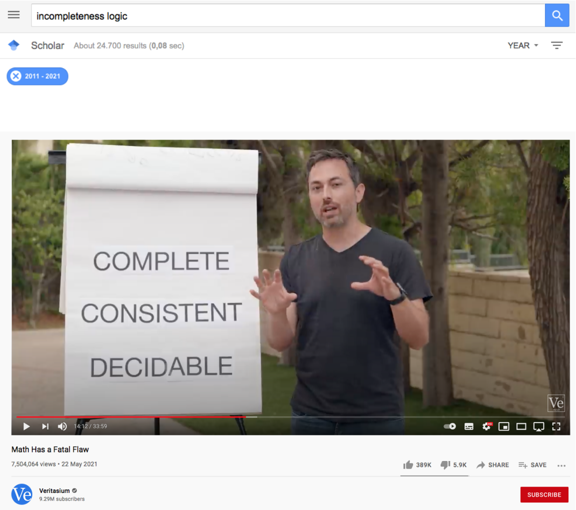
Synthetic Undecidability and Incompleteness of First-Order Axiom Systems in Coq

Dominik Kirst and Marc Hermes

Interactive Theorem Proving
June 30th, 2021

SAARLAND UNIVERSITY
COMPUTER SCIENCE
SIC Saarland Informatics Campus

Why Revisit Undecidability and Incompleteness?



- Still fascinates broad audience in and outside of science
- Prominent benchmark for interactive theorem proving
- Showcases synthetic approach to computability theory

From Undecidability of First-Order Logic...

Decision problems on first-order formulas φ:

- Is φ provable in a deduction system ($\vdash \varphi$)?
- Is φ valid in model-theoretic semantics ($\vDash \varphi$)?
- Is φ satisfiable by some model $\mathcal{M} \vDash \varphi$?
- Is φ satisfiable by a finite model $\mathcal{M} \vDash \varphi$?

All of them are undecidable (for non-trivial signatures):

- Classical papers: Turing (1937), Church (1936), Trakhtenbrot (1950)
- Coq mechanisations: Forster, K., and Smolka (2019), K. and Larchey-Wendling (2020)

...to Undecidability of First-Order Axiom Systems

Decision problems relativised to an axiomatisation $\mathcal{A}$:

- Is φ derivable from $\mathcal{A}$, i.e. $\mathcal{A} \vdash \varphi$?
- Is φ semantically entailed by $\mathcal{A}$, i.e. $\mathcal{A} \vDash \varphi$?

Call $\mathcal{A}$ (un)decidable if these problems are (un)decidable.

- Connected to the general decision problems
- Some are decidable: Presburger arithmetic, Boolean algebras, real closed fields etc.
- Some are undecidable: Peano arithmetic, ZF set theory, etc.
- Several mechanisations of decidability, none of undecidability (of PA/ZF)

Links to Consistency and Incompleteness

By contraposition of two facts:

Fact

Inconsistent axiomatisations ($\mathcal{A} \vdash \bot$) are decidable.

- Mechanising undecidability is at least as hard as mechanising consistency
- Our strategy is to work with standard models anyway

Fact

(Negation-)complete axiomatisations (for all closed φ either $\mathcal{A} \vdash \varphi$ or $\mathcal{A} \vdash \neg\varphi$) are decidable.

- Mechanising undecidability is at least as hard as mechanising incompleteness
- Disclaimer: no construction of an independent Gödel/Rosser sentence

Previous Mechanisations of Incompleteness

- Shankar (1986)
  - First full mechanisation of Gödel's 1st (G1) in the Boyer-Moore theorem prover
- O'Connor (2005)
  - Constructive mechanisation of G1 in Coq
- Paulson (2015)
  - Mechanisation of G1 and G2 in Isabelle/HOL
- Popescu and Traytel (2019)
  - Abstract preconditions for G1 and G2 in Isabelle/HOL

None of them approach incompleteness via undecidability.

Plan of the Talk

1. Framework:
   Synthetic undecidability and incompleteness

2. Case studies:
   Arithmetic (PA/HA) and set theory (ZF/IZF)

3. Conclusion:
   Coq mechanisation and future directions

Framework

Synthetic Undecidability (Forster, K., and Smolka (2019))

Every function definable in constructive type theory is computable.

A predicate/decision problem $p : X \to \mathbb{P}$...

- is decidable: $\exists f : X \to \mathbb{B}.\, \forall x.\, p\,x \leftrightarrow f\,x = \mathrm{tt}$
- is enumerable: $\exists g : \mathbb{N} \to X_\perp.\, \forall x.\, p\,x \leftrightarrow \exists n.\, g\,n = x$
- is reducible to $q : Y \to \mathbb{P}$: $\exists h : X \to Y.\, \forall x.\, p\,x \leftrightarrow q\,(h\,x)$

$\Rightarrow$ No need to encode f, g, and h as Turing machines!

Definition

A predicate p is undecidable if decidability of p implies *falsity*.**decidability of HALT**.

Lemma

A predicate p is undecidable if there is a reduction HALT $\preceq p$.

First-Order Axiom Systems (e.g. K. and Larchey-Wendling (2020))

Given a signature $\Sigma = (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$, we represent terms and formulas inductively by:

$$t : \mathsf{Term}_\Sigma \ ::= \ x \mid f\,\vec{t} \qquad\qquad (x : \mathbb{N},\ f : \mathcal{F}_\Sigma,\ \vec{t} : \mathsf{Term}_\Sigma^{|f|})$$

$$\varphi, \psi : \mathsf{Form}_\Sigma \ ::= \ \bot \mid P\,\vec{t} \mid \varphi \,\Box\, \psi \mid \nabla \varphi \qquad\qquad (P : \mathcal{P}_\Sigma,\ \vec{t} : \mathsf{Term}_\Sigma^{|P|})$$

- Interpretation ($\vDash$) in models $\mathcal{M} = (D, \forall f : \mathcal{F}_\Sigma.\, D^{|f|} \to D, \forall P : \mathcal{P}_\Sigma.\, D^{|P|} \to \mathbb{P})$
  - Map all connectives $\Box$ and quantifiers ∇ to their (constructive) counterparts in $\mathbb{P}$
- Provability ($\vdash$) characterised by intuitionistic ($\vdash_i$) and classical ($\vdash_c$) deduction systems
  - Soundness of $\vdash_i$ constructive, soundness of $\vdash_c$ requires excluded middle (LEM)

Definition

An axiomatisation is an enumerable predicate $\mathcal{A} : \mathsf{Form} \to \mathbb{P}$.
The decision problem $\mathcal{A}^{\vDash}$ contains the closed formulas φ with $\mathcal{A} \vDash \varphi$, similarly for $\mathcal{A}^{\vdash}$.

Consistency and Incompleteness of Undecidable Axiomatisations

Fact (Consistency)

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

Proof.

Let f witness $p \prec \mathcal{A}^{\vdash}$. Then $\mathcal{A} \nvdash f\, x$ since f is a reduction. Thus $\mathcal{A} \nvdash \bot$ by explosion rule. □

Fact (Synthetic Incompleteness)

If $p \preceq \mathcal{A}^{\vdash}$ and $\mathcal{A}$ is complete and consistent, then p is decidable.

Proof.

1. Completeness of $\mathcal{A}^{\vdash}$ implies decidability of $\mathcal{A}^{\vdash}$ via Post's theorem. The premises are enumerability of $\mathcal{A}^{\vdash}$ (immediate), enumerability of its complement (as $\mathcal{A} \nvdash \varphi$ iff $\mathcal{A} \vdash \neg\varphi$), and logical decidability of $\mathcal{A}^{\vdash}$ (as $\mathcal{A} \vdash \varphi \vee \mathcal{A} \vdash \neg\varphi$ implies $\mathcal{A} \vdash \varphi \vee \mathcal{A} \nvdash \varphi$).
2. Decidability of p follows by transporting back along $p \preceq \mathcal{A}^{\vdash}$ (also if $\mathcal{A} \vdash \bot$). □

Undecidability: General Strategy for an axiomatisation $\mathcal{A}$

1. Pick a suitable undecidable seed $p : X \to \mathbb{P}$ problem
2. Define the reduction function $f : X \to \mathsf{Form}$
3. Isolate a minimal finite fragment $A \subseteq \mathcal{A}$
4. Show that $p\,x$ implies $\mathcal{M} \vDash f\,x$ for all models $\mathcal{M} \vDash A$
5. Show that $\mathcal{M} \vDash f\,x$ implies $p\,x$ if $\mathcal{M}$ is standard (i.e. well-behaved)
6. Construct a standard model, possibly relying on assumptions
7. Repeat step 4 deductively ($p\,x$ implies $A \vdash f\,x$)

Theorem (Generic Undecidability)

Given an axiomatisation $\mathcal{A}$, a problem $p : X \to \mathbb{P}$, and an encoding $f : X \to \mathsf{Form}$ such that:

- *$\mathcal{M} \vDash f\,x$ implies $p\,x$ if $\mathcal{M}$ is standard*
- *$p\,x$ implies $A \vdash f\,x$*

Then for all $\mathcal{B} \supseteq \mathcal{A}$ admitting a standard model, $p \preceq \mathcal{B}^{\vDash}$ and $p \preceq \mathcal{B}^{\vdash_i}$. With LEM also $p \preceq \mathcal{B}^{\vdash_c}$.

Case Studies

Peano Arithmetic

Signature with zero, successor, addition, multiplication, and equality:

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

1. Seed problem: solvability of diophantine equations (H10)[1]
2. Reduction function: polynomial equation $p = q$ encoded as $\exists^* \overline{p} \equiv \overline{q}$
3. Core axiomatisation Q′: Dedekind equations characterising $\oplus$ and $\otimes$
4. Verification: straightforward using the canonical homomorphism $\mathbb{N} \hookrightarrow$ Term
5. Standard model: $\mathcal{N} = (\mathbb{N}, +, \times)$

Theorem

Q′ and all its extensions satisfied by $\mathcal{N}$ like Robinson arithmetic Q or full PA are undecidable and incomplete. Without LEM, these hold for the respective fragments of Heyting arithmetic.

[1] Reduction HALT $\preceq$ H10 mechanised by Larchey-Wendling and Forster (2019)

ZF Set Theory

Signature with empty set, pairing, union, power set, infinite set, equality, and membership:

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

1. Seed problem: Post correspondence problem (PCP)[2]
2. Reduction function: encode numbers, Booleans, strings, recursion (backup slide)
3. Core axiomatisation Z′: extensionality and characterisations of set operations
4. Verification: develop basic set theory, inline recursion theorem (backup slide)
5. Standard model: $\mathcal{M}$ where $\omega^{\mathcal{M}} \cong \mathbb{N}$, needs assumptions for full ZF (backup slide)

Theorem

Z′ and all its extensions satisfied by standard models like Z or full ZF are undecidable and incomplete. Without LEM, these hold for the respective fragments of intuitionistic ZF.

[2] Reduction HALT $\preceq$ PCP mechanised by Forster, Heiter, and Smolka (2018)

ZF Set Theory without Function Symbols

Core axiomatisation $Z'_\in$ minimal signature $\Sigma = (\_ \in \_)$ not even containing equality.

- Extensionality axiom: $\forall xy.\, (\forall z.\, z \in x \leftrightarrow z \in y) \rightarrow (\forall z.\, x \in z \leftrightarrow y \in z)$
- Set operations existentially guaranteed: $\forall x.\, \exists u.\, \forall y.\, y \in u \leftrightarrow y \subseteq x$

Direct reduction from PCP unfeasible, instead verify translation from previous signature:

- Encode terms t as formulas F_t^x stating that variable x behaves like t: $F_\emptyset^x := \forall y.\, y \notin x$
- Encode formulas accordingly: $F_{t \in t'} := \exists xy.\, F_t^x \wedge F_{t'}^y \wedge x \in y$
- Verify only needed directions: $Z'_\in \vDash F_\varphi \;\rightarrow\; Z' \vDash \varphi$ and $Z' \vdash \varphi \;\rightarrow\; Z'_\in \vdash F_\varphi$

Theorem

The axiomatisation $Z'_\in$ is undecidable and incomplete. LEM needed for $\vdash_c$.

Corollary (Improving on Forster, K., and Smolka (2019))

First-order logic with a single binary relation symbol is undecidable. LEM needed for $\vdash_c$.

Conclusion

Coq Mechanisation

- Mostly axiom-free, only local use of LEM and axioms for models of ZF
- FOL mechanisation synthesis of previous developments
- 5300 new lines of code, 1300 reused
  - 700loc for reduction from H10 to PA
  - 1600loc for reduction from PCP to ZF with function symbols
  - 3000loc for elimination of function symbols
- Inspiration for tooling: related talk @ Coq Workshop (Friday, 11:35)
- Included in the Coq Library of Undecidability Proofs (Forster et al. (2020))

Future Directions

- Strengthening and generalisation
  - Friedman translation to obtain data from classical deductions without LEM
  - Extract reduction functions to computational model for negated completeness
  - Eliminate power set and infinity axioms from set theory reduction
  - Mechanise the conservativity of FOL with definable symbols
- Find the most economical undecidability proof for FOL
  - Direct reduction into FOL with only $\perp$, $\rightarrow$, and $\forall$ over a single binary relation
- Mechanise Tennenbaum's theorem ($\mathbb{N}$ is the only recursive model of PA)
  - Connected to incompleteness, characteristic of constructive Tarski semantics
- Undecidability and incompleteness of second-order logic
  - By incompleteness and categoricity of second-order Peano arithmetic

Wrap-Up

- Synthetic approach eases mechanised undecidability proofs
- Synthetic approach eases mechanised incompleteness proofs
- Synthetic approach available in most constructive foundations (and even Coq + LEM)

`www.ps.uni-saarland.de/extras/axiomatisations/`

Thank you!

Bibliography I

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

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

Church, A. (1936). A note on the entscheidungsproblem. *J. Symb. Log.*, 1(1):40–41.

Forster, Y., Heiter, E., and Smolka, G. (2018). Verification of PCP-related computational reductions in Coq. In *International Conference on Interactive Theorem Proving*, pages 253–269. Springer.

Forster, Y., Kirst, D., and Smolka, G. (2019). On synthetic undecidability in Coq, with an application to the Entscheidungsproblem. In *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 38–51.

Forster, Y., Larchey-Wendling, D., Dudenhefner, A., Heiter, E., Kirst, D., Kunze, F., Smolka, G., Spies, S., Wehr, D., and Wuttke, M. (2020). A coq library of undecidable problems. In *CoqPL 2020*.

Kirst, D. and Larchey-Wendling, D. (2020). Trakhtenbrot's theorem in coq. In *International Joint Conference on Automated Reasoning*, pages 79–96. Springer.

Kirst, D. and Smolka, G. (2018). Large model constructions for second-order ZF in dependent type theory. *Certified Programs and Proofs - 7th International Conference, CPP 2018, Los Angeles, USA, 2018*.

Bibliography II

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

O'Connor, R. (2005). Essential incompleteness of arithmetic verified by Coq. In Hurd, J. and Melham, T., editors, *Theorem Proving in Higher Order Logics*, pages 245–260, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

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

Shankar, N. (1986). *Proof-checking metamathematics*. The University of Texas at Austin. PhD Thesis.

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

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

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

Encoding PCP in Set Theory (Construction)

PCP characterised inductively over a finite stack S of pairs (s, t) of Boolean strings:

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

Ingredients expressible in set theory via standard encodings:

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

Solvability condition of PCP expressed by accumulating all derivations recursively:

- "$\exists x.\, (x,x) \in \bigcup_{k\in\omega} \overline{S}^k$" where $\overline{S}^0 \triangleq \overline{S}$ and $\overline{S}^{k+1} \triangleq S \boxtimes \overline{S}^k \triangleq \bigcup_{s/t \in S} \{(\overline{s}x, \overline{t}y) \mid (x,y) \in \overline{S}^k\}$
- $\varphi_S := \exists k, f, B, x.\, k \in \omega \wedge (\forall (l,B),(l,B') \in f.\, B = B') \wedge f \gg k \wedge (k,B) \in f \wedge (x,x) \in B$

Encoding PCP in Set Theory (Verification)

With basic results about binary union and ordered pairs obtain (for $n, m : \mathbb{N}$ and $s, t : \mathbb{B}^*$):

1. $\mathcal{M} \vDash \overline{n} \in \omega$
2. $\mathcal{M} \vDash \overline{n} \notin \overline{n}$
3. $\mathcal{M} \vDash \overline{n} \equiv \overline{m}$ implies $n = m$
4. $\mathcal{M} \vDash \overline{s} \equiv \overline{t}$ implies $s = t$

Lemma

For $n : \mathbb{N}$ and $f_S^n := \{(\emptyset, \overline{S}), \dots, (\overline{n}, \overline{S^n})\}$ we have $\mathcal{M} \vDash f_S^n \gg \overline{n}$ in every model $\mathcal{M} \vDash \mathsf{Z}'$.

Corollary

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

Lemma

If in a standard model $\mathcal{M} \vDash \mathsf{Z}'$ there is a functional approximation $f \gg k$ for $k \in \omega$ with $(k, B) \in f$, then for all $p \in B$ there are $s, t : \mathbb{B}^*$ with $p = (\overline{s}, \overline{t})$ and $S \triangleright (s, t)$.

Corollary

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

Standard Models of Set Theory

Aczel's sets-as-trees interpretation (Aczel (1978); Werner (1997); Barras (2010)):

- Inductive type of well-founded trees $\mathcal{T}$ with constructor $\tau : \forall X. (X \to \mathcal{T}) \to T$
- Equality interpreted as bisimulation $t \approx t'$
- Membership interpreted by $t \in (\tau\, X\, f) := \exists x.\, t \approx f\, x$
- Models constructive set theory, assumptions needed for classical ZF

Previous work isolates assumptions for fragments (Kirst and Smolka (2018)):

$$\mathsf{CE} := \forall (P, P' : \mathcal{T} \to \mathbb{P}). (\forall t.\, P\, t \leftrightarrow P'\, t) \to P = P'$$
$$\mathsf{TD} := \exists (\delta : (\mathcal{T} \to \mathbb{P}) \to \mathcal{T}). \forall P. (\exists t.\, P = [t]_{\approx}) \to P\, (\delta\, P)$$

- Setoid models of Z′ and Z for free
- Quotiented models of Z′ and Z require CE
- Model of ZF requires both CE and TD