Post's Problem in Constructive Mathematics

Haoyi Zeng, Yannick Forster, Dominik Kirst, Takako Nemoto

CCC Workshop
Nice, October 2nd, 2024

Framework

Constructive Type Theory

Computational foundation centred around typing judgements $x : X$

Features included in the Calculus of Inductive Constructions (CIC):

- Inductive types: $\mathbb{B}$, $\mathbb{N}$, lists X^*, vectors X^n, ...
- Standard type formers: $X \to Y$, $X \times Y$, $X + Y$, $\forall x.\, F\, x$, $\Sigma x.\, F\, x$
- Propositional universe $\mathbb{P}$ with logical connectives: $\to$, $\land$, $\lor$, $\forall$, $\exists$

Features not included in CIC:

- Choice principles turning total relations $R : X \to Y \to \mathbb{P}$ into functions $f : X \to Y$
- Classical axioms that allow case distinctions of the form $P \lor \neg P$

The Arithmetical Hierarchy and Classical Axioms

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

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

With LEM $:= \forall P : \mathbb{P}.\ P \lor \neg P$ and DNE $:= \forall P : \mathbb{P}.\ \neg\neg P \to P$ we have (Akama et al. (2004)):

$$\begin{tikzcd}
 & \Sigma_n\text{-LEM} \arrow[dl] \arrow[dr] & \\
\Pi_n\text{-LEM} \arrow[r] \arrow[dr] & \Sigma_{n-1}\text{-LEM} \arrow[dl] \arrow[dr] & \Sigma_n\text{-DNE} \arrow[l] \arrow[dl] \\
\Pi_{n-1}\text{-LEM} \arrow[dr] & \Pi_n\text{-DNE} \arrow[r, leftrightarrow] & \Sigma_{n-1}\text{-DNE}, \arrow[dl] \\
 & \Pi_{n-1}\text{-DNE} &
\end{tikzcd}$$

Synthetic Computability[1]

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

$$P : X \to \mathbb{P} \text{ is decidable} \quad := \quad \exists d : X \to \mathbb{B}.\, \forall x.\, P\,x \leftrightarrow d\,x = \mathsf{true}$$
$$P : X \to \mathbb{P} \text{ is semi-decidable} \quad := \quad \exists s : X \to \mathbb{N} \to \mathbb{B}.\, \forall x.\, P\,x \leftrightarrow (\exists n.\, s\,x\,n = \mathsf{true})$$

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
- Some attempts: Bauer (2021); Forster (2021); Forster and Kirst (2022); Mück (2022)

[1] Richman (1983); Bauer (2006); Forster, Kirst and Smolka (2019)

Synthetic Oracle Computability

Definition (Forster, Kirst and Mück (2023))

An oracle computation is a functional $F\colon (Q \to A \to \mathbb{P}) \to I \to O \to \mathbb{P}$ captured by a computation tree $\tau\colon I \to A^* \rightharpoonup Q + O$ and its induced interrogation relation $\tau i \,; R \vdash qs \,; as$ as follows:

$$\frac{}{\sigma \,;\, R \vdash [\,] \,;\, [\,]} \qquad \frac{\sigma \,;\, R \vdash qs \,;\, as \qquad \sigma\, as \downarrow \mathsf{ask}\ q \qquad Rqa}{\sigma \,;\, R \vdash qs \mathbin{+\!\!+} [q] \,;\, as \mathbin{+\!\!+} [a]}$$

$$F\, R\, i\, o \;\leftrightarrow\; \exists qs\, as.\ \tau i \,; R \vdash qs \,; as \wedge \tau\, x\, as \downarrow \mathsf{out}\ o$$

$$P \preceq_T Q \;:=\; \text{there is an oracle computation } F\colon (\mathbb{N} \to \mathbb{B} \to \mathbb{P}) \to \mathbb{N} \to \mathbb{B} \to \mathbb{P} \text{ with } F\,Q = P$$

$$\mathcal{S}_Q(P) \;:=\; \text{there is an oracle computation } F\colon (\mathbb{N} \to \mathbb{B} \to \mathbb{P}) \to \mathbb{N} \to \mathbb{1} \to \mathbb{P} \text{ with } \mathrm{dom}(F\,Q) = P$$

Continuity of Oracle Computations

Our employed notion of sequential continuity is strictly stronger than modulus-continuity:

Lemma (Forster, Kirst and Mück (2023))

1. *Every oracle computation F is modulus-continuous:*

$$F\,R\,i\,o \;\to\; \exists qs \subseteq \mathrm{dom}(R).\, \forall R'.\, (\forall q \in qs.\, \forall a.\ Rqa \leftrightarrow R'qa) \to F\,R'\,i\,o$$

2. *Not every modulus-continuous functional is an oracle computation.*

Proof.

1. From a terminating run $F\,R\,i\,o$ we obtain an interrogation $\tau\,i; R \vdash qs; as$ and can easily show that qs is a modulus of continuity.
2. The modulus-continuous functional $F\,R\,i\,o \;:=\; \exists q.\, R\,q\,\mathsf{true}$ is not an oracle computation as for any computation tree τ we can define a suitably blocking oracle. □

Enumerating Oracle Computations

We need an enumeration of oracle computations for diagonalisations / Turing jump...

To ensure consistency, we start from a standard axiom (Kreisel (1965); Forster (2021)):

$$\mathsf{EPF} := \exists \theta \colon \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N}).\, \forall f \colon \mathbb{N} \rightharpoonup \mathbb{N}.\, \exists e \colon \mathbb{N}.\, \forall x v.\, \theta_e\, x \downarrow v \leftrightarrow f\, x \downarrow v$$

Theorem (Forster et al. (2024))

There is an enumerator of functionals $\Phi \colon \mathbb{N} \to (\mathbb{N} \to \mathbb{B} \to \mathbb{P}) \to \mathbb{N} \to \mathbb{B} \to \mathbb{P}$ *such that*

1. Φ_e *is an oracle computation for all* $e : \mathbb{N}$
2. *Given an oracle computation* F *there is* $e : \mathbb{N}$ *such that* $\forall R x b.\, \Phi_e^R(x) \downarrow b \leftrightarrow F\, R\, x\, b$
3. *The Turing jump* $P'\, x := \Phi_x^P(x) \downarrow \mathsf{true}$ *of* P *is strictly harder than* P
4. *The halting problem* $H := \emptyset'$ *is undecidable*

Post's Problem

Post's Problem

Is there a semi-decidable yet undecidable set S with $H \not\preceq_T S$?

- Left as an open problem by Post (1944)
- Positive solution by Friedberg (1957) and Muchnik (1956)
- Low simple set construction by Lerman and Soare (1980)

- Synthetic proof mechanised in Coq by Zeng et al. (2024), relying on Σ_2-LEM
- Analytic proof given by Nemoto (2024), relying only on Σ_1-LEM / LPO
- Combination yields a synthetic and mechanised proof using LPO

Low Simple Sets and Limit Computability

Definition (Lerman and Soare (1980) and Post (1944))

$P : X \to \mathbb{P}$ is low if $P' \preceq_T H$ and simple if it is semi-decidable and for W_e being the e-th enumerable set we have $W_e \cap P \neq \emptyset$ whenever W_e is infinite.

$\Rightarrow$ Every low simple set is a solution to Post's problem!

Definition (Shoenfield (1959) and Gold (1965))

$P : X \to \mathbb{P}$ is limit-computable if there exists a function $f : X \to \mathbb{N} \to \mathbb{B}$ with

$$Px \;\leftrightarrow\; \exists n. \forall m > n.\ f(x, m) = \text{true} \quad \wedge \quad \neg Px \;\leftrightarrow\; \exists n. \forall m > n.\ f(x, m) = \text{false}.$$

$\Rightarrow$ Limit-computability provides easy way to prove lowness...

Limit Lemma

Lemma (1)

If $\mathcal{S}_Q(P)$ and $\mathcal{S}_Q(\overline{P})$ then $P \preceq_T Q$.

Lemma (2)

Assuming Σ_n-LEM, if P is Σ_{n+1} and Q is Σ_n, then $\mathcal{S}_Q(P)$.

Lemma (Limit Lemma)

Assuming LPO, if P is limit computable, then $P \preceq_T H$.

Proof.

If P is limit computable, then immediately by definition both P and $\overline{P}$ are Σ_2. Moreover, since the halting problem H is Σ_1, Lemma 2 together with LPO yields both $\mathcal{S}_H(P)$ and $\mathcal{S}_H(\overline{P})$. From there we conclude $P \preceq_T H$ with Lemma 1. $\square$

The Priority Method

Fix step function $\gamma : \mathbb{N}^* \to \mathbb{N} \to \mathbb{N} \to \mathbb{P}$, approximate S inductively:

$$\frac{}{0 \rightsquigarrow [\,]} \qquad \frac{n \rightsquigarrow L \quad \gamma_n^L\ x}{n+1 \rightsquigarrow x :: L} \qquad \frac{n \rightsquigarrow L \quad \forall x.\ \neg\ \gamma_n^L\ x}{n+1 \rightsquigarrow L}$$

Depending on properties of γ we obtain for $S\,x := \exists n, L.\ n \rightsquigarrow L \land x \in L$ that:

- γ is computable $\Rightarrow$ S is semi-decidable
- S satisfies $P_e :=$ W_e is infinite $\to W_e \cap S \neq \emptyset$ $\Rightarrow$ S is simple
- S satisfies $N_e := (\exists^\infty n.\ \Phi_e^S(e)[n]\downarrow) \to \Phi_e^S(e)\downarrow$ $\Rightarrow$ S' is limit computable (using LPO)

Wall Functions

Definition

The use function $U_e^P(x)$ approximates the continuity information of the oracle computation $\Phi_e^P(x)$ in a step-indexed way.

Define suitable γ again relative to a wall function ω of same type:

- $\omega_n^L(e) \geq 2 \cdot e \;\Rightarrow\; S$ satisfies the requirements P_e
- $\omega_n^L(e) \geq \max_{e' \leq e} U_{e'}^L(e')[n] \;\Rightarrow\; S$ satisfies the requirements N_e (using LPO)

Theorem

Assuming LPO, a low simple set exists.

Proof.

Choose the wall function $\omega := \max(2 \cdot e, \max_{e' \leq e} U_{e'}^L(e')[n])$. $\square$

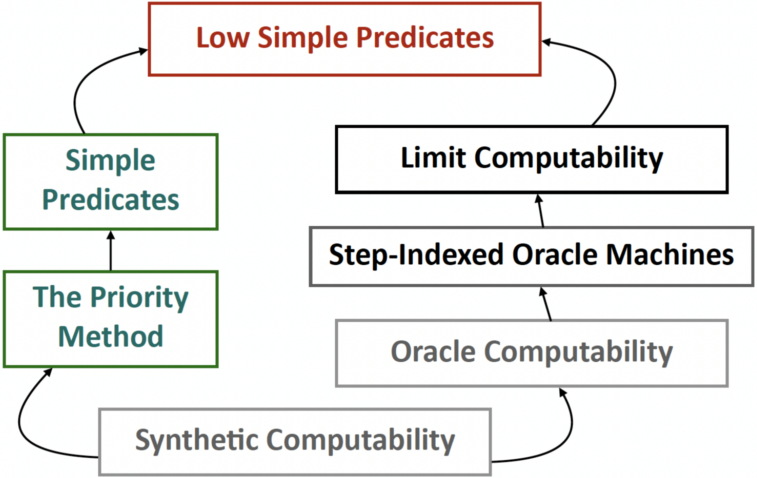
Conclusion

Coq Mechanisation

Lines of code: ~ 1200

Lines of code: ~ 1000

Lines of code: ~ 400



Ongoing Work

Reverse analysis:

- LPO needed for limit lemma?
- LPO needed to show that S' is limit computable?
- LPO needed to construct a low simple set?

Generalisation:

- Friedberg-Muchnik theorem
- Low basis theorem
- Connections to true second-order arithmetic

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).

Forster, Y., Kirst, D., and Mück, N. (2023). Oracle computability and turing reducibility in the calculus of inductive constructions. In *Asian Symposium on Programming Languages and Systems*. Springer.

Forster, Y., Kirst, D., and Mück, N. (2024). The kleene-post and post's theorem in the calculus of inductive constructions. In *32nd EACSL Annual Conference on Computer Science Logic (CSL 2024)*.

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*.

Bibliography II

Friedberg, R. M. (1957). Two recursively enumerable sets of incomparable degrees of unsovlability (solution of Post's problem), 1944. *Proceedings of the National Academy of Sciences*, 43(2):236–238.

Gold, E. M. (1965). Limiting recursion. *The Journal of Symbolic Logic*, 30(1):28–48.

Kreisel, G. (1965). Mathematical logic. In Kreisel, G., editor, *Lectures on Modern Mathematics*, pages 95–195. Wiley.

Lerman, M. and Soare, R. (1980). d-simple sets, small sets, and degree classes. *Pacific Journal of Mathematics*, 87(1):135–155.

Muchnik, 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.

Mück, N. (2022). *The Arithmetical Hierarchy, Oracle Computability, and Post's Theorem in Synthetic Computability*. Bachelor's thesis, Saarland University.

Nemoto, T. (2024). Computability theory over intuitionistic logic. Logic Colloquium 2024, European Summer Meeting of the Association for Symbolic Logic, Gothenburg, Sweden.

Post, E. L. (1944). Recursively enumerable sets of positive integers and their decision problems. *bulletin of the American Mathematical Society*, 50(5):284–316.

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

Bibliography III

Shoenfield, J. R. (1959). On degrees of unsolvability. *Annals of mathematics*, 69(3):644–653.

Zeng, H., Forster, Y., and Kirst, D. (2024). Post's problem and the priority method in cic. In *30th International Conference on Types for Proofs and Programs TYPES 2024–Abstracts*, page 27.