A Synthetic Definition of the Turing Jump

Second Bachelor Seminar Talk

Niklas Mück

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

Programming Systems Lab, Saarland University

March 14, 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]

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

What if we could solve the Halting Problem?

UNIVERSITÄT DES SAARLANDES

Oracle Machine [Turing (PhD thesis), 1939]

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

Turing jump [Post, 1948][Kleene and Post, 1954]

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

- ☞ Q' is undecidable by oracle machines with an oracle for Q.
- ☞ Repeated jumping gives rise to a hierarchy of unsolvability.

Last time: 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 quantifier prefix and the Turing jump

This time: Outline

- Synthetic Computability
  - Halting Problem
  - Turing Reduction
- My Work
  - Oracle Semi-decidability
  - Turing Jump
  - Formulation of Post’s Theorem

Synthetic Computability

Synthetic Computability

UNIVERSITÄT DES SAARLANDES

Observation

In Coq only computable functions can be defined

- ☞ treat all functions $\mathbb{N} \to \mathbb{N}$ as computable
- ☞ no need to work with a concrete model of computation
- ☞ partial functions $\mathbb{N} \rightharpoonup \mathbb{N}$ instead of diverging Turing machines

Approach by [Richman, 1983] [Bridges-Richman, 1987] [Bauer, 2006]

In constructive type theory by [Forster Kirst Smolka, 2019] [Forster (PhD), 2021]

Synthetic Computability – Halting Problem

“Does a partial function output a value?”

UNIVERSITÄT DES SAARLANDES

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

☞ Gödel encoding cannot be constructed

Axiom: Enumerability of Partial Functions [Richman, 1983][Forster, 2020]

$$\mathsf{EPF} := \Sigma\theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N}).\ \forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists c : \mathbb{N}.\ \theta_c \equiv f$$

$\theta_c\ x \triangleright y \triangleq$ “function with code c terminates on x with output y”

Self-halting problem $\mathcal{K} c := \exists y.\ \theta_c\ c \triangleright y$

Synthetic Computability – Turing Reduction

Problem: All (partial) functions are computable

☞ Functional relations $\mathbb{N} \rightsquigarrow \mathbb{B}$ are the uncomputable counterpart

Turing Reduction [Forster (PhD), 2021] joint work with Kirst, idea by Bauer

- Functional relation transformer: $r : (\mathbb{N} \rightsquigarrow \mathbb{B}) \to (\mathbb{N} \rightsquigarrow \mathbb{B})$
- Computational core: $r' : (\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{B})$
- Computable up to oracle:

$$\forall R\ f.\ f \text{ computes } R \to (r'\ f) \text{ computes } (r\ R)$$

- Continuity (i.e. termination $\to$ only finitely many oracle queries)

My Work

Oracle Semi-decidability

Oracle Machines for semi-decision $\mathbb{M}$

- Halting relation: $M_{\text{halts}} : (\mathbb{N} \rightsquigarrow \mathbb{B}) \to (\mathbb{N} \rightsquigarrow \mathbb{1})$
- Computational core: $M_{\text{core}} : (\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{1})$
- Computable up to oracle:

$$\forall R\ f.\ f \text{ computes } R \to (M_{\text{core}}\ f) \text{ computes } (M_{\text{halts}}\ R)$$

- Continuity (i.e. termination $\to$ only finitely many oracle queries)

P is semi-decidable relative to Q:

$$\mathcal{S}_Q(P) := \exists M : \mathbb{M}.\ \forall x.\ x \in P \leftrightarrow M_{\text{halts}}\ Q\ x$$

Turing Jump – "Halting Problem of Oracle Machines"

UNIVERSITÄT DES SAARLANDES

Lemma ("oracle machines with the same core behave the same")

$$\forall M\ M'.\ M_{\text{core}} = M'_{\text{core}} \to \neg M_{\text{halts}} \to \neg M'_{\text{halts}}$$

☞ enumerating computational cores is sufficient

We assume an enumerator

$$\zeta : \mathbb{N} \to ((\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{1}))$$

Turing Jump

$$Q' := \{c \in \mathbb{N} \mid \exists M : \mathbb{M}.\ M_{\text{core}} = \zeta c \ \land\ M_{\text{halts}}\ Q\ c\}$$

Turing Jump – Results

We assume an enumerator

$$\zeta : \mathbb{N} \to ((\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{1}))$$

Lemma

$$\forall c.\ \exists M : \mathbb{M}\,.\ M_{\textsf{core}} = \zeta c$$

Theorem (Turing jump is oracle semi-decidable)

$$\mathcal{S}_Q(Q')$$

Theorem (Complement of Turing jump is not oracle semi-decidable)

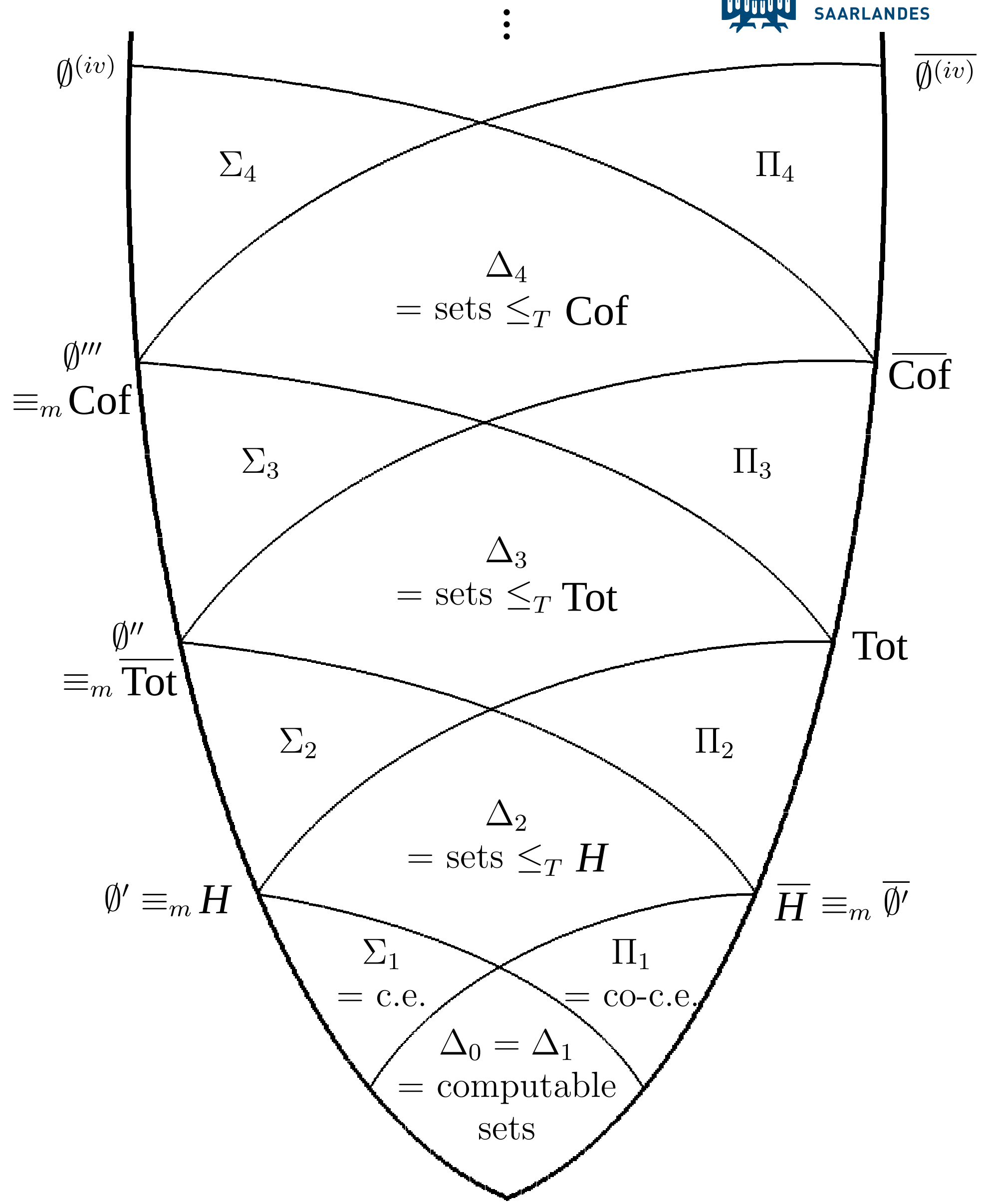
$$\neg\mathcal{S}_Q(\overline{Q'})$$

☞ All proofs in appendix and Coq

Formulation of Post's Theorem

Post's Theorem

- $P \in \sum_{n+1} \leftrightarrow \exists Q.\, \mathcal{S}_Q(P) \ \wedge \ Q \in \prod_n$
- $\emptyset^{(n+1)} \in \sum_{n+1}$
- $P \in \sum_{n+1} \rightarrow P \preceq_m \emptyset^{(n+1)}$
- $P \in \sum_{n+1} \leftrightarrow \mathcal{S}_{\emptyset^{(n)}}(P)$



Overview of my work

- Model arithmetical hierarchy in Coq ✓
  - in Peano arithmetic and in meta-theory ✓
  - prove interesting properties ✓
- Synthetic definition of oracle machines and Turing jump ✓
  - prove interesting results ✓
- Post's theorem ✍
  - formulation ✓
  - proof ✍
- Isolate the weakest set of assumptions ?

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.

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.

References III

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

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

Douglas Bridges, Fred Richman, et al.
Varieties of constructive mathematics, volume 97.
Cambridge University Press, 1987.

References IV

- Andrej Bauer.
  First steps in synthetic computability theory.
  Electronic Notes in Theoretical Computer Science, 155:5–31, 2006.

- 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.

References V

- Yannick Forster.

  Computability in Constructive Type Theory.

  PhD thesis, PhD thesis. Saarland University, 2021.: https://ps.uni-saarland.de/~forster/thesis/phd-thesis-yforster-printblack.pdf.

- 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, 2021.

References VI

Andrej Bauer.
Synthetic mathematics with an excursion into computability theory (slide set).
University of Wisconsin Logic seminar, 2020.
http://math.andrej.com/asset/data/madison-synthetic-computability-talk.pdf.

Synthetic Computability – Turing Reduction

Problem: All (partial) functions are computable

☞ Functional relations $\mathbb{N} \rightsquigarrow \mathbb{B}$ are the uncomputable counterpart

Turing Reduction [Forster (PhD), 2021] joint work with Kirst, idea by Bauer

- Functional relation transformer $r : (\mathbb{N} \rightsquigarrow \mathbb{B}) \to (\mathbb{N} \rightsquigarrow \mathbb{B})$
- Computational core $r' : (\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{B})$
- Computable up to oracle

$$\forall R\, f.\ f \text{ computes } R \to (r'\ f) \text{ computes } (r\ R)$$

- Continuous (i.e. termination $\to$ only finitely many oracle queries)

$$\forall Rx.\ \neg\neg\exists L.\ \forall R'.\ (\forall yb.\ y{\in}L \to R\ y\ b \to R'\ y\ b) \to \forall b.\ r\ R\ x\ b \to r\ R'\ x\ b$$

- Monotonic

$$\forall RR'.(\forall y\, b.\ R\ y\ b \to R'\ y\ b) \to \forall x\, b.\ r\ R\ x\ b \to r\ R'\ x\ b$$

Oracle Semi-decidability

Oracle Machines for semi-decision $\mathbb{M}$

- Halting relation: $M_{\text{halts}} : (\mathbb{N} \rightsquigarrow \mathbb{B}) \to (\mathbb{N} \rightsquigarrow \mathbb{1})$
- Computational core: $M_{\text{core}} : (\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{1})$
- Computable up to oracle:

$$\forall R\ f.\ f \text{ computes } R \to (M_{\text{core}}\ f) \text{ computes } (M_{\text{halts}}\ R)$$

- Continuous (i.e. termination $\to$ only finitely many oracle queries)

$$\forall Rx. \neg\neg\exists L. \forall R'.\ (\forall yb.\ y \in L \to R\ y\ b \to R'\ y\ b) \to M_{\text{halts}} R x \star \to M_{\text{halts}} R' x \star$$

- Monotonic

$$\forall RR'. (\forall y\ b.\ R\ y\ b \to R'\ y\ b) \to \forall x.\ M_{\text{halts}}\ R\ x \star \to M_{\text{halts}}\ R'\ x \star$$

P is semi-decidable relative to Q:

$$\mathcal{S}_Q(P) := \exists M : \mathbb{M}.\ \forall x.\ x \in P \leftrightarrow M_{\text{halts}}\ Q\ x$$

Lemma 1

We assume an enumerator

$$\zeta : \mathbb{N} \to ((\mathbb{N} \rightharpoonup \mathbb{B}) \to (\mathbb{N} \rightharpoonup \mathbb{1}))$$

and ζc continuous:

$$\forall c\ f\ x.\ \exists L.\ \forall f'.\ (\forall y\ b.\ y \in L \to f\ y \triangleright b \to f'\ y \triangleright b) \to \zeta c\ f\ x \to \zeta c\ f'\ x$$

Lemma 1

$$\forall c.\ \exists M : \mathbb{M}\ .\ M_{\text{core}} = \zeta c$$

Proof.

- Choose: $M_{\text{halts}} R\ x := \exists L. \forall f. (\forall y\ b.\ y \in L \to R\ y\ b \to f\ y \triangleright b) \to \zeta c\ f\ x$
- To show: f computes $R \to M_{\text{halts}} R\ x \leftrightarrow \zeta c\ f\ x$

$\to$ easy

$\leftarrow$ needs ζc continuous $\square$

Turing jump is oracle semi-decidable

Theorem (Turing jump is oracle semi-decidable)

$$\mathcal{S}_Q(Q') \equiv \mathcal{S}_Q(\{c \in \mathbb{N} \mid \exists M' : \mathbb{M}.\ M'_{\text{core}} = \zeta c \ \wedge\ M'_{\text{halts}}\ Q\ c\})$$

Proof.

- Need to construct $M : \mathbb{M}$ such that $M_{\text{halts}} Q\ c \leftrightarrow c \in Q'$
- Choose: $M_{\text{halts}} R_o\ c := \exists M'.\ M'_{\text{core}} = \zeta c \wedge M'_{\text{halts}} R_o\ c$
- Choose: $M_{\text{core}}\ f_o\ c := \zeta c\ f_o\ c$
- To show: f computes $R \to \forall c.(\exists M'.\ M'_{\text{core}} = \zeta c \wedge M'_{\text{halts}} R\ c) \leftrightarrow \zeta c\ f\ c$

$\to$ We get M' such that M'_{halts} and $M'_{\text{core}} = \zeta c$. Follows by core spec of M'

$\leftarrow$ Lemma 1 gives us M' with $M'_{\text{core}} = \zeta c$. Core spec gives $M'_{\text{halts}} R\ c$ $\square$

“oracle machines with the same core behave the same”

UNIVERSITÄT DES SAARLANDES

Lemma 2 (“oracle machines with the same core behave the same”)

$$\forall M\ M'.\ M_{\text{core}} = M'_{\text{core}} \to \forall R\ x.\ \neg M_{\text{halts}} R\ x \to \neg M'_{\text{halts}} R\ x$$

Proof.

- Equal cores $\triangleq$ M and M' behave the same when oracle is computable
- Continuity gives a list L where oracle is queried by M' for given R and x
- Claim is stable, we can “classically” construct a function that computes R on L, we call the corresponding computational relation f
- $M'_{\text{halts}} R\ x \overset{cont.}{\to} M'_{\text{halts}} f\ x \xrightarrow{f\ comp.,eq.\ cores} M_{\text{halts}} f\ x \overset{mono.}{\to} M_{\text{halts}} R\ x$ $\square$

Complement of Turing jump is **not** oracle semi-decidable

UNIVERSITÄT DES SAARLANDES

Theorem (Complement of Turing jump is **not** oracle semi-decidable)

$$\neg\mathcal{S}_Q(\overline{Q'}) \equiv \mathcal{S}_Q(\{c \in \mathbb{N} \mid \neg\exists M' : \mathbb{M}.\ M'_{\text{core}} = \zeta c \ \wedge \ M'_{\text{halts}}\ Q\ c\}) \to \bot$$

Proof.

- Assuming $\mathcal{S}_Q(\overline{Q'})$ gives M such that $M_{\text{halts}}\ Q\ c \leftrightarrow c \notin Q'$
- Let c be code of $M_{\text{core}} = \zeta c$
- Showing $c \notin Q' \leftrightarrow \neg M_{\text{halts}} Q\ c$ gives a contradiction

$\to$ Assume $c \notin Q'$ and $M_{\text{halts}} Q\ c$. But $M_{\text{core}} = \zeta c \wedge M_{\text{halts}} Q\ c$ means $c \in Q'$

$\leftarrow$ Assume $\neg M_{\text{halts}} Q\ c$ and $c \in Q' \equiv \exists M' : \mathbb{M}.\ M'_{\text{core}} = \zeta c \ \wedge \ M'_{\text{halts}}\ Q\ c$

But if M doesn't halt, all M' with same core also don't halt by Lemma 2

□