing to LEM. Note that in Sect. 5 we explain how this conclusion can be made constructively.

**Theorem 27** *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.*

***Proof*** By Strategy 10 as in Theorem 26, with Fact 9 and the reductions in [25]. □

In fact, all axiomatisations satisfied by $\mathbb{N}$ are undecidable and incomplete:

**Fact 28** *$\mathsf{H}_{10} \preceq \mathcal{A}$ for any axiomatisation $\mathcal{A}$ satisfied by the standard model $\mathbb{N}$.*

***Proof*** By Strategy 10 as in Theorem 26 we obtain $\mathsf{H}_{10} \preceq \mathcal{A} \cup \mathsf{Q}'$ and by Fact 12 we obtain $\mathcal{A} \cup \mathsf{Q}' \preceq \mathcal{A}$. Thus $\mathsf{H}_{10} \preceq \mathcal{A}$ by transitivity. □

We close this section with a few remarks about the theories $\mathsf{Q}'$, $\mathsf{Q}$, and $\mathsf{PA}$. The theory $\mathsf{Q}'$ is trivially incomplete under $\mathsf{LEM}$: using soundness of classical deduction, we have $\mathsf{Q}' \nvdash_c \forall xy.\, x = y$ because of the standard model $\mathbb{N}$ and $\mathsf{Q}' \nvdash_c \neg\forall xy.\, x = y$ because of the trivial model. Similarly, the formula $\forall x.\, Sx \neq x$ is independent in $\mathsf{Q}$, for instance violated by the model $\mathbb{N}^\infty$ extending $\mathbb{N}$ with a maximal number $\infty$. Note that these models in particular show that the theories $\mathsf{Q}'$, $\mathsf{Q}$, and $\mathsf{PA}$ are all distinct.

## 5 Eliminating the Law of Excluded Middle

We can strengthen the result of Theorem 27 and remove its reliance on $\mathsf{LEM}$ by utilising a combination of the double negation and Friedman translations [16]. Given any signature $\Sigma = (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$ we add a new 0-ary predicate $F$ to $\mathcal{P}_\Sigma$, giving us the new signature $\Sigma^F := (\mathcal{F}_\Sigma, \mathcal{P}_\Sigma \cup \{F\})$. This way of setting up the Friedman transform is easier to mechanise compared to the syntactic version where $\bot$ is replaced by a formula, and sufficient for our purpose here.

**Definition 29** We recursively define the $F$-translation $(\cdot)^F : \mathbb{F}_\Sigma \to \mathbb{F}_{\Sigma^F}$ by:

$$\begin{array}{lll} \bot^F := F & (\varphi \to \psi)^F := \varphi^F \to \psi^F & (\forall\varphi)^F := \forall\varphi^F \\ (P\vec{t})^F := \dot\neg\dot\neg(P\vec{t}) & (\varphi \wedge \psi)^F := \varphi^F \wedge \psi^F & (\exists\varphi)^F := \dot\neg\dot\neg\exists\varphi^F \\ & (\varphi \vee \psi)^F := \dot\neg\dot\neg(\varphi^F \vee \psi^F) & \end{array}$$

where $\dot\neg\varphi$ is short for $\varphi \to F$. We extend $(\cdot)^F$ to contexts $\Gamma$ as expectable.

We will state the crucial results concerning the $F$-translation with respect to minimal natural deduction $\Gamma \vdash_m \varphi$, which is natural deduction $\vdash_i$ without the explosion rule and restricted to formulas without the $\bot$ symbol.

**Lemma 30** *For any formula $\varphi$ we have $\vdash_m \dot\neg\dot\neg\varphi^F \to \varphi^F$.*

***Proof*** By induction on the size of $\varphi$. □

**Lemma 31** *For any formula $\varphi$ and context $\Gamma$, if $\Gamma \vdash_c \varphi$ then $\Gamma^F \vdash_m \varphi^F$.*

***Proof*** By induction on the deduction $\Gamma \vdash_c \varphi$, some cases need Lemma 30. □

**Definition 32** Given a proposition $P : \mathbb{P}$ and model $\mathcal{M}$ of the signature $\Sigma$, we can extend $\mathcal{M}$ to a model $\mathcal{M}^P$ of the extended signature $\Sigma^F$ by setting $F^{\mathcal{M}} := P$ and following the interpretation of $\mathcal{M}$ in all other cases. We will then write $\mathcal{M} \vDash \mathcal{T}^F$ to express that for every $\Gamma \subseteq \mathcal{T}$ and $P$ we have $\mathcal{M}^P \vDash \Gamma^F$.

We now apply the $F$-translation to the particular case of the $\mathsf{PA}$ signature to derive an improved version of Theorem 27, eliminating the usage of $\mathsf{LEM}$.

**Lemma 33** *If $\mathcal{M}^P \vDash (\varphi_{p,q})^F$ then $\mathcal{M}^P \vDash \dot\neg\dot\neg\varphi_{p,q}$.*

***Proof*** By $\mathcal{M}^P \vDash \exists^N \dot\neg\dot\neg(p^* \equiv q^*) \to \dot\neg\dot\neg\exists^N p^* \equiv q^*$, proved inductively on $N$. □

**Theorem 34** *Any axiomatisation $\mathcal{A} \supseteq \mathsf{Q}'$ with $\mathbb{N} \vDash \mathcal{A}^F$ witnesses $\mathsf{H}_{10} \preceq \mathcal{A}^{\vdash_c}$. Hence, its completeness would imply the decidability of the halting problem.*

***Proof*** First we will show $\mathsf{H}_{10} \preceq \mathcal{A}^{\vdash_c}$, by verifying that $\varphi_{p,q}$ is a reduction, where the first part of the verification follows from Fact 25. In the converse we are given $\Gamma \subseteq \mathcal{A}$ with $\Gamma \vdash_c \varphi_{p,q}$ and need to find a solution for $p = q$ or equivalently (Fact 22) need to show $\mathbb{N} \vDash \varphi_{p,q}$. Utilising Lemma 31 we get $\Gamma^F \vdash_m (\varphi_{p,q})^F$ which by soundness gives $\mathcal{M}^P \vDash \bigwedge \Gamma^F \to (\varphi_{p,q})^F$ in every model $\mathcal{M}^P$. Since $\mathbb{N} \vDash \mathcal{A}^F$ we have $\mathbb{N}^P \vDash \Gamma^F$ and therefore $\mathbb{N}^P \vDash (\varphi_{p,q})^F$. By Lemma 33 this gives us $\mathbb{N}^P \vDash \dot{\neg}\dot{\neg}\varphi_{p,q}$, which reduces to $((\mathbb{N} \vDash \varphi_{p,q}) \to P) \to P$. The model with $P := \mathbb{N} \vDash \varphi_{p,q}$ then proves that $\mathbb{N} \vDash \varphi_{p,q}$.

Secondly, we can show that $\mathcal{A}$ is consistent (with respect to $\vdash_c$) by the fact that $\mathcal{A} \vdash_c \bot$ together with Lemma 31 and soundness implies $\mathbb{N}^P \vDash \bot^F$, which reduces to $\bot$ in the model with $P := \bot$. Therefore by Fact 9, completeness of $\mathcal{A}$ would imply the decidability of $\mathsf{H}_{10}$ and thus also of the halting problem. □

## 6 ZF Set Theory with Skolem Functions

Turning to set theory, we first work in a signature providing function symbols for the operations of ZF. So 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. [40]). 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:

- $\mathsf{Z}'$ contains extensionality and the specifications of the function symbols.
- $\mathsf{Z}$ is obtained by adding all instances of the separation scheme.
- ZF is 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 [28]. Alternatively, one could add the more constructive axiom for $\epsilon$-induction, but instead we opt for staying more general and just leave the well-foundedness of sets unspecified. So in particular we do not rule out the addition of the anti-foundation axiom [2].

Following the general outline for the undecidability proofs in this paper, we first focus on verifying a reduction to the base theory $\mathsf{Z}'$ and then extend to the stronger axiomatisations

by use of Strategy 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 $\mathsf{H}_{10}$ 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 $\mathsf{Z}'$ 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. $S$ admits a solution, written $\mathsf{PCP}\, S$, if a matching pair $(s, s)$ can be derived.

Encoding data like numbers and Booleans in set theory is standard, using usual 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: $\overline{\mathtt{tt}} := \{\emptyset\}$ and $\overline{\mathtt{ff}} := \emptyset$
- Strings: $\overline{b_1, ..., b_n} := (\overline{b_1}, (...(\overline{b_n}, \emptyset)...))$
- Stacks: $\overline{S} := \{(\overline{s_1}, \overline{t_1}), ..., (\overline{s_m}, \overline{t_m})\}$

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

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

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 $\omega$ 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 stack $S$ component-wise to the elements of the set $B$ as specified above. Note that this operation is not definable as a function $\mathbb{L}(\mathbb{L}(\mathbb{B}) \times \mathbb{L}(\mathbb{B})) \to \mathbb{T} \to \mathbb{T}$ 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 the notation $S \boxtimes B$.

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 Strategy 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 35** *Let $n, m : \mathbb{N}$ and $s, t : \mathbb{L}(\mathbb{B})$ be given, then the following hold:*

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

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

***Proof***

(i) By induction on $n$, employing the infinity axiom characterising $\omega$.
(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 $\overline{s}$. □

In order to match the structure of iterated derivations encoded in $\varphi_S$, we reformulate $S \rhd (s, t)$ 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 36** *$S \rhd (s, t)$ iff there is $n : \mathbb{N}$ with $(s, t) \in S^n$.*

Then $S^n$ 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 37** *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 $\boxtimes$ on lists respectively sets interacts well with string encodings. □

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

**Fact 38** *If* PCP $S$ *then* $\mathsf{Z}' \vDash \varphi_S$.

***Proof*** Assuming PCP $S$, there are $s : \mathbb{L}(\mathbb{B})$ and $n : \mathbb{N}$ with $(s, s) \in S^n$ using Lemma 36. Now to prove $\mathsf{Z}' \vDash \varphi_S$ we assume $\mathcal{M} \vDash \mathsf{Z}'$ and need to show $\mathcal{M} \vDash \varphi_S$. Instantiating the leading existential quantifiers of $\varphi_S$ 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 35.
- Functionality of $f_S^n$, straightforward by construction of $f_S^n$.
- $\mathcal{M} \vDash f_S^n \gg \overline{n}$, immediate by Lemma 37.
- $\mathcal{M} \vDash (\overline{n}, \overline{S^n}) \in f_S^n$, 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 39** *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* $S \rhd (s, t)$.

***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 $S \rhd (s, t)$ with Lemma 36. □

**Fact 40** *Every standard model* $\mathcal{M} \vDash \mathsf{Z}'$ *with* $\mathcal{M} \vDash \varphi_S$ *yields* PCP $S$.

***Proof*** A standard model of $\mathsf{Z}'$ 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 39 there are $s, t : \mathbb{L}(\mathbb{B})$ with $(x, x) = (\overline{s}, \overline{t})$ and $S \rhd (s, t)$. By the injectivity of ordered pairs and string encodings ((iv) of Lemma 35) we obtain $s = t$ and thus $S \rhd (s, s)$. □

Finally, we just record the fact that the semantic argument in Fact 40 can be repeated deductively with an analogous intermediate structure.

**Fact 41** *If* PCP $S$ *then* $\mathsf{Z}' \vdash \varphi_S$.

With the three facts verifying $\varphi_S$, we conclude reductions as follows:

**Theorem 42** *We have the following reductions.*

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

***Proof*** By Facts 38, 40 and 41 as well as Strategy 10. □

In a previous paper [23] based on Aczel's sets-as-trees interpretation [1, 3, 48], 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 42 can be reformulated as follows.

**Corollary 43** *Assuming* CE *implies both* PCP $\preceq$ Z′ *and* PCP $\preceq$ Z*, and assuming both* CE *and* TD *implies* PCP $\preceq$ ZF.

***Proof*** By Fact 5.4 and Theorem 5.9 of [23] 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 [23]. This variant is included in the Coq development but we focus on the simpler case of extensional models in this text.

By these reductions, we can conclude the incompleteness of ZF.

**Theorem 44** *Assuming* LEM*, completeness of any extension* $\mathcal{A} \supseteq$ Z′ *satisfied by a standard model would imply the decidability of the halting problem.*

***Proof*** By Corollary 43, Strategy 10, Fact 9, and the reductions verified in [10]. □

In principle, it should be possible to derive a constructive version of Theorem 44 using the same technique as in Theorem 34. However, the reduction formula $\varphi_S$ we use for the undecidability of set theory is much more complex than the one for Peano arithmetic and not immediately in the necessary syntactic fragment applicable to the Friedman translation. We therefore leave a constructivisation of Theorem 44 as future work.

## 7 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 $\Sigma$ 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 $\widetilde{\mathsf{Z}}'$, $\widetilde{\mathsf{Z}}$, and $\widetilde{\mathsf{ZF}}$ as the respective counterparts of $\mathsf{Z}'$, $\mathsf{Z}$, and $\mathsf{ZF}$. 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 $\widetilde{\mathsf{Z}}' \vDash \tilde{\varphi}$ implies $\mathsf{Z}' \vDash \varphi$ (Fact 48) and that $\mathsf{Z}' \vdash \varphi$ implies $\widetilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ (Fact 51), which is enough to deduce the undecidability of $\widetilde{\mathsf{Z}}'$, $\widetilde{\mathsf{Z}}$, and $\widetilde{\mathsf{ZF}}$ (Theorem 52).

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 $\mathsf{x}_0$ to behave like $t$, for instance:

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

The formula expressing $\mathcal{P}(t)$ first asserts that there is a set satisfying $\varphi_t$ (where the substitution $\uparrow^n$ shifts all indices by $n$) and then characterises $\mathsf{x}_0$ (appearing as $\mathsf{x}_2$ 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:

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

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 $\widetilde{\mathcal{M}}$ the $\tilde{\Sigma}$-model obtained from a $\Sigma$-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 $\mathsf{Z}'$:

**Lemma 45** *$x = \hat{\rho}\, t$ iff $(x;\rho) \vDash_{\widetilde{\mathcal{M}}} \varphi_t$ in all models $\mathcal{M} \vDash \mathsf{Z}'$.*

***Proof*** By induction on $t$ with $x$ generalised. We consider the cases $\mathsf{x}_n$ and $\emptyset$:

- We need to show $x = \hat{\rho}\, \mathsf{x}_n$ iff $(x;\rho) \vDash_{\widetilde{\mathcal{M}}} \mathsf{x}_0 \equiv \mathsf{x}_{n+1}$ 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 46** *$\rho \vDash_{\mathcal{M}} \varphi$ iff $\rho \vDash_{\widetilde{\mathcal{M}}} \tilde{\varphi}$ in all models $\mathcal{M} \vDash \mathsf{Z}'$.*

***Proof*** By induction on $\varphi$ with $\rho$ 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 $\varphi_t$ and $\varphi_{t'}$, respectively. By Lemma 45 the choice $x := \hat{\rho}\, t$ and $x' := \hat{\rho}\, t'$ is enough. Now conversely, if there are such $x$ and $x'$, by Lemma 45 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 semantic implication follows since pruned models $\widetilde{\mathcal{M}}$ satisfy $\widetilde{\mathsf{Z}}'$:

**Lemma 47** *If $\mathcal{M} \vDash \mathsf{Z}'$ then $\widetilde{\mathcal{M}} \vDash \widetilde{\mathsf{Z}}'$.*

***Proof*** We only need to consider the axioms concerned with set operations, where we instantiate the existential quantifiers introduced in $\widetilde{\mathsf{Z}'}$ with the respective operations available in $\mathcal{M}$. For instance, to show $\widetilde{\mathcal{M}} \vDash \exists u.\, \forall x.\, x \notin u$ 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 48** *$\widetilde{\mathsf{Z}'} \vDash \tilde{\varphi}$ implies $\mathsf{Z}' \vDash \varphi$.*

***Proof*** Straightforward by Lemmas 47 and 46. □

We now turn to the more involved deductive verification of the translation, beginning with the fact that $\widetilde{\mathsf{Z}'}$ proves the unique existence of sets satisfying the term characterisations of terms $t \;:\; \mathbb{T}$ in the set-theoretic signature:

**Lemma 49** *For all $t \;:\; \mathbb{T}$ we have $\widetilde{\mathsf{Z}'} \vdash \exists\, \varphi_t$ and $\widetilde{\mathsf{Z}'} \vdash \varphi_t[x] \to \varphi_t[x'] \to x \equiv x'$.*

***Proof*** Both claims are by induction on $t$, the latter with $x$ and $x'$ generalised. The former is immediate for variables and $\emptyset$, so here we just discuss the case of $\mathcal{P}(t)$. By induction we know $\widetilde{\mathsf{Z}'} \vdash \exists\, \varphi_t$ yielding a set $x$ simulating $t$ and need to show $\widetilde{\mathsf{Z}'} \vdash \exists \exists\, \varphi_t[\mathsf{x}_0; \uparrow^2] \wedge \forall\, \mathsf{x}_0 \in \mathsf{x}_2 \leftrightarrow \mathsf{x}_0 \subseteq \mathsf{x}_1$. 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 $\varphi_t$ 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, term can be simulated by variables:

**Lemma 50** *For all $\varphi \;:\; \mathbb{F}$ and $t \;:\; \mathbb{T}$ we have $\widetilde{\mathsf{Z}'} \vdash \varphi_t[x] \to (\tilde{\varphi}[x] \leftrightarrow \widetilde{\varphi[t]})$.*

***Proof*** By induction on $\varphi$, 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 $\varphi_t[x]$, the rest is routine. □

This is the main ingredient to verify the desired proof transformation:

**Fact 51** *$\mathsf{Z}' \vdash \varphi$ implies $\widetilde{\mathsf{Z}'} \vdash \tilde{\varphi}$.*

***Proof*** We prove the more general claim that $\Gamma{+}{+}\mathsf{Z}' \vdash \varphi$ implies $\widetilde{\Gamma}{+}{+}\widetilde{\mathsf{Z}'} \vdash \tilde{\varphi}$ by induction on the first derivation. All rules but the assumption rule (A), ∀-elimination (AE), and ∃-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 $\tilde{\Gamma}{+}{+}\widetilde{\mathsf{Z}'} \vdash \tilde{\varphi}$ by (A). Regarding the latter case, we can verify $\widetilde{\mathsf{Z}'} \vdash \tilde{\varphi}$ 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 $\tilde{\Gamma}{+}{+}\widetilde{\mathsf{Z}}' \vdash \forall\, \tilde{\varphi}$. Given Lemma 49 we may assume $\varphi_t[x]$ for a fresh variable $x$. Then by instantiating the inductive hypothesis to $x$ via (AE) we obtain $\tilde{\Gamma}{+}{+}\widetilde{\mathsf{Z}}' \vdash \tilde{\varphi}[x]$ and conclude the claim $\tilde{\Gamma}{+}{+}\widetilde{\mathsf{Z}}' \vdash \widetilde{\varphi[t]}$ with Lemma 50. □

Now we obtain the undecidability of the symbol-free axiomatisations.

**Theorem 52** *Assuming* CE *implies both* $\mathsf{PCP} \preceq \widetilde{\mathsf{Z}}'$ *and* $\mathsf{PCP} \preceq \widetilde{\mathsf{Z}}$*, and assuming both* CE *and* TD *implies* $\mathsf{PCP} \preceq \widetilde{\mathsf{ZF}}$.

***Proof*** As Strategy 10, using Facts 48 and 51 and the reduction from Sect. 6. □

Note that Fact 51 almost yields *deductive conservativity*, i.e. the fact that if $\mathsf{Z}'$ proves a symbol-free formula over $\widetilde{\Sigma}$ then so does $\widetilde{\mathsf{Z}}'$. The missing lemma is that from $\widetilde{\mathsf{Z}}'$ such a formula $\varphi$ is provably equivalent to its translation $\tilde{\varphi}$ (after tacitly embedding $\varphi$ into the full signature $\Sigma$):

**Lemma 53** $\widetilde{\mathsf{Z}}' \vdash \varphi \leftrightarrow \tilde{\varphi}$ *for all* $\varphi$ *over* $\widetilde{\Sigma}$.

***Proof*** By induction on $\varphi$, all composite cases are trivial. For the atom $x \in y$, we have to show its equivalence to $\exists x'.\, x \equiv x' \wedge \exists y'.\, y \equiv y' \wedge x \in y$, similarly for $x \equiv y$. □

We can then record conservativity results as follows:

**Fact 54** *If* $\mathsf{Z}'/\mathsf{Z}/\mathsf{ZF}$ *proves a formula* $\varphi$ *over* $\widetilde{\Sigma}$*, then so does* $\widetilde{\mathsf{Z}}'/\widetilde{\mathsf{Z}}/\widetilde{\mathsf{ZF}}$.

***Proof*** First let $\mathsf{Z}' \vdash \varphi$. Then by Fact 51 we have $\widetilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ and thus $\widetilde{\mathsf{Z}}' \vdash \varphi$ by Lemma 53.

If we instead suppose $\mathsf{Z} \vdash \varphi$, we have in particular $\mathsf{Z}'{+}{+}\Gamma \vdash \varphi$, where $\Gamma$ contains finitely many instances of the separation scheme. Then by the generalised goal used in the proof of Fact 51 also $\widetilde{\mathsf{Z}}'{+}{+}\tilde{\Gamma} \vdash \tilde{\varphi}$ and therefore $\widetilde{\mathsf{Z}}'{+}{+}\tilde{\Gamma} \vdash \varphi$ again using Lemma 53. We hence conclude $\widetilde{\mathsf{Z}} \vdash \varphi$ since every translated instance of separation for a formula $\psi$ can be proved from the respective instance for $\tilde{\psi}$ available in $\widetilde{\mathsf{Z}}$.

The case for ZF is analogous by further decomposing into the finitely many used instances of the replacement scheme. □

For the sake of completeness, we also establish the converse directions. To this end, we first verify a deductive counterpart of Lemma 47:

**Lemma 55** $\mathsf{Z}' \vdash \widetilde{\mathsf{Z}}'$*, i.e.* $\mathsf{Z}'$ *proves every axiom from* $\widetilde{\mathsf{Z}}'$ *(embedded into* $\Sigma$*).*

***Proof*** By instantiating every existentially formulated axiom from $\widetilde{\mathsf{Z}}'$ with the respective symbol available in $\mathsf{Z}'$. □

**Fact 56** *If* $\widetilde{\mathsf{Z}}'/\widetilde{\mathsf{Z}}/\widetilde{\mathsf{ZF}}$ *proves a formula* $\varphi$ *over* $\widetilde{\Sigma}$*, then so does* $\mathsf{Z}'/\mathsf{Z}/\mathsf{ZF}$.

***Proof*** If $\widetilde{\mathsf{Z}}' \vdash \varphi$, we obtain the same deduction if we consider both $\widetilde{\mathsf{Z}}'$ and $\varphi$ embedded into the full signature. Then by Lemma 55 we can conclude that $\mathsf{Z}' \vdash \varphi$.

The respective results for $\widetilde{\mathsf{Z}}$ and $\widetilde{\mathsf{ZF}}$ follow by similar decompositions regarding the axiom schemes as used in the proof of Fact 54. □

Note that in the absence of unique choice there is no direct proof for *semantic conservativity*, i.e. the fact that if $\mathsf{Z}'$ validates a symbol-free formula over $\widetilde{\Sigma}$ then so does $\widetilde{\mathsf{Z}}'$, since this would involve constructing a $\Sigma$-model from a $\widetilde{\Sigma}$-model only existentially exhibiting the set operations.

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 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 48 and 51 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 [23] 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 57** *FOL with a single binary relation symbol is undecidable.*

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

## 8 Finitary Set Theories

In this section, we consider various finitary set theories, i.e. axiomatisations of set theory that do not guarantee infinite sets or do even refute their existence. Given our setting, the undecidability and incompleteness of such systems can be established either by indirectly reducing from set theories such as $\mathsf{Z}'$ or by modifying the direct reduction function $\mathsf{PCP} \preceq \mathsf{Z}'$. We discuss both of these strategies where applicable.

A first way to axiomatise finite set theory is to work in the full signature used in Sect. 6 and simply leave the set $\omega$ unspecified. Then on top, one can add an axiom ruling out any inductive sets like $\omega$, i.e. sets containing $\emptyset$ and being closed under successors $x \cup \{x\}$.

- $\mathsf{FZ}'$ denotes $\mathsf{Z}'$ without the axioms specifying $\omega$ as the least inductive set.
- $\mathsf{FZ}' + \neg\mathsf{Inf}$ denotes $\mathsf{FZ}'$ plus the axiom that no set is inductive.

That $\mathsf{FZ}'$ as a mere subset of $\mathsf{Z}'$ is undecidable follows immediately by Fact 12:

**Fact 58** $\mathsf{Z}' \preceq \mathsf{FZ}'$ *and therefore, provided* CE*, also* $\mathsf{PCP} \preceq \mathsf{FZ}'$.

***Proof*** By (2) of Fact 12 and Corollary 43. □

However, this direct result is unsatisfactory by the reliance on the extensional standard model $\mathcal{T}$ of $\mathsf{Z}'$ requiring CE and containing infinite sets. So in order to show $\mathsf{FZ}' + \neg\mathsf{Inf}$ undecidable and dispense with CE, we have to rework the reduction $\mathsf{PCP} \preceq \mathsf{Z}'$ from Sect. 6 to avoid mention of $\omega$ such that the constructive model of hereditarily finite sets [39] can be employed.

In this model, the numerals are exactly the hereditarily transitive sets (i.e. sets $x$ that are transitive, meaning $y \subseteq x$ for all $y \in x$, and every element of $x$ is transitive, written $\mathsf{HT}(x)$), allowing us to modify the reduction formula $\varphi_S$ given a PCP-instance as follows:

$$\varphi_S := \exists k, f, B, x.\, k \in \omega \wedge f \gg k \wedge \ldots \quad \Rightarrow \quad \psi_S := \exists k, f, B, x.\, \mathsf{HT}(k) \wedge f \gg k \wedge \ldots$$

Note that the bound $k \in \omega$ was only used to express that $k$ is a natural number such that (at least in standard models) the approximation $f \gg k$ corresponds to a faithful accumulation of PCP-solutions. This bound can be replaced by any defining property of numerals in the intended model and in the present case, $\mathsf{HT}(x)$ is particularly easy to express.

By according modification of the proofs for $\varphi_S$ we can verify the new reduction $\psi_S$ with respect to all standard models, i.e. models where every hereditarily transitive set is a numeral:

**Lemma 59** *The following facts about $\psi_S$ hold:*

1. If PCP $S$ then $\mathsf{FZ}' \vDash \psi_S$.
2. Every standard model $\mathcal{M} \vDash \mathsf{FZ}'$ with $\mathcal{M} \vDash \psi_S$ yields PCP $S$.
3. PCP $S$ then $\mathsf{FZ}' \vdash \psi_S$.

***Proof*** Analogous to Facts 38, 40 and 41, using the fact that $\mathsf{HT}(\overline{n})$ for all $n : \mathbb{N}$. □

Following the construction from [39], adopted more recently for, [22], a model $\mathcal{T}_2$ of $\mathsf{FZ}'$ can be obtained by taking the inductive type of binary trees quotiented by tree equivalence and implementing the set operations by suitable tree manipulations. In particular, this model is standard in the above sense and does not contain inductive sets:

**Lemma 60** *$\mathcal{T}_2$ is a standard model of $\mathsf{FZ}' + \neg\mathsf{Inf}$.*

***Proof*** To establish that $\mathcal{T}_2$ is standard, we show that for every $x : \mathcal{T}_2$ we can compute a number $n_x : \mathbb{N}$ such that $x = \overline{n_x}$. By induction on the well-foundedness of $x$ we may assume that every element $y \in x$ is a numeral $\overline{n_y}$. Since $x$ is finite, we can compute a bound $n$ such that $n_y < n$ for all $y \in x$. Then we can obtain that $x$ is a numeral (and in fact compute $n_x$) since $x$ is a transitive subset of the numeral $\overline{n}$ by induction on $n$.

Regarding the second claim, suppose $x$ were inductive. By finiteness of $x$ we obtain the cardinality $N$ of distinct elements in $x$. But since $x$ is inductive, it must contain the set of the first $N + 1$ numerals that are distinct by construction, yielding a contradiction. □

So we can conclude the undecidability of $\mathsf{FZ}'$ and $\mathsf{FZ}' + \neg\mathsf{Inf}$ as usual:

**Theorem 61** $\mathsf{PCP} \preceq \mathsf{FZ}'$ *and* $\mathsf{PCP} \preceq \mathsf{FZ}' + \neg\mathsf{Inf}$.

***Proof*** By applying Strategy 10 to Lemmas 59 and 60. □

An alternative, more incisive formulation of finitary set theory just axiomatises the empty set in addition to the adjunction operation $\{x\} \cup y$ (usually definable from union and pairing) [20], i.e. we work in the signature

$$\Sigma_{\mathsf{PS}} := (\emptyset, \_.\_ ; \_ \equiv \_, \_ \in \_)$$

where the term $x.y$ is enforced to behave like $\{x\} \cup y$ by the axiom

$$\forall z.\, z \in x.y \leftrightarrow z \equiv x \vee z \in y.$$

Moreover, to rule out infinite sets, one can require an induction scheme on top:

$$\lambda\varphi.\, \varphi[\emptyset] \to (\forall xy.\, \varphi[x] \to \varphi[y] \to \varphi[x.y]) \to \forall x.\, \varphi[x]$$

- PS denotes the axioms characterising $\emptyset$ and $x.y$ as well as extensionality.
- PS + Ind denotes PS plus all intances of the induction scheme.

We again begin with the indirect argument to establish undecidability of the core axiomatisation PS still compatible with $\mathsf{Z}'$. First note that, while the usual ZF-operations can define adjunction, the converse does not hold since the ZF-operations are strictly stronger on infinite models. We can therefore not directly translate formulas in the ZF-signature to the new signature $\Sigma_{\mathsf{PS}}$. Instead, the translation has to go through the function-free signature $\widetilde{\Sigma} := (\_ \equiv \_, \_ \in \_)$ used in Sect. 7, reusing the verified translation $\tilde{\varphi}$.

**Fact 62** $\mathsf{PCP} \preceq \mathsf{PS}$

***Proof*** We use the reduction formula $\varphi_S^{\mathsf{PS}} := \bigwedge \widetilde{\mathsf{Z}}' \to \tilde{\varphi}_S$ tacitly embedding the translated formulas from $\widetilde{\mathsf{Z}}'$ and $\tilde{\varphi}_S$ in $\widetilde{\Sigma}$ into the signature $\Sigma_{\mathsf{PS}}$. Then the sufficient facts are that $\mathsf{PCP}\, S$ implies $\mathsf{PS} \vdash \varphi_S^{\mathsf{PS}}$ and that $\mathsf{PS} \vDash \varphi_S^{\mathsf{PS}}$ implies $\mathsf{PCP}\, S$.

Regarding the former, from $\mathsf{PCP}\, S$ we obtain $\widetilde{\mathsf{Z}}' \vdash \tilde{\varphi}_S$ from Facts 51 and 41. So in particular $\vdash \bigwedge \widetilde{\mathsf{Z}}' \to \tilde{\varphi}_S$ and by weakening (and correctness of the tacit embedding) $\mathsf{PS} \vdash \varphi_S^{\mathsf{PS}}$.

Regarding the latter, suppose $\mathsf{PS} \vDash \varphi_S^{\mathsf{PS}}$. The (intensional) standard model $\mathcal{T}$ from Facts 38 interprets the full ZF-signature, so in particular $\Sigma_{\mathsf{PS}}$ and the axioms of PS. We therefore obtain that $\mathcal{T} \vDash \varphi_S^{\mathsf{PS}}$. Then by Lemmas 46 and 47 we can deduce that $\mathcal{T}$ (now equipped with the full ZF-structure again) satisfies $\varphi_S$ and conclude $\mathsf{PCP}\, S$ with Fact 40.

□

As with Fact 58 before, this indirect method does not extend to the axiomatisation PS + Ind, which is not satisfied by the standard model $\mathcal{T}$. We therefore sketch the direct reduction from PCP obtained by further modifying the formula $\psi_S$, full detail is given in the Coq formalisation.

First, the encodings of numbers and strings is mostly unaffected since the adjunction operation is exactly the natural successor function and can define unordered pairs $\{x, y\}$ by $x.y.\emptyset$, from which we obtained the ordered pairs used for strings. Secondly, the only other usage of a ZF-function in $\psi_S$ is the (binary) union used to implement the operation $S \boxtimes B$ recursively, which can be replaced by any set enforced to behave accordingly. Thus we obtain a formula $\psi_S^{\mathsf{PS}}$ in the signature $\Sigma_{\mathsf{PS}}$ that we can verify to capture PCP as usual:

**Lemma 63** *The following facts about $\psi_S^{\mathsf{PS}}$ hold:*

1. If $\mathsf{PCP}\, S$ then $\mathsf{PS} \vDash \psi_S^{\mathsf{PS}}$.
2. Every standard model $\mathcal{M} \vDash \mathsf{PS}$ with $\mathcal{M} \vDash \psi_S^{\mathsf{PS}}$ yields $\mathsf{PCP}\, S$.

3. If PCP $S$ then $\mathsf{PS} \vdash \psi_S^{\mathsf{PS}}$.

***Proof*** Analogous to Lemma 59 with the expectable differences regarding the altered data encodings and the elimination of binary unions. □

**Lemma 64** *$\mathcal{T}_2$ is a standard model of* PS + Ind.

***Proof*** That $\mathcal{T}_2$ is standard was already part of Lemma 60 and that it models PS was shown in [40]. They also established the higher-order induction principle

$$\forall P : \mathcal{T}_2 \to \mathbb{P}.\, P\,\emptyset \to (\forall xy.\, P\,x \to P\,y \to P\,(x.y)) \to \forall x.\, P\,x$$

which is easily seen to entail the first-order induction scheme. □

**Theorem 65** PCP $\preceq$ PS *and* PCP $\preceq$ PS + Ind.

***Proof*** By applying Strategy 10 to Lemmas 63 and 64. □

We conclude with a formulation of PS in the binary signature $\check{\Sigma} := (\_ \in \_)$ introduced in Sect. 7. As done with $\mathsf{Z}'$ to obtain $\check{\mathsf{Z}}'$, we can replace the two axioms from PS specifying $\emptyset$ and $x.y$ by existentially quantified versions, express equality via membership, and hence obtain the axiomatisation $\check{\mathsf{PS}}$ over $\check{\Sigma}$. This is a particularly compact system showing a single binary relation symbol undecidable, by virtue of the following reduction:

**Fact 66** $\check{\mathsf{Z}}' \preceq \check{\mathsf{PS}}$ *and thus also* PCP $\preceq \check{\mathsf{PS}}$.

***Proof*** To obtain $\check{\mathsf{Z}}' \preceq \check{\mathsf{PS}}$ we use (1) of Fact 12, so we have to show $\check{\mathsf{Z}}' \vdash \check{\mathsf{PS}}$. The only axiom of $\check{\mathsf{PS}}$ not already present in $\check{\mathsf{Z}}'$ is the existential specification of adjunction, which can be established by the existential specification of union and pairing available in $\check{\mathsf{Z}}'$. The full reduction PCP $\preceq \check{\mathsf{PS}}$ is obtained by composition with the reduction PCP $\preceq \check{\mathsf{Z}}'$ underlying Theorem 57. □

## 9 Abstract Undecidability and Incompleteness

We conclude the technical part of this paper by isolating the synthetic arguments underlying Fact 9 and Strategy 10, abstracting from the concrete formalism of FOL. This abstraction is in the spirit of Popescu and Traytel's [31] analysis of the abstract preconditions for Gödel's two incompleteness theorems. Given our computational approach, much less internal structure like substitution or numerals needs to be assumed, at the cost of essential incompleteness and Gödel's second incompleteness theorem remaining out of reach.

Overwriting all notation from before, our base setup is to assume an arbitrary discrete type $\mathbb{F}$ representing formulas as well as an enumerable predicate $\lambda\varphi : \mathbb{F}.\ \vdash \varphi$ considered the provable formulas. We do not have to commit to $\mathbb{F}$ only containing a specific sort of formulas (e.g. the closed formulas) or to $\vdash$ being defined over a particular context (e.g. an axiomatisation of arithmetic) or coming in a specific flavour (e.g. intuitionistic or classical).

If we add a reasonably well-behaved negation operation, we obtain an abstract version of the fact that negation-completeness implies decidability:

**Fact 67** *We assume a negation operation* $\neg : \mathbb{F} \to \mathbb{F}$ *as follows:*

- Discriminability: given $\varphi$ it is decidable if $\varphi$ is a negation $\neg\psi$ for some $\psi$.
- Injectivity: we have $\varphi = \psi$ whenever $\neg\varphi = \neg\psi$.
- Consistency: there is no $\varphi$ with both $\vdash \varphi$ and $\vdash \neg\varphi$.

Then if $\vdash$ is complete (i.e. either $\vdash \varphi$ or $\vdash \neg\varphi$ for all $\varphi$), then it is decidable.

***Proof*** As in the proof of Fact 9 we use Post's theorem, leaving us to show logical decidability and co-enumerability of provability (given enumerability by assumption):

- Given $\varphi$, to (logically) decide whether $\vdash \varphi$ or $\nvdash \varphi$ is the case, we analyse completeness for $\varphi$. In the non-trivial case where $\vdash \neg\varphi$ we obtain $\nvdash \varphi$ by consistency.
- For co-enumerability, by completeness and consistency it suffices to enumerate $\lambda\varphi. \vdash \neg\varphi$ instead of $\lambda\varphi. \nvdash \varphi$. This is obtained by the enumerator of $\vdash$, using discriminability to check for each $\varphi$ if it is a negation, and injectivity for the correctness proof. □

If instead of a negation operation we add an abstract notion of (standard) models, we obtain an abstract undecidability result analogous to Strategy 10:

**Fact 68** *We assume a type* $\mathbb{M}$ *of models together with the following data:*

- Satisfaction: a relation $\mathcal{M} \vDash \varphi$ inducing validity $\vDash \varphi$ as $\mathcal{M} \vDash \varphi$ for all $\mathcal{M}$.
- Soundness: all provable formulas are valid, i.e. $\vdash \varphi$ implies $\vDash \varphi$.
- Standardness: a predicate $S : \mathbb{M} \to \mathbb{P}$ with at least one standard model $S\,\mathcal{M}$.

If we further assume $P : X \to \mathbb{P}$ and $F : X \to \mathbb{F}$ satisfying

- Whenever $P\,x$ holds, we have a derivation $\vdash F\,x$, and
- Whenever $\mathcal{M} \vDash F\,x$ in a standard model $S\,\mathcal{M}$, we obtain $P\,x$,

then the function $F$ induces reductions $P \preceq (\lambda\varphi. \vdash \varphi)$ and $P \preceq (\lambda\varphi. \vDash \varphi)$.

***Proof*** The assumed standard model justifies that $P\,x$ whenever $\vDash F\,x$. We hence obtain the two reductions, with soundness used for the missing directions. □

Note that if we extend the setting of Fact 68 with the negation operation from Fact 67, we arrive at the conclusion that completeness of $\vdash$ would entail the decidability of $P$.

It is easy to instantiate Fact 68 to obtain Strategy 10 concerning first-order axiomatisations $\mathcal{B}$. We simply let $\mathbb{F}$ be the first-order formulas, $\vdash$ the formulas (intuitionistically) provable from $\mathcal{B}$, and $\mathbb{M}$ be the type of first-order models $\mathcal{M}$ with environments $\rho$ such that $\rho \vDash \mathcal{B}$. Then the remaining assumptions of Strategy 10 imply the assumptions of Fact 68.

Slightly more involved (at least on mechanisation level) is the instantiation of Fact 68 to Fact 9, since this time we pick $\mathbb{F}$ as the type of closed first-order formulas, to which we have to adopt the negation operation and the (classical) deduction system as well as the discreteness and enumerability proofs for arbitrary formulas.

Although these comments only show the applicability of our abstract analysis to the case of first-order logic as examined in this paper, we remark that Facts 67 and 68 could

as well be instantiated to extended formalisms such as second- or higher-order logic, or systems based on completely different primitives such as dependent type theories.

## 10 Discussion

### 10.1 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 to case-studies concerned with arithmetical theories in the family of PA as the typical systems considered in the investigation of incompleteness, and with various formulations of 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 $\mathsf{H}_{10}$ 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 [32]. Although perhaps 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 27 and 44 provable without LEM. Employing the negative translation, incompleteness of classical systems could then be considered from the perspective of intuitionistic systems.

In alignment with [11] but in contrast to [15], 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 leans towards 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′.

### 10.2 Coq Mechanisation

Our axiom-free mechanisation contributes about 10k lines of code (loc) to the Coq Library of Undecidability Proofs [13], on top of about 1500loc that could be reused from previous developments [15, 23]. Remarkably, the axiomatisation, undecidability, and incompleteness

of PA add up to only 800loc, while already the initial reduction from PCP to ZF in the skolemised signature is above 1800loc. The remaining development is mostly concerned with the signature reduction for ZF (2500loc) and the material on finitary set theories (3000loc). Both contain files with very similar proofs, especially the reduction files for $\mathsf{Z}'$ and $\mathsf{FZ}'$ are nearly identical and therefore it should be possible to reduce the development size by reorganisation (at the cost of a less transparent presentation). The abstract development outlined in Sect. 9 is below 300loc, including the instantiation to FOL.

Our mechanisation of first-order logic unifies ideas from previous versions [11, 15, 22] 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 [42] and the latter are based on the named reformulations of (AI) and (EE) given in Sect. 2.3. We are currently working on a more scalable proof mode for deductive goals including a HOAS input language hiding de Bruijn encodings [19], implementing a two-level approach in comparison to the one-level compromise proposed by Laurent [26].

### 10.3 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 [37] using the Nqthm prover. O'Connor [29] implements the same result fully constructively in Coq, and Paulson [30] provides an Isabelle/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 [6, 34, 38], and Popescu and Traytel [31] investigate the abstract preconditions of the incompleteness theorems using Isabelle/HOL. With the independence of the continuum hypothesis, Han and van Doorn [17] 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 [22] give a Coq mechanisation of Trakhtenbrot's theorem [46], 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 small signature containing a single binary relation symbol. We are not aware of any previous mechanisations of the undecidability of PA or ZF.

### 10.4 Future Work

There are two ways how our incompleteness results (Theorems 27 and 44) could be strengthened. First, while we were able to eliminate the use of LEM in the case of PA (Sect. 5), it is unclear whether the same technique applies to the concrete reduction formulas used for ZF

and the related systems. It might be necessary to reformulate (and streamline) the reduction to make the argument feasible for mechanisation. 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 $\lambda$-calculus $\mathsf{L}$ [12]. Then the actual contradiction of an $\mathsf{L}$-decider for $\mathsf{L}$-halting could be derived.

We plan to continue the work on $\mathsf{PA}$ with a constructive analysis of Tennenbaum's theorem [45], stating that no computable non-standard model of $\mathsf{PA}$ exists. Translated to the synthetic setting where all functions are computable by construction, this would mean that no non-standard model of $\mathsf{PA}$ can be defined in Coq's type theory as long as function symbols are interpreted with type-theoretic functions. It would be interesting to investigate which assumptions of synthetic computability [4] are necessary to derive this observation as an actual theorem inside of Coq.

Complementing Theorem 57 and Fact 66, 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 directly.

Regarding the signature translations and conservativity results for $\mathsf{ZF}$ discussed in Sect. 7, it should be possible to mechanise similar results for arbitrary axiom systems with definable extensions. Results like these would pave the way for an abstract mechanisation of undecidable theories as outlined by Tarski [43].

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

## A Deduction Systems

Intuitionistic natural deduction $\Gamma \vdash_i \varphi$ is defined 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 the Peirce rule $((\varphi \to \psi) \to \varphi) \to \varphi$.

## B Axioms of Set Theory

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

**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 $\mathsf{Z}'$ contains extensionality and the set operation axioms, $\mathsf{Z}$ adds the separation scheme, and $\mathsf{ZF}$ also adds the replacement scheme. The equality axioms are added when working with the deduction system or in an intensional model.

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

**Funding** Open Access funding enabled and organized by Projekt DEAL.

## References


1. Aczel, P.: The type theoretic interpretation of constructive set theory. In: Macintyre, A., Pacholski, L., Paris, J. (eds.) Studies in Logic and the Foundations of Mathematics, vol. Vol. 96, pp. 55–66. Springer, Heidelberg (1978)
2. Aczel, P.: Non-Well-Founded Sets. CSLI Lecture Notes, Palo Alto (1988)
3. Barras, B.: Sets in Coq, Coq in sets. J. Formaliz. Reason. **3**(1), 29–48 (2010)
4. Bauer, A.: First steps in synthetic computability theory. Electron. Notes Theor. Comput. Sci. **155**, 5–31 (2006)
5. Braibant, T., Pous, D.: An efficient Coq tactic for deciding Kleene algebras. In: International Conference on Interactive Theorem Proving, 163–178. Springer, Berlin, Heidelberg (2010)
6. Bundy, A., Giunchiglia, F., Villafiorita, A., Walsh, T.: An incompleteness theorem via abstraction. Technical Report (1996)
7. Church, A.: A note on the Entscheidungsproblem. J. Symb. Log. **1**(1), 40–41 (1936)
8. de Bruijn, N.G.: Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. Indag. Math. **75**(5), 381–392 (1972)
9. Doner, J., Hodges, W.: Alfred Tarski and decidable theories. J. Symb. Logic **53**(1), 20–35 (1988)
10. Forster, Y., Heiter, E., Smolka, G.: Verification of PCP-related computational reductions in Coq. In: International Conference on Interactive Theorem Proving, pp. 253–269 (2018). Springer
11. Forster, Y., Kirst, D., Smolka, G.: 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 (2019)
12. Forster, Y., Kunze, F.: A Certifying Extraction with Time Bounds from Coq to Call-By-Value Lambda Calculus. In: Harrison, J., O'Leary, J., Tolmach, A. (eds.) 10th International Conference on Interactive Theorem Proving. Leibniz International Proceedings in Informatics (LIPIcs), Vol.141, pp. 17–11719. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2019). https://doi.org/10.4230/LIPIcs.ITP.2019.17. http://drops.dagstuhl.de/opus/volltexte/2019/11072
13. Forster, Y., Larchey-Wendling, D., Dudenhefner, A., Heiter, E., Kirst, D., Kunze, F., Smolka, G., Spies, S., Wehr, D., Wuttke, M.: A Coq library of undecidable problems. In: CoqPL 2020, New Orleans, LA, United States (2020). https://github.com/uds-psl/coq-library-undecidability
14. Forster, Y.: Church's Thesis and related axioms in Coq's type theory. In: Baier, C., Goubault-Larrecq, J. (Eds.) 29th EACSL Annual Conference on Computer Science Logic (CSL 2021). LIPIcs, Vol. 183, pp. 21–12119. Dagstuhl, Germany (2021)
15. Forster, Y., Kirst, D., Wehr, D.: Completeness theorems for first-order logic analysed in constructive type theory: extended version. J. Logic Comput. **31**(1), 112–151 (2021)
16. Friedman, H.: Classically and intuitionistically provably recursive functions. In: Scott, D., Muller, G. (eds.) Higher Set Theory, pp. 21–27. Springer, Berlin (1978)
17. Han, J., van Doorn, F.: A formal proof of the independence of the continuum hypothesis. In: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, pp. 353–366 (2020)
18. Hilbert, D., Ackermann, W.: Grundzüge der Theoretischen Logik. Springer, Berlin (1928)
19. Hostert, J., Koch, M., Kirst, D.: A toolbox for mechanised first-order logic. In: Coq Workshop, vol. 2021 (2021)
20. Kirby, L.: Finitary set theory. Notre Dame J. Form. Log. **50**(3), 227–244 (2009)
21. Kirst, D., Hermes, M.: Synthetic undecidability and incompleteness of first-order axiom systems in coq. In: 12th International Conference on Interactive Theorem Proving (ITP 2021) (2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik
22. Kirst, D., Larchey-Wendling, D.: Trakhtenbrot's theorem in Coq: a constructive approach to finite model theory. In: International Joint Conference on Automated Reasoning (IJCAR 2020), Paris, France. Springer, Paris, France (2020)
23. Kirst, D., Smolka, G.: Large model constructions for second-order ZF in dependent type theory. CertifiedPrograms and Proofs—7th International Conference, CPP 2018, Los Angeles, USA, 2018 (2018)
24. Kreisel, G.: Church's thesis: a kind of reducibility axiom for constructive mathematics. In: Studies in Logic and the Foundations of Mathematics, Vol. 60, pp. 121–150 (1970)
25. Larchey-Wendling, D., Forster, Y.: Hilbert's tenth problem in Coq. In: 4th International Conference on Formal Structures for Computation and Deduction. LIPIcs, **131**, pp. 27–12720 (2019)
26. Laurent, O.: An anti-locally-nameless approach to formalizing quantifiers. In: Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs, pp. 300–312 (2021)
27. Maksimović, P., Schmitt, A.: HOCore in Coq. In: International Conference on Interactive Theorem Proving, pp. 278–293. Springer, Berlin (2015)

28. Myhill, J.: Some properties of intuitionistic Zermelo-Frankel set theory. In: Cambridge Summer School in Mathematical Logic, pp. 206–231. Springer, Berlin (1973)
29. O'Connor, R.: Essential incompleteness of arithmetic verified by Coq. In: Hurd, J., Melham, T. (eds.) Theorem Proving in Higher Order Logics, pp. 245–260. Springer, Berlin (2005)
30. Paulson, L.C.: A mechanised proof of Gödel's incompleteness theorems using Nominal Isabelle. J. Autom. Reason. **55**(1), 1–37 (2015)
31. Popescu, A., Traytel, D.: A formally verified abstract account of Gödel's incompleteness theorems. In: International Conference on Automated Deduction, pp. 442–461 (2019). Springer
32. Post, E..L.: Recursively enumerable sets of positive integers and their decision problems. Bull. Am. Math. Soc. **50**(5), 284–316 (1944)
33. Presburger, M..z, Jabcquette, D.: On the completeness of a certain system of arithmetic of whole numbers in which addition occurs as the only operation. Hist. Philos. Logic **12**(2), 225–233 (1991)
34. Quaife, A.: Automated proofs of Löb's theorem and Gödel's two incompleteness theorems. J. Autom. Reason. **4**(2), 219–231 (1988)
35. Richman, F.: Church's thesis without tears. J. Symbol. Logic **48**(3), 797–803 (1983)
36. Schäfer, S., Smolka, G., Tebbi, T.: Completeness and decidability of de Bruijn substitution algebra in Coq. In: Proceedings of the 2015 Conference on Certified Programs and Proofs, pp. 67–73. ACM, New York, NY, USA (2015)
37. Shankar, N.: Proof-checking Metamathematics, The University of Texas at Austin (1986). PhD Thesis
38. Sieg, W., Field, C.: Automated search for Gödel's proofs. In: Deduction, Computation, Experiment, pp. 117–140. Springer, Berlin (2008)
39. Smolka, G., Stark, K.: Hereditarily finite sets in constructive type theory. In: Interactive Theorem Proving - 7th International Conference, ITP 2016, Nancy, France, August 22–27, 2016. LNCS, vol. 9807, pp. 374–390. Springer, Cham (2016)
40. Smullyan, R.M., Fitting, M.: Set Theory and the Continuum Problem. Dover Publications, Mineola (2010)
41. Sozeau, M., Anand, A., Boulier, S., Cohen, C., Forster, Y., Kunze, F., Malecha, G., Tabareau, N., Winterhalter, T.: The MetaCoq Project. J. Autom. Reason. **64**(5), 947–999 (2020)
42. Stark, K., Schäfer, S., Kaiser, J.: Autosubst 2: reasoning with multi-sorted de Bruijn terms and vector substitutions. In: International Conference on Certified Programs and Proofs, pp. 166–180 (2019). ACM
43. Tarski, A.: I: A general method in proofs of undecidability. In: Tarski, A. (ed.) Undecidable Theories. Studies in Logic and the Foundations of Mathematics, **13**, pp. 1–34 (1953)
44. Team, T.C.D.: The Coq Proof Assistant, version 8.12.0. Zenodo (2020). https://doi.org/10.5281/zenodo.4021912
45. Tennenbaum, S.: Non-Archimedean models for arithmetic. Not. Am. Math. Soc. **6**(270), 44 (1959)
46. Trakhtenbrot, B.A.: The impossibility of an algorithm for the decidability problem on finite classes. Dokl. Akad. Nok. SSSR **70**(4), 569–572 (1950)
47. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society **2**(1), 230–265 (1937)
48. Werner, B.: Sets in types, types in sets. In: Ito, T., Abadi, M. (eds.) Theoretical Aspects of Computer Software, pp. 530–546. Springer, Berlin (1997)