Synthetic Versions of the Kleene-Post and Post's Theorem

Dominik Kirst,[1] Niklas Mück,[1] Yannick Forster[2]

[1] Saarland University, Saarland Informatics Campus
[2] Inria, Gallinette Project-Team, Nantes, France

TYPES'22, June 23rd

Synthetic Computability Theory[1]

Exploit that in constructive foundations, every definable function is computable:

$$A : X \to \mathbb{P} \text{ is decidable} \quad := \quad \exists d : X \to \mathbb{B}.\, \forall x.\, A\,x \leftrightarrow d\,x = \text{true}$$

$$A : X \to \mathbb{P} \text{ many-one-reduces to } B : Y \to \mathbb{P} \quad := \quad \exists r : X \to Y.\, \forall x.\, A\,x \leftrightarrow B\,(r\,x)$$

[1] Richman (1983); Bauer (2006)

Synthetic Computability Theory[1]

Exploit that in constructive foundations, every definable function is computable:

$$
\begin{aligned}
A : X \to \mathbb{P} \text{ is decidable} \quad &:= \quad \exists d : X \to \mathbb{B}.\, \forall x.\, A\,x \leftrightarrow d\,x = \text{true} \\
A : X \to \mathbb{P} \text{ many-one-reduces to } B : Y \to \mathbb{P} \quad &:= \quad \exists r : X \to Y.\, \forall x.\, A\,x \leftrightarrow B\,(r\,x)
\end{aligned}
$$

Pros:

- Avoid manipulating Turing machines or equivalent model of computation
- Elegant formalisation (e.g. in CIC), feasible mechanisation (e.g. in Coq)

[1] Richman (1983); Bauer (2006)

Synthetic Computability Theory[1]

Exploit that in constructive foundations, every definable function is computable:

$$
\begin{aligned}
A : X \to \mathbb{P} \text{ is decidable} \quad &:= \quad \exists d : X \to \mathbb{B}.\, \forall x.\, A\,x \leftrightarrow d\,x = \text{true} \\
A : X \to \mathbb{P} \text{ many-one-reduces to } B : Y \to \mathbb{P} \quad &:= \quad \exists r : X \to Y.\, \forall x.\, A\,x \leftrightarrow B\,(r\,x)
\end{aligned}
$$

Pros:

- Avoid manipulating Turing machines or equivalent model of computation
- Elegant formalisation (e.g. in CIC), feasible mechanisation (e.g. in Coq)

Cons:

- Finding a correct synthetic rendering of Turing reductions not so straightforward
- But Turing reductions are needed for interesting results like Kleene-Post and Post

[1] Richman (1983); Bauer (2006)

Synthetic Oracle Machines

We had some failed attempts, Andrej Bauer's proposal (Bauer (2021)) came to our rescue[2]

[2] See Forster (2021) and the related TYPES abstract Forster and Kirst (2022)

Synthetic Oracle Machines

We had some failed attempts, Andrej Bauer's proposal (Bauer (2021)) came to our rescue[2]

A synthetic oracle machine is an operation on functional relations $\mathbb{N} \to \mathbb{B} \to \mathbb{P}$

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

[2]See Forster (2021) and the related TYPES abstract Forster and Kirst (2022)

Synthetic Oracle Machines

We had some failed attempts, Andrej Bauer's proposal (Bauer (2021)) came to our rescue[2]

A synthetic oracle machine is an operation on functional relations $\mathbb{N} \to \mathbb{B} \to \mathbb{P}$

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

factoring through a computational core on partial functions $\mathbb{N} \rightharpoonup \mathbb{B}$

$$r : (\mathbb{N} \rightharpoonup \mathbb{B}) \to \mathbb{N} \rightharpoonup \mathbb{B} \qquad \text{with} \qquad \forall f : \mathbb{N} \rightharpoonup \mathbb{B}.\, R\, f = r\, f$$

[2] See Forster (2021) and the related TYPES abstract Forster and Kirst (2022)

Synthetic Oracle Machines

We had some failed attempts, Andrej Bauer's proposal (Bauer (2021)) came to our rescue[2]

A synthetic oracle machine is an operation on functional relations $\mathbb{N} \to \mathbb{B} \to \mathbb{P}$

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

factoring through a computational core on partial functions $\mathbb{N} \rightharpoonup \mathbb{B}$

$$r : (\mathbb{N} \rightharpoonup \mathbb{B}) \to \mathbb{N} \rightharpoonup \mathbb{B} \qquad \text{with} \qquad \forall f : \mathbb{N} \rightharpoonup \mathbb{B}.\, R\, f = r\, f$$

satisfying the requirement that R be continuous:

$$R\, A\, n\, b \to \exists L : \mathbb{N}^*.\, L \subseteq \text{dom}(A) \wedge \forall A'.\, A' =_L A \to R\, A'\, n\, b$$

[2]See Forster (2021) and the related TYPES abstract Forster and Kirst (2022)

Synthetic Oracle Machines

We had some failed attempts, Andrej Bauer's proposal (Bauer (2021)) came to our rescue[2]

A synthetic oracle machine is an operation on functional relations $\mathbb{N} \to \mathbb{B} \to \mathbb{P}$

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

factoring through a computational core on partial functions $\mathbb{N} \rightharpoonup \mathbb{B}$

$$r : (\mathbb{N} \rightharpoonup \mathbb{B}) \to \mathbb{N} \rightharpoonup \mathbb{B} \qquad \text{with} \qquad \forall f : \mathbb{N} \rightharpoonup \mathbb{B}.\, R\, f = r\, f$$

satisfying the requirement that R be continuous:

$$R\, A\, n\, b \to \exists L : \mathbb{N}^*.\, L \subseteq \text{dom}(A) \wedge \forall A'.\, A' =_L A \to R\, A'\, n\, b$$

Axiom: there is an enumeration R_n of oracle machines

[2]See Forster (2021) and the related TYPES abstract Forster and Kirst (2022)

Synthetic Oracle Machines

We had some failed attempts, Andrej Bauer's proposal (Bauer (2021)) came to our rescue[2]

A synthetic oracle machine is an operation on functional relations $\mathbb{N} \to \mathbb{B} \to \mathbb{P}$

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

factoring through a computational core on partial functions $\mathbb{N} \rightharpoonup \mathbb{B}$

$$r : (\mathbb{N} \rightharpoonup \mathbb{B}) \to \mathbb{N} \rightharpoonup \mathbb{B} \qquad \text{with} \qquad \forall f : \mathbb{N} \rightharpoonup \mathbb{B}.\, R\, f = r\, f$$

satisfying the requirement that R be continuous:

$$R\, A\, n\, b \to \exists L : \mathbb{N}^*.\, L \subseteq \mathrm{dom}(A) \land \forall A'.\, A' =_L A \to R\, A'\, n\, b$$

Axiom: there is an enumeration R_n of oracle machines

$$A \preceq_T B \;:=\; A \text{ Turing-reduces to } B \text{ if there is an oracle machine } R \text{ with } R\, B = A$$

[2]See Forster (2021) and the related TYPES abstract Forster and Kirst (2022)

The Kleene-Post Theorem (à la Odifreddi (1992))

Goal: construct incomparable Turing degrees $A := \bigcup_{n:\mathbb{N}} \sigma_n$ and $B := \bigcup_{n:\mathbb{N}} \tau_n$

The Kleene-Post Theorem (à la Odifreddi (1992))

Goal: construct incomparable Turing degrees $A := \bigcup_{n:\mathbb{N}} \sigma_n$ and $B := \bigcup_{n:\mathbb{N}} \tau_n$

Characterise σ_n and τ_n inductively by a predicate $n \triangleright (\sigma, \tau)$ with:

The Kleene-Post Theorem (à la Odifreddi (1992))

Goal: construct incomparable Turing degrees $A := \bigcup_{n:\mathbb{N}} \sigma_n$ and $B := \bigcup_{n:\mathbb{N}} \tau_n$

Characterise σ_n and τ_n inductively by a predicate $n \triangleright (\sigma, \tau)$ with:

- If $n \triangleright (\sigma, \tau)$ and $n \triangleright (\sigma', \tau')$, then $\sigma = \sigma'$ and $\tau = \tau'$
- For every n there not not exist σ and τ with $n \triangleright (\sigma, \tau)$

The Kleene-Post Theorem (à la Odifreddi (1992))

Goal: construct incomparable Turing degrees $A := \bigcup_{n:\mathbb{N}} \sigma_n$ and $B := \bigcup_{n:\mathbb{N}} \tau_n$

Characterise σ_n and τ_n inductively by a predicate $n \triangleright (\sigma, \tau)$ with:

- If $n \triangleright (\sigma, \tau)$ and $n \triangleright (\sigma', \tau')$, then $\sigma = \sigma'$ and $\tau = \tau'$
- For every n there not not exist σ and τ with $n \triangleright (\sigma, \tau)$
- If $2n \triangleright (\sigma, \tau)$, then $R_n\, A$ differs from B at position $|\tau|$
- If $2n + 1 \triangleright (\sigma, \tau)$, then $R_n\, B$ differs from A at position $|\sigma|$

The Kleene-Post Theorem (à la Odifreddi (1992))

Goal: construct incomparable Turing degrees $A := \bigcup_{n:\mathbb{N}} \sigma_n$ and $B := \bigcup_{n:\mathbb{N}} \tau_n$

Characterise σ_n and τ_n inductively by a predicate $n \triangleright (\sigma, \tau)$ with:

- If $n \triangleright (\sigma, \tau)$ and $n \triangleright (\sigma', \tau')$, then $\sigma = \sigma'$ and $\tau = \tau'$
- For every n there not not exist σ and τ with $n \triangleright (\sigma, \tau)$
- If $2n \triangleright (\sigma, \tau)$, then $R_n\, A$ differs from B at position $|\tau|$
- If $2n+1 \triangleright (\sigma, \tau)$, then $R_n\, B$ differs from A at position $|\sigma|$

Theorem (Kleene-Post)

There are predicates A and B such that neither $A \preceq_T B$ nor $B \preceq_T A$.

Post's Theorem (à la Odifreddi (1992))

Goal: connect the arithmetical hierarchy with iterated Turing jumps $\emptyset^{(n)}$

Post's Theorem (à la Odifreddi (1992))

Goal: connect the arithmetical hierarchy with iterated Turing jumps $\emptyset^{(n)}$

Represent the arithmetical hierarchy on predicates $p : \mathbb{N}^k \to \mathbb{P}$ inductively:

$$\frac{f : \mathbb{N}^k \to \mathbb{B}}{\Sigma_0(\lambda \vec{x}.\, f\,\vec{x} = \mathsf{true})} \qquad \frac{f : \mathbb{N}^k \to \mathbb{B}}{\Pi_0(\lambda \vec{x}.\, f\,\vec{x} = \mathsf{true})} \qquad \frac{\Pi_n\, p}{\Sigma_{n+1}(\lambda \vec{x}.\, \exists y.\, p\,(y :: \vec{x}))} \qquad \frac{\Sigma_n\, p}{\Pi_{n+1}(\lambda \vec{x}.\, \forall y.\, p\,(y :: \vec{x}))}$$

Post's Theorem (à la Odifreddi (1992))

Goal: connect the arithmetical hierarchy with iterated Turing jumps $\emptyset^{(n)}$

Represent the arithmetical hierarchy on predicates $p : \mathbb{N}^k \to \mathbb{P}$ inductively:

$$\frac{f : \mathbb{N}^k \to \mathbb{B}}{\Sigma_0(\lambda\vec{x}.\, f\,\vec{x} = \mathsf{true})} \qquad \frac{f : \mathbb{N}^k \to \mathbb{B}}{\Pi_0(\lambda\vec{x}.\, f\,\vec{x} = \mathsf{true})} \qquad \frac{\Pi_n\, p}{\Sigma_{n+1}(\lambda\vec{x}.\, \exists y.\, p\,(y :: \vec{x}))} \qquad \frac{\Sigma_n\, p}{\Pi_{n+1}(\lambda\vec{x}.\, \forall y.\, p\,(y :: \vec{x}))}$$

$$\begin{aligned} \text{Turing jump of } A &\quad:=\quad \lambda n.\, R_n\, A\, n\, \mathsf{true} \\ A \text{ is semi-decidable relative to } B &\quad:=\quad \exists R.\, \forall n.\, A\, n \leftrightarrow R\, B\, n\, \mathsf{true} \end{aligned}$$

Post's Theorem (à la Odifreddi (1992))

Goal: connect the arithmetical hierarchy with iterated Turing jumps $\emptyset^{(n)}$

Represent the arithmetical hierarchy on predicates $p : \mathbb{N}^k \to \mathbb{P}$ inductively:

$$\frac{f : \mathbb{N}^k \to \mathbb{B}}{\Sigma_0(\lambda \vec{x}.\, f\, \vec{x} = \mathsf{true})} \qquad \frac{f : \mathbb{N}^k \to \mathbb{B}}{\Pi_0(\lambda \vec{x}.\, f\, \vec{x} = \mathsf{true})} \qquad \frac{\Pi_n\, p}{\Sigma_{n+1}(\lambda \vec{x}.\, \exists y.\, p\,(y :: \vec{x}))} \qquad \frac{\Sigma_n\, p}{\Pi_{n+1}(\lambda \vec{x}.\, \forall y.\, p\,(y :: \vec{x}))}$$

$$\begin{aligned} \text{Turing jump of } A &:= \lambda n.\, R_n\, A\, n\, \mathsf{true} \\ A \text{ is semi-decidable relative to } B &:= \exists R.\, \forall n.\, A\, n \leftrightarrow R\, B\, n\, \mathsf{true} \end{aligned}$$

Theorem (Post)

Assuming LEM ($\forall p.\, p \vee \neg p$), the following can be shown:

- *A predicate A is Σ_{n+1} iff it is semi-decidable relative to $\emptyset^{(n)}$.*
- *If A is Σ_n, then $A \preceq_T \emptyset^{(n)}$. If $n > 0$ already $A \preceq_m \emptyset^{(n)}$ for synthetic many-one reductions.*

Outlook

1. Investigate if the enumeration R_n can be obtained using Church's thesis (Kreisel (1965))
   $\Rightarrow$ Maybe possible using Kleene's second algebra (Kleene (1952))
2. Analyse use of LEM in Post's theorem (though deemed consistent with enumeration R_n)
   $\Rightarrow$ Avoid switching between Σ_n and Π_n via complementation (Akama et al. (2004))
3. Tackle Post's problem regarding an undecidable but enumerable degree below $\emptyset^{(1)}$
   $\Rightarrow$ Following Friedberg (1957) and Mučnik (1956) or Kučera (1986)

Outlook

1. Investigate if the enumeration R_n can be obtained using Church's thesis (Kreisel (1965))
   $\Rightarrow$ Maybe possible using Kleene's second algebra (Kleene (1952))

2. Analyse use of LEM in Post's theorem (though deemed consistent with enumeration R_n)
   $\Rightarrow$ Avoid switching between Σ_n and Π_n via complementation (Akama et al. (2004))

3. Tackle Post's problem regarding an undecidable but enumerable degree below $\emptyset^{(1)}$
   $\Rightarrow$ Following Friedberg (1957) and Mučnik (1956) or Kučera (1986)

Thanks for your attention!

Bibliography I

Akama, Y., Berardi, S., Hayashi, S., and Kohlenbach, U. (2004). An arithmetical hierarchy of the law of excluded middle and related principles. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pages 192–201. IEEE.

Bauer, A. (2006). First steps in synthetic computability theory. *Electronic Notes in Theoretical Computer Science*, 155:5–31.

Bauer, A. (2021). Synthetic mathematics with an excursion into computability theory. University of Wisconsin Logic seminar.

Forster, Y. (2021). *Computability in Constructive Type Theory*. PhD thesis, Saarland University.

Forster, Y. and Kirst, D. (2022). Synthetic Turing reducibility in constructive type theory. 28th International Conference on Types for Proofs and Programs (TYPES 2022).

Friedberg, R. M. (1957). Two recursively enumerable sets of incomparable degrees of unsolvability (solution of post's problem, 1944). *Proceedings of the National Academy of Sciences of the United States of America*, 43(2):236.

Kleene, S. C. (1952). Introduction to metamathematics.

Kreisel, G. (1965). Mathematical logic. *Lectures in modern mathematics*, 3:95–195.

Bibliography II

Kučera, A. (1986). An alternative, priority-free, solution to post's problem. In *International Symposium on Mathematical Foundations of Computer Science*, pages 493–500. Springer.

Mučnik, A. A. (1956). On the unsolvability of the problem of reducibility in the theory of algorithms. In *Dokl. Akad. Nauk SSSR*, volume 108, page 1.

Odifreddi, P. (1992). *Classical recursion theory: The theory of functions and sets of natural numbers*. Elsevier.

Richman, F. (1983). Church's thesis without tears. *The Journal of symbolic logic*, 48(3):797–803.

Backup Kleene-Post

Characterise σ_n and τ_n inductively by $\triangleright : \mathbb{N} \to \mathbb{B}^* \to \mathbb{B}^* \to \mathbb{P}$ with $0 \triangleright (\epsilon, \epsilon)$ and:

$$\frac{2n \triangleright (\sigma, \tau) \quad \sigma' \text{ least extension of } \sigma \text{ with } b = r_n\, \sigma'\, |\tau|}{2n + 1 \triangleright (\sigma', \tau +\!\!+ [\neg b])}$$

$$\frac{2n \triangleright (\sigma, \tau) \quad \neg(\exists \sigma' b.\, \sigma' \geq \sigma \land b = r_n\, \sigma'\, |\tau|)}{2n + 1 \triangleright (\sigma, \tau +\!\!+ [\mathsf{false}])}$$

$$\frac{2n + 1 \triangleright (\sigma, \tau) \quad \tau' \text{ least extension of } \tau \text{ with } b = r_n\, \tau'\, |\sigma|}{2n + 2 \triangleright (\sigma +\!\!+ [\neg b], \tau')}$$

$$\frac{2n + 1 \triangleright (\sigma, \tau) \quad \neg(\exists \tau' b.\, \tau' \geq \tau \land b = r_n\, \tau'\, |\sigma|)}{2n + 2 \triangleright (\sigma +\!\!+ [\mathsf{false}], \tau)}$$

Backup Post

Theorem (Post')

Assuming LEM ($\forall p.\, p \vee \neg p$), the following can be shown:

- *A predicate A is Σ_{n+1} iff it is semi-decidable relative to some B in Π_n.*
- *If A is Σ_n, then $A \preceq_T \emptyset^{(n)}$. If $n > 0$ already $A \preceq_m \emptyset^{(n)}$ for synthetic many-one reductions.*

Lemma

Given an oracle machine R with core r, termination $R\,A\,n\,b$ is equivalent to

$$\exists L_{\mathsf{true}} L_{\mathsf{false}}.\, (\forall n \in L_{\mathsf{true}}.\, A\,b\,\mathsf{true}) \wedge (\forall n \in L_{\mathsf{false}}.\, A\,b\,\mathsf{false}) \wedge r\,(\mathsf{lookup}\,L_{\mathsf{true}}\,L_{\mathsf{false}})\,n = b$$

where $\mathsf{lookup}\,L_{\mathsf{true}}\,L_{\mathsf{false}}\,n$ *returns* true *if* $n \in L_{\mathsf{true}}$, false *if* $n \in L_{\mathsf{false}}$, *and diverges otherwise.*