Post's Problem in Constructive Mathematics

Haoyi Zeng, Yannick Forster, Dominik Kirst, Takako Nemoto

Post's Problem in Constructive Mathematics

Haoyi Zeng
Saarland University
Saarbrücken, Germany
haze00001@stud.uni-saarland.de

Yannick Forster, Dominik Kirst
Inria
Paris, France
firstname.lastname@inria.fr

Takako Nemoto
Tohoku University
Sendai, Japan
nemototakako@gmail.com

***Abstract*—We study a solution to Post's problem, i.e. the existence of a semi-decidable but undecidable Turing degree strictly below the halting problem, from the perspective of constructive mathematics. We use the finite injury priority method due to Friedberg and Muchnik to carry out the construction of a low simple set due to Soare.**

Our proof reveals that as only non-constructive logical principle it suffices to assume $(\neg\neg\Sigma_1)$-LEM, stating that for any Σ_1 set A, $\neg\neg\forall x.\ x \in A \vee x \notin A$. In usual settings of constructive reverse mathematics, this principle is classified as strictly weaker than LEM, LPO, or even $\neg\neg$LPO.

Our proof is explained in (analytic) textbook computability theory based on a model of computation, but additionally formally backed by a machine-checked development in synthetic computability using the Rocq proof assistant.

I. Introduction

Most if not all undecidability proofs for natural problems, i.e. occurring in mathematics or theoretical computer science, are *by reduction* from the halting problem K, or potentially a generalised variant thereof. There are various forms of reduction used in the literature, such as many-one or truth-table truth-table reductions. The most general form of computable reduction was introduced in Turing's PhD thesis [1] and is thus called Turing reduction in Post's seminal paper establishing a theory of computability and reduction [2].

Turing reductions generalise computability to be relative to an oracle B, which can be an arbitrary and in particular undecidable set. The underlying model of computation is extended with an instantaneous oracle querying operation, which allows determining $x \in B$. A Turing reduction of A to B, written $A \preceq_T B$, is a decision procedure for A relative to B, and thus is considered the most general form of intuitive computable reducibility – an insight on the same level as the Church-Turing thesis, i.e. relating an intuitive concept with a formal definition.

In his 1944 paper, Post asked a famous question: Can all possible undecidability proofs for semi-decidable by undecidable sets given as some form of computable reduction from the halting problem and thus by Turing reduction? I.e., he asked whether there exists a semi-decidable but undecidable set which is not (Turing-)reducible from the halting problem. Post was able to answer his own question for many-one and truth-table reducibility, but the most general case for Turing reducibility staid open and became known as *Post's problem*.

It was quickly identified as a fundamental question in the rising field of computer science: Are reductions the only tool necessary, or is there finer structure to the emerging subfield of computability theory? The quest for a solution to Post's problem led to the comprehensive development of an entire discipline, and was completed independently by Friedberg [3] and Muchnik [4]. They both proved a stronger theorem by constructing two Turing-incomparable semi-decidable sets A and B. Since if A would be undecidable by reduction from the halting problem, i.e. $\mathsf{K} \preceq_T A$, then also $B \preceq_T \mathsf{K} \preceq_T A$, because all semi-decidable sets reduce *to* the halting problem.

Both Friedberg and Muchnik introduced a general technique for their proof, which remains influential even in modern day computability: the priority method. In general, priority arguments can get highly complicated, and even the proof of Friedberg and Muchnik is well-known to be opaque and technical and thus often left out in introductions to computability.

For Post's problem, simpler solutions exist, both priority-based and priority-free. One of the former kind is a construction due to Soare establishing the existence of a so-called low simple set [5, Theorem 4.1]. The terminology *simple* stems from Post's seminal paper [2], where he constructed such a set to solve Post's problem for many-one reducibility. The terminology *low* refers to the fact that the set ends up to be strictly below the halting problem. The formal definition of lowness is that A is low if the Turing jump A' of A, defined as the halting problem for machines with access to oracle A, Turing reduces to K, i.e. $A' \preceq_T \mathsf{K}$ (actually, even $A' \equiv_T \mathsf{K}$). The key property of Turing jumps is that they strictly increase the complexity of a set, i.e. that $A' \not\preceq_T A$ while $A \preceq_T A'$. Thus, $\mathsf{K} \not\preceq_T A$ is ensured by lowness. Soare uses the notion of *limit computability*, introduced by Shoenfield [6] and independently re-discovered by Gold [7] as an elegant and economical way to establish lowness. Soare calls the theorem establishing a low simple set via Turing jumps the "jump theorem", we follow his terminology and refer to this specific theorem with proof via limit computability as Soare's jump theorem.

In our work, we show that the construction of a low simple set can be modularly given, and thus make the proof more accessible because properties can be shown one after the other.

Contribution 1: *We present a modular solution for Post's problem based on Soare's jump theorem, providing a low simple set parameterised in a wall function determining a lower bound for the elements to be added during the priority argument. Simpleness and lowness can then be independently established via requirements on this wall function.*

Constructive mathematics is a school which takes a different perspective on the interpretation of logical disjunction and existence. In classical mathematics, where the law of excluded middle is available as logical rule, existence arguments are often carried out using proofs by contradiction, where one assumes that an object does not exist and then deduces a logical contradiction from this assumption, i.e. proofs do not provide a method to construct witnesses of existentials.

In principle, a classical solution for Post's problem might show the existence of a semi-decidable but undecidable set not reducible from the halting problem without giving any information on how this set looks like.

In contrast, constructive mathematics enforces providing witnesses for existential arguments explicitly, and similarly forces to provide a concrete method to prove disjunctions of the form $p \vee q$ where one side needs to be explicitly chosen. Not all classical theorems will have constructive proofs. Thus, the motivations of constructive mathematics align closely with those of computability theory: understanding what are the boundaries of what can be computed or proved with or without access to oracles. In fact, constructive logic has a natural interpretation in so-called *realisability models* [8], [9], where formulas, in particular disjunctions and existential formulas, are assigned computational content.

It can be argued that the direct nature of constructive proofs is easier to understand, since unnecessary indirections via proofs by contradictions are avoided as much as possible. As such, constructive proofs, if available, can also be seen to contribute to a simpler presentation of well-known material.

Whenever an argument cannot be carried out fully constructively, there are two ways out: The first is to enrich intermediate statements with double negations, since for proving a doubly negated existence one may use arbitrary case distinctions, and, by the very meaning of double negation, proof by contradiction. However, this might lead to weakened or hard to read theorems. The second way can be seen as analogous to what one does to classify uncomputable problems in computability theory. The analogue to oracles in computability theory for constructive mathematics is the explicit use of local assumptions: if a proof requires a non-constructive case analysis, the ability to perform these can be assumed as a hypothesis to the theorem.

Usually, it suffices to add the law of excluded middle $P \vee \neg P$ to constructive logic to recover the full strength of classical mathematics. This is not fully accurate, as constructive logic often also comes without strong function existence principles such as the axiom of choice or even countable choice, but since for computability theory choice principles play virtually no role we do not discuss them here. Of course, the law of excluded middle, while sufficient, is not satisfying to understand the logical status of theorems: we could have used in classical logic directly. There is however a multitude of weakenings of the law of excluded middle which are still non-constructive, such as the limited principle of omniscience (LPO), corresponding to an oracle for the halting problem. We discuss these principles in detail later in this paper, see e.g. the book by Troelstra and van Dalen for a full overview [10].

The last and most conclusive step in the endeavour to classify foundational mathematical theorems constructively is the programme of constructive *reverse* mathematics [11], [12], where one proves that not only does a theorem follow from certain logical assumptions, but even that the assumptions are necessary in the sense that they follow from assuming the statement of the theorem.

In this paper, we do not do any reverse analysis, but we lay a foundation for it: We try to minimise our use of assumptions as much as possible, and in the end can show that LPO is sufficient to prove the existence of a low simple set, where in fact LPO is only needed for the lowness part. For the solution to Post's problem, where after constructing a candidate set we need to prove that a certain reduction is *not* possible, it then suffices to assume the double negation of LPO, because in a negative context where one wants to prove a contradiction, $\neg\neg$LPO suffices to deduce LPO. Other weaker variants of LPO appear in the literature: First, Fujiwara and Kohlenbach [13] identify other doubly negated principles related to LPO, which are weaker than LPO. One such instance is $(\neg\neg\Sigma_1)$-LEM, which states that for $A \in \Sigma_1$, $\neg\neg\forall x.\ x \in A \vee x \notin A$, i.e. places the double negation behind the quantification of A. $(\neg\neg\Sigma_1)$-LEM, in the formulation of Fujiwara and Kohlenbach, is strictly weaker than $\neg\neg$LPO. Secondly, Cohen et al. [14], [15] show that by varying the definition of Σ_1 to define decidability either via a computational model (as is standard in computability theory) or via the existence of boolean functions (as is standard in constructive mathematics) changes the strength of LPO. In particular, $\mathsf{LPO}_{\mathbb{B}}$ defined via boolean functions is strictly stronger than $\mathsf{LPO}_{\mathsf{PR}}$, defined via a (primitive recursive or recursive) model of computation. In this setting, however, $(\neg\neg\Sigma_1)$-$\mathsf{LEM}_{\mathsf{PR}}$ and $\neg\neg(\Sigma_1$-$\mathsf{LEM}_{\mathsf{PR}})$ become equivalent.

This leads us to the following, where we write $(\neg\neg\Sigma_1)$-LEM for $(\neg\neg\Sigma_1)$-$\mathsf{LEM}_{\mathsf{PR}}$ for the rest of the introduction.

Contribution 2: *We are careful to make our proof as constructive as possible, and only assume weak additional principles to enable apparently non-constructive components of the proof, paving the way for a constructive reverse analysis of the problem in the future. Concretely, we show that* $(\neg\neg\Sigma_1)$-LEM *suffices to solve Post's problem.*

Due to their similar motivational nature, computability theory and constructive mathematics have a key aspect in presentations in common: They rely on subtle and detailed arguments that require a lot of experience to carry out correctly and verify accurately. As a consequence, a tension arises between intuitive explanations to give reasons *why* theorems hold, and verifiable proofs showing *how* they can be proved. It may be the case that intuitive explanations for constructive computability arguments are not detailed enough to be verifiable, while fully verifiable formal proofs need to discuss too many details to remain easily surveyable.

One way out, which is on the rise in many subcommunities of mathematics, is mechanisation in a so-called proof assistant such as Rocq, Isabelle, or Lean. However, there is a central problem for the mechanisation of computability theory: the reliance on models of computation. Post himself comments:

> *That mathematicians generally are oblivious to the importance of this work of Gödel, Church, Turing, Kleene, Rosser and others as it affects the subject of their own interest is in part due to the forbidding, diverse and alien formalisms in which this work is embodied.*

While computability theory in textbooks and articles makes heavy use of the (informal) Church-Turing thesis, fully formal,

mechanised work would have to carry out computability arguments in all details – an infeasible overhead [16]–[21].

An elegant alternative is proposed by synthetic methods, explored for computability theory by Richman [22] and elaborated further by Bridges and Richman [23] and Bauer [24], [25]. Ultimately, this approach goes back to Kreisel [26], who proposed the axiom CT (Church's thesis) in constructive foundations stating that any function $\mathbb{N} \to \mathbb{N}$ is computable, i.e. has a code w.r.t. the effective enumeration φ_e of (partial) computable functions. CT has been shown consistent for arithmetic [8], [26] and type theory [27], [28], through variants of the effective topos [29]. Richman's insight was that it is philosophically unsatisfying to introduce a model of computation just to discard it via CT, and instead proposed a fully synthetic axiom just postulating an enumeration of the partial function space from $\mathbb{N}$ to $\mathbb{N}$. Via realisability models mentioned before, theorems in synthetic computability correspond directly to analytic theorems, with concise but precise proofs, reduced to their essence by sparing all overhead introduced via particular models of computability. Synthetic computability hides what ought to be hidden without sacrificing rigour.

However, synthetic computability in the style of Richman, Bridges, and Bauer is anti-classical: axioms like LPO are disprovable. Two more recently proposed approaches avoid this problem: First, Swan [30] suggests oracle modalities in homotopy type theory which essentially nest double negations in definition everywhere, allowing classical reasoning at any point. This suggestion carries over to non-relative computability and allows proving inherently classical theorems, but it does no longer allow classifying the exact assumptions necessary. Secondly, Forster [31] suggests axioms for computability theory in the Calculus of Inductive Constructions [32]–[34], the type theory underlying the Rocq proof assistant [35], which are compatible even with strong classical assumptions such as the law of excluded middle as long as no choice axioms in the form of function existence principles are assumed.

In this setting, Forster [36] proves Rice's theorem, building on synthetic definitions given in [37]. Forster and Jahn [38] construct simple and hypersimple sets, solving Post's problem for many-one and truth-table reducibility, and discuss constructive notions of infinity. Forster, Lauermann, and Kunze [39] show that a simple set can also be constructed based on Kolmogorov complexity. Forster, Mück, and Kirst explain how oracle computability can be synthetically captured in this setting [40], and prove the Kleene-Post and Post's theorem for the arithmetical hierarchy [41].

Contribution 3: *We introduce and explain all relevant notions in synthetic computability, building on the framework of Forster, Mück, and Kirst [40].*

Starting from this synthetic perspective on oracle computation, it is then feasible to mechanise our proofs with a proof assistant. In fact, this not only yields correctness guarantees, allowing to focus on intuitive explanations in the paper proof, but it also helps keeping track of exact logical dependencies.

Contribution 4: *We give a machine-checked proof in the Rocq proof assistant, available online.*

II. Proof Strategy

We recall standard notions from (relative) computability theory that are necessary to state Post's problem and some more advanced notions necessary to state Soare's jump theorem. We conclude this section by a proof that Soare's jimp theorem solves Post's problem. The proof of Soare's jump theorem will then be developed in the course of the paper, and more notions from relative computability are introduced when needed.

We work with a universal function φ for any Turing-complete model of computation. We write $\varphi_e(x) = v$ if the e-th computable function terminates on value x with value v. We write $\varphi_e(x)\downarrow$ if we want to indicate that such a v exists. We write $\langle x, y\rangle$ for a computable pairing function.

As an example for the use of both, we call a code u *universal* if for any e, $\varphi_u\langle e, x\rangle = \varphi_e(x)$. This equation also demonstrates a notation of equality for partial functions: We here mean, formally, $\forall v.\ \varphi_u\langle e, x\rangle = v \leftrightarrow \varphi_e(x) = v$. In particular, we then also have that if one side is undefined, so is the other.

A code e is *total* if for any x there exists v with $\varphi_e(x) = v$.

A set A of natural numbers is *decidable* if there is a total e such that $\forall x.\ x \in A \leftrightarrow \varphi_e(x) = 1$. Note that one can equivalently require $\varphi_e(x) \leq 1$ for all x, i.e. then have $\neg(x \in A) \leftrightarrow \varphi_e(x) = 0$ instead of just $\neg(x \in A) \leftrightarrow \varphi_e(x) \neq 1$.

A set A of natural numbers is *semi-decidable* if there is a (not necessarily total) e such that $\forall x.\ x \in A \leftrightarrow \varphi_e(x)\downarrow$. The crucial difference to decidability comes from the fact that φ_e is partial and we have $\forall x.\ \neg(x \in A) \leftrightarrow \neg(\varphi_e(x)\downarrow)$. Note that one can equivalently ask $\varphi_e(x) = 1$ instead of termination, but then $\neg(x \in A)$ can be indicated by termination with any other value or non-termination, so we usually use the first definition unless explicitly mentioned.

To prove decidability, semi-decidability, and other computability results we will make frequent use of the Church-Turing thesis, i.e. argue informally that a construction is computable to obtain a corresponding code e. At no point we actually need to discuss details of the model of computation, a style of proofs pioneered in the textbook by Rogers [42].

As an instructive example, we prove that the self-halting problem $\mathsf{K} := \{e \mid \varphi_e(e)\downarrow\}$ undecidable: Assume it would be decidable via a code e. Then in particular, we could define the function f which on input x diverges if $\varphi_e(x) = 1$ and otherwise returns 1. Now by the Church-Turing thesis, there is a code e_f for f because f is intuitively computable, but then $\varphi_{e_f}(e_f)\downarrow \leftrightarrow e_f \notin \mathsf{K} \leftrightarrow \neg(\varphi_{e_f}(e_f)\downarrow)$, a logical contradiction.

We now turn to relative notions of computability. We write $\varphi_e^B x$ for the value of running the e-th oracle computable function on input x with an oracle for the set B. During a terminating computation, only finitely many questions may be asked to the oracle, i.e. it can be seen as continuous, see e.g. the discussion in Odifreddi's book [43, §II.3].

We now introduce Turing reductions as notion of relative decidability. A set A of natural numbers Turing-reduces to B – we write $A \preceq_T B$ – if there exists e such that

$$\forall x.\ x \in A \leftrightarrow \varphi_e^B x = 1 \quad \text{and} \quad \forall x.\ \neg(x \in A) \leftrightarrow \varphi_e^B x = 0.$$

Similarly, a set A of natural numbers is semi-decidable relative to B if there is e such that $\forall x.\ x \in A \leftrightarrow \varphi_e^B x \downarrow$.

The Turing jump of a set A is the self-halting problem relativised to A:

$$A' := \{e \mid \varphi_e^A(e) \downarrow\}$$

In principle, any definition here will serve the purpose as long as $A \preceq_T A'$ but $A' \not\preceq_T A$. The former holds because in order to decide A with access to an oracle for A', on input x one just has to query the oracle for the code of e_x of the oracle computation which ignores its argument, queries its oracle at position x, and returns the outcome. The latter just a generalisation of the undecidability of the halting problem.

Now turning to the main topic of this paper, Post's problem asks for a semi-decidable but undecidable problem A that does not Turing-reduce from the halting problem, i.e. $\mathsf{K} \not\preceq_T A$.

Soare's jump theorem [5] proposes a solution to Post's problem based on low simple sets. A set of natural numbers is simple if it is semi-decidable, and its complement is infinite and does not have a semi-decidable infinite subset. A set of natural numbers A is low if its Turing jump A' is Turing reducible to K, i.e. if $A' \preceq_T \mathsf{K}$.

Theorem 1. *Soare's jump theorem solves Post's problem.*

Proof. The first observation here is that simple sets are semi-decidable but undecidable, since decidability would yield that the infinite complement is semi-decidable as well and thus would contain itself as semi-decidable infinite subset. The second observation is that low sets A do not fulfill $\mathsf{K} \preceq_T A$, because otherwise $A' \preceq_T A$ would hold by transitivity. □

III. Constructive Mathematics

In this section, we introduce foundations of constructive mathematics, in particular we introduce which logical principles are available natively, and which principles can be assumed axiomatically. Precisely we introduce Markov's principle MP [44] and the limited principle of omniscience LPO, as well as their generalisations Σ_n-DNE, Π_n-DNE, Σ_n-LEM, and Π_n-LEM [45]. For expository purposes, in this section we connect these axioms directly to results in computability theory, either showing how they can be used to prove results, or whenever possible by immediately demonstrating how they are *equivalent* to computability results.

For most of this paper, we work in informal constructive mathematics, not committing to a foundation. Our results are applicable to type theories such as Martin-Löf type theory, the Calculus of Inductive Constructions, or variants of homotopy type theory, systems of constructive arithmetic like HA, EL_0, EL, or HA^ω. Note that we work with unrestricted comprehension, that is the formation of subsets for arbitrary properties is allowed.

To state axioms, depending on the base system one might have to resort to axiom schemes for systems that cannot quantify over propositions internally. We gloss over this potential issue and write $\mathbb{P}$ for the set of all propositions, readily available e.g. in CIC. We treat propositions impredicatively, leaving open whether our proof could be given predicatively, which is well possible.

We will mainly be concerned with axioms that take the general form of the law of excluded middle or double negation elimination. The law of excluded middle LEM states that any proposition is definite, i.e. holds or does not hold:

$$\mathsf{LEM} := \forall P \in \mathbb{P}.\ P \vee \neg P.$$

The principle of double negation elimination states that any proposition is stable under double negation:

$$\mathsf{DNE} := \forall P \in \mathbb{P}.\ \neg\neg P \to P$$

If $\neg\neg P$ holds we say that P *holds classically*. We frequently say that a number n classically exists for $\neg\neg\exists n$ or that a function is classically terminating, in the latter case meaning that pointwise for every input an output classically exists.

A short side remark on terminology: We use "definite" here to indicate that one can logically determine the status of P. In constructive mathematics, this is often called a "decidable" proposition, which is confusing in our context since the assumption of a proposition being logically definite does not yield a computational decider. Other frequently used terminology is "classical", which we refrain from mainly because it can otherwise be confused with a proposition being classically true.

Note first that equivalently both principles can be stated in terms of sets of natural numbers:

Fact 1. *We have* $\mathsf{LEM} \leftrightarrow \forall x \in \mathbb{N}.\ \forall A \subseteq \mathbb{N}.\ x \in A \vee x \notin A$ *and* $\mathsf{DNE} \leftrightarrow \forall x \in \mathbb{N}.\ \forall A \subseteq \mathbb{N}.\ \neg(x \notin A) \to x \in A$.

In the first case, we then also say that A is definite, i.e. when $x \in A$ is definite. In fact the equivalences holds for every non-empty base set in place of $\mathbb{N}$ but the upcoming restrictions of these principles are usually stated for $\mathbb{N}$ alone.

Furthermore, since any definite proposition is stable under double negation, it is trivial to see that $\mathsf{LEM} \to \mathsf{DNE}$. For the converse direction, the crucial observation is that the double negation of a case distinction holds constructively:

Fact 2. $\forall P \in \mathbb{P}.\ \neg\neg(P \vee \neg P)$

Corollary 3. $\mathsf{LEM} \leftrightarrow \mathsf{DNE}$

We emphasise that while definite propositions are stable under double negation, the converse direction does not hold. This plays a role when next restricting the classes of considered propositions, as then restricted DNE in general will be weaker than restricted LEM.

In computability theory it is common to speak of Σ_1^0 sets as variant of semi-decidable sets. Because we do not use the second-order variants Σ_1^1 etc. we will only write Σ_1 in this paper. A set A of natural numbers is Σ_1 if there exists a decidable set B with $x \in A \leftrightarrow \exists n.\ \langle n, x\rangle \in B$. We treat Σ_1 as a set of sets of natural numbers, then exactly coinciding with our definition of semi-decidable sets. In constructive logic and meta-mathematics, often other slightly more liberal definitions of Σ_1 are used, we discuss these in Section VIII.

One can then weaken LEM and DNE to only consider Σ_1 sets. The two resulting principles are commonly called the *limited principle of omniscience* and *Markov's principle*.

$$\mathsf{LPO} := \forall A \in \Sigma_1.\, \forall x.\, x \in A \vee x \notin A$$
$$\mathsf{MP} := \forall A \in \Sigma_1.\, \forall x.\, \neg(x \notin A) \to x \in A$$

As before, it is trivial to see that LPO implies MP. The converse does not hold by the above mentioned lack of locality, because Σ_1 is neither closed under disjunction nor complementation – the halting problem is the canonical example.

Both principles have natural equivalents in the language of computability theory:

Fact 4. LPO *is equivalent to* K *being definite.*

Proof. The direction from left to right requires $\mathsf{K} \in \Sigma_1$, which follows because K is semi-decidable. The converse requires many-one completeness of K, i.e. for any $A \in \Sigma_1$ there exists a total e such that $\forall x.\, x \in A \leftrightarrow \varphi_e(x) \in \mathsf{K}$. □

The equivalent to MP in the language of computability theory is a results usually attributed to Post:

Fact 5 (Post's Theorem)**.** MP *is equivalent to the statement that if both A and its complement are semi-decidable, then A is decidable.*

Proof. See e.g. [10], [24]. □

In Fact 2 we have already established that the double negation of any case analysis is constructively provably. More generally, one can show that double negation is a form of modality allowing for classical reasoning:

Fact 6. 1) $P \to \neg\neg P$.
2) *To prove $\neg Q$ it suffices to prove $(P \vee \neg P) \to \neg Q$ for any P.*
3) *To prove $\neg Q$ from assumption $\neg\neg P$ it suffices to prove $P \to \neg Q$.*

We now generalise LPO and MP which are concerned with Σ_1 sets even further to the full arithmetical hierarchy of Σ_n and Π_n sets, following Akama, Berardi, Hayashi, and Kohlenbach [45]. A set A is Σ_0 or Π_0 if it is decidable. A set A is Σ_{n+1} if there is $B \in \Pi_n$ with $x \in A \leftrightarrow \exists n.\, \langle n, x\rangle \in B$. A set A is Π_{n+1} if there is $B \in \Sigma_n$ with $x \in A \leftrightarrow \forall n.\, \langle n, x\rangle \in B$.

One can deduce principles Σ_n-LEM, Π_n-LEM, Σ_n-DNE, and Π_n-DNE from these:

$$\Sigma_n\text{-LEM} := \forall A \in \Sigma_n.\, \forall x.\, x \in A \vee x \notin A$$
$$\Pi_n\text{-LEM} := \forall A \in \Pi_n.\, \forall x.\, x \in A \vee x \notin A$$
$$\Sigma_n\text{-DNE} := \forall A \in \Sigma_n.\, \forall x.\, \neg(x \notin A) \to x \in A$$
$$\Pi_n\text{-DNE} := \forall A \in \Pi_n.\, \forall x.\, \neg(x \notin A) \to x \in A$$

We refer to the paper by Akama, Berardi, Hayashi, and Kohlenbach [45] for a detailed discussion of their interrelation.

We make use of the following aspect of Post's hierarchy theorem [41].

Fact 7 (Post's hierarchy theorem)**.** *Given Σ_1-LEM, if $A \in \Sigma_2$ then A is semi-decidable in* K.

Lastly, we discuss two forms of weakenings: Double negations of these principles, and variants with an intermediate use of double negation.

First, clearly every principle implies its double negation:

Lemma 8. *Σ_n-LEM $\to \neg\neg(\Sigma_n$-LEM$)$ as well as Σ_n-DNE $\to \neg\neg(\Sigma_n$-DNE$)$, but the converse does not hold.*

Proof. By Fact 6 (1). The converse does not hold because double negation shift implies $\neg\neg(\Sigma_1\text{-LEM})$ [46] and thus $\neg\neg(\Sigma_1\text{-DNE})$, but not Σ_1-DNE and thus not Σ_1-LEM, because it holds in Berger and Oliva's model based on modified bar recursion, which does not validate Σ_1-DNE analogously to how Kreisel's modified realisability does not validate it [47]. □

Fujiwara and Kohlenbach introduce an even further weakening [13]: principles $(\neg\neg\mathcal{C})$-LEM and $(\neg\neg\mathcal{C})$-DNE for $\mathcal{C}$ being either Σ_n or Π_n:

$$(\neg\neg\mathcal{C})\text{-LEM} := \forall A \in \mathcal{C}.\, \neg\neg\forall x.\, x \in A \vee x \notin A$$
$$(\neg\neg\mathcal{C})\text{-DNE} := \forall A \in \mathcal{C}.\, \neg\neg\forall x.\, \neg(x \notin A) \to x \in A$$

We from here on focus on the LEM variants because only they are relevant for the present paper.

Lemma 9. *$\neg\neg(\mathcal{C}$-LEM$) \to (\neg\neg\mathcal{C})$-LEM for any $\mathcal{C}$.*

For the chosen definition of Σ_n making use of decidability and thus a model of computation, the converse actually holds, while other definitions of decidability of a set e.g. in terms of (non-computable but constructive) functions $f \in \mathbb{N} \to \mathbb{N}$ with $x \in A \leftrightarrow fx = 1$ this proof does not go through. We discuss this in detail in Section VIII.

Note that the following holds in general:

Fact 10. $(\neg\neg\mathcal{C})\text{-LEM} \leftrightarrow \neg\neg((\neg\neg\mathcal{C})\text{-LEM})$

Proof. Because $(\forall x \in A.\neg Px) \leftrightarrow \neg\neg(\forall x \in A.\neg Px)$ for any set A and predicate P. □

To summarise, we have that Σ_n-LEM strictly implies $\neg\neg(\Sigma_n\text{-LEM})$ which implies $\neg\neg\Sigma n$-LEM, where the implication is an equivalence in our setting, but not in all settings of constructive reverse mathematics, see Section VIII.

Lastly, there are several notions of finiteness in constructive mathematics. We are mainly interested however in weak notions of infinity in this paper, thus we only introduce one notion of finiteness: A set of natural numbers A is finite if it is bounded, i.e. if $\exists n.\, \forall x \in A.\, x \leq n$. We say that a set A is infinite if it is not finite. This notion is (constructively) equivalent to some other notions of infinity that we do not discuss here, but there are also stronger notions of infinity, see the paper by Forster and Jahn [38], which do not allow the constructive construction of a simple set, and the paper by Nemoto [48] for a more general discussion of infinity in constructive mathematics.

IV. Limit Computability

The notion of limit computability was introduced by Shoenfield [6] and independently rediscovered by Gold [7]. It relies on a notion of convergence. We say that a total computable function f between natural numbers is convergent to a limit z if $fy = v$ for all $y > x$ for some x. We then write $\lim_{y\to\infty} fy = v$.

Building on our terminology from before, we say that a family of functions f_i is classically convergent if a limit classically exists for every i, i.e. $\forall i.\ \neg\neg\exists v.\ \lim_{y\to\infty} fy = v$.

A partial function F is limit computable via e if φ_e is total and $Fx = v \leftrightarrow \lim_{y\to\infty} \varphi_e\langle x, y\rangle = v$. Consequently, a set A is limit-computable if its characteristic function is limit computable via e. Spelt out, this means that A is limit-computable if there exists e with

$$x \in A \leftrightarrow \lim_{y\to\infty} \varphi_e\langle x, y\rangle = 1$$
$$x \notin A \leftrightarrow \lim_{y\to\infty} \varphi_e\langle x, y\rangle = 0$$

Note that classically, this is equivalent to stating that for every x, the limit $\lim_{y\to\infty} \varphi_e\langle x, y\rangle$ exists and reflects $x \in A$. Constructively, our definition is however *weaker*: it does not imply the existence of the limit for every x, because otherwise this value could be used to determine $x \in A$, i.e. A would be definite.

We now prove the limit lemma, stating that a predicate is limit computable if and only if it is Turing-reducible to K. Both directions of this proof rely on Σ_1-LEM.

The forward direction makes use of Post's hierarchy theorem, but also of a relativised version of Post's theorem about bi-semi-decidable sets:

Fact 11 (Relativised Post's Theorem)**.** *Assuming membership in B definite (i.e. $\forall x.\ x \in B \vee x \notin B$), if both A and its complement are semi-decidable in B, then $A \preceq_T B$.*

Proof. See [40]. □

Lemma 12. *Given Σ_1-LEM, if A is limit computable, then $A \preceq_T \mathsf{K}$.*

Proof. Let A be limit computable, i.e. take e with

$$x \in A \leftrightarrow \lim_{y\to\infty} \varphi_e\langle x, y\rangle = 1$$
$$x \notin A \leftrightarrow \lim_{y\to\infty} \varphi_e\langle x, y\rangle = 0$$

Now it is obvious that both A and its complement are in Σ_2, since $\lim_{y\to\infty} fy = v$ is of the form $\exists x.\forall y. Dxy$ for a decidable formula D. Thus, using Π_1-LEM and the corollary of the Post hierarchy theorem discussed in Fact 7, they are both semi-decidable in K. Now by relativised Post's theorem (Fact 11), $A \preceq_T \mathsf{K}$, since Σ_1-LEM entails $\forall x.\ x \in \mathsf{K} \vee x \notin \mathsf{K}$. □

We now turn to the converse direction. Here, we need a way to computationally approximate the result of an oracle computation. For decidable oracles, one can use the decider to simulate an oracle computation with this decidable oracle. For semi-decidable but undecidable oracles, this is not possible because the semi-decider will not terminate in some cases. However, one can still approximate the result of such an oracle computation in the limit, independently of the concrete semi-decider.

To do so, we define $\varphi_e^B(x)[n]$ to run the computation denoted by e for n steps. If a question to the oracle is queried, the semi-decider for B is run for n steps. If it terminates, the computation is resumed with answer "yes", otherwise with answer "no". If the overall computation terminates with value v in n steps like this, $\varphi_e^B(x)[n] := \ulcorner v \urcorner$. Otherwise, $\varphi_e^B(x)[n] := \mathsf{none}$.

Note that the value of $\varphi_e^B(x)[n]$ for small n might be wrong and depend on the concrete semi-decider used. However, in the limit, it is correct, i.e.

$$\varphi_e^B(x) = v \leftrightarrow \lim_{n\to\infty} n.\ \varphi_e^B(x)[n] = \ulcorner v \urcorner.$$

The converse direction of this is easy to prove under the assumption that B is definite, i.e. Σ_1-LEM. We did not investigate whether this assumption is strictly necessary, but since it is used in the forward direction of the limit lemma anyways, this will not affect the overall strength of the result.

In one part of the paper, we need the exact results of running $\varphi_e^B(x)[n]$ with a concrete semi-decider. We are explicit about this abuse of notation at the point (see Theorem 3).

We now use this construction for the backward direction of the limit lemma.

Lemma 13. *Assuming Σ_1-LEM, if $A \preceq_T \mathsf{K}$, then A is limit computable.*

Proof. Let A be Turing-reducible to K via e. We define

$$f\langle x, n\rangle := \begin{cases} v & \text{if } \ \varphi_e^{\mathsf{K}}(x)[n] = \ulcorner v \urcorner \\ 0 & \text{otherwise} \end{cases}$$

which is computable by the Church-Turing thesis. Because e is a Turing reduction, it can be shown total on oracle K using Σ_1-LEM, i.e. $\lim_{n\to\infty} \varphi_e^{\mathsf{K}}(x)[n] = \ulcorner v \urcorner$ for some v. Now

$$\lim_{n\to\infty} f\langle x, n\rangle = v \leftrightarrow \lim_{n\to\infty} \varphi_e^{\mathsf{K}}(x)[n] = \ulcorner v \urcorner \leftrightarrow \varphi_e^{\mathsf{K}}(x) = v.$$

In particular then

$$x \in A \leftrightarrow \varphi_e^{\mathsf{K}}(x) = 1 \leftrightarrow \lim_{n\to\infty} f\langle x, n\rangle = 1$$
$$x \notin A \leftrightarrow \varphi_e^{\mathsf{K}}(x) = 0 \leftrightarrow \lim_{n\to\infty} f\langle x, n\rangle = 0$$

where the first one holds because e is a Turing reduction and the second we just proved. □

V. Constructing a Simple Set

The (finite injury) priority method was introduced independently by Friedberg [3] and Muchnik [4] to construct a solution to Post's problem. In general, the priority method yields a construction of a set in finite stages Γ_n such that the final set $\Gamma = \bigcup_{n\in\mathbb{N}} \Gamma_n$ satisfies a family of so-called requirements R_e. Requirements are sorted according to a certain priority, and in the stage-wise construction elements are added trying to satisfy requirements with highest priority first. In this section we will

only deal with positive requirements that remain satisfied by later stages once they are satisfied by a stage Γ_n.

The core of the construction is to define an extension function $\gamma_n^l \in \mathbb{N}$ where n is a number and l is the (encoding of) a finite set of numbers. We treat the domain of γ as optional, i.e. write $\gamma_n^l = \ulcorner b \urcorner$ and $\gamma_n^l = \mathsf{none}$ as before. $\gamma_n^l = \ulcorner b \urcorner$ indicates that b is supposed to be added for stage l in step n.

We will give a construction of such a γ parametric in a total, computable wall function $\omega_n^l(e) \in \mathbb{N}$ computing a lower bound for elements to be added in a stage. The terminology wall function is due to Soare [49]. Keeping ω as a parameter allows modularly requiring increasingly stronger properties of ω which in turn render Γ semi-decidable, or simple, or simple and low.

For the construction of the final set, we define the n-th stage of γ as $\Gamma_0 := \{\}$, $\Gamma_{n+1} := \Gamma_n \cup \{b \mid \gamma_n^{\Gamma_n} = \ulcorner b \urcorner\}$.

Note that $|\Gamma_n| \leq n$ and that Γ_n is cumulative, i.e. $n \leq m$ implies $\Gamma_n \subseteq \Gamma_m$. Recall that the result Γ of the priority method is $\Gamma := \{x \mid \exists n.\ x \in \Gamma_n\}$. By definition, Γ is semi-decidable, since Γ_n is decidable.

We now turn to the construction of a simple set.

Requirements: Following the definition of a simple set, the key property we want to ensure for Γ is that it intersects every infinite semi-decidable set, (because then its complement cannot contain an infinite semi-decidable subset) while ensuring that the complement of Γ stays infinite.

Recall that we definite infinity as the negation of finiteness. This is crucial here for constructive proofs: If we could for instance constructively prove that $\forall n.\ \exists x > n.\ x \in \overline{\Gamma}$, then meta-theoretically we know that this constructive proof corresponds to an enumeration of an infinite set in $\overline{\Gamma}$ – since enumerable sets are semi-decidable, this would mean any such infinite set contains an infinite semi-decidable subset and we cannot establish the existence of any simple set [38].

We first define a family of requirements P_e for a given set A ensuring this first property, while infinity is handled by the construction of γ.

This family of requirements is parameterised in a code e with the ultimate goal to show that Γ satisfies all P_e. In the proof, we will also discuss which requirements are satisfied by a stage Γ_i.

The definition makes use of the notation $W_e := \{x \mid \varphi_e(x)\downarrow\}$ for the set semi-decided by code e. We define P_e to say: If W_e is infinite, then A and W_e are not disjoint.

$$P_e \ := \ W_e \text{ infinite } \ \rightarrow \neg(A \# W_e)$$

Here we write $B \# C$ for two sets being disjoint. Note that being not disjoint is the same as classically intersecting, i.e. $\neg\neg\exists x. x \in B \cap C$.

The negative conclusion of P_e is crucial to make the following fact constructively provable. The fact basically already occurs in Post's seminal construction of a simple set [2], Post calls the established property "immunity".

Fact 14. *If a set A meets the requirement P_e for all e, then its complement does not contain any infinite semi-decidable set.*

Proof. Assume a set A meets all requirements P_e, but its complement has an infinite semi-decidable subset. Let e be the index of this set $W_e \subseteq \overline{A}$. We need to prove a contradiction.

Now P_e states that A is not disjoint from W_e, but because we need to prove a contradiction, we may assume that A intersects $W_e \subseteq \overline{A}$ – a contradiction. □

Construction: We now construct a suitable extension function γ, for which it will later be possible to prove that the resulting set Γ meets all requirements P_e, and that it is infinite. Recall that semi-decidability of Γ follows by construction, and thus Γ will be simple. The construction remains parametric in a total, computable wall function ω_n^l.

To define γ we need finite, decidable, cumulative, exhausting approximations $W_e[n]$ of W_e. We can define $W_e[n]$ as the set of all $x \leq n$ such that $\varphi_e(x)$ terminates in n steps.

We define γ_n^l as the function which searches for the least pair (e, x) in lexical ordering such that

1) $e < n$
2) $l \# W_e[n]$
3) $x \in W_e[n]$
4) $\omega_n^l(e) < x$

and then returns $\ulcorner x \urcorner$ and none otherwise. Note that the search process is finite due to conditions (1) and (3), i.e. γ is total. By the Church-Turing thesis, γ is computable.

Since we always choose the smallest e, we prioritise the requirements where e is smaller, i.e. they have a higher priority.

Conditions (1)-(3) are relevant to ensure that every infinite semi-decidable set is intersected as imposed by the requirements P_e, while controlling the definition of ω allows condition (4) to ensure that not too many elements are added to Γ so its complement stays infinite.

Verification of Requirements: We say that the requirement P_e *receives attention at stage* n, if e is the least number such that conditions (1)-(4) from above are fulfilled for $l := \Gamma_n$ and for *any* x, not necessarily the least such x.

Once P_e receives attention at stage n, a new element x is added to Γ_n at the next stage to ensure that P_e is satisfied by Γ_{n+1}. P_e satisfies the following lemma:

Lemma 15. 1) *Once P_e is satisfied by Γ_n, it remains so for all Γ_m with $m > n$.*
2) *If P_e receives attention at n, it does not receive attention anymore for any $m > n$.*

We can now prove that all P_e are satisfied by Γ by imposing a condition on the wall function. The condition ensures that the wall function actually allows adding elements eventually. To provide some intuition, note that setting $\omega_n^l(e) = n$ entails Γ to be empty, since $n < x$ from condition (4) cannot be fulfilled if $x \in W_e[n]$ which entails $x \leq n$. A sufficient condition is that from some point on $\omega_n^{\Gamma_n}(e)$ does not depend on n anymore, i.e. is convergent. To keep the proof modular, we only require classical convergence since our final wall function is not constructively convergent in n.

Lemma 16. *If for all e, the wall function $\omega_n^{\Gamma_n}(e)$ is classically convergent in n, then for any infinite W_e, there classically exists n such that Γ_n intersects $W_e[n]$.*

Proof. We will assume that there classically exists n such that for all $e' < e$ and $m > n$, $P_{e'}$ does not receive attention at stage m, and prove this later in Corollary 18. Since we have to prove a classical existence in the main proof, we can assume a stage n after which no $P_{e'}$ with $e' < e$ receives attention.

Since the wall function converges classically and W_e is infinite, there classically exists a stage m such that conditions (1)(3)(4) are fulfilled for $l := \Gamma_m$ and $n := m$.

Thus, if at step $\max(n, m)$, $W_e[m]$ intersects Γ_m, then the proof is done. Otherwise, P_e receives attention at stage m. Consequently, we have that Γ_{n+1} intersects $W_e[n+1]$ – as claimed. □

Note that the pre-condition could be weakened to require that $\omega_n^{\Gamma_n}(e)$ is classically eventually bounded in n, i.e. $\forall e.\ \neg\neg\exists n_0 b. \forall n > n_0.\ \omega_n^{\Gamma_n}(e) < b$, but we refrain from introducing this notion formally.

Theorem 2. *Γ satisfies P_e if the wall function $\omega_n^{\Gamma_n}(e)$ is classically convergent in n for all e.*

Proof. We need to prove that for any infinite W_e, Γ classically intersects W_e. By Lemma 16, we have that for any infinite W_e there classically exists n such that Γ_n intersects $W_e[n]$, which is sufficient since $\Gamma_n \subseteq \Gamma$ and $W_e[n] \subseteq W_e$. □

We now prove the missing Corollary 18, i.e. for any e there classically exists n with certain properties. To do so, we prove a generalisation that makes the finitely many necessary classical case distinctions explicit and proves a (constructive) existence. This generalisation can be used to prove the classical existence by performing the finitely many classical case distinctions upfront (c.f. Fact 6 (2)), but also can be instantiated under the assumption of a suitable axiom in the next section.

Lemma 17. *For any e, if for all $e' < e$, it is definite whether there exists k such that $P_{e'}$ receives attention at stage k, then exists there n such that for all $e' < e$ and $m > n$, $P_{e'}$ does not receive attention at stage m.*

Proof. By induction on e. There is nothing to prove for $e = 0$. For the case $e + 1$, by induction hypothesis we have n such that for all $e' < e$ and $m > n$, e' does *not* receive attention at stage m. It suffices to find an n' such that for all $m > n'$, P_e does *not* receive attention at m, because we can then pick the maximum of n and n' for the main claim.

So we need to prove that for any given e, there exists n' such that for all $m > n'$, P_e does *not* receive attention at m. We do so by case analysis allowed by the assumption: If there exists n such that P_e receives attention at n, we can pick this n because then for any $m > n$, P_e does not receive attention anymore by Lemma 15 (2). If no such n exists, we are done anyways. □

Corollary 18. *For any e there classically exists n such that for all $e' < e$ and $m > n$, $P_{e'}$ does not receive attention at stage m.*

Proof. One can prove by induction on k that for all $A \subseteq \mathbb{N}$

$$((\forall x < k.\ x \in A \vee x \notin A) \to \neg Q) \to \neg Q,$$

i.e. that finitely many case distinctions are allowed to prove negative statements. Because we want to prove a classical existence, this allows us to do the e many case analyses necessary to use Lemma 17. □

Constructing a simple set: We are missing two ingredients for a simple set: First, we have to prove that the complement of Γ is infinite. This proof is constructive, mainly because the statement is a negation due to the definition of infinity. Secondly, we have to instantiate the parametric construction with a classically convergent wall function.

Once again, remember that infinity crucially is the negation of boundedness.

Lemma 19. *The complement of Γ is infinite if $\omega_n^{\Gamma_n}(e) \geq 2 \cdot e$.*

Proof. An element is added to Γ at stage n if $x \geq \omega_n^{\Gamma_n}(e) \geq 2 \cdot e$. Thus, for the predicate Γ, there classically exists *at most* n elements of all less than $2n$, and thus at least n elements in $\overline{\Gamma}$, so $\overline{\Gamma}$ is infinite. □

We now collect all missing pieces together by instantiating the wall function. Overall, we now have:

Lemma 20. *Whenever $\omega_n^{\Gamma_n}(e) \geq 2 \cdot e$ and $\omega_n^{\Gamma_n}(e)$ is classically convergent, Γ is simple.*

Corollary 21. *Γ is simple for $\omega_n^l(e) := 2 \cdot e$.*

Proof. Because $\lim_{n\to\infty} 2 \cdot e = 2 \cdot e$, i.e. $\omega_n^{\Gamma_n}(e)$ is classically convergent in n. □

VI. Constructing a Low Simple Set

In the previous section we have constructed a simple set using requirements to enforce the "immunity" property that the complement of the iteratively constructed set does not have an infinite semi-decidable subset.

In this section, we add additional requirements N_e to establish lowness of Γ, i.e. to obtain that $\Gamma' \preceq_T \mathsf{K}$, and introduce conditions on the wall function such that all N_e are satisfied. The conditions are called negative, because they avoid certain elements to be added to Γ.

We can modularly use the proof from the last section to maintain immunity, semi-decidability, and infinity of the constructed set.

Requirements: We define N_e for a given semi-decidable set A to state that the approximation $\varphi_e^A(e)[n]$ is convergent in n for all e:

$$N_e \ :=\ \exists v.\ \lim_{n\to\infty} \varphi_e^A(e)[n] = v$$

Note that v is an optional value here, and that crucially $\varphi_e^A(e)[n]$ is approximative, i.e. it might yield wrong values because the semi-decider for A is only executed for n steps

and the oracle (potentially wrongly) answers "no" on non-termination in n steps. For A being decidable, the semi-decider is instantaneous and thus does not depend on the step-index, so for A every N_e is satisfied. For semi-decidable A, wrong "no" answers might be given and thus wrong values returned by the computation. Since with higher n wrong answers might flip to a correct "yes" during a computation, this can affect what further questions are asked. If these asked questions keep flipping from a wrong "no" to a correct "yes", e.g. in a potentially infinite search process for a "no", the returned value can keep changing for larger n, and thus the limit would not exist. Thus not every semi-decidable set satisfies all N_e, even classically, but the construction of Γ will, relying on the canonical semi-decider which for input x in step n checks whether $x \in \Gamma_n$.

For N_e to satisfy Γ, there has to be n such that $\forall n' > n.\ \varphi_e^\Gamma(e)[n'] = v$ for *the same* v. N_e might be finitely injured in the sense that $\varphi_e^\Gamma(e)[n] = v$, i.e., by definition of the semi-decider of Γ, $\varphi_e^{\Gamma_n}(e)[n] = v$, but then $\varphi_e^{\Gamma_{n+1}}(e)[n+1] = v'$, for $v \neq v'$. This can happen because an element was added to Γ_{n+1} to satisfy $P_{e'}$ for $e' < e$ which changes the computation. Since the requirements can be prioritised as $P_0 > N_0 > P_1 > N_1 \ldots$, any N_e will only be injured finitely often, meaning eventually the value v remains the same and the limit exists.

Note that if the limit exists, and $v = \mathsf{none}$, the computation $\varphi_e^A(e)$ diverges, and in the case $v = \ulcorner r \urcorner$, we have $\varphi_e^A(e) = r$. Thus, the existence of this limit implies constructively that $\varphi_e^A(e)$ either eventually terminates or diverges, but it is a strictly stronger condition.

There is of course no hope to be able to constructively prove that Γ satisfies the requirement N_e for all e in our formulation. This is because from the existence of the limit we could then show the Turing jump of Γ to be definite. Meta-theoretically we however know that a constructive proof of definiteness would allow a constructive proof of decidability. Thus, we know that the verification of the requirements cannot be fully constructive.

Our requirements N_e are classically equivalent to the requirements given by Soare [5, Theorem 4.1] [49, Theorem VII.1.1.]. We carefully change the definition of N_e so it can be proven with Σ_1-LEM while still implying lowness. Similar to the section before, we now prove the following fact constructively, and then later use Σ_1-LEM and the limit lemma to deduce lowness.

Fact 22. *If a semi-decidable set A satisfies all N_e, then A' is limit-computable.*

Proof. Define a function

$$f\langle e, n\rangle := \begin{cases} 1 & \text{if } \varphi_e^A(e)[n] = \ulcorner m \urcorner \\ 0 & \text{if } \varphi_e^A(e)[n] = \mathsf{none} \end{cases}$$

which is computable by the Church-Turing thesis via a code e_f. To show limit computability of the jump, we need to show

$$e \in A' \leftrightarrow \ \varphi_e^A(e) \downarrow \overset{*}{\leftrightarrow} (\exists m.\ \lim_{n\to\infty} \varphi_e^A(e)[n] = \ulcorner m \urcorner) \leftrightarrow \lim_{n\to\infty} f\langle e, n\rangle = 1$$

$$e \notin A' \leftrightarrow \neg\varphi_e^A(e) \downarrow \overset{*}{\leftrightarrow} \lim_{n\to\infty} \varphi_e^A(e)[n] = \mathsf{none} \qquad \leftrightarrow \lim_{n\to\infty} f\langle e, n\rangle = 0$$

where the equivalences marked by $*$ are immediate from the specification of the approximation function and the others are by definition. □

Construction: To verify all requirements N_e for Γ we have to ensure that $\lim_{n\to\infty} \varphi_e^\Gamma(e)[n]$ exists. By construction, this is equivalent to $\lim_{n\to\infty} \varphi_e^{\Gamma_n}(e)[n]$ existing. To ensure that, we want that questions asked in a computation $\varphi_e^{\Gamma_m}(e)[m]$ for $m > n$ do not flip an oracle answer asked to Γ_n from negative to positive, meaning while $\varphi_e^{\Gamma_n}(e)[n]$ might run out of fuel in its computation, questions x to the oracle will always be accurately answered. We can make sure this is the case by asking that elements x added to Γ_{n+1} are larger than all questions asked in the computation of $\varphi_{e'}^{\Gamma_n}(e')[n]$ for all $e' < e$.

We will do so by requiring that the wall function is larger than the so-called use of the computation. For this, we define total approximative *use functions* $u_e^A(x)[n]$ for any semi-decidable set A. It returns none if the computation of $\varphi_e^A(x)[n]$ does not terminate, and otherwise $\ulcorner m \urcorner$ where m is the maximum of questions asked to A in $\varphi_e^A(x)[n]$. Note that use functions also appear without the n in the literature, then they are partial functions.

Fact 23. *Let A be a semi-decidable set.*

If $u_e^A(x)[n] = \ulcorner m \urcorner$ and for all $m' \leq m$, the semi-decider for A on input m' returns the same result in n and $n' > n$ steps, then the use function and approximative step-index execution return the same result on n and n':

1) $u_e^A(x)[n'] = \ulcorner m \urcorner$
2) $\varphi_e^A(x)[n] = \ulcorner v \urcorner \to \varphi_e^A(x)[n'] = \ulcorner v \urcorner$

Furthermore,

3) *If $u_e^A(x)[n] = \ulcorner m \urcorner$ there exists v such that $\varphi_e^A(x)[n] = \ulcorner v \urcorner$*
4) $u_e^A(x)[n] = \mathsf{none} \leftrightarrow \varphi_e^A(x)[n] = \mathsf{none}$

Lemma 24. *Assuming Σ_1-LEM, if $\lim_{n\to\infty} u_e^A(x)[n] = \ulcorner m \urcorner$, then there exists v with $\lim_{n\to\infty} \varphi_e^A(x)[n] = \ulcorner v \urcorner$.*

Proof. Let the limit for $u_e^A(x)[n'] = \ulcorner m \urcorner$ for $n' > n_0$. Now for all numbers $m' < m$, by Σ_1-LEM we either have a number $s_{m'}$ in which the semi-decider for A halts on m' or it runs forever. We can build the maximum s over all these $s_{m'}$ and n_0.

Now we have that $u_e^A(x)[s] = \ulcorner m \urcorner$ by Fact 23 (1) and thus $\varphi_e^A(x)[s] = \ulcorner v \urcorner$ by Fact 23 (3).

By Fact 23 (2) we then also have $\varphi_e^A(x)[s'] = \ulcorner v \urcorner$ for all $s' > s$ since the computation only asks questions smaller than m, but they will be correctly approximated by the semi-decider by construction of s as the maximum over the necessary steps. □

Verification of Requirements: We now prove the analogon of Theorem 2. The proof is similar to Lemma 16. Recall that the return value of u is a natural number, where $\mathsf{none} = 0$ and $\ulcorner m \urcorner = m + 1$, and that there is no hope to constructively establish that all N_e are satisfied.

Theorem 3. *Assuming Σ_1-LEM, $u_e^\Gamma(x)[n]$ is convergent in n for all e and x if $\omega_n^{\Gamma_n}(e) \geq u_{e'}^\Gamma(e')[n]$ for all $e' < e$.*

Proof. We will assume that there exists n such that for all $e' < e$, $P_{e'}$ does not receive attention after stage n and prove this later in Corollary 26.

We then know that every element x added to Γ at stage $m > n$ is greater than the use function $u_e^\Gamma(e')[m]$ for all $e' < e$, since $\gamma^{\Gamma_m}(m)$ only adds elements greater than the wall function.

In order to show the convergence of $u_e^\Gamma(x)[n]$, we need to show that the limit $\lim_{n\to\infty} u_e^\Gamma(e)[n]$ exists. We fix the n from above. We use Σ_1-LEM to do a case analysis on $\exists m > n.\ \exists r.\ u_e^\Gamma(e)[m] = \ulcorner r \urcorner$. Note that this is Σ_1 because termination is and Σ_1 is closed under existential.

If no such $m > n$ exists, then $\lim_{n\to\infty} u_e^\Gamma(e)[n] = \mathsf{none}$. If such an m exists, we have r with $u_e^\Gamma(e)[n] = \ulcorner r \urcorner$. We show that $\lim_{n\to\infty} u_e^\Gamma(e)[n] = \ulcorner r \urcorner$ by proving $\forall m' > m.\ u_e^\Gamma(e)[m'] = \ulcorner r \urcorner$.

Let $m' > m$. Since for all elements x added to Γ after stage n we have $x > \ulcorner r \urcorner$ and $m' > m > n$, we also have $\Gamma_{m'}$ and Γ_n are identical for elements below r. Then we have $\varphi_e^\Gamma(e)[m']$ will not change at any later stage m by Fact 23 (2). Therefore, $u_e^\Gamma(x)[n]$ is convergent in n. □

Corollary 25. *Assuming Σ_1-LEM, Γ satisfies N_e if $\omega_n^\Gamma(e) \geq u_{e'}^\Gamma(e')[n]$ for all $e' < e$.*

Proof. According to Theorem 3, we have that $u_e^\Gamma(x)[n]$ is convergent in n. If $\lim_{n\to\infty} u_e^\Gamma(e)[n]$ is none, then $\lim_{n\to\infty} \varphi_e^\Gamma(e)[n] = \mathsf{none}$ by Fact 23 (4). When the use function converges to $\ulcorner r \urcorner$, there must be a v, such that the approximate computation $\varphi_e^\Gamma(e)[n]$ is convergent to $\ulcorner v \urcorner$ due to Lemma 24. □

Now the missing corollary can again be deduced from Lemma 17.

Corollary 26. *Assuming Σ_1-LEM, for any e, there exists n such that for all $e' < e$ and $m > n$, $P_{e'}$ does not receive attention at stage m.*

Proof. Direct from Lemma 17 because the pre-condition that it is definite whether there exists k such that P_e receives attention at stage k follows from Σ_1-LEM since given e and k it is decidable whether P_e receives attention at stage k. □

Finally, we show that the use function is classically convergent without assumptions, which is necessary to have a constructive proof of simpleness.

Lemma 27. *$u_e^\Gamma(x)[n]$ is classically convergent in n for all e and x if $\omega_n^\Gamma(e) \geq u_{e'}^\Gamma(e')[n]$ for all $e' < e$.*

Proof. One can generalise the proof of Theorem 3 to state that if (1) there exists n such that for all $e' < e$, $P_{e'}$ does not receive attention after stage n, and (2) $\exists m > n.\ \exists r.\ u_e^\Gamma(e)[m] = \ulcorner r \urcorner$ is definite, then the conclusion of the theorem holds. Then Theorem 3 follows from Corollary 26 and the fact that the case distinction is in Σ_1. This lemma follows from Corollary 18 and the fact that only one additional case distinction is necessary. □

Constructing a low simple set: Overall, we now have:

Lemma 28. *Whenever $\omega_n^{\Gamma_n}(e)$ is classically convergent, $\omega_n^{\Gamma_n}(e) \geq 2 \cdot e$, and $\omega_n^{\Gamma_n}(e) \geq u_{e'}^l(e')[n]$ for all $e' \leq e$, then Γ is simple, and that given Σ_1-LEM, Γ' is limit-computable.*

We now deduce lowness by constructing a wall function:

Corollary 29. *Γ is simple, and given Σ_1-LEM also low, for $\omega_n^l(e)$ set to the maximum of $2 \cdot e$ and $u_{e'}^l(e')[n]$ for $e' \leq e$.*

Proof. By the forward direction of the Limit Lemma (Lemma 12) lowness follows from Γ' being limit-computable.

We thus only need to prove that $\omega_n^{\Gamma_n}(e)$ is classically convergent in n. By Theorem 3, $u_{e'}^\Gamma(e')[n]$ is convergent in n, and thus $\omega_n^{\Gamma_n}(e)$ is because it is the maximum over convergent functions. □

VII. Post's Problem

We conclude by formally deducing that a low set solves Post's problem.

Theorem 4. *Assuming $(\neg\neg\Sigma_1)$-LEM, there is a solution to Post's problem.*

Proof. Γ is simple as discussed in the last section, and thus semi-decidable but undecidable. We just have to prove that it does not reduce from the halting problem. Assume it does to obtain a contradiction.

Now $(\neg\neg\Sigma_1)$-LEM implies $\neg\neg(\Sigma_1\text{-LEM})$, which because we want to prove a contradiction yields Σ_1-LEM, which by the last corollary yields that Γ is low and thus Γ does not reduce to the halting problem by the fact that Soare's jump theorem solves Post's problem (Theorem 1). □

VIII. Synthetic Computability

In this section we quickly recap synthetic computability in the Calculus of Inductive Constructions (CIC) as developed by Forster [31] and explain how we extend the synthetic framework for oracle computability by Forster, Mück, and Kirst [40] to be suitable for our setting.

In contrast to the rest of the paper, we commit to a foundation here: CIC. Being a type theory, the choice of CIC entails two central changes. First, we model sets A of natural numbers A as predicates $\mathbb{N} \to \mathbb{P}$. Switching from sets to predicates is largely inconsequential. The major difference is that predicates do not behave extensionally without further assumptions, but none of our result actually depends on extensional behaviour. Talking about Turing degrees, i.e. equivalence classes of sets, would become tedious, but we anyways do not do so in the technical part of the paper. Secondly, working in type theory allows us to give more precise type structure to functions. For instance, a decision function can be a function of type $X \to \mathbb{B}$, i.e. it takes as input an element of an arbitrary set X (not just a natural number) and produce as output either true or false (rather than a natural number which needs to be constrained to 1 or 0). Composite structure like list types X^* to model

finite sets and option types $X^?$ become native, for instance the type of γ becomes $\mathbb{N}^* \to \mathbb{N} \to \mathbb{N}^?$.

The basic idea of synthetic computability is to identify function spaces with computable function spaces. Already without introducing an axiom, one can give new definitions of underlying concepts, e.g. of decidability of a predicate $p : X \to \mathbb{P}$ as

$$\exists f : X \to \mathbb{B}.\ \forall x.\ px \leftrightarrow fx = \mathsf{true}.$$

Similarly simple, many-one reducibility of a predicate $p : X \to \mathbb{P}$ to $q : Y \to \mathbb{P}$ can be defined as

$$\exists f : X \to Y.\ \forall x.\ px \leftrightarrow q(fx).$$

In this setting, it is also natural to define Σ_1 predicates $p : X \to \mathbb{P}$ as

$$\exists f : X \to \mathbb{N} \to \mathbb{B}.\ \forall x.\ px \leftrightarrow \exists n.\ fxn = \mathsf{true}.$$

This is, in fact, a usual definition of Σ_1 in constructive (reverse) mathematics, even not considering synthetic computability: For instance LPO and MP are routinely stated like this [50], [51].

As discussed in the introduction, Σ_1-$\mathsf{LEM}_{\mathbb{B}}$ with this definition is strictly stronger than Σ_1-$\mathsf{LEM}_{\mathsf{PR}}$ with the definition we used before in the paper [15], and in turn both are individually stronger than $\neg\neg(\Sigma_1$-$\mathsf{LEM}_{\mathbb{B}})$ and $\neg\neg(\Sigma_1$-$\mathsf{LEM}_{\mathsf{PR}})$ respectively.

While $\neg\neg(\Sigma_1$-$\mathsf{LEM}_{\mathbb{B}})$ is strictly stronger than $(\neg\neg\Sigma_1)$-$\mathsf{LEM}_{\mathbb{B}}$ [13], in our setting with synthetic computability axioms they become equivalently again, analogous to how it was equivalent in the analytic setting.

The proof of both facts can be given agnostically with respect to concrete definitions of Σ_1:

Lemma 30. *Let $\mathcal{C} : (\mathbb{N} \to \mathbb{P}) \to \mathbb{P}$, i.e. $\mathcal{C}$ is a class of predicates on natural numbers. If there exists a predicate K such that*

1) *$\mathcal{C}(K)$ and*
2) *for all p with $\mathcal{C}(p)$, p many-one reduces to K,*

then $(\neg\neg\mathcal{C})$-LEM implies $\neg\neg(\mathcal{C}$-LEM$)$.

Proof. Assume (3) $(\neg\neg\mathcal{C})$-LEM and (4) $\neg(\mathcal{C}$-LEM$)$. We need to deduce a contradiction. We can instantiate (3) with K because of (1), i.e. know $\neg\neg\forall x.\ \mathsf{K}x \vee \neg\mathsf{K}x$. Because we need to prove a contradiction, we may even assume $\forall x.\ \mathsf{K}x \vee \neg\mathsf{K}x$ (5). Using (4), we need to prove $\mathcal{C}$-LEM.

Fix p with $\mathcal{C}(p)$ and use (2) to obtain a many-one reduction f with $\forall x.\ px \leftrightarrow \mathsf{K}(fx)$. We need to prove $\forall x.\ px \vee \neg px$, equivalent to $\forall x.\ \mathsf{K}(fx) \vee \neg\mathsf{K}(fx)$ and following from (5). □

Now K from the analytic section fulfills (1,2) for $\mathcal{C}$ being the analytic variant of Σ_1. Note that no analytic variant of many-one completeness is necessary, the lemma remains useful with just the synthetic version.

Turning to axioms for synthetic computability, we assume a universal function for the function space $\mathbb{N} \to \mathbb{N}$, i.e. morally identify functions with computable functions. This is particularly natural in type theory, where functions are programs anyways. The key axiom we use is the axiom EPF due to Richman [22], which talks about partial functions $\mathbb{N} \rightharpoonup \mathbb{N}$. Partial functions can be modeled in various forms in type theory, see [31, §4.5]. It is just crucial that termination, denoted by $fx\downarrow$ as before, is in the boolean version of Σ_1.

$$\mathsf{EPF} := \exists\varphi : \mathbb{N} \to \mathbb{N} \rightharpoonup \mathbb{N}.\ \forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists e.\ \forall x.\ \varphi_e(x) \equiv fx$$

This axiom is useful, but it requires an s-m-n like theorem on top to be powerful. Instead of such an additional assumption, we can deduce a seemingly stronger variant from EPF:

Fact 31. EPF *is equivalent to its following parametric form:*

$$\exists\varphi : \mathbb{N} \to \mathbb{N} \rightharpoonup \mathbb{N}.\ \forall f : \mathbb{N} \to \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists\gamma.\ \forall ix.\ \varphi_{\gamma i}(x) \equiv f_i x$$

We can define the self halting problem K as

$$\mathsf{K}e := \varphi_e(e)\downarrow$$

as before, but with φ from the last fact.

We can show that K is in the boolean version of Σ_1 and that K is many-one complete:

Lemma 32. *Let $p : \mathbb{N} \to \mathbb{P}$ be in Σ_1. Then p many-one reduces to K.*

Proof. Because p is in Σ_1 we have f with $\forall x.\ px \leftrightarrow \exists n. fxn = \mathsf{true}$. It is possible to define a partial function gxy which ignores y and searches for an n such that $fxn = \mathsf{true}$ and returns it. Now $\forall y.\ gxy \downarrow\leftrightarrow px$. Also g has a coding function γ w.r.t. φ via the last fact.

Now γ serves as many-one reduction, because $px \leftrightarrow gx(\gamma x)\downarrow\leftrightarrow \varphi_{\gamma x}(\gamma x)\downarrow\leftrightarrow \mathsf{K}(\gamma x)$ as claimed. □

We now recap the synthetic definition of oracle computability due to Forster, Kirst, and Mück [40].

A functional $F : (\mathbb{N} \to \mathbb{B} \to \mathbb{P}) \to \mathbb{N} \to \mathbb{B} \to \mathbb{P}$, mapping characteristic relations to characteristic relations, is considered oracle-computable if there is an underlying computation tree $\tau : \mathbb{N} \to \mathbb{B}^* \rightharpoonup \mathbb{N} + \mathbb{B}$ capturing the extensional behaviour of F by

$$\forall Rxb.\ FRxb \ \leftrightarrow\ \exists qs\ as.\ \tau x\,;R \vdash qs\,;as \wedge \tau\ x\ as = \mathsf{inr}\ b$$

where the *interrogation relation* $\sigma; R \vdash qs; as$ is inductively defined for $\sigma : \mathbb{B}^* \rightharpoonup \mathbb{N} + \mathbb{B}$ as

$$\frac{}{\sigma\,;R \vdash [\,]\,;[\,]} \qquad \frac{\sigma\,;R \vdash qs\,;as \qquad \sigma as = \mathsf{inl}\ q \qquad Rqa}{\sigma\,;R \vdash qs +\!\!+ [q]\,;as +\!\!+ [a]}$$

and where we write $\mathsf{inl}\ q$ and $\mathsf{inr}\ o$ for the respective injections into the sum type $\mathbb{N} + \mathbb{B}$.

Computation trees provide a notion of *sequential continuity* that singles out the functionals operating like oracle machines. The general definition yields a notion of Turing reductions $P \preceq_T Q$, by requiring an oracle computation F that maps Q to P, and a notion of relative semi-decidability $\mathcal{S}_Q(P)$, by requiring an oracle computation F that maps Q to the positive part of P.

Forster, Mück, and Kirst [41] use EPF to derive an enumeration for oracle computations, just as in the analytic setting, i.e. $\varphi_e^p(x)$ with an oracle p, code e, and input x.

For the present work, we need to give two novel constructions in their framework: approximative step-indexed execution and use functions, but otherwise can user their framework through specifications and closure properties and do not need to consider definitions of e.g. $\varphi_e^p(x)$.

Concretely, we define approximative step-indexed execution for any sequentially continuous computation (oracle machine) $\tau^f : A^* \rightharpoonup Q + O$, where f is a total function from Q to A. The execution of τ^f indexed by step i and depth j, denoted by $\tau^f[i,j] : (Q+O)^?$, is defined as

$$\tau^f[i,j] \;:=\; \begin{cases} \ulcorner \mathsf{out}\ (o) \urcorner & \text{if } (\tau[]) \downarrow_j \mathsf{out}\ (o) \\ \ulcorner \mathsf{ask}\ (q) \urcorner & \text{if } (\tau[]) \downarrow_j \mathsf{ask}\ (q) \\ & \text{and } \; i = 0 \\ \lambda r.\ \tau^f(fq :: r)[i', j] & \text{if } (\tau[]) \downarrow_j \mathsf{ask}\ (q) \\ & \text{and } \; i = i' + 1 \\ \mathsf{none} & \text{otherwise} \end{cases}$$

where $\tau \downarrow_j o$ denotes that the partial function terminates with output o within j steps.

For any semi-decidable oracle p, the step-indexed execution of φ_e^p is defined as

$$\varphi_e^p(x)[n] \;:=\; \varphi_e^{p_n}(x)[n,n]$$

where $p_n : Q \to \mathbb{B}$ returns true for input y if and only if $y \leq n$ and the semi-decider of p terminates in n steps on input y.

We omit the related detailed definition of the use function. Intuitively, similar to the step-indexed execution, the use function $u_e^p(x)[n]$ is indexed by n. If the step-indexed execution $\varphi_e^p(x)[n]$ returns an output o with an underlying computation τ, there exist answers list as questions list qs, such that

$$\tau\ x\ as \downarrow \mathsf{out}\ o \;\wedge\; \forall q.\ q \equiv_{qs} p_n \to \tau\ x; q \vdash as; qs$$

where $\equiv_{qs}$ denotes the equivalence relation up to the list qs. Then $u_e^p(x)[n]$ is defined as follows: if the computation returns an output, the use function returns $\ulcorner m \urcorner$ of the maximum m of the questions list qs mentioned above. Otherwise, it returns none. With this definition, we can show that the use function fulfills the properties in Fact 23.

With these definitions, one can give a fully synthetic rendering of all notions, theorems, and proofs in this paper. The translation is largely routine, taking care to synthetically express computable functions as functions, and potentially uncomputable functions as their graph relation. Ultimately, this gives rise to a fully formal account of a solution to Post's problem, serving as basis for feasible mechanisation.

IX. Rocq mechanisation

The Rocq mechanisation closely follows a synthetic rendering of the analytic proofs explained in this paper. At some points we take shortcuts and do not prove intermediate results in full strength, saving proof engineering time while keeping the final theorems at full strength.

In particular, in the Rocq development we only prove one direction of the specification of approximative step-indexed execution, i.e. that if the oracle machine terminates, the output corresponds to the limit of the step-indexed execution. For proving properties in the reverse direction as required by Lemma 13 and Fact 22, we inline a proof for the special cases when needed, which is simpler.

Overall, the Rocq development comprises around 2700 lines of code (25% spec, 75% proofs). It does not make use of advanced techniques in Rocq, apart from some custom tactics to make working in the double negation modality seamless.

X. Discussion

In future work, we want to tackle a constructive *reverse* analysis of Soare's jump theorem, of solutions to Post's theorem, and then also on results like Post's hierarchy theorem.

Further results that would be interesting to understand constructively, synthetically, and to mechanise, are for instance Sacks' jump theorem [52], a generalisation of Soare's jump theorem discussing the existence of sets between A and A' for more general A, or priority-free solutions to Post's problem, e.g. by Kucera [53].

For Soare's jump theorem, establishing the existence of a low simple set, our proof uses Σ_1-LEM. Reusing the analogy that axioms in constructive mathematics play the role of oracles in relative computability, this means that we need a logical oracle to determine membership in the halting problem K.

That for the solution to Post's problem via Soare's jump theorem actually a slightly exotic principle in the form of $(\neg\neg\Sigma_1)$-LEM suffices can be considered as surprising. The mechanised Rocq development however made it trivial to check that only a negative result depends on Σ_1-LEM, because Rocq (as other proof assistants with support for constructive mathematics such as Agda) tracks dependencies on axioms automatically. The Rocq development in turn was only possible due to an underlying formalisation in synthetic computability, which crucially had to support assuming (sub-)classical axioms consistently.

Acknowledgements

Dominik Kirst received funding from the European Union's Horizon research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101152583.

Takako Nemoto's research was supported by the Japan Society for the Promotion of Science (JSPS), specifically JSPS KAKENHI Grant Numbers JP21KK0045 and JP24K06823.

References

[1] A. M. Turing, "Systems of logic based on ordinals," *Proceedings of the London mathematical society*, vol. 2, no. 1, pp. 161–228, 1939.

[2] E. L. Post, "Recursively enumerable sets of positive integers and their decision problems," *Bulletin of the American Mathematical Society*, vol. 50, no. 5, pp. 284–316, 1944.

[3] R. M. Friedberg, "Two recursively enumerable sets of incomparable degrees of unsovlability (solution of Post's problem), 1944," *Proceedings of the National Academy of Sciences*, vol. 43, no. 2, pp. 236–238, Feb. 1957. [Online]. Available: https://doi.org/10.1073/pnas.43.2.236

[4] A. A. Muchnik, "On strong and weak reducibility of algorithmic problems," *Sibirskii Matematicheskii Zhurnal*, vol. 4, no. 6, pp. 1328–1341, 1963. [Online]. Available: doi.org/10.3233/COM-150042

[5] R. I. Soare, "The infinite injury priority method," *The Journal of Symbolic Logic*, vol. 41, no. 2, pp. 513–530, 1976. [Online]. Available: https://doi.org/10.2307/2272252

[6] J. R. Shoenfield, "On degrees of unsolvability," *Annals of mathematics*, vol. 69, no. 3, pp. 644–653, 1959. [Online]. Available: https://doi.org/10.2307/1970028

[7] E. M. Gold, "Limiting recursion," *The Journal of Symbolic Logic*, vol. 30, no. 1, pp. 28–48, 1965. [Online]. Available: https://doi.org/10.2307/2270580

[8] S. C. Kleene, "On the interpretation of intuitionistic number theory," *The journal of symbolic logic*, vol. 10, no. 4, pp. 109–124, 1945.

[9] J. Van Oosten, *Realizability: an introduction to its categorical side*. Elsevier, 2008.

[10] A. S. Troelstra and D. van Dalen, "Constructivism in mathematics. vol. i," *Studies in Logic and the Foundations of Mathematics*, vol. 26, 1988.

[11] H. Ishihara, "Reverse mathematics in Bishop's constructive mathematics," *Philosophia Scientae*, no. CS 6, pp. 43–59, Sep. 2006. [Online]. Available: https://doi.org/10.4000/philosophiascientiae.406

[12] H. Diener, "Constructive Reverse Mathematics," *arXiv:1804.05495 [math]*, 2020. [Online]. Available: https://arxiv.org/abs/1804.05495

[13] M. Fujiwara and U. Kohlenbach, "Interrelation between weak fragments of double negation shift and related principles," *The Journal of Symbolic Logic*, vol. 83, no. 3, pp. 991–1012, 2018. [Online]. Available: https://doi.org/10.1017/jsl.2017.63

[14] L. Cohen, Y. Forster, D. Kirst, B. Da Rocha Paiva, and V. Rahli, "Separating markov's principles," in *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, 2024, pp. 1–14. [Online]. Available: https://doi.org/10.1145/3661814.3662104

[15] B. da Rocha Paiva, L. Cohen, Y. Forster, D. Kirst, and V. Rahli, "Limited principles of omniscience in constructive type theory," in *30th International Conference on Types for Proofs and Programs TYPES 2024–Abstracts*, 2024, p. 23. [Online]. Available: https://types2024.itu.dk/abstracts.pdf

[16] M. Norrish, "Mechanised computability theory," in *ITP 2011*, ser. LNCS, vol. 6898. Springer, 2011, pp. 297–311. [Online]. Available: https://doi.org/10.1007/978-3-642-22863-6_22

[17] J. Xu, X. Zhang, and C. Urban, "Mechanising Turing machines and computability theory in Isabelle/HOL," in *International Conference on Interactive Theorem Proving*. Springer, 2013, pp. 147–162. [Online]. Available: https://doi.org/10.1007/978-3-642-39634-2_13

[18] Y. Forster and G. Smolka, "Weak call-by-value lambda calculus as a model of computation in coq," in *Interactive Theorem Proving - 8th International Conference, ITP 2017, Brasília, Brazil, September 26-29, 2017, Proceedings*, ser. Lecture Notes in Computer Science, M. Ayala-Rincón and C. A. Muñoz, Eds., vol. 10499. Springer, 2017, pp. 189–206. [Online]. Available: https://doi.org/10.1007/978-3-319-66107-0_13

[19] Y. Forster, F. Kunze, and M. Wuttke, "Verified programming of turing machines in coq," in *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*. ACM, Jan. 2020. [Online]. Available: https://doi.org/10.1145/3372885.3373816

[20] Y. Forster, F. Kunze, G. Smolka, and M. Wuttke, "A mechanised proof of the time invariance thesis for the weak call-by-value λ-calculus," vol. 193, no. 29, 2021. [Online]. Available: https://doi.org/10.4230/LIPIcs.ITP.2021.19

[21] M. Carneiro, "Formalizing Computability Theory via Partial Recursive Functions," in *10th International Conference on Interactive Theorem Proving (ITP 2019)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), J. Harrison, J. O'Leary, and A. Tolmach, Eds., vol. 141. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 12:1–12:17. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/11067

[22] F. Richman, "Church's thesis without tears," *The Journal of symbolic logic*, vol. 48, no. 3, pp. 797–803, 1983. [Online]. Available: https://doi.org/10.2307/2273473

[23] D. Bridges and F. Richman, *Varieties of constructive mathematics*. Cambridge University Press, 1987, vol. 97.

[24] A. Bauer, "First steps in synthetic computability theory," *Electronic Notes in Theoretical Computer Science*, vol. 155, pp. 5–31, 2006. [Online]. Available: https://10.1016/j.entcs.2005.11.049

[25] ——, "On fixed-point theorems in synthetic computability," *Tbilisi Mathematical Journal*, vol. 10, no. 3, pp. 167–181, 2017.

[26] G. Kreisel, "Mathematical logic," *Lectures in modern mathematics*, vol. 3, pp. 95–195, 1965.

[27] A. Swan and T. Uemura, "On Church's thesis in cubical assemblies," *arXiv preprint arXiv:1905.03014*, 2019. [Online]. Available: https://arxiv.org/abs/1905.03014

[28] P. Pédrot, ""upon this quote I will build my church thesis"," in *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, P. Sobocinski, U. D. Lago, and J. Esparza, Eds. ACM, 2024, pp. 65:1–65:12. [Online]. Available: https://doi.org/10.1145/3661814.3662070

[29] J. M. E. Hyland, "The effective topos," in *The L. E. J. Brouwer Centenary Symposium, Proceedings of the Conference held in Noordwijkerhout*. Elsevier, 1982, pp. 165–216. [Online]. Available: https://doi.org/10.1016/s0049-237x(09)70129-6

[30] A. W. Swan, "Oracle modalities," 2024. [Online]. Available: https://arxiv.org/abs/2406.05818

[31] Y. Forster, "Computability in constructive type theory," Ph.D. dissertation, Saarland University, 2021. [Online]. Available: https://doi.org/10.22028/D291-35758

[32] T. Coquand, "Metamathematical investigations of a calculus of constructions," INRIA, Tech. Rep. RR-1088, 1989. [Online]. Available: https://hal.inria.fr/inria-00075471

[33] T. Coquand and G. P. Huet, "The calculus of constructions," *Information and Computation*, vol. 76, no. 2/3, pp. 95–120, 1988. [Online]. Available: https://doi.org/10.1016/0890-5401(88)90005-3

[34] C. Paulin-Mohring, "Inductive definitions in the system Coq rules and properties," in *International Conference on Typed Lambda Calculi and Applications*. Springer, 1993, pp. 328–345.

[35] The Coq Development Team, "The Coq proof assistant," Jun. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8161141

[36] Y. Forster, "Parametric church's thesis: Synthetic computability without choice," in *Lecture Notes in Computer Science*, 2022, pp. 70–89. [Online]. Available: https://doi.org/10.1007/978-3-030-93100-1_6

[37] Y. Forster, D. Kirst, and G. 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 - CPP 2019*. ACM Press, 2019. [Online]. Available: https://doi.org/10.1145/3293880.3294091

[38] Y. Forster and F. Jahn, "Constructive and Synthetic Reducibility Degrees: Post's Problem for Many-One and Truth-Table Reducibility in Coq," in *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), B. Klin and E. Pimentel, Eds., vol. 252. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, pp. 21:1–21:21. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2023/17482

[39] Y. Forster, F. Kunze, and N. Lauermann, "Synthetic Kolmogorov Complexity in Coq," in *13th International Conference on Interactive Theorem Proving (ITP 2022)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), J. Andronick and L. de Moura, Eds., vol. 237. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, pp. 12:1–12:19. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2022/16721

[40] Y. Forster, D. Kirst, and N. Mück, "Oracle computability and turing reducibility in the calculus of inductive constructions," in *Programming Languages and Systems - 21st Asian Symposium, APLAS 2023, Taipei, Taiwan, November 26-29, 2023, Proceedings*, ser. Lecture Notes in Computer Science, C. Hur, Ed., vol. 14405. Springer, 2023, pp. 155–181. [Online]. Available: https://doi.org/10.1007/978-981-99-8311-7_8

[41] ——, "The Kleene-Post and Post's theorem in the Calculus of Inductive Constructions," in *32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19-23, 2024, Naples, Italy*, ser. LIPIcs, A. Murano and A. Silva, Eds., vol. 288. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 29:1–29:20. [Online]. Available: https://doi.org/10.4230/LIPIcs.CSL.2024.29

[42] H. Rogers, "Theory of recursive functions and effective computability," 1987.

[43] P. Odifreddi, *Classical recursion theory: The theory of functions and sets of natural numbers*. Elsevier, 1992.

[44] A. A. Markov, "The theory of algorithms," *Trudy Matematicheskogo Instituta Imeni VA Steklova*, vol. 42, pp. 3–375, 1954.

[45] Y. Akama, S. Berardi, S. Hayashi, and U. Kohlenbach, "An arithmetical hierarchy of the law of excluded middle and related principles," in *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*. IEEE Computer Society, 2004, pp. 192–201. [Online]. Available: https://doi.org/10.1109/LICS.2004.1319613

[46] D. Ilik, "An interpretation of the sigma-2 fragment of classical analysis in system t," 2015. [Online]. Available: https://arxiv.org/abs/1301.5089

[47] U. Berger and P. Oliva, "Modified bar recursion," *Mathematical Structures in Computer Science*, vol. 16, no. 2, p. 163, 2006. [Online]. Available: https://doi.org/10.1017/S0960129506005093

[48] T. Nemoto, "Finite sets and infinite sets in weak intuitionistic arithmetic," *Archive for Mathematical Logic*, vol. 59, no. 5-6, pp. 607–657, 2020. [Online]. Available: https://doi.org/10.1007/s00153-019-00704-8

[49] R. I. Soare, *Recursively enumerable sets and degrees: A study of computable functions and computably generated sets*. Springer Science & Business Media, 1999.

[50] Y. Forster, "Church's Thesis and Related Axioms in Coq's Type Theory," in *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), C. Baier and J. Goubault-Larrecq, Eds., vol. 183. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, pp. 21:1–21:19. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2021/13455

[51] T. Coquand and B. Mannaa, "The independence of markov's principle in type theory," *Log. Methods Comput. Sci.*, vol. 13, no. 3, 2017. [Online]. Available: https://doi.org/10.23638/LMCS-13(3:10)2017

[52] G. E. Sacks, "Recursive enumerability and the jump operator," in *Journal of Symbolic Logic 29 (4):204-204*, 1964. [Online]. Available: https://doi.org/10.1142/9789812812926_0002

[53] A. Kučera, "An alternative, priority-free, solution to Post's problem," in *Lecture Notes in Computer Science*, 1986, pp. 493–500. [Online]. Available: https://doi.org/10.1007/BFb0016275