UNIVERSITÄT DES SAARLANDES

The Arithmetical Hierarchy, Oracle Computability, and Post's Theorem in Synthetic Computability

Final Bachelor Talk

Niklas Mück

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

Programming Systems Lab, Saarland University

July 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 (1939)]

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

Turing reducibility [Turing (1939); Post (1944)]

$P \leq_T Q$:="P can be solved by an oracle machine for Q"

☞ "P is decidable *relative* to Q"

What if we could solve the Halting Problem?

Well-known problems that are undecidable relative to the halting problem:

Totality

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

☞ H $\leq_T$ Tot, but Tot $\not\leq_T$ H
Best one can do: Semi-decider for $\overline{\text{Tot}}$ relative to H.

Cofiniteness

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

☞ Tot $\leq_T$ Cof, but Cof $\not\leq_T$ Tot
Best one can do: Semi-decider for Cof relative to $\overline{\text{Tot}}$.

What if we could solve the Halting Problem?

UNIVERSITÄT DES SAARLANDES

For each problem there exists a relatively undecidable problem:

Turing jump [Post (1948); Kleene and Post (1954)]

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

- ☞ Q' is semi-decidable by oracle machines with an oracle for Q.
- ☞ Repeated jumping gives rise to a hierarchy of undecidability.
- ☞ $\emptyset^{(n)} :=$ "the n-th Turing jump starting with the empty predicate"

Arithmetical Hierarchy [Kleene (1943); Mostowski (1947)]

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

Halting Problem	$\mathsf{H}(M, i) := \exists s.\ h(M, i, s)$	$\in \Sigma_1$
$\lneq_T$		
Totality	$\mathsf{Tot}(M) := \forall i.\ \exists s.\ h(M, i, s)$	$\in \Pi_2$
$\lneq_T$		
Cofiniteness	$\mathsf{Cof}(M) := \exists n.\ \forall i \geq n.\ \exists s.\ h(M, i, s)$	$\in \Sigma_3$

☞ Post's Theorem [Post (1948)]:
Connection between the arithmetical hierarchy and the Turing jump

Saarland University
Faculty of Mathematics and Computer Science

Bachelor's Thesis

The Arithmetical Hierarchy, Oracle Computability, and Post's Theorem in Synthetic Computability

Author
Niklas Mück

Supervisor
Prof. Dr. Gert Smolka

Advisors
Dr. Yannick Forster
Dominik Kirst

Reviewers
Prof. Dr. Gert Smolka
Dr. Yannick Forster

Synthetic Computability[1]

- ☞ Consider all (partial) functions e.g. $\mathbb{N} \to \mathbb{N}$ as computable
- ☞ In constructive type theory only computable functions can be defined

Definition

A predicate $P : \mathbb{N} \to \mathbb{P}$ is

- decidable: $\mathcal{D}(P) := \exists f : \mathbb{N} \to \mathbb{B}.\, P\,x \leftrightarrow f\,x = \texttt{true}$
- semi-decidable: $\mathcal{S}(P) := \exists f : \mathbb{N} \rightharpoonup \mathbb{1}.\, P\,x \leftrightarrow f\,x \triangleright \star$
- many-one reducible to a predicate $Q : \mathbb{N} \to \mathbb{P}$:
  $P \preceq_m Q := \exists f : \mathbb{N} \to \mathbb{N}.\, P\,x \leftrightarrow Q\,(f\,x)$

- ☞ Native reasoning without manipulating concrete models of computation

[1] Approach by [Richman (1983); Bridges and Richman (1987); Bauer (2005)]
In constructive type theory by [Forster et al. (2019); Forster (2021b)]

Synthetic Computability – Halting Problem

“Does a partial function output a value?”

Problem: (Partial) functions are not associated with their source code

☞ Gödel encoding cannot be constructed

Axiom: Enumerability of Partial Functions [Richman (1983); Forster (2021a)]

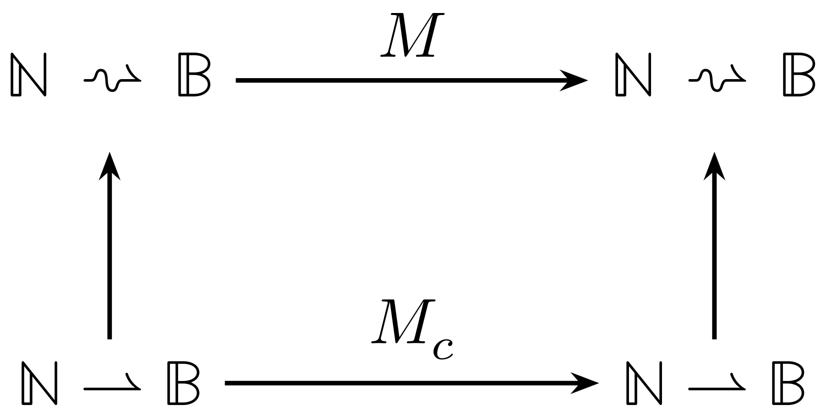
$$\text{EPF} := \Sigma\theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N}).\ \forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists c : \mathbb{N}.\ \theta\, i \approx f$$

$\theta\, i\, x \triangleright y \triangleq$ “i-th partial function terminates on x with output y”

Self-halting problem $\mathcal{K} i := \exists y.\ \theta\, i\, i \triangleright y$

Synthetic Computability – Turing Reductions[2]

$P \preceq_T Q :=$ "M that maps the characteristic relation of Q to the one of P"



$\mathbb{N} \rightsquigarrow \mathbb{B} := \{R : \mathbb{N} \to \mathbb{B} \to \mathbb{P} \mid R \text{ functional}\}$

require diagram to commute

Theorem

If Q is undecidable then for any P: $P \preceq_T Q$

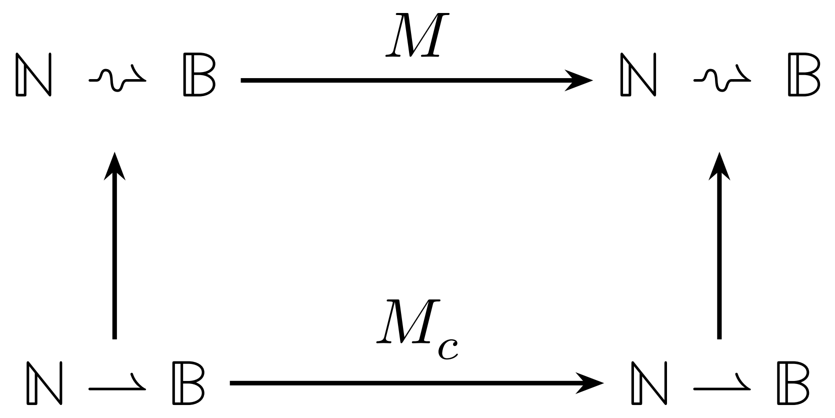
Proof.

Define $M\,R\,x\,b$ as: If R is decidable then `true` else reflect P. □

[2] Forster (2021b) in joint work with Kirst following two-layer idea by Bauer (2021)

Synthetic Computability – Turing Reductions[2]

$P \preceq_T Q :=$ "M that maps the characteristic relation of Q to the one of P"



$\mathbb{N} \rightsquigarrow \mathbb{B} := \{R : \mathbb{N} \to \mathbb{B} \to \mathbb{P} \mid R \text{ functional}\}$

require diagram to commute

- ☞ Prevent M from inspecting the oracle globally
- ☞ Forster-Krist: require M to be weakly continuous and monotonic
  - $\forall R\, x.\, \neg\neg\exists L \in \mathscr{L}(\mathbb{N}).\, \forall R' \supseteq_L R \to \forall y.\, M\, R\, x\, y \to M\, R'\, x\, y$
  - $\forall R\, R'.\, R \subseteq R' \to M\, R \subseteq M\, R'$

[2]Forster (2021b) in joint work with Kirst following two-layer idea by Bauer (2021)

Synthetic Oracle Computability

Before my Bachelor's thesis:

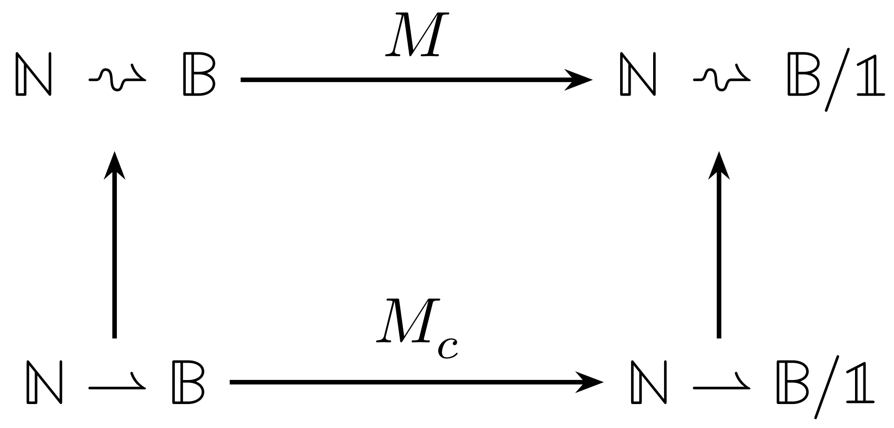
- There was only a proposal of synthetic Turing reductions
- Forster has shown that it differs from truth-table reductions

My Bachelor's thesis:

- Advance definition of Turing reducibility (constructively strengthen continuity requirement)
- Constructive results: can be expressed solely by a continuous higher order partial function
- Connection to a textbook presentation of the arithmetical hierarchy by proving Post's theorem synthetically

Advancing Synthetic Oracle Computability

Synthetic Oracle Computability



$\mathbb{N} \rightsquigarrow \mathbb{B} := \{R : \mathbb{N} \to \mathbb{B} \to \mathbb{P} \mid R \text{ functional}\}$

require diagram to commute

☞ Require M to be constructively continuous

- $\forall R\, x\, y.\ M\, R\, x\, y \to \exists L \subseteq \mathsf{Dom}(R).\ \forall R' =_L R.\ M\, R'\, x\, y$

Turing Reducibility

$P \preceq Q := \exists M : \mathbb{M}_{\mathbb{B}}.\ M\, Q \approx P$

Oracle Semi-decidability

$\mathcal{S}_Q(P) := \exists M : \mathbb{M}_{\mathbb{1}}.\ M\, Q\, x \star \leftrightarrow P\, x$

Determinacy of Oracle Machines by Their Cores

UNIVERSITÄT DES SAARLANDES

Theorem

$$M\,R\,x\,y \leftrightarrow \exists L_{\texttt{true}}\,L_{\texttt{false}}.\,(\forall a \in L_{\texttt{true}}.\ R\,a\,\texttt{true}) \land (\forall a \in L_{\texttt{false}}.\ R\,a\,\texttt{false}) \land M_c\,(\texttt{lookup}\,L_{\texttt{true}}\,L_{\texttt{false}})\,x \triangleright y$$

where $\texttt{lookup}\,L_{\texttt{true}}\,L_{\texttt{false}}\,a := \begin{cases} \texttt{true} & \text{if } a \in L_{\texttt{true}} \\ \texttt{false} & \text{if } a \in L_{\texttt{false}} \\ \texttt{undef.} & \text{else} \end{cases}$

Proof.

M is continuous:

$$\forall R\,x\,y.\ M\,R\,x\,y \to \exists L \subseteq \mathsf{Dom}(R).\ \forall R' =_L R.\ M\,R'\,x\,y$$

- ☞ $L \subseteq \mathsf{Dom}(R) := \forall a \in L.\,\exists b.\ R\,a\,b$
- ☞ split L into $L_{\texttt{true}}$ and $L_{\texttt{false}}$
- ☞ Proof is constructive

Synthetic Turing Jump

Theorem

$$M\,R\,x\,y \leftrightarrow \exists L_{\texttt{true}}\,L_{\texttt{false}}.\,(\forall a \in L_{\texttt{true}}.\ R\,a\,\texttt{true}) \land (\forall a \in L_{\texttt{false}}.\ R\,a\,\texttt{false}) \land M_c\,(\texttt{lookup}\,L_{\texttt{true}}\,L_{\texttt{false}})\,x \triangleright y$$

☞ One-to-one correspondence between oracle machines and continuous higher-order partial functions.

Axiom

Assume an enumeration of continuous higher-order partial functions $\xi : \mathbb{N} \to ((\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{1}))$

☞ Construct enumeration of oracle machines $\Xi : \mathbb{N} \to \mathbb{M}_{\mathbb{1}}$

Turing jump

$Q'\,i := (\Xi\,i)\,Q\,i\,\star$

Fact

$\mathsf{S}_Q(Q')$ but $\neg\mathsf{S}_Q(\overline{Q'})$

Arithmetical Hierarchy

Arithmetical Hierarchy in First-Order Logic

Definition in first-order arithmetic[3] following Odifreddi (1992) without relying on a concrete model of computation:

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

Classify formulas in prenex normal form[4] (all quantifier in front):

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

☞ same definition for $\prod_n$, mutually inductive

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

$$\sum_n p := \exists\varphi.\ \sum_n \varphi \wedge (\forall\vec{v}.\ p\vec{v} \leftrightarrow \vec{v} \vDash_{\mathbb{N}} \varphi)$$

[3] Using the mechanization from the "Coq Library for Mechanised First-Order Logic" [Kirst et al. (2022)]
[4] I have mechanized an algorithm for that

Arithmetical Hierarchy – Synthetic Definition

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

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

☞ same definition for $\tilde{\Pi}_n$, mutually inductive

☞ Both definitions are equivalent when assuming a CT[5]-like axiom

Axiom

$\forall f : \mathbb{N}^k \to \mathbb{B}.\ \Sigma_1(\lambda \vec{v}.\ f\vec{v} = \texttt{true})$

☞ The synthetic definition is more elegant to establish synthetic results

[5] Kreisel (1965)

Proving Post’s Theorem Synthetically

Post's Theorem

UNIVERSITÄT DES SAARLANDES

Theorem

LEM $\rightarrow P \in \tilde{\Sigma}_{n+1} \leftrightarrow \exists Q \in \tilde{\Pi}_n .\, \mathcal{S}_Q(P)$

Proof.

$\rightarrow$ Linearly search for the $\exists$-quantified value

$\leftarrow$ "There exists a number of steps"-intuition does not work.

Instead: follow proof of Odifreddi (1992) and show: Given $Q \in \tilde{\Pi}_n$

$$\exists L_{\mathtt{t}}\, L_{\mathtt{f}}.\ \underbrace{(\forall a \in L_{\mathtt{t}}.\ Q\, a)}_{\substack{\text{bounded quantifier} \\ \in \tilde{\Pi}_n}} \land \underbrace{(\forall a \in L_{\mathtt{f}}.\ \overline{Q}\, a)}_{\substack{\text{requires LEM} \\ \in \tilde{\Sigma}_n}} \land \underbrace{M_c\, (\mathtt{lookup}\, L_{\mathtt{t}}\, L_{\mathtt{f}})\, x \rhd y}_{\substack{\text{partial functions are stepwise} \\ \in \tilde{\Sigma}_1}} \in \tilde{\Sigma}_{n+1}$$

$\square$

Post's Theorem

Theorem

$\text{LEM} \to P \in \tilde{\sum}_{n+1} \leftrightarrow \exists Q \in \tilde{\prod}_n \, . \, \mathcal{S}_Q(P)$

Corollary

$\text{LEM} \to P \in \tilde{\sum}_{n+1} \leftrightarrow \exists Q \in \tilde{\sum}_n \, . \, \mathcal{S}_Q(P)$

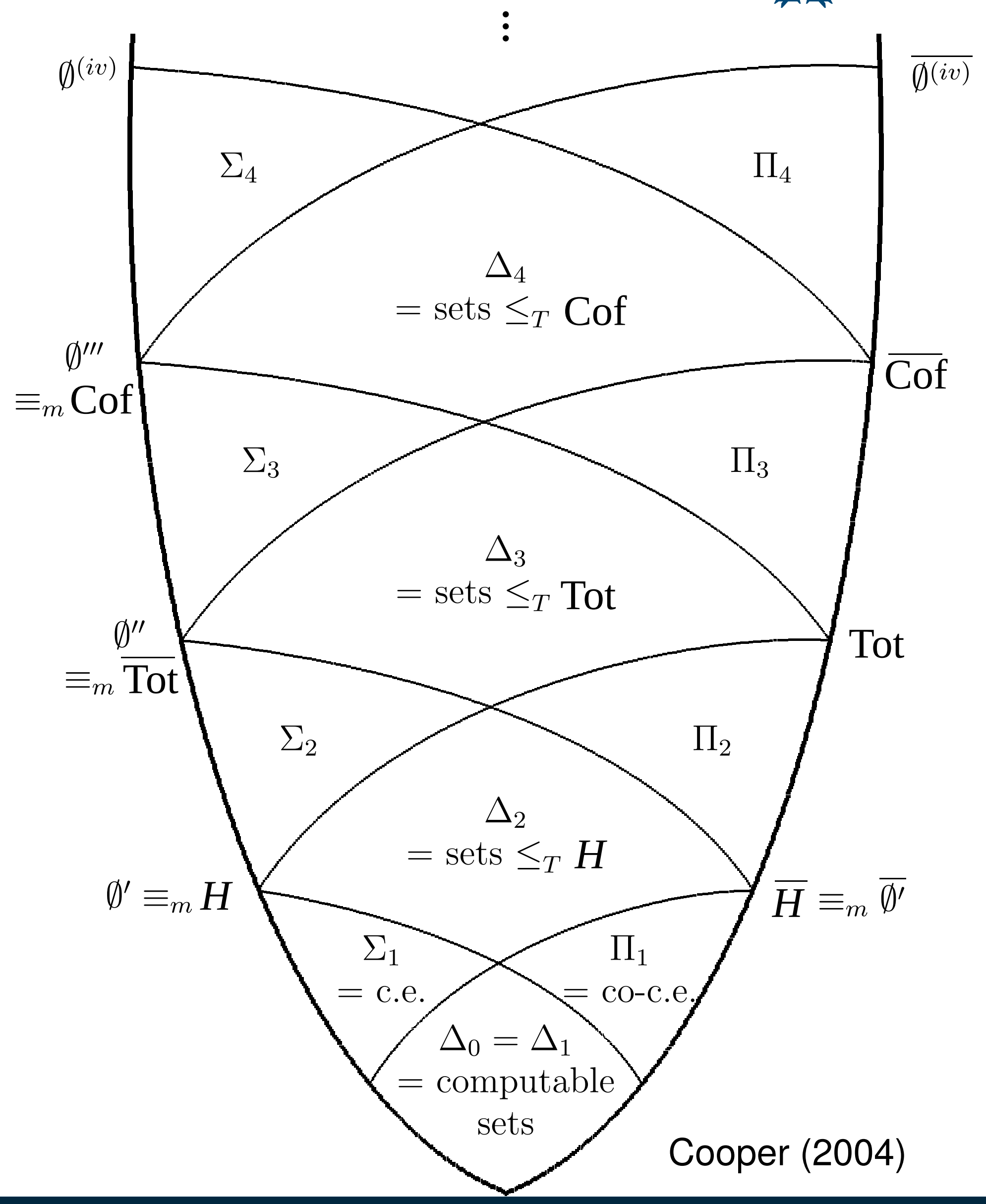
Corollary

$\text{LEM} \to \emptyset^{(n+1)} \in \tilde{\sum}_{n+1}$ is many-one complete

$\text{LEM} \to \emptyset^{(n)} \in \tilde{\sum}_n$ is Turing complete

Corollary

$\text{LEM} \to P \in \tilde{\sum}_{n+1} \leftrightarrow \mathcal{S}_{\emptyset^{(n)}}(P)$



Cooper (2004)

Overview of My Contributions

- Advance definition of synthetic oracle machines (constructive continuity)
  ☞ can be expressed solely by a continuous higher order partial function

- Identify axiom needed for the first synthetic definition of the Turing jump

- Validate synthetic definitions by proving Post's theorem, connecting to
  - Synthetic definition of the arithmetical hierarchy, shown equivalent[6] to
  - The arithmetical hierarchy in first-order logic found in Odifreddi (1992)
    - A mechanized and structurally recursive algorithm for PNF conversion

☞ All in all, I ratify the existing definition of synthetic Turing reductions but propose a constructive refinement

[6] using a CT-like axiom

Bibliography I

Andrej Bauer. First steps in synthetic computability theory. In Martín Hötzel Escardó, Achim Jung, and Michael W. Mislove, editors, *Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics, MFPS 2005, Birmingham, UK, May 18-21, 2005*, volume 155 of *Electronic Notes in Theoretical Computer Science*, pages 5–31. Elsevier, 2005. 10.1016/j.entcs.2005.11.049.

Andrej Bauer. Synthetic mathematics with an excursion into computability theory. University of Wisconsin Logic seminar, 2021. URL http://math.andrej.com/asset/data/madison-synthetic-computability-talk.pdf.

Douglas Bridges and Fred Richman. *Varieties of Constructive Mathematics*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1987. 10.1017/CBO9780511565663.

S. Barry Cooper. *Computability theory*. Chapman & Hall/CRC Press, 2004. ISBN 1584882379.

Yannick Forster. Church's thesis and related axioms in Coq's type theory. In Christel Baier and Jean Goubault-Larrecq, editors, *29th EACSL Annual Conference on Computer Science Logic, CSL 2021, January 25-28, 2021, Ljubljana, Slovenia (Virtual Conference)*, volume 183 of *LIPIcs*, pages 21:1–21:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021a. 10.4230/LIPIcs.CSL.2021.21.

Yannick Forster. *Computability in Constructive Type Theory*. PhD thesis, Saarland University, 2021b. doi:10.22028/D291-35758.

Bibliography II

Yannick Forster. Parametric Church's thesis: Synthetic computability without choice. In Sergei N. Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2022, Deerfield Beach, FL, USA, January 10-13, 2022, Proceedings*, volume 13137 of *Lecture Notes in Computer Science*, pages 70–89. Springer, 2022. 10.1007/978-3-030-93100-1_6. URL https://doi.org/10.1007/978-3-030-93100-1_6.

Yannick Forster, Dominik Kirst, and Gert Smolka. On synthetic undecidability in Coq, with an application to the Entscheidungsproblem. In Assia Mahboubi and Magnus O. Myreen, editors, *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2019, Cascais, Portugal, January 14-15, 2019*, pages 38–51. ACM, 2019. 10.1145/3293880.3294091.

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

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

Stephen C. Kleene. Recursive predicates and quantifiers. *Transactions of the American Mathematical Society*, 53 (1):41–73, 1943. 10.1090/S0002-9947-1943-0007371-8.

Stephen C. Kleene and Emil L. Post. The upper semi-lattice of degrees of recursive unsolvability. *Annals of mathematics*, pages 379–407, 1954. 10.2307/1969708.

Georg Kreisel. Mathematical logic. *Lectures in modern mathematics 3*, pages 95–195, 1965.

Bibliography III

Dominique Larchey-Wendling and Yannick Forster. Hilbert's tenth problem in coq (extended version). *Log. Methods Comput. Sci.*, 18(1), 2022. 10.46298/lmcs-18(1:35)2022.

Andrzej Mostowski. On definable sets of positive integers. *Fundamenta Mathematicae*, 34(1):81–112, 1947. 10.4064/fm-34-1-81-112.

Piergiorgio Odifreddi. *Classical recursion theory. The theory of functions and sets of natural numbers.*, volume 125 of *Stud. Logic Found. Math.* Amsterdam etc.: North-Holland, paperback ed. edition, 1992. ISBN 0-444-89483-7.

Benjamin Petres. *Gödel's Theorem Without Tears - Essential Incompleteness in Synthetic Computability*. Bachelor's thesis, Saarland University, June 2022. URL https://www.ps.uni-saarland.de/~peters/bachelor/resources/thesis.pdf.

Emil L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society*, 50(5):284–316, 1944. 10.1090/s0002-9904-1944-08111-1.

Emil L. Post. Degrees of recursive unsolvability-preliminary report. In *Bulletin of the American Mathematical Society*, volume 54, pages 641–642, 1948.

Fred Richman. Church's thesis without tears. *J. Symb. Log.*, 48(3):797–803, 1983. 10.2307/2273473.

Thomas Streicher. Realizability. Lecture Notes, WS 17/18. URL https://www2.mathematik.tu-darmstadt.de/~streicher/REAL/REAL.pdf.

Bibliography IV

UNIVERSITÄT DES SAARLANDES

Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *J. of Math*, 58:345–363, 1936.

Alan M. Turing. Systems of logic based on ordinals. *Proceedings of the London mathematical society*, 2(1):161–228, 1939.

Coq Development

	Specification	Proofs
Prenex Normal Form	185	326
Arithmetical Hierarchy in First-order Logic	45	236
Arithmetical Hierarchy in Type Theory	103	459
Arithmetical Hierarchy – Equivalence	15	105
Oracle Computability	170	649
Turing Jump	64	152
Post's Theorem	49	168
Total	631	2095

Dependencies: Formalization of partial functions by Forster (2021b) and syntax and semantics of first-order logic by Kirst et al. (2022)

Prenex Normal Form

UNIVERSITÄT DES SAARLANDES

Prenex Normal Form

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

Textbooks

Inductive argument showing these rules:

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

Some directions only hold in classical logic

Mechanization of PNF

First-order logic from Coq FOL library [Kirst et al. (2022)]

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

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

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

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

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

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 – $\texttt{convert}: \mathbb{F} \to \mathbb{F}$

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 functions 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: $\forall\varphi.\ \mathsf{PNF}(\texttt{convert}\ \varphi)$
- Result is an equivalent formula: $\mathsf{LEM} \leftrightarrow \forall\varphi\ \rho.\ \rho \vDash (\varphi \dot{\leftrightarrow} \texttt{convert}\ \varphi)$

  ☞ you need the right de Bruijn lemmas

Arithmetical Hierarchy – Equivalence proof (1)

Theorem (first-order definition $\rightarrow$ synthetic definition)

$$(\forall p\ n.\ \textstyle\sum_n p \rightarrow \tilde{\sum}_n p) \quad \wedge \quad (\forall p\ n.\ \prod_n p \rightarrow \tilde{\prod}_n p)$$

Proof.

Enough to show

$$(\forall \varphi\ n\ k.\ \textstyle\sum_n \varphi \rightarrow \tilde{\sum}_n (\lambda \vec{v}.\ \vec{v} \vDash_{\mathbb{N}} \varphi)) \ \wedge\ (\forall \varphi\ n\ k.\ \prod_n \varphi \rightarrow \tilde{\prod}_n (\lambda \vec{v}.\ \vec{v} \vDash_{\mathbb{N}} \varphi))$$

by `predicate_ext`. Proof by mutual induction:

- base case: quantifier-free formulas are decidable
- $\sum_n$ allows stacking same quantifiers, but $\tilde{\sum}_n$ does not
  ☞ use pairing function and that $\tilde{\sum}_n$ is closed under many-one reduction

Arithmetical Hierarchy – Equivalence proof (2)

Theorem (synthetic definition → synthetic definition)

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

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 CT-like axiom [Kreisel, 1965] ("Church's thesis")

Axiom

$$\forall f : \mathbb{N}^k \to \mathbb{B}\,.\ \sum_1 (\lambda \vec{v}.\ f\vec{v} = \mathtt{true})$$

☞ the same for $\prod_1$ follows

Axioms

Axiom

$\forall f : \mathbb{N}^k \to \mathbb{B}\,.\ \sum_1(\lambda\vec{v}.\ f\vec{v} = \texttt{true})$

Petres (2022) has derived a similar fact by combining the mechanization of the DPRM theorem by Larchey-Wendling and Forster (2022) with representability results by Hermes and Kirst (2022).

Axiom

Assume an enumeration of continuous higher-order partial functions
$\xi : \mathbb{N} \to ((\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{1}))$

Lecture notes by Streicher (WS 17/18) on Kleene's second algebra hint how an enumeration of higher-order continuous *total* functions can be constructed by assuming EPF.

Usage of the Law of Excluded Middle

UNIVERSITÄT DES SAARLANDES

The following lemmas are classical:

Lemma

$\tilde{\prod}_n p \to \tilde{\sum}_n \overline{p} \;\wedge\; \tilde{\sum}_n p \to \tilde{\prod}_n \overline{p}$

Lemma

$\mathcal{S}_Q(P) \to \mathcal{S}_{\overline{Q}}(P)$

Those are used for Post's theorem at exactly two places:

Theorem

$P \in \tilde{\sum}_{n+1} \leftrightarrow \exists Q \in \tilde{\prod}_n \,.\, \mathcal{S}_Q(P)$

Corollary

$P \in \tilde{\sum}_{n+1} \leftrightarrow \exists Q \in \tilde{\sum}_n \,.\, \mathcal{S}_Q(P)$

Forster (2022) investigates the consistency of LEM in CIC when assuming CT due to the lack of the axiom of countable choice.
This is in contrast to the settings of Richman (1983), Bridges and Richman (1987), and Bauer (2005) where LEM is inconsistent.