Modeling the Arithmetical Hierarchy in Coq

First Bachelor Seminar Talk

Niklas Mück

Advisors: Yannick Forster and Dominik Kirst
Supervisor: Prof. Gert Smolka

Programming Systems Lab, Saarland University

January 06, 2022

What if we could solve the Halting Problem?

UNIVERSITÄT DES SAARLANDES

Halting Problem [Turing, 1936]

"Does a Turing machine halt on a given input?"

☞ The halting problem is undecidable.

Oracle Machine [Turing (PhD thesis), 1939]

"A Turing machine having a black box for solving a given problem"

Turing reducibility [Turing (PhD thesis), 1939] [Post, 1944]

$A \leq_T B :=$"A can be solved by an oracle machine for B"

What if we could solve the Halting Problem?

Totality

Tot :=“Does a Turing machine halt on **all** inputs?”

☞ H $\leq_T$ Tot, but Tot $\not\leq_T$ H
Even with an oracle for the halting problem, the complement $\overline{\text{Tot}}$ is only semi-decidable.

Cofiniteness

Cof :=“Does a Turing machine halt on **all but finitely many** inputs?”

☞ Tot $\leq_T$ Cof, but Cof $\not\leq_T$ Tot
Even with an oracle for Totality, the problem Cof is only semi-decidable.

An interesting observation

$h(M, i, s)$:= "Turing machine M halts on input i after $\leq s$ steps"

Halting Problem	$\mathsf{H}(M, i) := \exists s.\ h(M, i, s)$

$\not\leq_T$

Totality	$\mathsf{Tot}(M) := \forall i.\ \exists s.\ h(M, i, s)$

$\not\leq_T$

Cofiniteness	$\mathsf{Cof}(M) := \exists n.\ \forall i \geq n.\ \exists s.\ h(M, i, s)$

☞ Post's Theorem [Post, 1948]:
Connection between level of undecidability and quantifier prefix

Arithmetical Hierarchy

Arithmetical Hierarchy

Arithmetical Hierarchy [Kleene, 1943]

Let p be a decidable predicate on numbers:

- $\underbrace{\exists x_1 \; \forall x_2 \; \exists x_3 \; \dots}_{n} \; p(x_1, \dots, x_n, y) \in \sum_n$
- $\underbrace{\forall x_1 \; \exists x_2 \; \forall x_3 \; \dots}_{n} \; p(x_1, \dots, x_n, y) \in \prod_n$
- $\Delta_n := \sum_n \cap \prod_n$

☞ computable predicates can be expressed in Peano Arithmetic

Arithmetical Hierarchy – first-order Peano Arithmetic [Mostowski, 1947]

Let φ be a first-order formula with all quantifiers in the front.

$n :=$ number of quantifier alternations, then $\varphi \in \begin{cases} \sum_n & \text{first quantifier is } \exists \\ \prod_n & \text{first quantifier is } \forall \end{cases}$

Prenex Normal Form

Prenex Normal Form

For each formula there is an equivalent formula with all quantifiers in the front.

☞ only holds in classical logic

In Coq (see more)

- you need a trick in order to define PNF conversion structurally recursive
- and a lemma for renaming de Bruijn indices

Coq Development

First-order Peano Arithmetic

First-order Peano Arithmetic from the undecidability library[1]

$$t \quad ::= O \mid S\,t \mid t + t \mid t \cdot t$$

$$\varphi : \mathbb{F} ::= t = t \mid \bot \mid \varphi \to \psi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \forall \varphi \mid \exists \varphi$$ (de Bruijn)

Tarski semantics in the standard model: $\rho \vDash_{\mathbb{N}} \varphi$

1 https://github.com/uds-psl/coq-library-undecidability

Arithmetical Hierarchy in Coq – Syntactically

$\sum_n : \mathbb{F} \to \mathbb{P}$

$$\frac{\text{noQuant } \varphi}{\sum_n \varphi} \qquad \frac{\prod_n \varphi}{\sum_{n+1} \exists\varphi} \qquad \frac{\sum_{n+1} \varphi}{\sum_{n+1} \exists\varphi}$$

☞ same definition for $\prod_n$, mutually inductive

For predicates: $p : \mathbb{N}^k \to \mathbb{P}$

$$\sum_n p := \exists\varphi.\ \sum_n \varphi \wedge \underbrace{\forall\vec{n}.\ p\vec{n} \leftrightarrow \vec{n} \vDash_{\mathbb{N}} \varphi}_{\texttt{reflects}\ p\ \varphi}$$

Example

$\sum_1$ even $\qquad \varphi := (\exists k.\ x = 2 \cdot k)$

$\forall n.$ even $n \leftrightarrow [x \mapsto n] \vDash_{\mathbb{N}} \exists k.\ x = 2 \cdot k$

$$\cfrac{\cfrac{\text{noQuant } (x = 2 \cdot k)}{\prod_0 (x = 2 \cdot k)}}{\sum_1 (\exists k.\ x = 2 \cdot k)}$$

Arithmetical Hierarchy in Coq – Semantically

UNIVERSITÄT DES SAARLANDES

$\tilde{\sum}_n^k : (\mathbb{N}^k \to \mathbb{P}) \to \mathbb{P}$

$$\frac{f : \mathbb{N}^k \to \mathbb{B}}{\tilde{\sum}_n^k (\lambda \vec{n}.\ f\vec{n} = \texttt{true})} \qquad \frac{\tilde{\prod}_n^{k+1} p}{\tilde{\sum}_{n+1}^k (\lambda \vec{n}.\ \exists x.\ p(x :: \vec{n}))}$$

☞ same definition for $\tilde{\prod}_n^k$, mutually inductive

Axiom:

$\texttt{predicate\_ext} := \forall pq.(\forall \vec{n}.\ p\vec{n} \leftrightarrow q\vec{n}) \to p = q$

Arithmetical Hierarchy in Coq – Equivalence proof (1)

UNIVERSITÄT DES SAARLANDES

Theorem (syntactic $\to$ semantic)

$$(\forall p\, n.\ \Sigma_n\, p \to \tilde{\Sigma}^k_n\, p) \quad \land \quad (\forall p\, n.\ \Pi_n\, p \to \tilde{\Pi}^k_n\, p)$$

Proof.

Enough to show

$$(\forall \varphi\, n\, k.\ \Sigma_n\, \varphi \to \tilde{\Sigma}^k_n(\lambda \vec{n}.\ \vec{n} \vDash_{\mathbb{N}} \varphi)) \ \land \ (\forall \varphi\, n\, k.\ \Pi_n\, \varphi \to \tilde{\Pi}^k_n(\lambda \vec{n}.\ \vec{n} \vDash_{\mathbb{N}} \varphi))$$

by `predicate_ext`. Proof by mutual induction:

- base case: quantifier-free formulas are decidable
- Σ_n allows stacking same quantifiers, but $\tilde{\Sigma}^k_n$ does not
  ☞ use pairing function and a generalized embedding lemma

Arithmetical Hierarchy in Coq – Equivalence proof (2)

UNIVERSITÄT DES SAARLANDES

Theorem (semantic → syntactic)

$$(\forall p\, n.\ \tilde{\sum}^{k}_{n+1} p \to \sum_{n+1} p) \quad \wedge \quad (\forall p\, n.\ \tilde{\prod}^{k}_{n+1} p \to \prod_{n+1} p)$$

We need to express decidable predicates in first-order logic

- ☞ i.e. translate meta logic into a concrete model of computation
- ☞ we have to assume a $\mathtt{CT}$-like axiom [Kreisel, 1965] ("Church's thesis")

Variant 1 (see other variants)

Assume:

$\forall k\ (f : \mathbb{N}^k \to \mathbb{B}).\Delta_1(\lambda \vec{n}.\ f\vec{n} = \mathtt{true}).$

decidable predicates are syntactically expressible as a Δ_1-formula

Discussion

Wrap-up

Arithmetical Hierarchy in Coq

Two definitions – equivalent when assuming a `CT`-like axiom

- concrete model of computation i.e. Peano Arithmetic
- lifted to meta-theory

☞ we can now start proving interesting properties

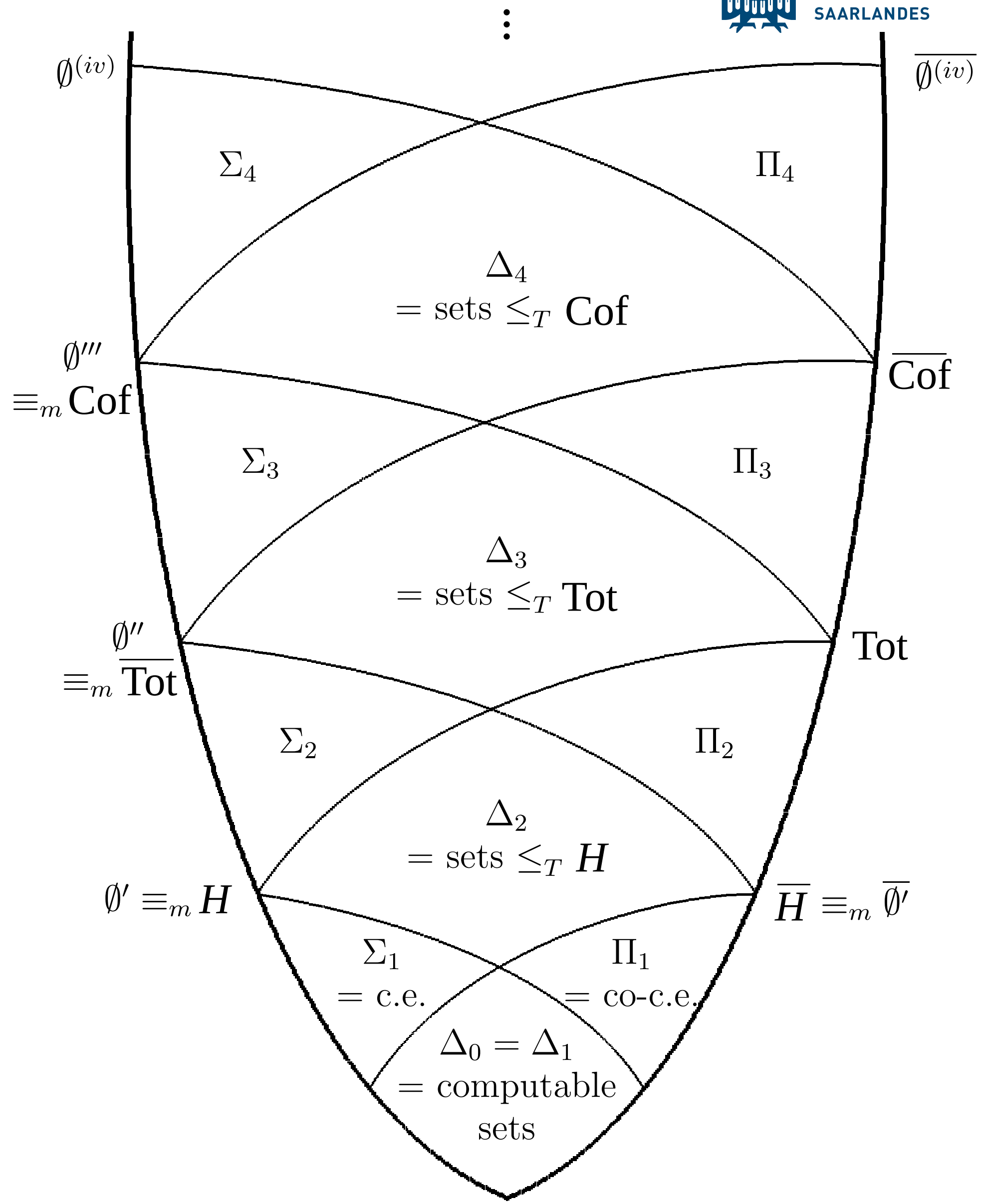
Outlook: Post's Theorem

Turing jump

$A' :=$ "halting problem of Turing machines with an oracle for A"

Post's Theorem [Post, 1948]

- $\emptyset^{(n+1)}$ is $\sum_{n+1}$-complete
- $A \in \sum_{n+1} \iff A$ is c.e. relative to $\emptyset^{(n)}$



References I

- Alan Mathison Turing.
  On computable numbers, with an application to the entscheidungsproblem.
  J. of Math, 58:345–363, 1936.
- Alan Mathison Turing.
  Systems of logic based on ordinals.
  Proceedings of the London mathematical society, 2(1):161–228, 1939.
- Emil L Post.
  Recursively enumerable sets of positive integers and their decision problems.
  bulletin of the American Mathematical Society, 50(5):284–316, 1944.

References II

- Stephen Cole Kleene.
  Recursive predicates and quantifiers.
  Transactions of the American Mathematical Society, 53(1):41–73, 1943.
- Andrzej Mostowski.
  On definable sets of positive integers.
  Fundamenta Mathematicae, 34(1):81–112, 1947.
- Georg Kreisel.
  Mathematical logic.
  Lectures in modern mathematics 3, pages 95–195, 1965.

References III

Emil L Post.

Degrees of recursive unsolvability-preliminary report.

In *Bulletin of the American Mathematical Society*, volume 54, pages 641–642. AMER MATHEMATICAL SOC 201 CHARLES ST, PROVIDENCE, RI 02940-2213, 1948.

Prenex Normal Form

Prenex Normal Form

For each formula there is an equivalent formula with all quantifiers in the front.

Textbooks

Inductive argument showing these rules:

$$
\begin{aligned}
(\forall x.\ \varphi_1) \wedge \varphi_2 &\iff \forall x.\ (\varphi_1 \wedge \varphi_2) & (\forall x.\ \varphi_1) \to \varphi_2 &\iff \exists x.\ (\varphi_1 \to \varphi_2)\\
(\exists x.\ \varphi_1) \wedge \varphi_2 &\iff \exists x.\ (\varphi_1 \wedge \varphi_2) & (\exists x.\ \varphi_1) \to \varphi_2 &\iff \forall x.\ (\varphi_1 \to \varphi_2)\\
(\forall x.\ \varphi_1) \vee \varphi_2 &\iff \forall x.\ (\varphi_1 \vee \varphi_2) & \varphi_1 \to (\forall x.\ \varphi_2) &\iff \forall x.\ (\varphi_1 \to \varphi_2)\\
(\exists x.\ \varphi_1) \vee \varphi_2 &\iff \exists x.\ (\varphi_1 \vee \varphi_2) & \varphi_1 \to (\exists x.\ \varphi_2) &\iff \exists x.\ (\varphi_1 \to \varphi_2)
\end{aligned}
$$

Some directions only hold in classical logic

PNF in Coq

UNIVERSITÄT DES SAARLANDES

First-order logic from undecidability library

For a fixed signature with relation symbols P and terms t we define $\varphi : \mathbb{F}$

$\varphi ::= P\vec{t} \mid \bot \mid \varphi \to \psi \mid \varphi \land \psi \mid \varphi \lor \psi \mid \forall\varphi \mid \exists\varphi$ (de Bruijn)

Tarski semantics over a given ρ and a fixed structure: $\rho \vDash \varphi$

PNF : $\mathbb{F} \to \mathbb{P}$

$$\frac{\text{PNF } \varphi}{\text{PNF } (\forall\ \varphi)} \qquad \frac{\text{PNF } \varphi}{\text{PNF } (\exists\ \varphi)} \qquad \frac{\text{noQuant } \varphi}{\text{PNF } \varphi}$$

PNF – full definition

UNIVERSITÄT DES SAARLANDES

PNF : $\mathbb{F} \to \mathbb{P}$

$$\frac{\text{PNF } \varphi}{\text{PNF } (\forall\ \varphi)} \qquad \frac{\text{PNF } \varphi}{\text{PNF } (\exists\ \varphi)} \qquad \frac{\text{noQuant } \varphi}{\text{PNF } \varphi}$$

noQuant : $\mathbb{F} \to \mathbb{P}$

$$\frac{}{\text{noQuant } \bot} \qquad \frac{}{\text{noQuant } (P\vec{t})} \qquad \frac{\text{noQuant } \varphi_1 \qquad \text{noQuant } \varphi_2}{\text{noQuant } (\varphi_1 \diamond \varphi_2)}$$

PNF conversion in Coq – $\texttt{convert}: \mathbb{F} \to \mathbb{F}$

UNIVERSITÄT DES SAARLANDES

Naive approach: by recursion on the formula

Problem: $(\forall\forall\varphi) \wedge (\exists\exists\exists\psi) \rightsquigarrow \forall \underbrace{(\forall\varphi) \wedge (\exists\exists\exists\psi[\uparrow])}_{\text{not structurally recursive}}$

My solution

Auxiliary function returning a **quantifier prefix as list** and a formula without quantifiers.

$[\forall,\forall]\ \varphi \wedge [\exists,\exists,\exists]\ \psi \rightsquigarrow [\forall,\forall,\exists,\exists,\exists]\ \varphi[\uparrow^3] \wedge \psi[0;1;2;\uparrow^2]$

☞ concatenate quantifier lists and rename de Bruijn indices

Proof.

- Result is a formula in PNF: $\forall\varphi.\ \mathrm{PNF}(\texttt{convert}\ \varphi)$
- Result is an equivalent formula: $\forall\varphi.\ \forall\rho.\ \rho \vDash \varphi \leftrightarrow \rho \vDash (\texttt{convert}\ \varphi)$
  ☞ you need the right de Bruijn lemmas

PNF conversion – de Bruijn lemma

UNIVERSITÄT DES SAARLANDES

Want to show

$$\forall \rho\ \varphi_1\ \varphi_2.\ \ \rho \vDash merge(qs_1 +\!\!+\ qs_2)(\varphi_1[\uparrow^{|qs_2|}] \land \varphi_2[0; \ldots; |qs_2| - 1; \uparrow^{|qs_1|}])$$
$$\leftrightarrow \rho \vDash (merge\ qs_1\ \varphi_1 \land merge\ qs_2\ \varphi_2)$$

by induction on $qs_1 +\!\!+\ qs_2$

We need

$$(merge\ qs\ \varphi)[\uparrow] = merge\ qs\ (\varphi[0; \ldots; |qs_2| - 1; \uparrow^1])$$

Lemma

$$(merge\ qs\ \varphi)[\sigma] = merge\ qs\ \left(\varphi\left[\lambda n. \begin{cases} \$n & \mathtt{if}\ \ n < |qs| \\ \sigma(n - |qs|)[\uparrow^{|qs|}] & \mathtt{else} \end{cases}\right]\right)$$

$\tilde{\sum}_n$ – embedding Lemma

Want to show

$$\forall k\ (p : \mathbb{N}^{k+1} \to \mathbb{P}).\ \tilde{\sum}_n^{k+1} p \to \tilde{\sum}_n^{k} (\lambda \vec{n}.\ \exists x.\ p(x :: \vec{n})))$$

Lemma

$$(\forall k\ (p : \mathbb{N}^{k} \to \mathbb{P})\ k'\ (p' : \mathbb{N}^{k'} \to \mathbb{P})\ (\iota : \mathbb{N}^{k'} \to \mathbb{N}^{k}).\ (\forall \vec{n}. p\,(\iota\ \vec{n}) \leftrightarrow p' \vec{n})$$
$$\to \tilde{\sum}_n^{k} p \to \tilde{\sum}_n^{k'} p')$$
$$\wedge\ (\forall k\ (p : \mathbb{N}^{k} \to \mathbb{P})\ k'\ (p' : \mathbb{N}^{k'} \to \mathbb{P})\ (\iota : \mathbb{N}^{k'} \to \mathbb{N}^{k}).\ (\forall \vec{n}. p\,(\iota\ \vec{n}) \leftrightarrow p' \vec{n})$$
$$\to \tilde{\prod}_n^{k} p \to \tilde{\prod}_n^{k'} p')$$

Variants of CT-like axiom

Variant 1

Assume:

$\forall k\ (f : \mathbb{N}^k \to \mathbb{B}).\ \Delta_1(\lambda \vec{n}.\ f\vec{n} = \texttt{true})$

computable predicates are syntactically in Δ_1

Variant 2

Assume:

(i) $\forall (p : \mathbb{N}^k \to \mathbb{P}).\ \tilde{\sum}_1^k p \to \sum_1 p$

(ii) $\forall (p : \mathbb{N}^k \to \mathbb{P}).\ \tilde{\prod}_1^k p \to \prod_1 p$ (a)

or Markov's principle (b)

Variant 1 is equivalent to Variant 2 (a)