On Synthetic Undecidability in Coq, with an Application to the Entscheidungsproblem

Yannick Forster, Dominik Kirst, Gert Smolka

CPP 2019
January 15

How to formalise decidability?

Classical approach:

- Pick a concrete model of computation
  (Turing machines, μ-recursive functions, untyped λ-calculus, etc.)
- Invent a decision procedure for the given problem
- Explicitly code the algorithm in the chosen model!

Synthetic approach:

- Work in a constructive type theory
- Define a decision procedure e.g. as a Boolean function
- Definable functions are computable, so that's it!

(Similar for other notions like enumerability and reducibility)

How to formalise undecidability?

Problem of the synthetic approach:

- Constructive type theories like MLTT or CIC are consistent with the assumption that every problem is decidable
- Proving a given problem undecidable is not outright possible

Possible solutions:

- Resort to a concrete model of computation
- Verify a synthetic reduction from an undecidable problem
- Deduce that the undecidability of the base problem implies the undecidability of the considered problem

(Again similar for other notions of computability theory)

Related Work

Andrej Bauer.

First steps in synthetic computability theory.

Electronic Notes in Theoretical Computer Science, 155:5–31, 2006.

Yannick Forster, Edith Heiter, and Gert Smolka.

Verification of PCP-Related Computational Reductions in Coq.

In *International Conference on Interactive Theorem Proving*, 2018.

Zohar Manna.

Mathematical Theory of Computation.

McGraw-Hill computer science series. McGraw-Hill, 1974.

Alan M. Turing.

On computable numbers, with an application to the entscheidungsproblem.

Proceedings of the London mathematical society, 2(1):230–265, 1937.

Emil L. Post.

A variant of a recursively unsolvable problem.

Bulletin of the American Mathematical Society, 52(4):264–268, 1946.

Elementary Synthetic Computability Theory

Decidability and Enumerability

A problem interpreted as a predicate $p : X \to \mathbb{P}$ on a type X is decidable if there is a function $f : X \to \mathbb{B}$ with

$$\forall x.\, p\, x \leftrightarrow f\, x = \mathrm{tt},$$

enumerable if there is a function $f : \mathbb{N} \to \mathcal{O}(X)$ with

$$\forall x.\, p\, x \leftrightarrow \exists n.\, f\, n = \ulcorner x \urcorner.$$

Fact

Let $p : X \to \mathbb{P}$ be a predicate, then p is

- *decidable iff $\forall x.\, p\, x + \neg(p\, x)$ is inhabited and*
- *enumerable iff there is $L : \mathbb{N} \to \mathcal{L}(X)$ s.t. $\forall x.\, p\, x \leftrightarrow \exists n.\, x \in L\, n$.*

Data Types

Computability theory is usually developed for computational domains.

A type X is called

- enumerable if $\lambda x.\, \top$ is enumerable,
- discrete if $\lambda xy.\, x = y$ is decidable, and
- data type if it is both enumerable and discrete.

Fact

Decidable predicates on data types are enumerable and co-enumerable.

Proof.

Let $f_X : \mathbb{N} \to \mathcal{O}(X)$ enumerate X and $f_p : X \to \mathbb{B}$ decide p. Then

$$f\ n := \text{match } f_X\ n \text{ with } \ulcorner x \urcorner \Rightarrow \text{if } f_p\, x \text{ then } \ulcorner x \urcorner \text{ else } \emptyset \mid \emptyset \Rightarrow \emptyset$$

defines an enumerator for p. □

Many-One Reductions

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 from p to q if

$$\forall x.\, p\, x \leftrightarrow q\, (f\, x).$$

We write $p \preccurlyeq q$ if a reduction from p to q exists.

Theorem (Reduction)

Let p and q be predicates on data types with $p \preccurlyeq q$.

- *If q is decidable/enumerable/co-enumerable, then so is p.*
- *If p is not co-enumerable, then q is not co-enumerable.*

Proof.

If f witnesses $p \preccurlyeq q$ and g decides q, then $g \circ f$ decides p. □

Post's Theorem and Markov's Principle

Post: Bi-enumerable predicates on data types are decidable.

Not directly provable, needs extra assumption:

Lemma

Post *holds for logically decidable predicates.*

Proof.

By (guarded) unbounded linear search and parallel enumeration. □

Markov: $\forall f : \mathbb{N} \to \mathbb{B}. \neg\neg(\exists n.\, f\ n = \mathrm{tt}) \to \exists n.\, f\ n = \mathrm{tt}$

Theorem

Markov *is equivalent to the logical decidability of bi-enumerable predicates on discrete types. So in particular,* Markov *is equivalent to* Post.

The Post Correspondence Problem

Recap: given a stack S of cards s/t, find a derivable match.

This (undecidable) problem can be expressed by an inductive predicate:

$$\frac{s/t \in S}{S \rhd s/t} \qquad \frac{S \rhd u/v \quad s/t \in S}{S \rhd su/tv} \qquad \frac{S \rhd s/s}{\mathrm{PCP}\ S}$$

Fact

The type $\mathbb{S}$ of stacks is a data type and PCP *is enumerable.*

Proof.

The former follows from closure properties and for the latter

$$\begin{aligned} L\,0 &:= [] \\ L\,(\mathrm{S}\,n) &:= L\,n +\!\!+ [(S,(s,t)) \mid S \in L_{\mathbb{S}}\,n, (s,t) \in S] \\ &\quad +\!\!+ [(S,(su,tv)) \mid (S,(u,v)) \in L\,n, (s,t) \in S] \end{aligned}$$

defines a list enumerator for $\lambda Sst.\, S \rhd s/t$. □

Undecidability of First-Order Logic

Syntax and Tarski Semantics

Terms and formulas are defined for a fixed signature:

$$\tau : \mathcal{T} := x \mid a \mid e \mid g_{\mathrm{tt}}\,\tau \mid g_{\mathrm{ff}}\,\tau \qquad x, a : \mathbb{N}$$
$$\varphi, \psi : \mathcal{F} := \dot{\bot} \mid Q \mid P\,\tau_1\,\tau_2 \mid \varphi \dot{\to} \psi \mid \dot{\forall} x.\,\varphi$$

Formulas are interpreted in models $\mathcal{I} = (D, \eta, e^{\mathcal{I}}, g_{\mathrm{tt}}^{\mathcal{I}}, g_{\mathrm{ff}}^{\mathcal{I}}, Q^{\mathcal{I}}, P^{\mathcal{I}})$ given a variable environment $\rho : \mathbb{N} \to D$:

$$\rho \models_{\mathcal{I}} \dot{\bot} := \bot$$
$$\rho \models_{\mathcal{I}} Q := Q^{\mathcal{I}}$$
$$\rho \models_{\mathcal{I}} P\,\tau_1\,\tau_2 := P^{\mathcal{I}}\,(\hat{\rho}\,\tau_1)\,(\hat{\rho}\,\tau_2)$$
$$\rho \models_{\mathcal{I}} \varphi \dot{\to} \psi := \rho \models_{\mathcal{I}} \varphi \to \rho \models_{\mathcal{I}} \psi$$
$$\rho \models_{\mathcal{I}} \dot{\forall} x.\,\varphi := \forall d : D.\,\rho[x := d] \models_{\mathcal{I}} \varphi$$

A formula φ is valid if $\rho \models_{\mathcal{I}} \varphi$ for all $\mathcal{I}$ and ρ.

A Standard Model

Strings can be encoded as terms, e.g. $\overline{\mathsf{tt}\,\mathsf{ff}\,\mathsf{ff}\,\mathsf{tt}} = g_{\mathsf{tt}}\,(g_{\mathsf{ff}}\,(g_{\mathsf{ff}}\,(g_{\mathsf{tt}}\,e)))$.

The standard model $\mathcal{B}$ over the type $\mathcal{L}(\mathbb{B})$ of Boolean strings captures exactly the cards derivable from a fixed stack S:

$$e^{\mathcal{B}} := [] \qquad Q^{\mathcal{B}} := \mathsf{PCP}\,S$$
$$g_b^{\mathcal{B}}\,s := b :: s \qquad P^{\mathcal{B}}\,s\,t := S \triangleright s/t.$$

Lemma

Let $\rho : \mathbb{N} \to \mathcal{L}(\mathbb{B})$ be an environment for the standard model $\mathcal{B}$.
Then $\hat{\rho}\,\overline{s} = s$ and $\rho \models_{\mathcal{B}} P\,\tau_1\,\tau_2 \leftrightarrow S \triangleright \hat{\rho}\,\tau_1/\hat{\rho}\,\tau_2$.

Fact

Markov $\to \neg\neg\rho \models_{\mathcal{B}} \varphi \to \rho \models_{\mathcal{B}} \varphi$

Undecidability of Validity

We express the constructors of $S \triangleright s/t$ and PCP as formulas:

$$\varphi_1 := [P\,\overline{s}\,\overline{t} \mid s/t \in S]$$
$$\varphi_2 := [\dot{\forall} xy.\, P\,x\,y \dot{\to} P\,(\overline{s}x)\,(\overline{t}y) \mid s/t \in S]$$
$$\varphi_3 := \dot{\forall} x.\, P\,x\,x \dot{\to} Q$$
$$\varphi_S := \varphi_1 \dot{\to} \varphi_2 \dot{\to} \varphi_3 \dot{\to} Q$$

Theorem

PCP S *iff* φ_S *is valid.*

Proof.

Let φ_S be valid, so in particular $\mathcal{B} \models \varphi_S$. Since $\mathcal{B}$ satisfies all of φ_1, φ_2, and φ_3 it follows that $\mathcal{B} \models Q$ and thus PCP S.
Now suppose that $S \triangleright s/s$ for some s and that some model $\mathcal{I}$ satisfies all of φ_1, φ_2, and φ_3. Then $\mathcal{I} \models P\,\overline{s}\,\overline{s}$ by φ_1 and φ_2, hence $\mathcal{I} \models Q$ by φ_3, and thus $\mathcal{I} \models \varphi_S$. □

Undecidability of Minimal Provability

We define a minimal natural deduction system inductively:

$$\frac{\varphi \in A}{A \vdash \varphi}\ A \qquad \frac{\varphi :: A \vdash \psi}{A \vdash \varphi \dot{\to} \psi}\ II \qquad \frac{A \vdash \varphi \dot{\to} \psi \quad A \vdash \varphi}{A \vdash \psi}\ IE$$

$$\frac{A \vdash \varphi^x_a \quad a \notin \mathcal{P}(\varphi) \cup \mathcal{P}(A)}{A \vdash \dot{\forall} x.\, \varphi}\ AI \qquad \frac{A \vdash \dot{\forall} x.\, \varphi \quad \mathcal{V}(\tau) = \emptyset}{A \vdash \varphi^x_\tau}\ AE$$

A formula φ is provable if $\vdash \varphi$.

Fact (Soundness)

$A \vdash \varphi$ implies $A \models \varphi$, so provable formulas are valid.

Theorem

- PCP S *iff* φ_S *is provable.* *(as before using soundness)*
- *Provability is enumerable.* *(by giving a list enumerator)*

Undecidability of Classical Provability

We extend the deduction system by a classical rule for falsity:

$$\frac{A \vdash_C \dot{\neg}\dot{\neg}\varphi}{A \vdash_C \varphi}\ DN$$

Unfortunately, this rule is not sound constructively!

As a remedy, we define a Gödel-Gentzen-Friedman translation φ^Q of formulas φ such that $A \vdash_C \varphi$ implies $A^Q \vdash \varphi^Q$.

Theorem

PCP S *iff* φ_S *is classically provable.*

Proof.

If PCP S then $\vdash \varphi_S$ by the previous theorem and hence $\vdash_C \varphi_S$. Conversely, let $\vdash_C \varphi_S$ and hence $\vdash \varphi_S^Q$. Then by soundness $\mathcal{B} \models \varphi_S^Q$ which implies $\mathcal{B} \models Q$ and PCP S as before. □

Wrap-Up

Coq Formalisation

Structure:

- Elementary synthetic computability (900 loc)
- Metatheory of first-order logic (750 loc)
- Undecidability of first-order logic (550 loc)

Features:

- Tagged inductive types and predicates representing syntax and deduction systems to avoid code duplication
- Convenient method to define enumerators for inductive types and predicates via cumulative lists
- Type class inference for automated decidability proofs

`www.ps.uni-saarland.de/extras/fol-undec`

Future Work

- Constructive metatheory of first-order logic (completeness, syntax representations, ...)
- Realisability model of the calculus of inductive constructions witnessing (the propositional version) of excluded middle
- Automated translation of Coq function definitions into a concrete model of computation (e.g. call-by-value lambda calculus)
- Library of formalised undecidability results
  - Particular axiomatic theories (PA, ZF, ...)
  - Diophantine equations (Hilbert's 10th problem)
  - Higher-order unification
  - ...

`www.github.com/uds-psl/coq-library-undecidability`

Backup

Development Details

File	Spec	Proof
Prelim.v	402	456
DecidableEnumerable.v	124	257
Reductions.v	31	38
MarkovPost.v	38	82
PCP.v	42	50
Infinite.v	103	134
FOL.v	49	78
Semantics.v	107	21
Deduction.v	124	165
Kripke.v	89	142
Weakening.v	61	83
BPCP_FOL.v	151	163
BPCP_IFOL.v	58	16
BPCP_CND.v	65	93
Total	1461	1788

Undecidability of Satisfiability

Theorem

$\neg(\mathrm{PCP}\, S)$ *iff* $\dot{\neg}\varphi_S$ *is satisfiable.*

Proof.

Suppose that $\neg(\mathrm{PCP}\, S)$. We show that $\mathcal{B} \models \dot{\neg}\varphi_S$, so let $\mathcal{B} \models \varphi_S$. As before this implies that $\mathrm{PCP}\, S$, contradiction.
Now suppose that $\mathcal{I} \models \dot{\neg}\varphi_S$ and that $\mathrm{PCP}\, S$. The latter implies that φ_S is valid, contradicting the former. □

Kripke Semantics

Definition

Kripke models $\mathcal{M}$ consists of a domain D, an assignment $\eta : \mathbb{N} \to D$, and

- A preorder $(W, \leq)$ called accessibility relation,
- A function $\mathcal{W}$ mapping nodes $w : W$ to interpretations over D and η with $\mathcal{W}\, w \hookrightarrow \mathcal{W}\, w'$ whenever $w \leq w'$.

Definition

We define the forcing relation $\rho, w \Vdash_{\mathcal{M}} \varphi$ on Kripke models by

$$\begin{aligned}
\rho, w \Vdash_{\mathcal{M}} \bot &:= \bot \\
\rho, w \Vdash_{\mathcal{M}} Q &:= Q^w \\
\rho, w \Vdash_{\mathcal{M}} P\, \tau_1\, \tau_2 &:= P^w\, (\hat{\rho}\, \tau_1)\, (\hat{\rho}\, \tau_2) \\
\rho, w \Vdash_{\mathcal{M}} \varphi \dot{\to} \psi &:= \forall w'.\, w \leq w' \to \rho, w' \Vdash_{\mathcal{M}} \varphi \to \rho, w' \Vdash_{\mathcal{M}} \psi \\
\rho, w \Vdash_{\mathcal{M}} \dot{\forall} x.\varphi &:= \forall w'.\, w \leq w' \to \forall d.\, \rho[x := d], w' \Vdash_{\mathcal{M}} \varphi
\end{aligned}$$