Gödel's Theorem Without Tears

Essential Incompleteness in Synthetic Computability

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

Benjamin Peters
Universität des Saarlandes, Saarland Informatics Campus, Saarbrücken, Germany

Abstract

Gödel published his groundbreaking first incompleteness theorem in 1931, stating that a large class of formal logics admits independent sentences which are neither provable nor refutable. This result, in conjunction with his second incompleteness theorem, established the impossibility of concluding Hilbert's program, which pursued a possible path towards a single formal system unifying all of mathematics. Using a technical trick to refine Gödel's original proof, the incompleteness result was strengthened further by Rosser in 1936 regarding the conditions imposed on the formal systems.

Computability theory, which also originated in the 1930s, was quickly applied to formal logics by Turing, Kleene, and others to yield incompleteness results similar in strength to Gödel's original theorem, but weaker than Rosser's refinement. Only much later, Kleene found an improved but far less well-known proof based on computational notions, yielding a result as strong as Rosser's.

In this expository paper, we work in constructive type theory to reformulate Kleene's incompleteness results abstractly in the setting of synthetic computability theory and assuming a form of Church's thesis, an axiom internalising the fact that all functions definable in such a setting are computable. Our novel, greatly condensed reformulation showcases the simplicity of the computational argument while staying formally entirely precise, a combination hard to achieve in typical textbook presentations. As an application, we instantiate the abstract result to first-order logic in order to derive essential incompleteness and, along the way, essential undecidability of Robinson arithmetic.

This paper is accompanied by a Coq mechanisation covering all our results and based on existing libraries of undecidability proofs and first-order logic, complementing the extensive work on mechanised incompleteness using the Gödel-Rosser approach. In contrast to the related mechanisations, our development follows Kleene's ideas and utilises Church's thesis for additional simplicity.

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

Keywords and phrases incompleteness, undecidability, synthetic computability theory

Digital Object Identifier 10.4230/LIPIcs.CSL.2023.30

Supplementary Material To seamlessly integrate the mechanisation with the written text, each formal statement in the PDF version of this paper is hyperlinked with HTML documentation of the Coq files (signalled via a small Coq symbol).
InteractiveResource (Website): `https://www.ps.uni-saarland.de/extras/incompleteness`
Software (Source Code): `https://github.com/uds-psl/coq-synthetic-incompleteness`

1 Introduction

Shortly after Gödel published his celebrated completeness theorem of first-order logic [15, 17] in 1930, he discovered the surprising phenomenon of incompleteness [16] of sufficiently strong axiom systems. While completeness states that all valid formulas are provable, incompleteness (sometimes called negation-incompleteness for disambiguation) refers to the existence of independent sentences that are neither provable nor refutable from a given set of axioms. Considered from the programmatic perspective of metamathematics, completeness encouragingly entails that the formal method of syntactic, finitary deduction is an adequate

means to explore mathematical validities. In contrast, incompleteness establishes a principal limitation to axiomatic reasoning and therefore triggered a long tradition of interpretations (and sometimes misinterpretations [14]) in mathematics, philosophy, and even pop culture,[1] especially regarding the consequential observation that no such sufficiently strong axiom system can verify its own consistency (referred to as Gödel's second incompleteness theorem).

Concretely, Gödel showed that for all formal systems expressing enough properties of the natural numbers while being sound (i.e. all derivable arithmetical sentences are true for the standard model over $\mathbb{N}$) or at least ω-consistent (i.e. if $\varphi(\overline{n})$ is provable for all numerals $\overline{n}$, then $\exists x.\, \neg\varphi(x)$ is not provable) one can explicitly construct an independent sentence. For his elaborate construction, a lot of machinery regarding the arithmetisation of syntax and deduction systems as well as their interplay with substitution had to be developed, for instance Gödel numbering, the β-function, and the diagonal lemma. All this complexity obscures the underlying simple liar paradox of the constructed self-referential sentence, which is the reason why even full textbooks (e.g. Smith's monographs [44, 45]) are devoted to a formal exposition. Rosser later improved on the result by lifting the requirement of ω-consistency to plain consistency using a technically compact trick, entailing essential incompleteness meaning that independent sentences can be constructed in all consistent extensions of an incomplete system, but he still followed the same rather sophisticated strategy [42].

Only with the development of formal notions of computability and the resulting discovery of undecidability in 1936 by Church [4] and Turing [53], a much simpler proof strategy relying on a direct encoding of the halting problem was conceived, as directly remarked in Turing's paper. The underlying observation (already anticipated by Post, cf. [40]) is that complete axiom systems are decidable,[2] and thus systems able to express the halting problem and therefore inheriting its undecidability must be incomplete. To establish that a given system correctly expresses the halting problem, however, one typically relies on soundness to extract termination information from a formal derivation and, additionally, the proof does not readily yield a concrete independent sentence. Thus the nowadays well-known proof of incompleteness via undecidability, though elementary enough to be taught in basic courses on computability theory, yields a result even weaker than Gödel's original statement ahead of Rosser's refinement.

Far less well-known is the line of work pursued by Kleene [26, 27, 28, 29, 30], ultimately accomplishing a form of incompleteness as strong as Rosser's while still transparently showcasing the computational core of the argument.[3] Kleene's improved strategy is based on a switch from the encoded halting problem to encoding a pair of recursively inseparable sets via a stronger representability property, which is in turn established by a technique akin to Rosser's trick in [42]. By this switch the requirement of soundness instead of consistency can be avoided, since no termination information needs to be extracted from derivations but only existing derivations and refutations need to be preserved. Moreover, on more careful inspection already of the previous argument employing the halting problem, an explicit independent sentence can be extracted, similarly for the improved version. The only drawback of the computational variant of Gödel's first incompleteness theorem is that it no longer prepares the machinery for the second incompleteness theorem, but for the mere construction of independent sentences Kleene's argument seems superior and deserves wider popularity.

---

[1] See Douglas Hofstadter's classic "Gödel, Escher, Bach" [21] or far-reaching Youtube channels like Derek Muller's Veritasium (`https://www.youtube.com/watch?v=HeQX2HjkcNo`).

[2] Where we crucially consider the collection of axioms as enumerable, since the set of sentences satisfied in the standard model over $\mathbb{N}$ is a simple example of a complete but undecidable theory.

[3] For instance, a recent posting on the FOM mailing list (`https://cs.nyu.edu/pipermail/fom/2021-September/022872.html`) testified surprise "to discover such a proof laid out" in equally astonished blog posts and StackExchange threads.

Working in the constructive type theory CIC [5, 36], we translate Kleene's incompleteness proofs to the framework of synthetic computability of Richman and Bauer [41, 1], replacing the formal model of computation needed for the notions of enumerability and decidability by the implicit computation inherent to any intuitionistic meta-theory like CIC. Taking this perspective, Kleene's proofs can be further enhanced as no (often left informal) manipulation of Turing machines, μ-recursive functions, or untyped λ-terms is necessary to single out the computable functions $\mathbb{N} \to \mathbb{N}$. Instead, the necessary constructions can be (then directly formally) done with respect to all functions $\mathbb{N} \to \mathbb{N}$, as they are guaranteed to be computable by definability in our intuitionistic meta-theory. To enable the usual diagonalisation referring to universal machines for negative results, we assume variants of Church's thesis [31, 41, 9, 8], internalising the computability of all definable functions and inducing synthetic definitions of an undecidable halting problem and recursively inseparable sets.

With such a synthetic reformulation of Kleene's ideas, we contribute a strikingly simple yet fully formal proof of the strong Gödel-Rosser incompleteness theorem, isolating the computational essence at the core of the phenomenon. To this end, we first work with a fully abstract notion of formal systems to pin down their necessary properties and showcase the strategy free of any contingent overhead, an approach also followed by Beklemishev [2], Smullyan [46], Popescu and Traytel [38, 39], as well as Kirst and Hermes [23]. Subsequently, we instantiate the abstract development to the concrete case of first-order arithmetic, culminating in a proof of essential incompleteness of Robinson arithmetic Q, a finitely axiomatised fragment of Peano arithmetic PA. First, this conclusion is drawn, still maintaining the argument's simplicity, by assuming Church's thesis directly for Q as already employed by Hermes and Kirst [20]. Afterwards we replace this assumption by Church's thesis for μ-recursive functions and an application of the, naturally highly non-elementary, DPRM theorem [6, 33] to bring every μ-recognisable predicate into Diophantine and thus Q-expressible form.

On top of the mathematical contribution to formalise the computational incompleteness proofs in synthetic computability theory, especially the abstract proofs are straightforward to implement in the Coq proof assistant [49], suggesting that the chosen approach is well-suited for the notoriously hard mechanisation of incompleteness [43, 35, 19, 37, 39]. This approach was already exploited in [23], where only the weakest incompleteness result is derived from the undecidability of Q and PA. Following up on [23], the code for the abstract Gödel-Rosser theorem implemented as part of this paper spans merely about 200 lines, while the instantiation to Q adds roughly 2500 lines on top of the employed Coq libraries for first-order logic [24] and undecidability proofs [13]. The latter contains Larchey-Wendling and Forster's extensive mechanisation of the DPRM theorem [32], which could be replaced by a much weaker arithmetisation of a machine model to allow for a realistic comparison to the previous stand-alone mechanisations. Nevertheless, we deem it a valuable contribution to complement the extensive line of work regarding mechanisations of Gödel's original proof strategy with the first equally general mechanisation of the computational argument.

Outline. In Section 2 we summarise preliminary definitions and facts about constructive type theory, synthetic computability, and first-order logic. Then in the core technical part, we give synthetic and abstract proofs of the weak computational incompleteness theorem (Section 3) and Kleene's improvement (Section 4), the latter assuming a general form of Church's thesis. In Section 5, the abstract results are instantiated to Robinson arithmetic Q, assuming Church's thesis for Q to maintain a simple proof outline. Afterwards, this assumption is derived from a more conventional axiom referring to μ-recursive functions, now carrying out the core argument why Q can represent computation (Section 6). We close with some general remarks and further comments on related and future work in Section 7.

2 Preliminaries

In order to make this paper self-contained and accessible to a broader audience, we briefly outline the synthetic approach to computability theory and the representation of first-order logic in constructive type theory as used in prior work [10, 11, 25, 23].

2.1 Constructive Type Theory

We work in the framework of a constructive type theory such as CIC 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 unique 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 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 $=\colon \forall X.\, X \to X \to \mathbb{P}$ specified by the constructor $\forall(x : X).\, x = x$, as well as function definitions by well-founded recursion.

We further employ the basic inductive types of *Booleans* ($\mathbb{B} := \mathsf{tt} \mid \mathsf{ff}$), *Peano natural numbers* ($n : \mathbb{N} := 0 \mid n + 1$), as well as the *option type* ($\mathbb{O}(X) := \ulcorner x \urcorner \mid \emptyset$), and add further inductive types by need. Note that there is a canonical embedding of $\mathbb{B}$ into $\mathbb{N}$, encoding tt as 1 and ff as 0, which we sometimes use to interpret functions $X \to \mathbb{B}$ as functions $X \to \mathbb{N}$.

2.2 Synthetic Computability Theory

The base of the synthetic approach to computability theory of Richman and Bauer [41, 1] is the fact that all functions definable in an intuitionistic 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 predicative calculus of cumulative inductive constructions (pCuIC) [51], the variant of CIC currently implemented in Coq, satisfies this condition although there is no formal proof yet.

As a basis we can introduce decidability, semi-decidability, and enumerability of decision problems synthetically, i.e. without reference to a formal model of computation (cf. [10]):

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

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

On data types like $\mathbb{N}$, semi-decidability and enumerability coincide:

Fact 2. *Every predicate $P : \mathbb{N} \to \mathbb{P}$ is semi-decidable iff it is enumerable.*

Due to this fact, we will interchange both notions fluidly where appropriate to trigger different intuitions. In general, we prefer the view of semi-decidability as it harmonises with the much-used concept of partial functions.

Definition 3. *$f : X \to \mathbb{N} \to \mathbb{O}(Y)$ is a* partial function *if it is deterministic, i.e.:*

$$\forall x n n' y y'.\, f\, x\, n = \ulcorner y \urcorner \to f\, x\, n' = \ulcorner y' \urcorner \to y = y'$$

We write $f : X \rightharpoonup Y$ to denote that f is a partial function from X to Y. We write $f\, x \downarrow y$ if there is n with $f\, x\, n = \ulcorner y \urcorner$, $f\, x \downarrow$ if there is y with $f\, x \downarrow y$, and $f\, x \uparrow$ if $f\, x\, n = \emptyset$ for all n. The notation $f\, x \downarrow$ is meant to suggest termination while $f\, x \uparrow$ denotes divergence.

From every partial function $f : X \rightharpoonup Y$ that is total, i.e. satisfies $f\,x \downarrow$ for all x, one can extract a function $X \to Y$. Conversely, every function $X \to Y$ induces a total partial function $X \rightharpoonup Y$. We therefore freely change between both perspectives.

Finally, we introduce a notion of reductions capable of transporting decidability, foreshadowing the way how computational properties of formal systems can be expressed.

Definition 4. *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 *iff $P\,x$ iff $Q\,(f\,x)$ for all x. We write $P \preceq Q$ if such a function exists.*

Fact 5. *If $P \preceq Q$ and Q is decidable, then so is P.*

Note that all these synthetic notions are only meaningful as long as no classical axioms jeopardising the computational interpretation of the function space $X \to Y$ are assumed. Instead, we will consider several axioms internalising the computational interpretation later.

2.3 First-Order Logic

The abstract incompleteness theorems discussed in Sections 3 and 4 make no reference to a concrete formalism, but the instantiation subject to Sections 5 and 6 will be based on first-order logic. We therefore summarise the representation of first-order terms and formulas with inductive types $\mathbb{T}$ and $\mathbb{F}$, respectively, as underlying [24]:

$$t, t' : \mathbb{T} \;::=\; x \mid O \mid S\,t \mid t \oplus t' \mid t \otimes t' \qquad (x : \mathbb{N})$$
$$\varphi, \psi : \mathbb{F} \;::=\; t \equiv t' \mid \dot{\bot} \mid \varphi \dot{\to} \psi \mid \varphi \dot{\wedge} \psi \mid \varphi \dot{\vee} \psi \mid \dot{\forall} x.\, \varphi \mid \dot{\exists} x.\, \varphi \qquad (x : \mathbb{N})$$

Given a number $n : \mathbb{N}$, we write $\overline{n}$ for the numeral $S^n\,O$. Given formulas φ and ψ, we let $\dot{\neg}\varphi$ denote $\varphi \dot{\to} \dot{\bot}$ and $\varphi \dot{\leftrightarrow} \psi$ denote $(\varphi \dot{\to} \psi) \dot{\wedge} (\psi \dot{\to} \varphi)$. We write $\varphi(x)$ to indicate that x is the only variable occurring free (i.e. not bound by a quantifier) in φ and $\varphi(t)$ to denote the usual capture-avoiding substitution of x with t, similarly for formulas with more free variables.

Axiom systems are represented as enumerable predicates $\mathcal{A} : \mathbb{F} \to \mathbb{P}$. We will consider the standard axiomatisation of Peano arithmetic PA, consisting of the defining equations for $\oplus$ and $\otimes$, injectivity of S, disjointness of S and O, as well as the induction scheme. The weaker system of Robinson arithmetic Q is obtained by replacing the induction scheme with a formula expressing case distinction.

Deduction systems are represented as inductive predicates of type $(\mathbb{F} \to \mathbb{P}) \to \mathbb{F} \to \mathbb{P}$ relating a context with a formula. Concretely, we use classical ($\vdash_c$) and intuitionistic ($\vdash_i$) natural deduction but since all presented results are agnostic to the particular flavour we simply write $\mathcal{A} \vdash \varphi$ standing for both. Since we assume axiomatisations $\mathcal{A}$ to be enumerable, their deductive closure denoted by $\mathcal{A}^{\vdash}$ can be shown enumerable, too.

A general representation of (Tarski) semantics is based on types $\mathcal{M}$ providing the structure to interpret the function symbols of the term language, giving rise to the recursive entailment relation $\mathcal{M} \vDash \varphi$ embedding formulas into propositions of the meta-logic. In this paper, we are exclusively concerned with the standard model $\mathcal{N}$ with $\mathbb{N}$ as domain and the natural interpretations of the function symbols. In this model, $\mathcal{N} \vDash \varphi$ evaluates to ordinary arithmetical statements and in particular $\mathcal{N} \vDash \mathsf{PA}$ can be shown. Note that for $\mathcal{N} \vDash \mathsf{PA}$ to hold constructively, it is crucial that we do not by default include classical axioms in PA [54].

In previous work [23], reductions from the solvability of Diophantine equations (H_{10}) as formalised by Larchey-Wendling and Forster [32] to arithmetical systems were verified. We recollect this fact to include the resulting weak form of incompleteness (already observed in [23]) as motivating approximation in the uniformised framework of this paper. Again note that without additional axioms a predicate like H_{10} cannot be shown undecidable in the synthetic sense but, given its actual undecidability, serves as a suitable computational taboo.

Fact 6. *There is a reduction witnessing $\mathsf{H}_{10} \preceq \mathsf{Q}^{\vdash}$ and $\mathsf{H}_{10} \preceq \mathsf{PA}^{\vdash}$.*

3 Synthetic and Abstract Approach to Incompleteness

In this and the next section, we develop incompleteness results of various strengths in a purely abstract setting. Our exposition follows the computational approach described by Kleene [29, 30], which we translate to the setting of synthetic computability to achieve a highly condensed but still fully formal presentation. We begin with the underlying notion of a formal system, involving only modest assumptions about sentences, negation, and provability.

Definition 7 (Formal System). *A triple $\mathcal{S} = (\mathbb{S}, \dot\neg, \vdash)$ is called a formal system if:*

- *$\mathbb{S}$ is a type, considered the* sentences *of $\mathcal{S}$,*
- *$\dot\neg : \mathbb{S} \to \mathbb{S}$ is a function on sentences, considered the* negation operation*,*
- *$\vdash\colon \mathbb{S} \to \mathbb{P}$ is a semi-decidable predicate on sentences, considered the* provable sentences*.*
- *Consistency holds in the form that for all $\varphi : \mathbb{S}$ not both $\vdash \varphi$ and $\vdash \dot\neg\varphi$.*

A formal system $\mathcal{S}' = (\mathbb{S}, \dot\neg, \vdash')$ is called an extension of $\mathcal{S}$ if $\vdash \varphi$ implies $\vdash' \varphi$ for all φ. Moreover, $\mathcal{S}$ is called decidable if the provability predicate $\vdash$ is decidable.

This general definition captures first-order axiomatisations as will be made precise in Section 5, but also applies to many other formalisms including constructive type theories like CIC or classical systems like HOL.

(Negation-)completeness can be easily expressed as a property of such formal systems, contrasting an informative notion of incompleteness relying on independent sentences.

Definition 8 (Completeness). *We call $\mathcal{S}$ complete if for all φ either $\vdash \varphi$ or $\vdash \dot\neg\varphi$. In contrast, $\mathcal{S}$ admits an independent sentence if there is φ with neither $\vdash \varphi$ nor $\vdash \dot\neg\varphi$.*

To obtain a first weak form of incompleteness, it suffices to observe that complete formal systems are decidable, therefore deciding every decision problem they can encode. This observation is an immediate consequence of Post's theorem [1, 10], however, we prefer to give an alternative proof employing a partial decider that will be reused later.

Lemma 9 (Partial Decider). *One can construct a partial function $d_{\mathcal{S}} : \mathbb{S} \rightharpoonup \mathbb{B}$ with:*

$$\forall\varphi.\ (\vdash \varphi \leftrightarrow d_{\mathcal{S}}\,\varphi \downarrow \mathsf{tt}) \ \wedge\ (\vdash \dot\neg\varphi \leftrightarrow d_{\mathcal{S}}\,\varphi \downarrow \mathsf{ff})$$

Note that by this specification $d_{\mathcal{S}}$ exactly diverges on the independent sentences of $\mathcal{S}$.

Proof. By the definition of formal systems, we have semi-deciders f_1 for $\lambda\varphi.\vdash\varphi$ and f_2 for $\lambda\varphi.\vdash\dot\neg\varphi$, where the latter is obtained from the former by testing if a given negation $\dot\neg\varphi$ is derivable, i.e. by $f_2\,\varphi := f_1\,(\dot\neg\varphi)$. Then we construct $d_{\mathcal{S}} : \mathbb{S} \rightharpoonup \mathbb{B}$ to be the (partial) function that on input φ simultaneously runs $f_1\,\varphi$ and $f_2\,\varphi$, returns tt if the former terminates and ff if the latter terminates, and diverges otherwise:

$$d_{\mathcal{S}}\,\varphi\,n \ :=\ \mathsf{if}\ f_1\,\varphi\,n\ \mathsf{then}\ \ulcorner\mathsf{tt}\urcorner\ \mathsf{else\ if}\ f_2\,\varphi\,n\ \mathsf{then}\ \ulcorner\mathsf{ff}\urcorner\ \mathsf{else}\ \emptyset$$

Consistency is used as the crucial property to show that this function is deterministic. ◀

Now the connection of completeness and decidability can be established transparently:

Fact 10 (Decidability). *If $\mathcal{S}$ is complete, then it is decidable.*

Proof. By completeness the partial decider $d_{\mathcal{S}}$ is total, inducing a decider $\mathbb{S} \to \mathbb{B}$. ◀

To derive said weak form of incompleteness, it remains to clarify what it means for a formal system to encode a decision problem. An intuitive characterisation, called weak representability, exhibits the structure of many-one reductions.

Definition 11 (Weak Representability). *$\mathcal{S}$ weakly represents $P : X \to \mathbb{P}$ if $P \preceq \mathcal{S}$, i.e. if there is a function $r : X \to \mathbb{S}$ such that $P\,x \leftrightarrow \vdash r\,x$. If only $\vdash r\,x$ implies $P\,x$, then we call $\mathcal{S}$ sound for P and r (or simply sound for P if we leave r implicit).*

We can now derive incompleteness in the sense that systems weakly representing an undecidable problem cannot be complete. As the property of weak representability is preserved along sound extensions, we instantiate this result later to derive weak incompleteness of PA and other axiomatisations sound for $\mathcal{N}$.

Theorem 12 (Weak Incompleteness). *If $\mathcal{S}$ weakly represents $P : X \to \mathbb{P}$, then for any extension $\mathcal{S}'$ of $\mathcal{S}$ sound for P it holds that if $\mathcal{S}'$ is complete, then P is decidable. Therefore, if P is known to be undecidable, then $\mathcal{S}'$ must be incomplete.*

Proof. Note that any sound extension $\mathcal{S}'$ of $\mathcal{S}$ still weakly represents P. Since completeness induces decidability of $\vdash$ (Fact 10), we obtain decidability of P from Fact 5. ◀

4 Improving the Computational Incompleteness Result

Although Theorem 12 correctly identifies the computational essence of incompleteness, namely the connection to undecidability, it still falls short of the stronger Gödel-Rosser theorem:

1. The reliance on weak representability excludes consistent but unsound extensions and hence, for instance, essential incompleteness of Q cannot be achieved.
2. There is no concrete example of an independent sentence constructed since the global completeness assumption is needed to totalise the partial decider $d_{\mathcal{S}}$.
3. The result is presented only up to a computational taboo, i.e. the decidability of a problem known to be undecidable, instead of an actual contradiction.

In this section, we address these shortcomings one-by-one, yielding the strongest form of incompleteness possible. Regarding the third improvement, the only way to derive a contradiction from a computation taboo is to assume an axiom that restricts the ambient constructive type theory to a computational interpretation. Concretely, we now assume a variant of Church's thesis [31], namely EPF for "enumerability of partial functions" [41, 9, 8]. It postulates a universal function $\Theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$ computing all partial functions, i.e. for every $f : \mathbb{N} \rightharpoonup \mathbb{N}$ there is a code c such that Θ_c agrees with f (extensionally).

Axiom 13 (EPF). *There is a universal function $\Theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$ satisfying:*

$$\forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\, \exists c : \mathbb{N}.\, \forall xy.\, \Theta_c\, x \downarrow y \leftrightarrow f\, x \downarrow y$$

This assumption induces a canonical undecidable problem:

Definition 14 (Halting Problem). *We define the self-halting problem by $\mathsf{K}_\Theta\, x := \Theta_x\, x \downarrow$.*

The self-halting problem for Θ can be easily shown undecidable by the usual diagonalisation argument. Following this argument in a constructively more informative way, we show that every potential decider for K_Θ necessarily diverges on a concretely constructed input.

Fact 15. *K_Θ is enumerable, but for every candidate decider $d : \mathbb{N} \rightharpoonup \mathbb{B}$ with*

$$\forall x.\, \mathsf{K}_\Theta\, x \leftrightarrow d\, x \downarrow \mathsf{tt}$$

one can construct a concrete value x with $\neg \mathsf{K}_\Theta\, x$ such that $d\, x \uparrow$.

Proof. We first define the partial function $f : \mathbb{N} \rightharpoonup \mathbb{B}$ such that $f\,x \downarrow \mathsf{tt}$ whenever $d\,x \downarrow \mathsf{ff}$ and $f\,x \uparrow$ otherwise. Now using EPF we obtain a code c for f and deduce for $x := c$ that

$$d\,x \downarrow \mathsf{tt} \;\Leftrightarrow\; \mathsf{K}_\Theta\,x \;\Leftrightarrow\; \Theta_x\,x \downarrow \Leftrightarrow\; f\,x \downarrow \Leftrightarrow\; f\,x \downarrow \mathsf{tt} \;\Leftrightarrow\; d\,c \downarrow \mathsf{ff}$$

from which we conclude $d\,x \uparrow$. That K_Θ is not decidable follows since every decider $\mathbb{N} \to \mathbb{B}$ would induce a total candidate decider $\mathbb{N} \rightharpoonup \mathbb{B}$. Finally, enumerability of K_Θ is standard. ◀

We can now identify an intermediate refinement of the incompleteness theorem, providing a concrete independent sentence up to an actual contradiction, which corresponds to the result originally shown by Gödel (in the semantic form requiring soundness instead of ω-consistency).

Theorem 16 (Gödel's Incompleteness)**.** *If $\mathcal{S}$ weakly represents K_Θ, then any extension $\mathcal{S}'$ of $\mathcal{S}$ sound for K_Θ admits an independent sentence.*

Proof. Let $r : \mathbb{N} \to \mathbb{S}$ weakly represent K_Θ in $\mathcal{S}$, therefore also in all sound extensions $\mathcal{S}'$. The function $d := d_{\mathcal{S}'} \circ r$ is a candidate decider for K_Θ in the sense of Fact 15 since:

$$\mathsf{K}_\Theta\,x \;\Leftrightarrow \vdash r\,x \;\Leftrightarrow\; d_{\mathcal{S}'}(r\,x) \downarrow \mathsf{tt} \;\Leftrightarrow\; d \downarrow \mathsf{tt}$$

Then by Fact 15 there is a particular x with $d\,x \uparrow$ and we observe that the sentence $r\,x$ can neither be provable nor refutable since in either case $d\,x \downarrow$ by specification of $d_{\mathcal{S}'}$. ◀

In order to tackle the remaining improvement, namely the applicability to consistent extensions, we follow Kleene's idea to switch to a stronger notion of representability that is not affected by unsound formal systems. Since for weak representability of a predicate P it was crucial to obtain $P\,x$ from $\vdash r\,x$, so to extract information from a derivation, one might hope that this can be replaced by a proof of $\vdash \dot{\neg}(r\,x)$ from $\neg P\,x$, as this has a derivation in the conclusion and therefore transports along any extension. Unfortunately, this strong notion of representability can only be achieved for decidable predicates, thus ruling out the encoding of the undecidable K_Θ for a contradiction. However, it is possible to specify a very similar notion involving a second predicate Q, such that still all derivations appear in conclusions but P and Q can be instantiated with undecidable problems, respectively.

Definition 17 (Strong Separability)**.** *$\mathcal{S}$ strongly separates $P : X \to \mathbb{P}$ and $Q : X \to \mathbb{P}$ if there is a function $r : X \to \mathbb{S}$ such that $P\,x$ implies $\vdash r\,x$ and $Q\,x$ implies $\vdash \dot{\neg} r\,x$.*

The notion of strong separability can now be instantiated with any pair of recursively inseparable problems (i.e. problems excluding any total decider discriminating them) to derive essential incompleteness. The canonical pair of such recursively inseparable problems in the context of EPF refers to the self-halting problems for specific output.

Definition 18. *We define the problems $\mathsf{K}^1_\Theta\,x := \Theta_x\,x \downarrow 1$ and $\mathsf{K}^0_\Theta\,x := \Theta_x\,x \downarrow 0$.*

As done with the normal self-halting problem before (Fact 15), we do not just refute any discriminating decider but show that every partial decider actually diverges on an explicitly constructed input.

Fact 19. *K^1_Θ and K^0_Θ are enumerable, but for every candidate separator $s : \mathbb{N} \rightharpoonup \mathbb{B}$ with*

$$\forall x.\; (\mathsf{K}^1_\Theta\,x \to s\,x \downarrow \mathsf{tt}) \;\wedge\; (\mathsf{K}^0_\Theta\,x \to s\,x \downarrow \mathsf{ff})$$

one can construct a concrete value x with $\neg\mathsf{K}^1_\Theta\,x$ and $\neg\mathsf{K}^0_\Theta\,x$ such that $s\,x \uparrow$.

Proof. We define the partial function $f : \mathbb{N} \rightharpoonup \mathbb{B}$ such that $f\,x \downarrow \mathsf{ff}$ if $s\,x \downarrow \mathsf{tt}$, $f\,x \downarrow \mathsf{tt}$ if $s\,x \downarrow \mathsf{ff}$, and $f\,x \uparrow$ otherwise. Using EPF we obtain a code c for f and deduce for $x := c$ that

$$\begin{aligned} s\,x \downarrow \mathsf{tt} &\Leftrightarrow f\,x \downarrow \mathsf{ff} \Leftrightarrow \Theta_x\,x \downarrow 0 \Leftrightarrow \mathsf{K}^0_\Theta\,x \Rightarrow s\,x \downarrow \mathsf{ff} \\ s\,x \downarrow \mathsf{ff} &\Leftrightarrow f\,x \downarrow \mathsf{tt} \Leftrightarrow \Theta_x\,x \downarrow 1 \Leftrightarrow \mathsf{K}^1_\Theta\,x \Rightarrow s\,x \downarrow \mathsf{tt} \end{aligned}$$

from which we conclude $s\,x \uparrow$. Again, enumerability of K^1_Θ and K^0_Θ is standard. ◀

The desired strong incompleteness theorem, now corresponding to Rosser's refinement of Gödel's result, follows for all formal systems that capture enough computation to strongly separate K^1_Θ and K^0_Θ.

Theorem 20 (Gödel-Rosser Incompleteness)**.** *If $\mathcal{S}$ strongly separates K^1_Θ and K^0_Θ, then any extension $\mathcal{S}'$ of $\mathcal{S}$ admits an independent sentence, i.e. $\mathcal{S}$ is essentially incomplete.*

Proof. Let $r : \mathbb{N} \to \mathbb{S}$ strongly separate K^1_Θ and K^0_Θ in $\mathcal{S}$, therefore also in all consistent extensions $\mathcal{S}'$. The function $s := d_{\mathcal{S}'} \circ r$ is a candidate separator for K^1_Θ and K^0_Θ since:

$$\begin{aligned} \mathsf{K}^1_\Theta\,x \Rightarrow \vdash r\,x &\Leftrightarrow d_{\mathcal{S}'}(r\,x) \downarrow \mathsf{tt} \Leftrightarrow s \downarrow \mathsf{tt} \\ \mathsf{K}^0_\Theta\,x \Rightarrow \vdash \dot\neg r\,x &\Leftrightarrow d_{\mathcal{S}'}(r\,x) \downarrow \mathsf{ff} \Leftrightarrow s \downarrow \mathsf{ff} \end{aligned}$$

Then by Fact 19 there is a particular x with $s\,x \uparrow$ and we observe that the sentence $r\,x$ can neither be provable nor refutable since in either case $s\,x \downarrow$ by specification of $d_{\mathcal{S}'}$. ◀

To emphasise the connection with computational incompleteness, we observe essential undecidability of formal systems of the same expressivity as required in Theorem 20.

Theorem 21 (Essential Undecidability)**.** *If $\mathcal{S}$ strongly separates K^1_Θ and K^0_Θ, then any extension $\mathcal{S}'$ of $\mathcal{S}$ is undecidable, i.e. $\mathcal{S}$ is essentially undecidable.*

Proof. Given $r : \mathbb{N} \to \mathbb{S}$ strongly separating K^1_Θ and K^0_Θ and $d : \mathbb{S} \to \mathbb{B}$ deciding $\mathcal{S}'$, the (total) function $s := d \circ r$ would recursively separate K^1_Θ from K^0_Θ, contradicting Fact 19. ◀

5 Essential Incompleteness of Robinson Arithmetic

We next instantiate the abstract approach to incompleteness from the previous sections to the case of first-order arithmetic. To this end, we now make precise that every consistent axiomatisation $\mathcal{A}$ induces a formal system $\mathcal{S}_\mathcal{A} = (\mathbb{S}_\mathcal{A}, \dot\neg_\mathcal{A}, \vdash_\mathcal{A})$ where

- $\mathbb{S}_\mathcal{A}$ is the type of closed formulas $\varphi : \mathbb{F}$,
- $\dot\neg_\mathcal{A}$ is the negation function $\dot\neg\varphi$ restricted to closed φ,
- $\vdash_\mathcal{A}$ is the provability predicate $\mathcal{A} \vdash \varphi$ restricted to closed φ, and
- $\vdash_\mathcal{A} \varphi$ simultaneous to $\vdash_\mathcal{A} \dot\neg\varphi$ is ruled out by the consistency of $\mathcal{A}$.

We then say that $\mathcal{A}$ is complete if its induced formal system $\mathcal{S}_\mathcal{A}$ is complete, i.e. if either $\mathcal{A} \vdash \varphi$ or $\mathcal{A} \vdash \dot\neg\varphi$ for all closed φ. Similarly, we say that $\mathcal{A}$ admits an independent sentence if $\mathcal{S}_\mathcal{A}$ does, i.e. if there is some closed φ with neither $\mathcal{A} \vdash \varphi$ nor $\mathcal{A} \vdash \dot\neg\varphi$. Note that here, as our notational convention suggests, we deliberately include both the intuitionistic and the classical ND system, so our treatment of incompleteness applies to both flavours.

Since reductions $P \preceq \mathcal{A}^\vdash$ establish that $\mathcal{S}_\mathcal{A}$ weakly represents P, we can immediately derive a weak form of incompleteness from previous results.

Theorem 22 (Weak Incompleteness, cf. [23])**.** *If* PA *is complete, then* H_{10} *is decidable.*

Proof. We have $\mathsf{H}_{10} \preceq \mathsf{PA}^{\vdash}$ by Fact 6, so $\mathcal{S}_{\mathsf{PA}}$ weakly represents H_{10}. Then if PA were complete, H_{10} were decidable by Theorem 12. ◀

Note that this result also applies to all sound extensions of PA, i.e. extensions $\mathcal{A}$ such that from $\mathcal{A} \vdash \varphi$ one can derive $\mathcal{N} \vDash \varphi$, as well as to all weaker (and hence vacuously incomplete) fragments, in particular Q. We refer the reader to [23] for more detail on this weak form of incompleteness obtained from the reduction of H_{10} in a synthetic sense.

To obtain the stronger result concerning merely consistent extensions, we prepare to instantiate Theorem 20 to the case of Q, as this axiomatisation exactly provides the needed representability requirements. For this instantiation, note that although EPF is an axiom strong enough to yield undecidable problems, it does not necessarily restrict the function space $\mathbb{N} \rightharpoonup \mathbb{N}$ to a concrete model of computation expressible in Q. We therefore need to assume a more explicit form of Church's thesis to derive the desired representability within Q. An elegant strategy is to directly assume Church's thesis for Q itself (CT_{Q}) as introduced by Hermes and Kirst [20], instantiate Theorem 20 with elementary arguments, and afterwards deliver the rather involved argument that CT_{Q} follows from a more conventional explicit form of EPF for μ-recursive functions.

To state CT_{Q}, we first identify the semantically well-behaved class of Σ_1-formulas.

Definition 23 (Δ_1- and Σ_1-formulas, cf. [20]). *We say that $\varphi : \mathbb{F}$ is a Δ_1-formula if for all substitutions σ such that $\sigma\, n$ is closed for all $n : \mathbb{N}$ we have $\mathsf{Q} \vdash \varphi[\sigma]$ or $\mathsf{Q} \vdash \dot{\neg}\varphi[\sigma]$. Moreover, we say that $\psi : \mathbb{F}$ is a Σ_1-formula if there is a Δ_1-formula ψ such that $\varphi = \dot{\exists} \ldots \dot{\exists}\psi$.*

CT_{Q} then states that any function $\mathbb{N} \rightharpoonup \mathbb{N}$ is fully captured by a Σ_1-formula.

Axiom 24 (CT_{Q}). *For all partial $f : \mathbb{N} \rightharpoonup \mathbb{N}$ there exists a Σ_1-formula $\varphi(x, y)$ with:*

$$\forall xy.\, f\, x \downarrow y \;\leftrightarrow\; \mathsf{Q} \vdash \dot{\forall} y'.\, \varphi(\overline{x}, y') \dot{\leftrightarrow} y' \equiv \overline{y}$$

To enable the usage of the results from the previous section solely assuming CT_{Q}, we show that CT_{Q} yields a universal function Θ as formerly postulated with EPF.

Fact 25. *Given that we now assume CT_{Q}, in particular EPF holds.*

Proof. We choose as universal function $\Theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$ the partial function that on input c and x enumerates all derivations from Q and terminates with value y if a derivation $\mathsf{Q} \vdash \forall y'.\, \varphi_c(\overline{x}, y') \dot{\leftrightarrow} y' \equiv \overline{y}$ is found for φ_c being the c-th formula.

Then given a partial function $f : \mathbb{N} \rightharpoonup \mathbb{N}$, the assumption of CT_{Q} guarantees that f is captured by some Σ_1-formula $\varphi = \varphi_c$ for some c. Then we deduce for all x and y

$$\Theta_c\, x \downarrow y \;\Leftrightarrow\; \mathsf{Q} \vdash \forall y'.\, \varphi_c(\overline{x}, y') \dot{\leftrightarrow} y' \equiv \overline{y} \;\Leftrightarrow\; f\, x \downarrow y$$

as desired to establish that Θ is universal. ◀

In the case of total functions, the capturing condition can be slightly simplified, which yields the actual formulation of CT_{Q} used in [20].

Fact 26 (Total CT_{Q}, cf. [20]). *For all $f : \mathbb{N} \to \mathbb{N}$ there exists a Σ_1-formula $\varphi(x, y)$ with:*

$$\forall x.\, \mathsf{Q} \vdash \dot{\forall} y'.\, \varphi(\overline{x}, y') \dot{\leftrightarrow} y' \equiv \overline{f\, x}$$

From CT_{Q} we can derive all the representability conditions employed in Section 3. In fact, we obtain more precise conditions involving Σ_1-formulas $\varphi(x)$ providing uniform encoding functions $r\, n := \varphi(\overline{n})$.

Definition 27. *Given $P, P' : \mathbb{N} \to \mathbb{P}$ and a Σ_1-formula $\varphi(x)$ we say that*

- *φ weakly Σ_1-represents P if $P\,n \leftrightarrow \mathsf{Q} \vdash \varphi(\overline{n})$ and*
- *φ strongly Σ_1-separates P and P' if $P\,n \to \mathsf{Q} \vdash \varphi(\overline{n})$ and $P'\,n \to \mathsf{Q} \vdash \dot{\neg}\varphi(\overline{n})$.*

So if φ for instance Σ_1-represents $P : \mathbb{N} \to \mathbb{P}$, then $r\,n := \varphi(\overline{n})$ witnesses that the system $\mathcal{S}_{\mathsf{Q}}$ weakly represents P in the sense of Definition 11, analogously for strong Σ_1-separability.

Theorem 28 (Representability, cf. [20])**.** *Q can represent predicates as follows:*

1. *Every enumerable predicate over $\mathbb{N}$ is weakly Σ_1-representable.*
2. *Every pair of disjoint enumerable predicates over $\mathbb{N}$ is strongly Σ_1-separable.*

Proof. We establish both claims independently:

1. An enumerator e of P can be recast as a function $\mathbb{N} \to \mathbb{N}$ with $P\,x$ iff $\exists n.\, e\,n = x + 1$. Applying CT_{Q}, we obtain a Σ_1-formula φ capturing e and deduce:

$$P\,x \;\Leftrightarrow\; \exists n.\, e\,n = x + 1 \;\Leftrightarrow\; \exists n.\, \mathsf{Q} \vdash \overline{e\,n} = S\,\overline{x} \;\Leftrightarrow\; \exists n.\, \mathsf{Q} \vdash \varphi(\overline{n}, S\,\overline{x}) \;\Leftrightarrow\; \mathsf{Q} \vdash \dot{\exists} k.\, \varphi(k, S\,\overline{x})$$

   Thus $\psi(x) := \exists k.\, \varphi(k, S\,x)$ weakly Σ_1-represents P.
2. A partial decider $d : \mathbb{N} \rightharpoonup \mathbb{B}$ can be constructed with $P\,x$ iff $d\,x \downarrow \mathsf{tt}$, and $P'\,x$ iff $d\,x \downarrow \mathsf{ff}$, analogously to the partial decider defined in Lemma 9. Applying CT_{Q}, we obtain a Σ_1-formula φ capturing d and deduce:

$$\begin{aligned} P\,x \;&\Rightarrow\; d\,x \downarrow \mathsf{tt} \;\Rightarrow\; \mathsf{Q} \vdash \varphi(x, \overline{1}) \\ P'\,x \;&\Rightarrow\; d\,x \downarrow \mathsf{ff} \;\Rightarrow\; \mathsf{Q} \vdash \varphi(x, \overline{0}) \;\Rightarrow\; \mathsf{Q} \vdash \dot{\neg}\varphi(x, \overline{1}) \end{aligned}$$

   Thus $\psi(x) := \varphi(x, \overline{1})$ strongly Σ_1-separates P and P'. ◀

Note that the weak representability property (1) of Theorem 28 could be used to obtain independent sentences for all sound extensions of Q based on the intermediate result Theorem 16. Already given the strong separability property (2), however, we immediately conclude the stronger essential incompleteness of Q based on Theorem 20.

Theorem 29. *Any consistent axiomatisation $\mathcal{A} \supseteq \mathsf{Q}$ admits an independent sentence.*

Proof. We apply Theorem 20, so we only need to show that Q strongly separates K^1_Θ and K^0_Θ. Since these are enumerable, this follows from (2) of Theorem 28. ◀

Similarly, we can observe the essential undecidability of Q based on Theorem 21.

Theorem 30. *Any consistent axiomatisation $\mathcal{A} \supseteq \mathsf{Q}$ is undecidable.*

Proof. We apply Theorem 21 and then argue as in the proof of Theorem 29. ◀

6 Deriving Church's Thesis for Robinson Arithmetic

Arguably, by the assumption of CT_{Q} we have sidestepped much of the actual work needed to establish the essential incompleteness of Q. To showcase that most of this work concerned with the representability properties can actually be done feasibly and only an axiom connecting the synthetic level with a concrete model of computation is necessary, we now derive CT_{Q} from a common version of Church's thesis for μ-recursive functions (EPF_μ). Note that Church's thesis for any Turing complete model could be consistently assumed as discussed by Forster [9] and thus by the upcoming derivation we in particular justify the consistency of CT_{Q}. We also remark that our derivation relies on the heavy-weight DPRM theorem as mechanised by Larchey-Wendling and Forster [32], however, one could also give a less informative but more direct arithmetisation of formal computation.

We refer to [32] for full detail about an encoding of μ-recursive functions in CIC and only require a step-indexed interpreter $\Theta^\mu : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$. For Θ^μ we then state EPF_μ which will only be used to show that the graph of a given partial function is μ-enumerable, and therefore Diophantine by the DPRM theorem.

Definition 31. *EPF_μ states that Θ^μ is universal for all partial functions:*

$$\forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\, \exists c : \mathbb{N}.\, \forall xy.\, \Theta^\mu_c\, x \downarrow y \leftrightarrow f\, x \downarrow y$$

To prepare the result that EPF_μ implies CT_Q, we need a bit more machinery about Σ_1-formulas φ, especially the completeness property that for deriving $\mathsf{Q} \vdash \varphi$ it suffices to show $\mathcal{N} \vDash \varphi$. This and forthcoming observations can be simplified by the fact that a prefix of existential quantifiers can be compressed into a single existential quantifier:

Lemma 32. *For every Σ_1-formula φ there is a Δ_1-formula ψ with $\mathsf{Q} \vdash \varphi \dot{\leftrightarrow} \dot{\exists}\psi$.*

Proof. By induction on the length of the quantifier prefix of φ. For the inductive step it suffices to show that two quantifiers can be merged into one, i.e. that for a given Δ_1-formula φ there is a Δ_1-formula ψ with $\mathsf{Q} \vdash (\dot{\exists}x.\, \dot{\exists}y.\, \varphi(x,y)) \dot{\leftrightarrow} (\dot{\exists}z.\, \psi(z))$. We set:

$$\psi(z) \;:=\; \dot{\exists}x.\, (\dot{\exists}k.\, z \equiv x \oplus k) \mathbin{\dot{\wedge}} \dot{\exists}y.\, (\dot{\exists}k.\, z \equiv k \oplus y) \mathbin{\dot{\wedge}} \varphi(x,y)$$

The sought equivalence is not hard to establish as one can instantiate $z := x \oplus y$. Proving that ψ is Δ_1 is more tedious but less insightful as this requires to establish decidability of bounded quantifications via their equivalence to iterated disjunctions formally in Q. ◀

Note that from now on we use $x \dot{\leq} y$ as the common notation for $\dot{\exists}k.\, y \equiv x \oplus k$ but that we indeed also need to employ the symmetric variant $\dot{\exists}k.\, y \equiv k \oplus x$ in the previous proof since Q does not recognise addition as commutative.

Fact 33 (Σ_1-completeness, cf. [20])**.** *If φ is closed and Σ_1, then $\mathcal{N} \vDash \varphi$ implies $\mathsf{Q} \vdash \varphi$.*

Proof. By Lemma 32 we may assume that φ has the form $\dot{\exists}\psi$ where ψ is Δ_1. Then from $\mathcal{N} \vDash \varphi$ we obtain $n : \mathbb{N}$ such that $\mathcal{N} \vDash \psi(\overline{n})$. Now since $\psi(\overline{n})$ is closed we have either $\mathsf{Q} \vdash \psi(\overline{n})$ or $\mathsf{Q} \vdash \dot{\neg}\psi(\overline{n})$ by the definition of Δ_1, where the former immediately yields $\mathsf{Q} \vdash \varphi$ and where the latter contradicts $\mathcal{N} \vDash \varphi$ via soundness. ◀

We can now give a proof that EPF_μ implies CT_Q based on a technique resembling Rosser's trick in his refinement of Gödel's original incompleteness proof. To provide some intuition, the idea is to refine a formula weakly Σ_1-representing a predicate such that a witness not only guarantees a solution but also that all potential smaller solutions show similar behaviour.

Fact 34. *EPF_μ implies CT_Q.*

Proof. Let $f : \mathbb{N} \rightharpoonup \mathbb{N}$ be given, the goal is to capture f by some Σ_1-formula φ. From EPF_μ we obtain some c such that f is computed by Θ^μ_c. Now since Θ^μ_c is μ-recursive, we can apply the DPRM theorem to obtain a polynomial equation $p = q$ recognising the graph of Θ^μ_c. From the reduction verified in [23] we obtain that solvability of $p = q$ agrees with derivability of $\varphi_{p,q} = \dot{\exists}^N\, p^* \equiv q^*$ in Q:

$$f\, x \downarrow y \;\leftrightarrow\; \mathsf{Q} \vdash \varphi_{p,q}(\overline{x}, \overline{y})$$

This intermediate result states that the graph of f is weakly Σ_1-representable and can be refined to a capturing as needed in CT_Q using a general variant of Rosser's trick. First, with Lemma 32 we refine $\varphi_{p,q}(x,y)$ to a formula $\dot{\exists}k.\, \psi(x,y,k)$ where ψ is Δ_1. Secondly, we set

$$\varphi'(x,y,k) \;:=\; \psi(x,y,k) \mathbin{\dot{\wedge}} \dot{\forall}y'k'.\, y' \oplus k' \dot{\leq} y \oplus k \mathbin{\dot{\to}} \psi(x,y',k') \mathbin{\dot{\to}} y' \equiv y$$

followed by $\varphi(x,y) := \dot{\exists}k.\, \varphi'(x,y,k)$ and verify that φ captures f as desired for CT_Q:

- Assuming $f\,x \downarrow y$, we want to derive $\dot\forall y'.\, \varphi(\overline{x}, y') \dot\leftrightarrow y' \equiv \overline{y}$ formally within Q. Note that from $f\,x \downarrow y$ we obtain some natural number k with $\psi(\overline{x}, \overline{y}, \overline{k})$ as base. Using Σ_1-completeness, we can in fact derive $\varphi'(\overline{x}, \overline{y}, \overline{k})$ as this is straightforward to verify in the standard model $\mathcal{N}$.
  This establishes the backwards direction of the sought equivalence, for the forward direction assume $\varphi(\overline{x}, y')$ for some variable y'. Hence $\varphi'(\overline{x}, y', k')$ for some variable k', complementing $\varphi'(\overline{x}, \overline{y}, \overline{k})$ from before. As Q can derive that either $\overline{y} \oplus \overline{k} \leq y' \oplus k'$ or $y' \oplus k' \leq \overline{y} \oplus \overline{k}$, we obtain $y' \equiv \overline{y}$ in either case from the construction of φ'.
- If conversely $\mathsf{Q} \vdash \dot\forall y'.\, \varphi(\overline{x}, y') \dot\leftrightarrow y' \equiv \overline{y}$, then in particular $\mathsf{Q} \vdash \dot\exists k.\, \psi(\overline{x}, \overline{y}, k)$ from which we obtain $f\,x \downarrow y$ by the representability property of $\varphi_{p,q}$. ◀

In fact, we also expect that CT_{Q} implies EPF_μ as this basically boils down to the same proof as in Fact 25, with the difference that all computability arguments are done for μ-recursive functions instead of synthetically.

7 Discussion

In this paper, we first gave generic incompleteness proofs of different strengths for abstract formal systems with a negation operation, translating ideas of Kleene to the framework of synthetic computability. The strongest version states essential incompleteness of formal systems strongly separating canonical enumerable and disjoint predicates. Secondly, we instantiated our results to first-order logic over the axiomatisation of Robinson arithmetic Q. The instantiation was first approximated assuming CT_{Q} and then using EPF_μ, the DPRM theorem, and Rosser's trick to show strong Σ_1-separability of disjoint enumerable predicates.

The remaining assumption of EPF_μ is a common formulation of Church's thesis, already mentioned as a consistent axiom for constructive mathematics in the textbook by Troelstra and van Dalen [52]. Though no consistency proof for the specific case of EPF_μ in CIC has been conducted, equivalent formulations of Church's thesis have been shown consistent in closely related type theories by Swan and Uemura [47] and Yamada [55], see also Forster's discussion [9] for an overview of formulations of Church's thesis in CIC.

7.1 Coq Mechanisation

The mechanisation consists of two main parts: the abstract incompleteness proofs and their instantiation to first-order logic. The former consists of roughly 400 lines of code, of which only around 200 are required for the strongest incompleteness proofs, while the latter consists of around 2500 lines of code. The development is based on Coq libraries of undecidability proofs [13] and first-order logic [24], from which code particularly on synthetic computability, the DPRM theorem, as well as the encoding of first-order logic is reused, respectively.

Mechanising and working with partial functions and Church's thesis is straightforward. The paper proofs, however, tend to follow a slightly different structure than their mechanised counterparts, in particular when dealing with equivalences, such as in Fact 15. Otherwise, the mechanisation of Sections 3 and 4 is remarkably unremarkable.

Mechanising the instantiation to first-order logic, however, was a lot more work. We build upon an existing mechanisation of first-order logic by Kirst et al. [24] that includes most fundamental definitions and lemmas for working with first-order logic. As opposed to the definitions presented in this text, it defines formulas and terms to be parametric in a signature, i.e. types of predicate and function symbols with their corresponding arities, and uses de Bruijn indices instead of explicit naming to implement binding. While the former

difference did not affect the mechanisation other than requiring some boilerplate code, the latter repeatedly caused us problems. Mechanising structures that include binders, such as predicate logic or programming languages, is well known to be much more tedious than dealing with them on paper, where many lemmas on and properties of substitutions are largely glossed over.

Notably, a lot of work (almost half of the mechanisation of the instantiation, by lines of code) went into mechanising Q-decidability of bounded quantification and Σ_1-completeness due to the technicality of these results. These proofs relied heavily on the first-order proof mode for Coq by Koch, as described in [22], allowing us to use tactics similar to the ones included with Coq to show statements within first-order logic. The proof mode also provides translations between a de Bruijn representation of logical formulas and a named representation, which greatly improves the ergonomics of working with first-order logic. This project would have been much more tedious if we did not have the proof mode available.

7.2 Related Work

Variants of Gödel's incompleteness theorems. The Gödel-Rosser approach to incompleteness was developed in the 1930s, primarily by Gödel [16] and Rosser [42]. Kleene presented his approach to incompleteness prominently in both of his books [29, 30], as well as multiple papers [26, 27, 28, 29, 30]. Turing mentioned similar ideas to show incompleteness in his seminal paper on the *Entscheidungsproblem* [53].

Different proofs of Gödel's first incompleteness theorem, among them some abstract ones, have been considered by Beklemishev [2], Smullyan [46], as well as Popescu and Traytel [39]. Our approach especially shares similarities with the former two, as they also consider Kleene's computational proofs in an abstract setting, while the latter approach is mechanised but based on the Gödel-Rosser strategy. Another computational account of Gödel's incompleteness theorem was anticipated independently by Post [40].

Synthetic computability theory in CIC. The basic principles of synthetic computability theory as introduced by Richman and Bauer [41, 1] were first applied to CIC by Forster et al. [10]. An investigation of Church's thesis [31, 52] to enhance the expressivity and applicability of synthetic computability theory in CIC was conducted by Forster [7, 9, 8]. Note that Forster uses an axiomatic notion of partial functions which can be instantiated with our representation (Definition 3). Moreoever, the obtained framework was used to mechanise various undecidability results for several decision problems [13], including the solvability of Diophantine equations [32] by Larchey-Wendling and Forster.

Hermes and Kirst [20] use synthetic methods to analyse Tennenbaum's theorem [50] in constructive type theory, stating that the standard model over $\mathbb{N}$ is the only computable model of PA. In their development, they assume CT_{Q} for total functions (Fact 26) and leave the derivation of CT_{Q} from a more common axiom for synthetic computability such as EPF_{μ} for future work. They also introduce a related but stronger semantic notion of Σ_1-formulas based on decidability properties (compared to our Definition 23) and derive corresponding versions of weak Σ_1-representability (Theorem 28) and Σ_1-completeness (Fact 33).

Mechanisations of Gödel's incompleteness theorems. The earliest mechanisation of Gödel's first incompleteness theorem was developed by Shankar in 1994 [43] using Nqthm [3], also called the Boyer-Moore theorem prover, a proof assistant based on Lisp. He does not mechanise incompleteness of arithmetic, but of a finite set theory, which simplifies encoding recursive structures, such as formulas and proofs, immensely. His development consists of

around 20 000 lines of code. A mechanisation of incompleteness of first-order arithmetic, based on an axiomatisation similar to Robinson arithmetic, was first developed by O'Connor in 2005 [35] using Coq, consisting of almost 44 000 lines of code. Another mechanisation of incompleteness of arithmetic using HOL Light [18] was developed by Harrison in 2009 [19].

More recently, both of Gödel's incompleteness theorems were mechanised by Paulson in 2014 [37] in around 12 000 lines of Isabelle [34] code. He showed incompleteness of a finite set theory slightly different from the one used by Shankar. To our knowledge, he was the first to give a complete mechanisation of Gödel's second incompleteness theorem, relying on a proof outline by Swierczkowski [48]. Also using Isabelle, Popescu and Traytel [38, 39] in 2019 mechanised both incompleteness theorems using the Gödel-Rosser approach abstractly, based on a much more subtle notion of formal systems than ours, additionally incorporating substitutions, soundness, arithmetic, and more.

None of the mechanisations mentioned above used Kleene's approach to incompleteness, let alone a synthetic approach to computability arguments. However, for example O'Connor used the representability of primitive recursive functions as an intermediate step to show weak representability of first-order provability, similarly as in Gödel's original proof.

The weak computational form of incompleteness for first-order arithmetic and set theory in Coq was mechanised by Kirst and Hermes [23], as a by-product of a general approach to the undecidability of first-order axiom systems. Their result differs from ours in two ways: First, they do not obtain essential incompleteness since they rely on Kleene's early proof using the halting problem (see Theorem 12). Instead, they give an abstract notion of formal systems incorporating soundness, and use it to deduce incompleteness of all sound extensions of their axiomatisation. Secondly, their development does not deduce falsity from the assumption of incompleteness, instead constructing a decider for the halting problem of Turing machines, which also prevents them from constructing an independent sentence.

7.3 Future Work

We have not considered the conditions under which Rosser's trick is applicable abstractly but just gave the concrete proof of strong separability derived from weak representability for Q in Section 6. Generalising this proof could simplify future instantiations of the stronger incompleteness results, as long as the abstraction is sufficiently simple.

Similarly on the abstract level, it is conceivable that instead of working with EPF to internalise that every function $\mathbb{N} \rightharpoonup \mathbb{N}$ is computable from the start, one could also axiomatise a predicate $(\mathbb{N} \rightharpoonup \mathbb{N}) \to \mathbb{P}$ describing the computable functions with enough closure properties to perform the intermediate constructions. Then one can still assume EPF to obtain the same results for the trivially true predicate, but also an assumption-free version (then better comparable to the related mechanisations) could be obtained if the predicate refers to a specific model of computation for which the necessary closure properties are verified.

Our instantiation to first-order logic with Robinson's Q currently relies on Larchey-Wendling and Forster's mechanisation of the DPRM theorem [32]. The DPRM theorem, however, is a much stronger statement than the representability property we actually need, and is considerably harder to show. Using our mechanisation of Σ_1-completeness, it appears feasible to obtain weak representability of μ-enumerable predicates (or predicates enumerable in any equivalent model of computation) for Q directly by just finding first-order formulas that define these predicates in the standard model. Similar approaches have been taken by O'Connor [35] and Paulson [37].

In Section 6, we showed that EPF_μ implies Church's thesis for Q. Along the lines of Fact 25, we expect the converse to be provable as well by first showing that, given any partial function by Q, its graph is μ-enumerable, which suffices for its μ-computability.

Mechanising this fact, however, would be challenging because we would have to implement our first-order logic, that is, substitution, enumerability of provable formulas, etc. using μ-recursive functions. Automatic extractions of such functions for first-order logic, specifically into a lambda calculus, have already been investigated by Forster, Kirst, and Wehr [11] using a tool by Forster and Kunze [12].

References

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

2 Lev D. Beklemishev. Gödel incompleteness theorems and the limits of their applicability. i. *Russian Mathematical Surveys*, 65(5):857, 2010.

3 Robert S. Boyer, Matt Kaufmann, and J S. Moore. The Boyer-Moore theorem prover and its interactive enhancement. *Computers & Mathematics with Applications*, 29(2):27–62, 1995.

4 Alonzo Church. A note on the Entscheidungsproblem. *The journal of symbolic logic*, 1(1):40–41, 1936.

5 Thierry Coquand and Gérard Huet. *The calculus of constructions.* PhD thesis, INRIA, 1986.

6 Martin Davis, Hilary Putnam, and Julia Robinson. The decision problem for exponential Diophantine equations. *Annals of Mathematics*, pages 425–436, 1961.

7 Yannick Forster. Church's thesis and related axioms in Coq's type theory. In Christel Baier and Jean Goubault-Larrecq, editors, *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*, volume 183 of *LIPIcs*, pages 21:1–21:19, Dagstuhl, Germany, 2021.

8 Yannick Forster. *Computability in constructive type theory.* PhD thesis, Saarland University, 2021.

9 Yannick Forster. Parametric Church's thesis: Synthetic computability without choice. In *International Symposium on Logical Foundations of Computer Science*, pages 70–89. Springer, 2022.

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

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

12 Yannick Forster and Fabian Kunze. A certifying extraction with time bounds from Coq to call-by-value lambda calculus. In John Harrison, John O'Leary, and Andrew Tolmach, editors, *10th International Conference on Interactive Theorem Proving*, volume 141 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:19, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.ITP.2019.17`.

13 Yannick Forster, Dominique Larchey-Wendling, Andrej Dudenhefner, Edith Heiter, Dominik Kirst, Fabian Kunze, Gert Smolka, Simon Spies, Dominik Wehr, and Maximilian Wuttke. A Coq library of undecidable problems. In *CoqPL 2020*, New Orleans, LA, United States, 2020. URL: `https://github.com/uds-psl/coq-library-undecidability`.

14 Torkel Franzén. *Gödel's theorem: an incomplete guide to its use and abuse.* AK Peters/CRC Press, 2005.

15 Kurt Gödel. *Über die Vollständigkeit des Logikkalküls.* PhD thesis, University of Vienna, 1929.

16 Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für mathematik und physik*, 38(1):173–198, 1931.

17 Kurt Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37:349–360, 1930. URL: `https://zbmath.org/?q=an%3A56.0046.04`.

18 John Harrison. HOL Light: a tutorial introduction. In *Formal Methods in Computer-Aided Design*, pages 265–269. Springer Berlin Heidelberg, 1996.

19 John Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009.

20 Marc Hermes and Dominik Kirst. An analysis of Tennenbaum's theorem in constructive type theory. In *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, 2022.

21 Douglas R. Hofstadter. *Gödel, Escher, Bach*. Basic books New York, 1979.

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

23 Dominik Kirst and Marc Hermes. Synthetic undecidability and incompleteness of first-order axiom systems in Coq (extended version). To appear.

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

25 Dominik Kirst and Dominique Larchey-Wendling. Trakhtenbrot's Theorem in Coq: Finite Model Theory through the Constructive Lens. *Logical Methods in Computer Science*, Volume 18, Issue 2, June 2022. doi:10.46298/lmcs-18(2:17)2022.

26 Stephen C. Kleene. General recursive functions of natural numbers. *Mathematische annalen*, 112(1):727–742, 1936.

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

28 Stephen C. Kleene. A symmetric form of Gödel's theorem. *Journal of Symbolic Logic*, 16(2), 1951.

29 Stephen C. Kleene. Introduction to Metamathematics, 1952.

30 Stephen C. Kleene. *Mathematical Logic*. Dover books on mathematics. Dover Publications, 2002.

31 Georg Kreisel. Church's thesis: a kind of reducibility axiom for constructive mathematics, 1970.

32 Dominique Larchey-Wendling and Yannick Forster. Hilbert's Tenth Problem in Coq (Extended Version). *Logical Methods in Computer Science*, Volume 18, Issue 1, March 2022.

33 Juri V. Matijasevic. Enumerable sets are Diophantine. In *Soviet Math. Dokl.*, volume 11, pages 354–358, 1970.

34 Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283. Springer Science & Business Media, 2002.

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

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

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

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

39 Andrei Popescu and Dmitriy Traytel. Distilling the requirements of Gödel's incompleteness theorems with a proof assistant. *Journal of Automated Reasoning*, 65(7):1027–1070, 2021.

40 Emil L. Post. Absolutely unsolvable problems and relatively undecidable propositions–account of an anticipation (1941). *Collected Works of Post*, pages 375–441, 1994.

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

42 Barkley Rosser. Extensions of some theorems of Gödel and Church. *The journal of symbolic logic*, 1(3):87–91, 1936.

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

44 Peter Smith. *An introduction to Gödel's theorems*. Cambridge University Press, 2013.

45 Peter Smith. Gödel without (too many) tears, 2021.

46 Raymond M. Smullyan. *Gödel's incompleteness theorems*. Oxford University Press on Demand, 1992.

47 Andrew W. Swan and Taichi Uemura. On Church's thesis in cubical assemblies. *Mathematical Structures in Computer Science*, pages 1–20, 2019.

48 Stanislaw Swierczkowski. Finite sets and Gödel's incompleteness theorems. *Dissertationes Mathematicae*, 422:1–58, 2003.

49 The Coq Development Team. The Coq proof assistant, January 2022. doi:10.5281/zenodo.5846982.

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

51 Amin Timany and Matthieu Sozeau. Consistency of the predicative calculus of cumulative inductive constructions (pCuIC). *CoRR*, abs/1710.03912, 2017. arXiv:1710.03912.

52 Anne S. Troelstra and Dirk Van Dalen. *Constructivism in Mathematics*. Vol. 121 of Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, 1988.

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

54 Benno Van den Berg and Jaap Van Oosten. Arithmetic is categorical, 2011. Technical report.

55 Norihiro Yamada. Game semantics of Martin-Löf type theory, part III: its consistency with Church's thesis. *arXiv e-prints*, 2020. arXiv:2007.08094.