Post's Problem and The Priority Method in Synthetic Computability

Haoyi Zeng

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

Date:16.05.24

Background

Synthetic Computability

Recap

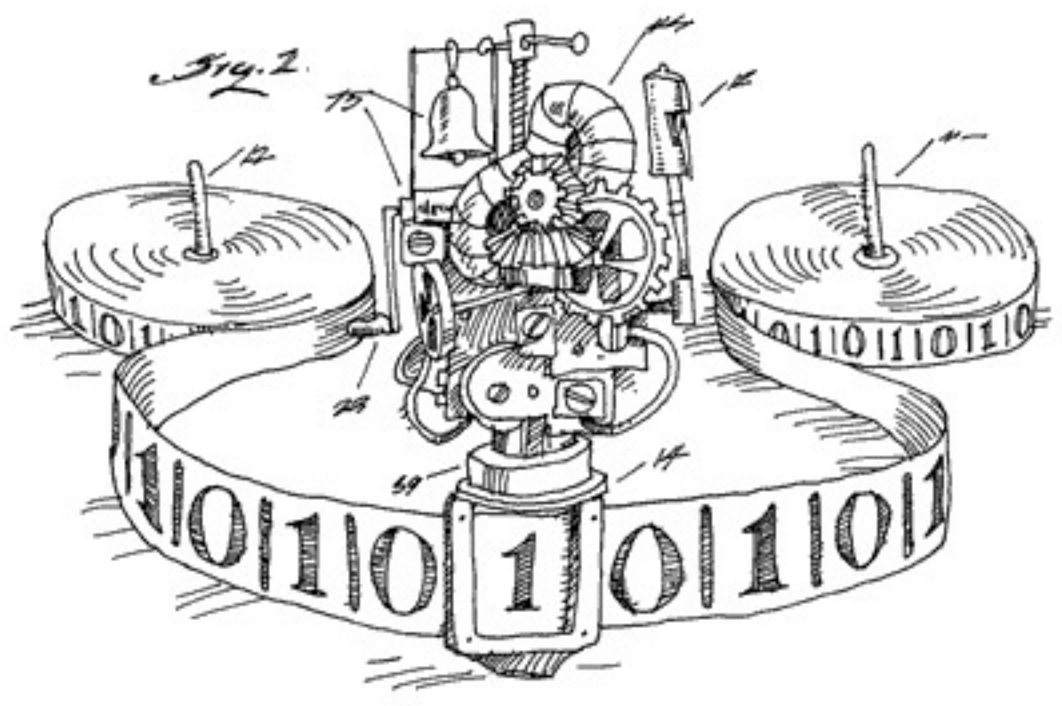
P is **Decidable**: $\exists f : X \to \mathbb{B} \,.\, P\ x \leftrightarrow f\ x = \mathsf{tt}$ $\cancel{\land\ f \text{ is computable}}$

What is computable ?

Turing Machine

λ-Calculus

Synthetic Computability



$$\begin{aligned} \mathsf{fix}\ F &:= \lambda x.\, \mathsf{fix}'\ \mathsf{fix}'\ F\ x \\ \mathsf{fix}' &:= \lambda f, F.\, F\ (\lambda x.\, f\ f\ F\ x) \end{aligned}$$

Synthetic Computability

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

Decidable $\exists\, f : X \to \mathbb{B}\,.\, P\ x \leftrightarrow\ f\ x = \mathsf{tt}$

Semi-decidable $\exists\, f : X \to \mathbb{N} \to \mathbb{B}\,.\, P\ x \leftrightarrow \exists n\,.\, f\ x\ n = \mathsf{tt}$

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

Halting Problem H

$H\ x \leftrightarrow x$-th partial function halts on x

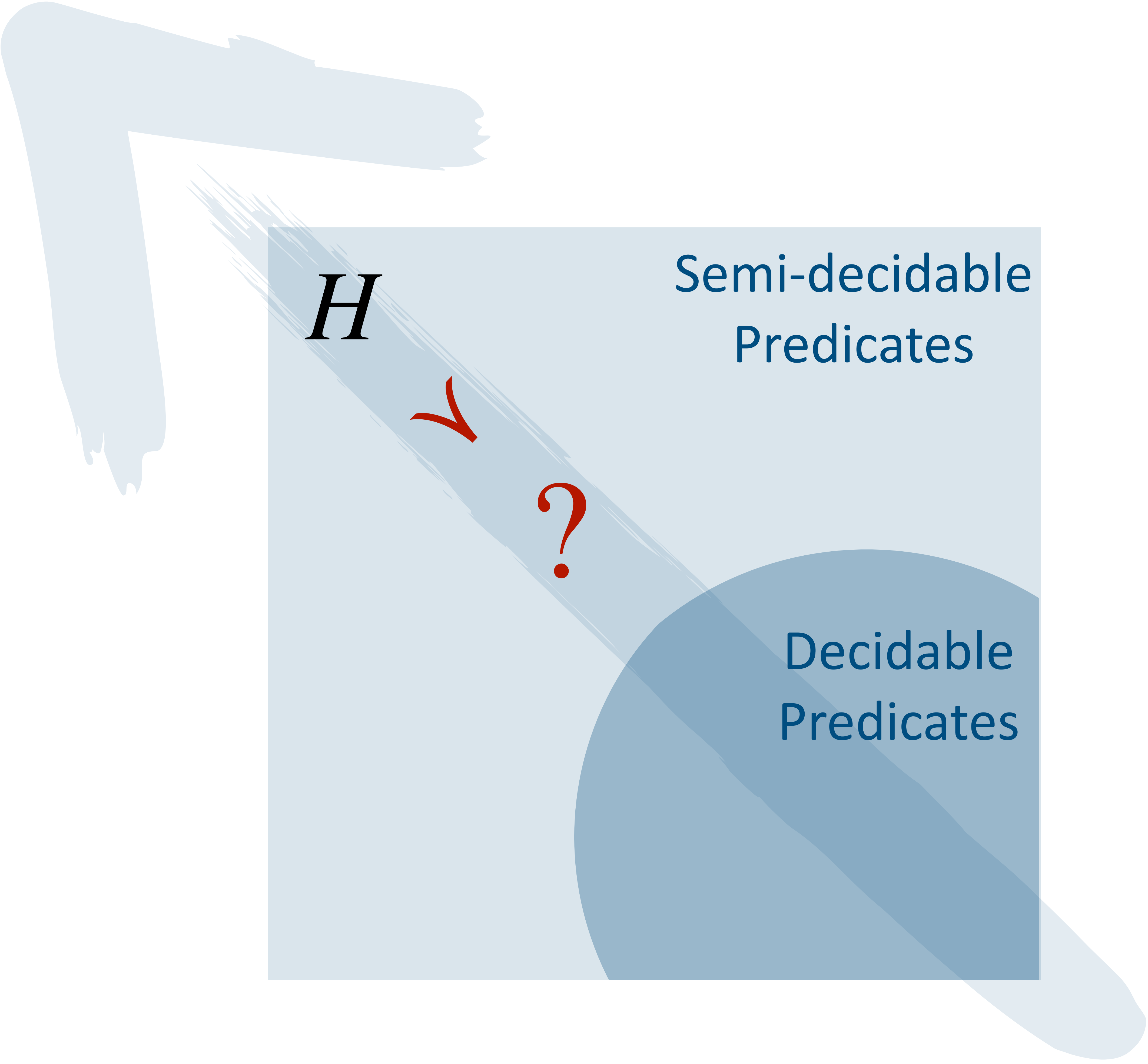
Post's Problem

"Is there an undecidable,

semi-decidable predicate

that is strictly easier than the Halting problem?"

- Post, 1944

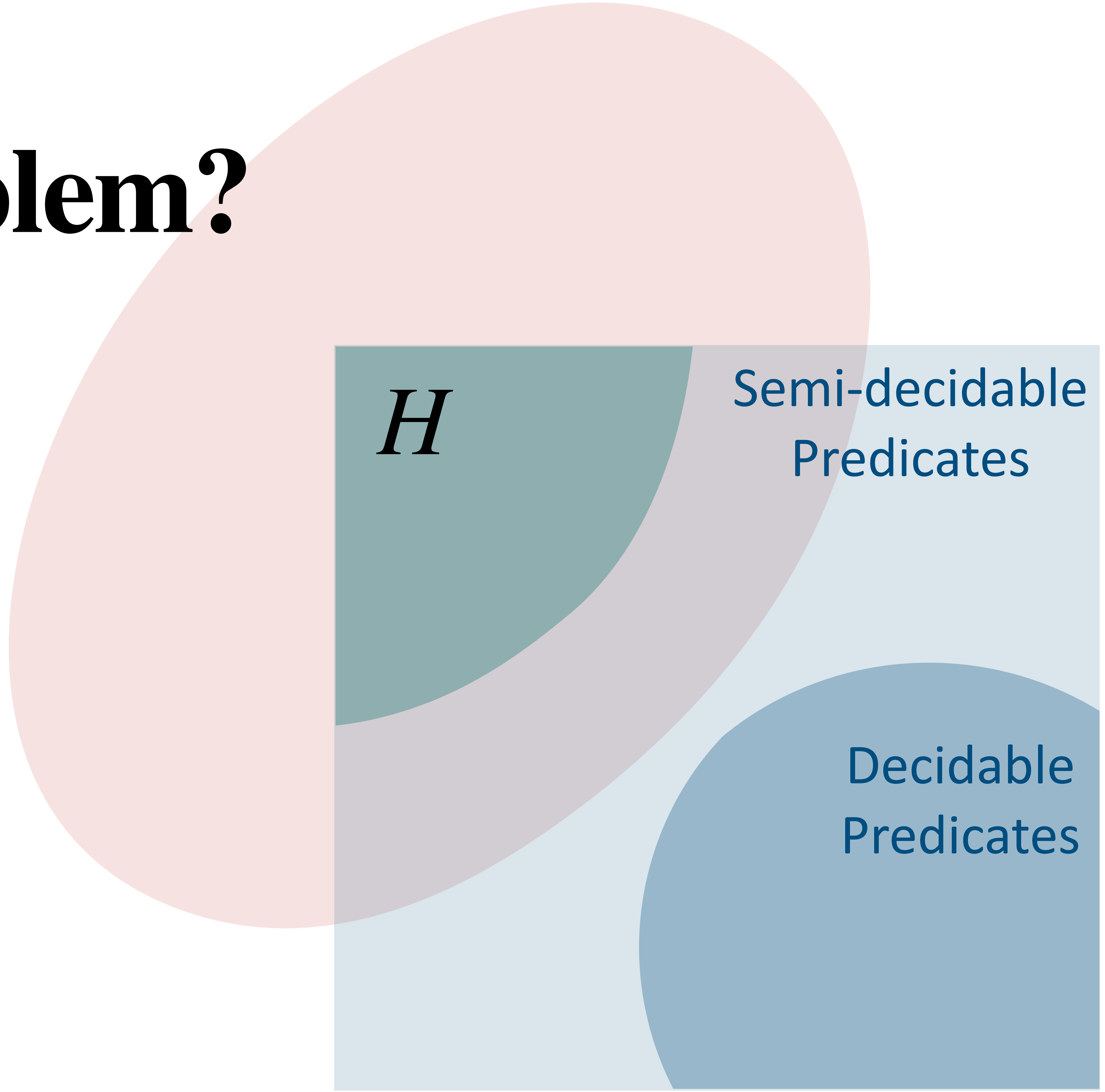


Easier than Halting Problem?

H is reducible to P

Many-one Reduction: $H \preceq_m P$

Turing Reduction: $H \preceq_T P$



Consider reductions in the most general sense, i.e., Turing reduction, which is also the problem Post left open in his paper.

Solutions to Post’s Problem

Finite extension method [Post 1944]

Simple Predicate

… **After 12 years**

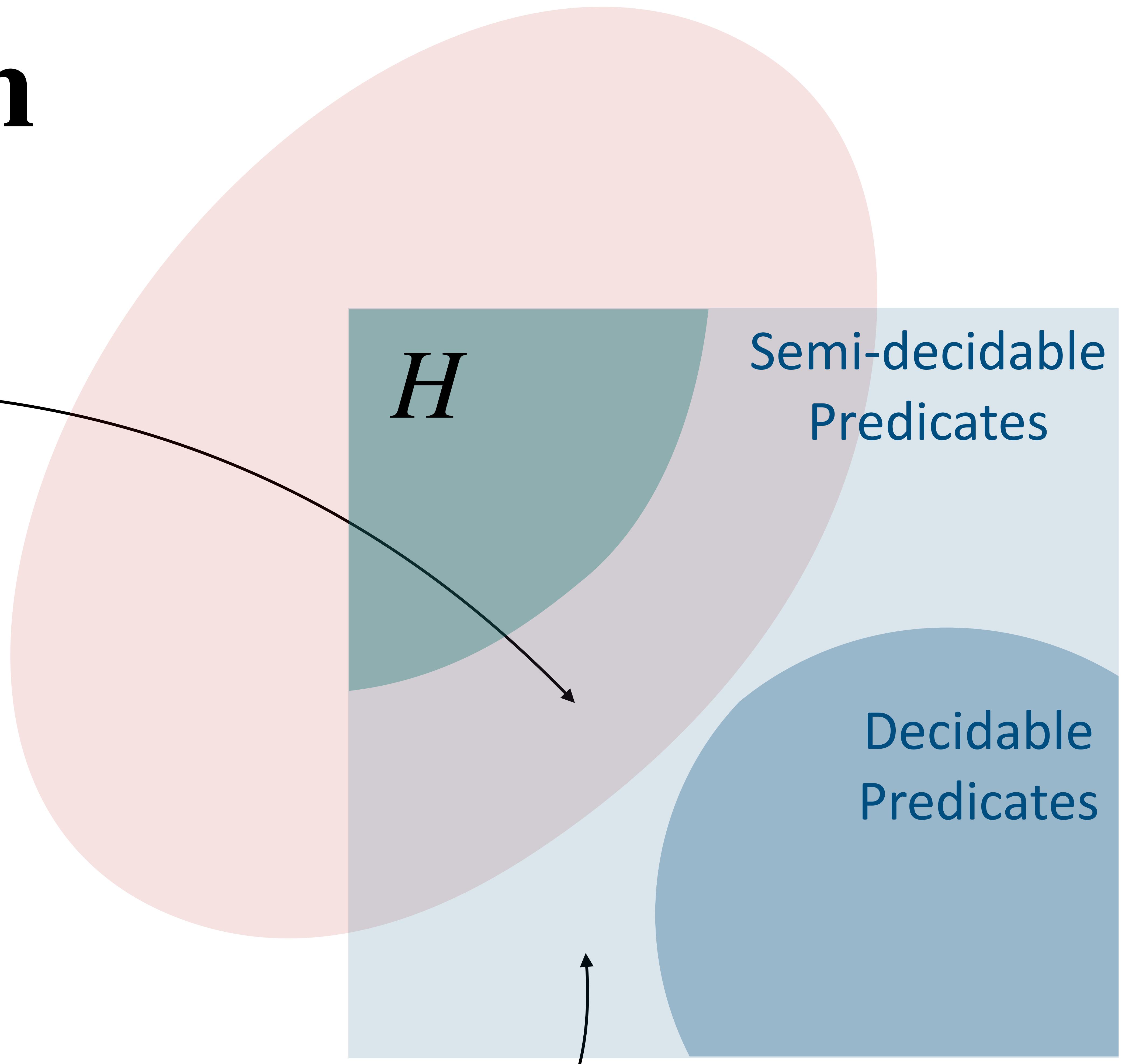
The Priority Method

Friedberg–Muchnik Theorem

[Mučnik 1956] [Friedberg 1957]

Low Simple Predicate

[Lerman & Soare 1980] [Soare 1999]

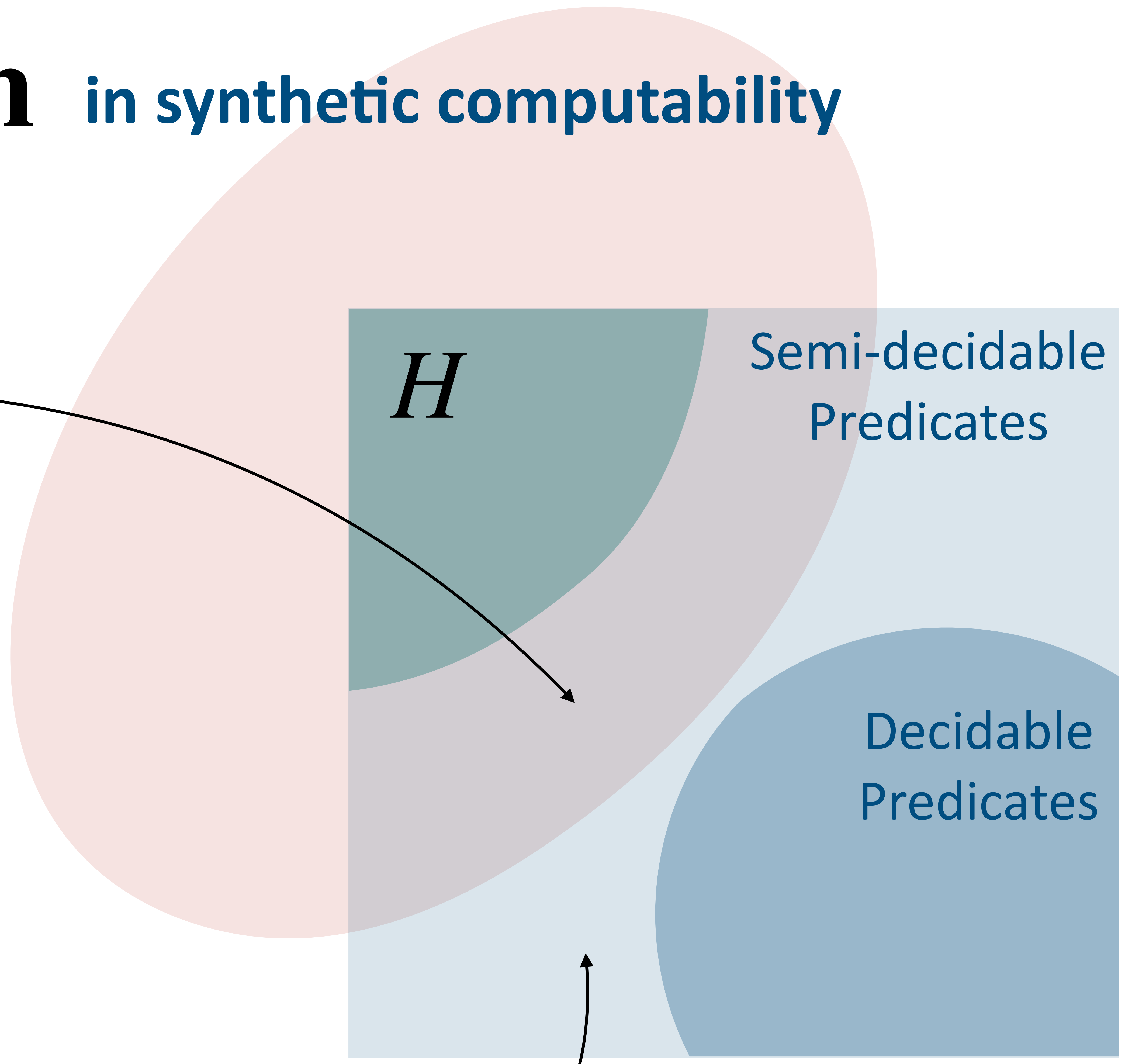


Solutions to Post's Problem in synthetic computability

in synthetic computability

[Forster & Jahn 2023]

Simple Predicate



Another 12 years ?

Low Simple Predicate

The Priority Method

The Priority Method

$$\gamma : \mathbb{N}^* \to \mathbb{N} \to \mathbb{N} \to \mathfrak{P}$$

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

The Priority Method

$$0 \rightsquigarrow \emptyset \qquad \gamma_0^{\emptyset}\ x_0 \qquad 1 \rightsquigarrow [x_0] \qquad \forall x.\ \neg\gamma_1^{[x_0]}\ x \qquad 2 \rightsquigarrow [x_0] \qquad \gamma_2^{[x_0]}\ x_1 \qquad [x_0, x_1] \qquad \dots$$

$$x \in P_\gamma \ := \ \exists n\ L.\ n \rightsquigarrow L \land x \in L$$

Let γ be an extension

γ is computable and unique $\implies$ P_γ is **semi-decidable**

Simple Extension

$$\begin{aligned} \pi_n^L\ e\ x &:= x \in \mathcal{W}_e[n] \land \omega_n^L(e) < x \\ \Pi_n^L\ e &:= L \cap \mathcal{W}_e[n] = \varnothing \land \exists x.\ \pi_n^L\ e\ x \\ \gamma_n^L\ x &:= \exists e.\ e < n \land \mathcal{L}\ e.\ \Pi_n^L \land \mathcal{L}\ x.\ \pi_n^L\ e \end{aligned}$$

$$\omega : \mathbb{N} \to \mathbb{N}^* \to \mathbb{N} \to \mathbb{N}$$

$$P_1 \prec P_2 \prec P_3 \prec P_4 \prec P_5 \prec P_6 \cdots$$

Let ω be a wall

ω is greater than $2e$ and convergent $\Rightarrow$ P_γ is **Simple**

Low Wall

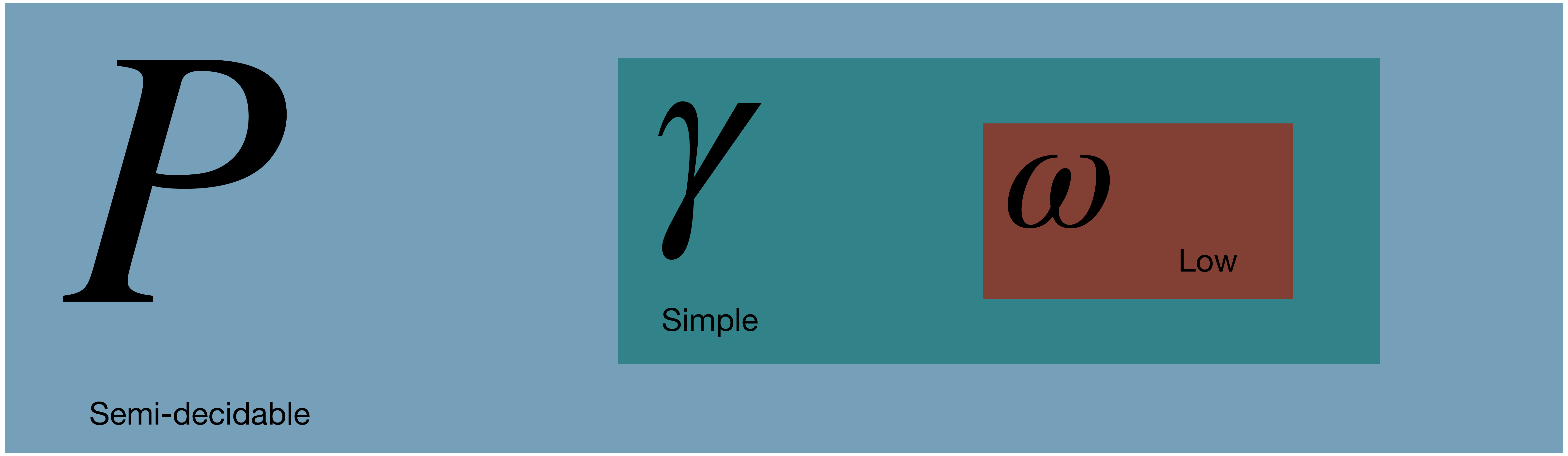
$$\omega_n^L(e) \;:=\; \max(2 \cdot e, \varphi_e^{x \in L}(e)[n]) \qquad N_e \;:=\; \exists^\infty s.\ \Phi_e^A(e)[s]\downarrow \to \Phi_e^A(e)\downarrow$$

$$P_1 \prec N_1 \prec P_2 \prec N_2 \prec P_3 \prec N_3 \cdots$$

By the Limit Lemma $\Rightarrow$ P_{γ_ω} is **Low**

Low Simple Predicate

P_{γ_ω} is **low** **simple** **predicate**



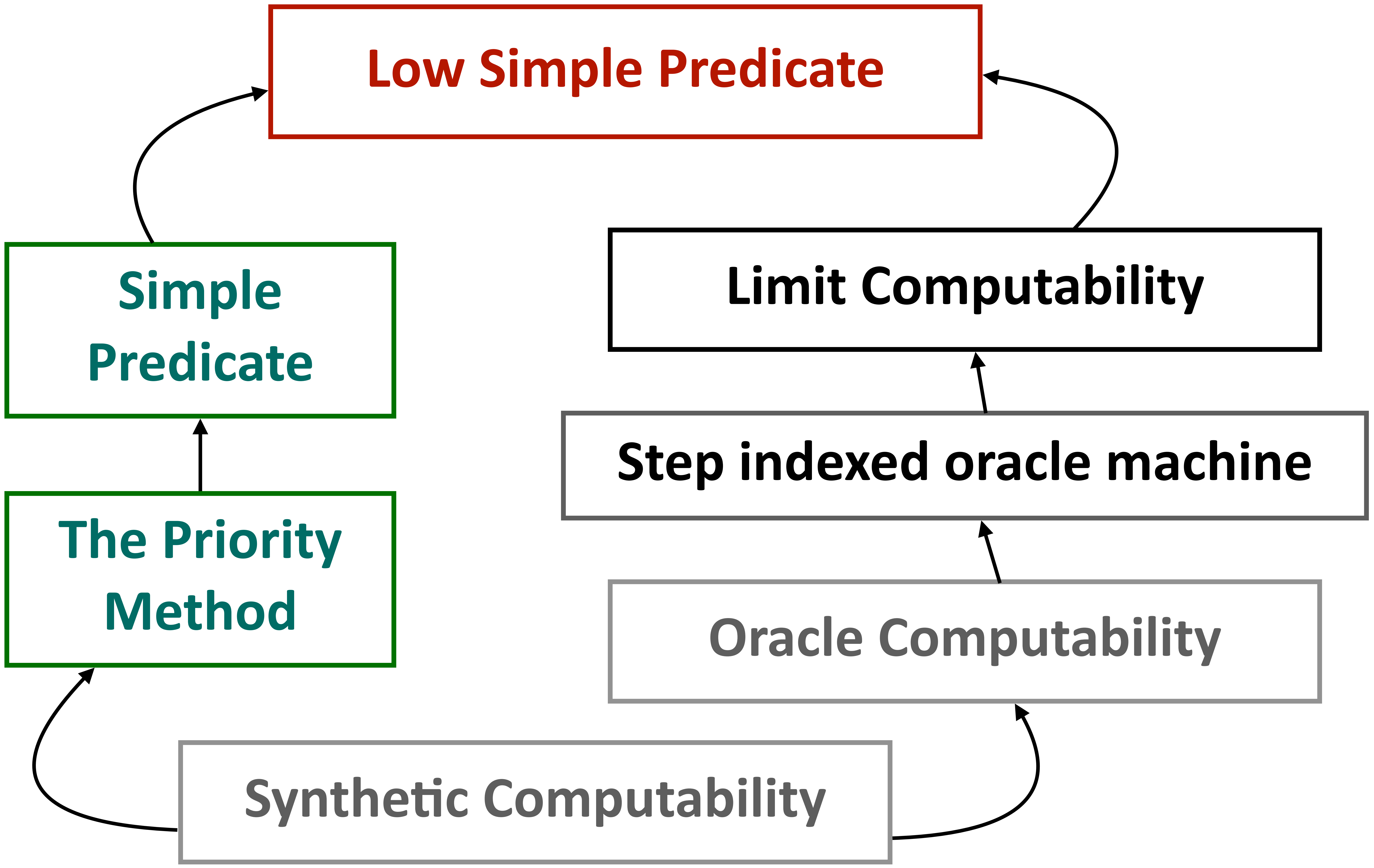
P_{γ_ω} is a positive solution to Post's Problem

Formalisation in Coq

Lines of code*: ~ 900

Lines of code: ~ 1000

Lines of code: ~ 250

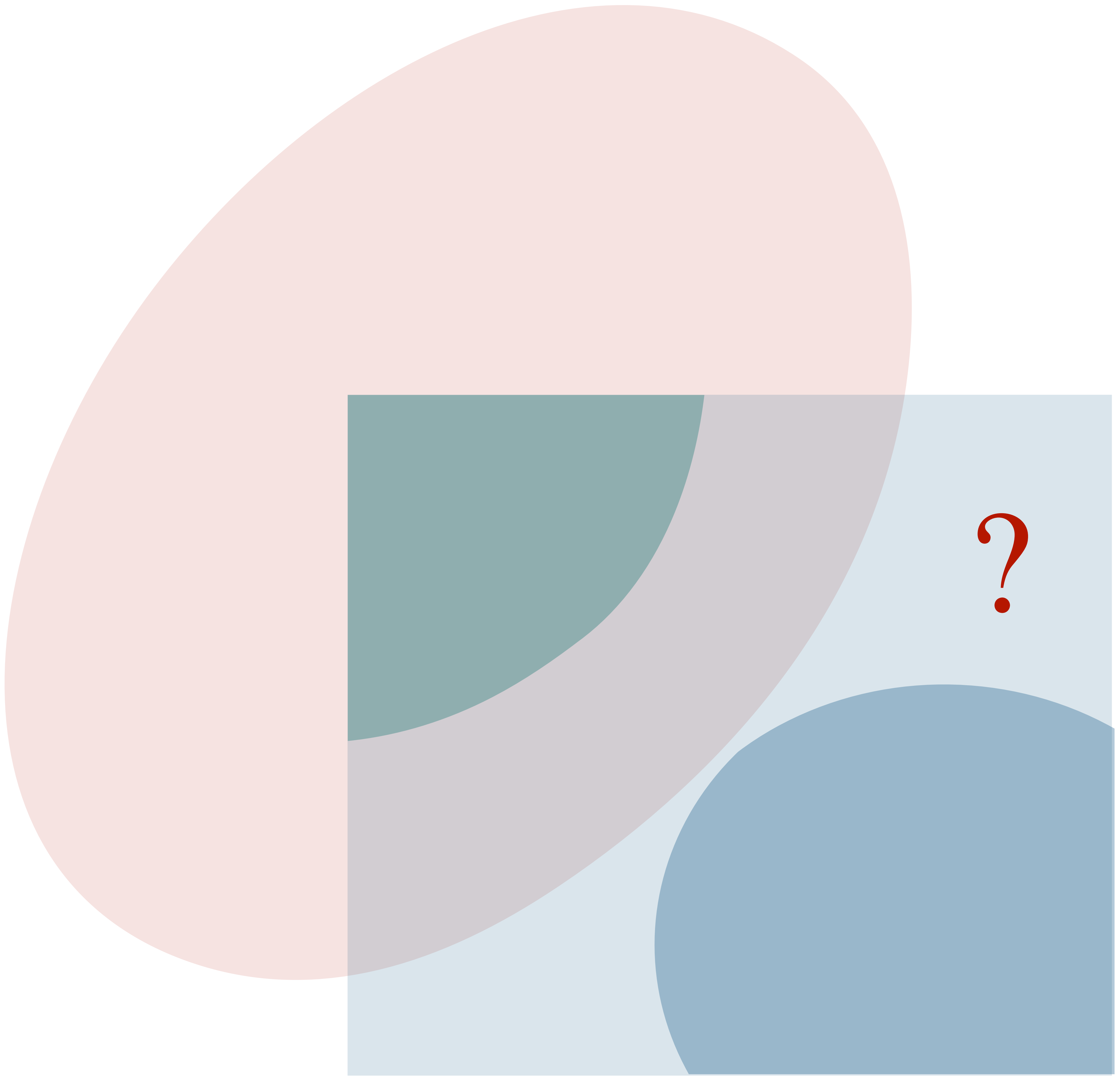


*: a technical lemma is missing

Goals (now)

We aim to show the following theorems and constructions in synthetic computability:

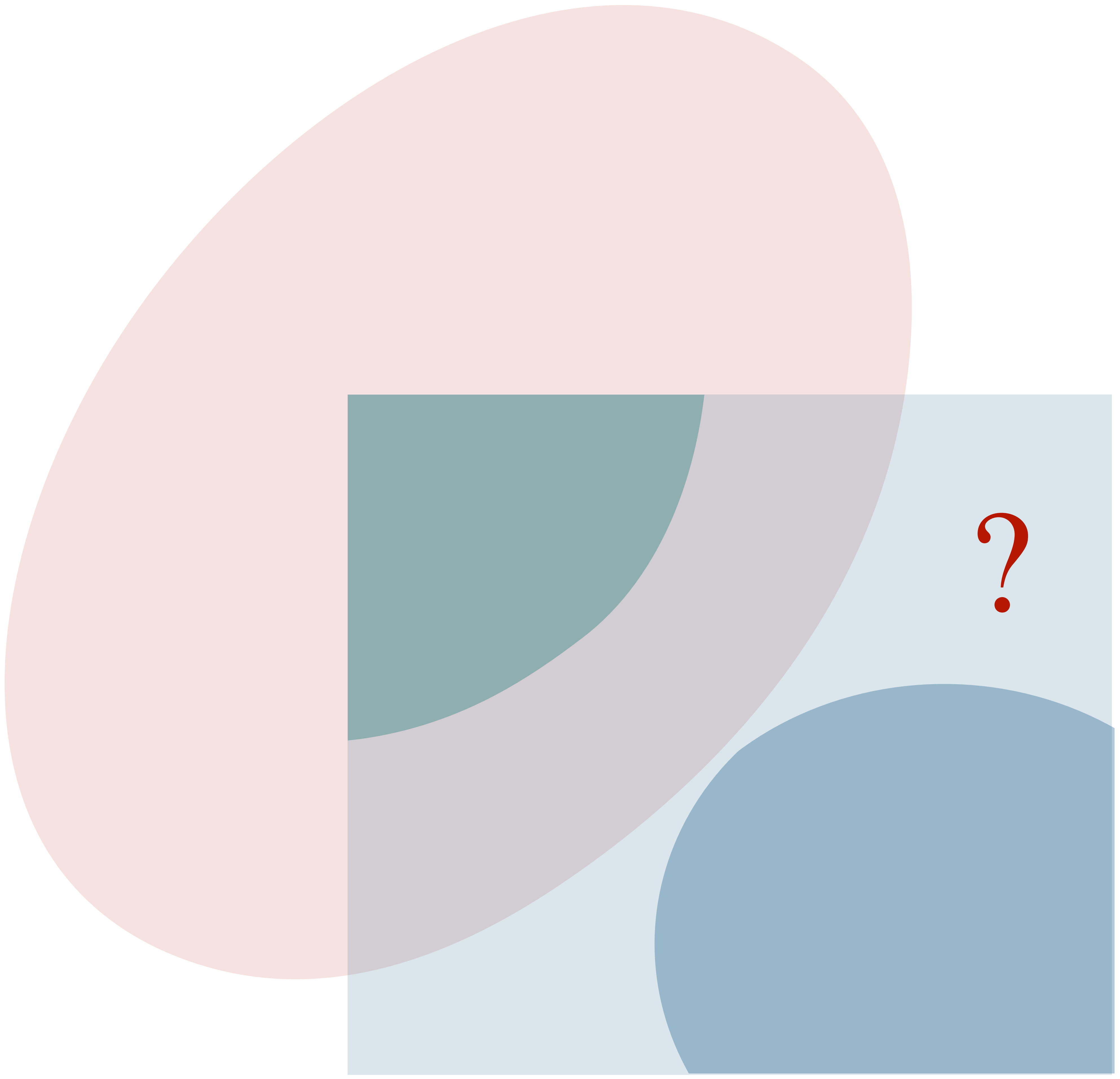
- Definition of limit computable
- Limit lemma
- The priority method
- Definition of low simple predicate
- Existence of low simple predicate
- Friedberg-Muchnik Theorem
- Constructive analysis



Goals (now)

We aim to show the following theorems and constructions in synthetic computability:

- Definition of limit computable
- Limit lemma
- The priority method
- Definition of low simple predicate
- Existence of low simple predicate
- Friedberg-Muchnik Theorem
- Constructive analysis



References

[Forster, Kirst & Mück 2023] Yannick Forster, Dominik Kirst and Niklas Mück. Oracle computability and Turing reducibility in the calculus of inductive constructions. *APLAS 2023*

[Post 1944] Post, E. L. Recursively enumerable sets of positive integers and their decision problems.

[Forster & Jahn 2023] Yannick Forster and Felix Jahn. Constructive and Synthetic Reducibility Degrees: Post's Problem for Many-one and Truth-table Reducibility in Coq. CSL 2023

References

[Friedberg 1957] Richard M. Friedberg. 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. Vol. 43, no. 2. pp. 236–238.

[Lerman & Soare 1980] LERMAN, M., AND SOARE, R. d-simple sets, small sets, and degree classes. Pacific Journal of Mathematics 87, 1 (1980), 135–155.

[Soare 1999] SOARE, R. Recursively enumerable sets and degrees: A study of computable functions and computably generated sets. Springer Science & Business Media, 1999.

References

[Mučnik 1956] Mučnik Albert Abramovich On the unsolvability of the problem of reducibility in the theory of algorithms. Doklady Akademii Nauk SSSR. 108: 194–197.

[Forster, Kirst & Mück 2024] Yannick Forster, Dominik Kirst and Niklas Mück. The kleene-post and post’s theorem in the calculus of inductive constructions CSL

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

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

Appendix A

Oracle Computable

Based on a notion of computability of functionals $F : (Q \to A \to \mathbb{P}) \to (I \to Q \to \mathbb{P})$, The argument $R : Q \to A \to \mathbb{P}$ is to be read as the oracle relating questions $q : Q$ to answers $a : A$, $i : I$ is the input to the computation, and $o : O$ is the output, such an F is considered (oracle)-computable if there is an underlying computation tree $\tau : I \to A^* \rightharpoonup (Q + O)$:

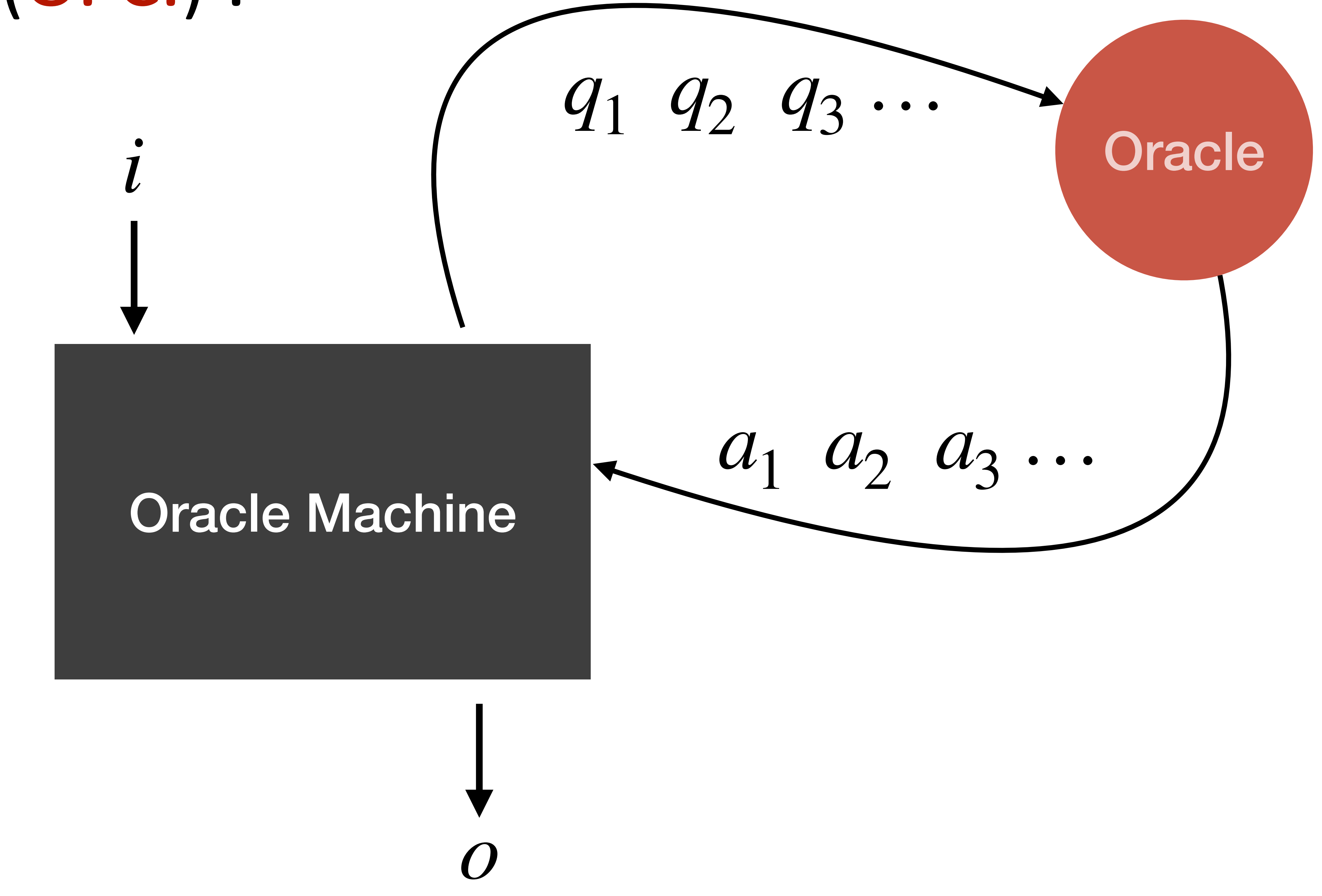
$$\forall R\ x\ b\,.\, F\ R\ x\ b \iff \exists qs\ as\,.\ \tau\ x; R \vdash qs; as \wedge \tau\ x\ as \triangleright\ \mathrm{out}\ b$$

where the interrogation relation $\sigma; R \vdash qs; as$ is inductively defined for $\sigma : A^* \rightharpoonup Q + O$ as:

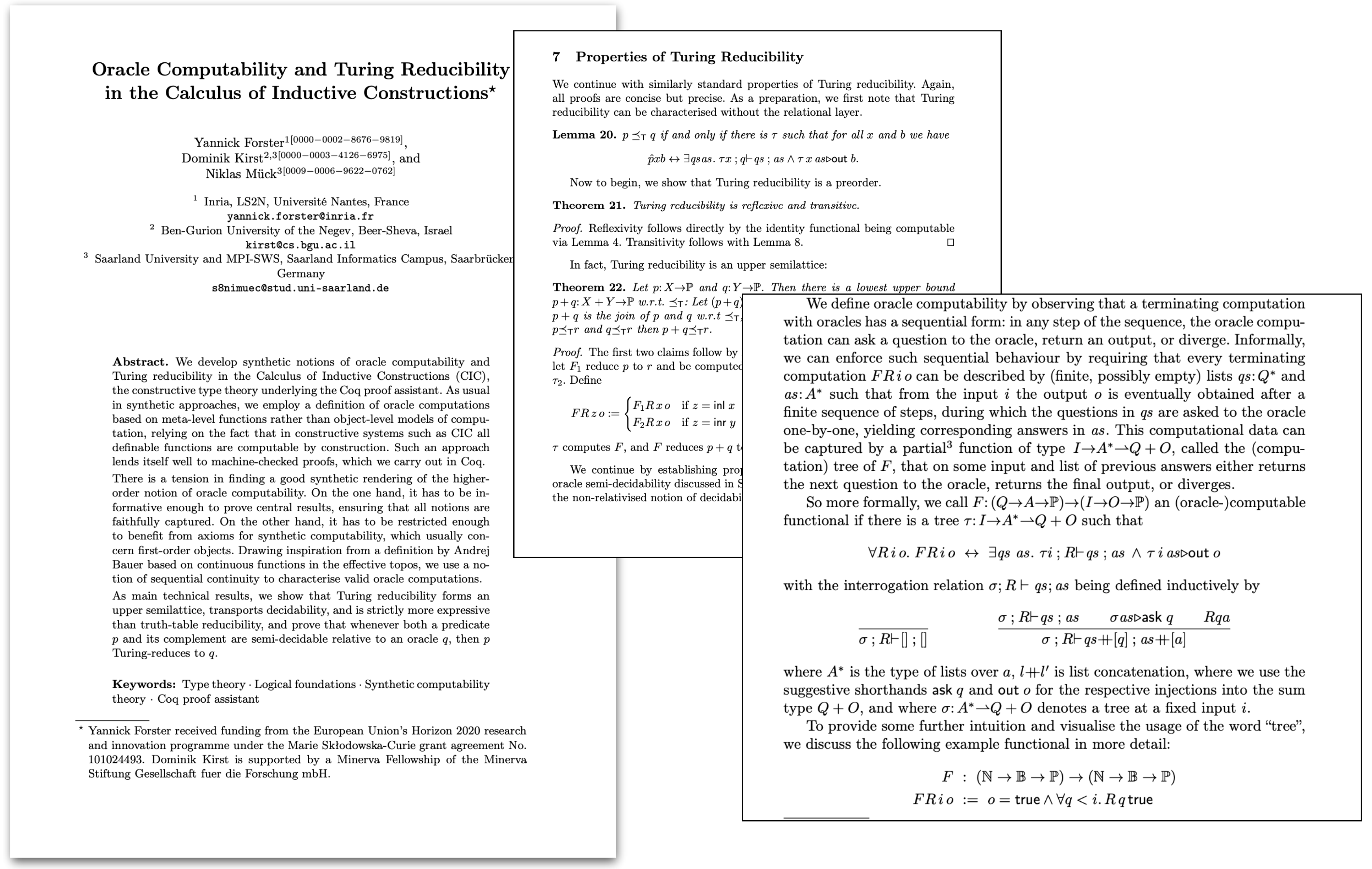
$$\frac{}{\sigma\ ; R \vdash [\,]; [\,]} \qquad \frac{\sigma\ ; R \vdash qs; as \quad \sigma\ as \triangleright \mathrm{ask}\ q \quad R(q, a)}{\sigma\ ; R \vdash qs @ [q]; as @ [a]}$$

Turing reducible in synthetic computability

Synthetic notation of Oracle Computable (O. C.) :



F is O.C. is capturing by some underlying computable object [Forster, Kirst & Mück 2023]



Turing reduction $P \leq Q := \exists F\,.\; F$ is O.C. $\wedge\ \forall x\,.\ \begin{cases} P\,x \leftrightarrow F\,\hat{Q}\,x\ \mathrm{tt} \\ \neg P\,x \leftrightarrow F\,\hat{Q}\,x\ \mathrm{ff} \end{cases}$

Appendix B

Step index function

We insert this oracle O into our Turing machine by fixing a n, and subsequently run τ. Based on this effectively computable oracle, we can define a total function Φ as follows:

$$\Phi_{\tau}^{O(n)}\ x\ i\ j\ :=\ \begin{cases} \ulcorner \mathsf{out}\ o \urcorner & \text{if } (\tau\ x\ []) \rightsquigarrow_j \mathsf{out}\ o \\ \ulcorner \mathsf{ask}\ q \urcorner & \text{if } (\tau\ x\ []) \rightsquigarrow_j \mathsf{ask}\ q \text{ and } i = 0 \\ \Phi_{\tau @ [\chi_O\ n\ q]}^{O(n)}\ x\ i'\ j & \text{if } (\tau\ x\ []) \rightsquigarrow_j \mathsf{ask}\ q \text{ and } i = S\ i' \\ \mathsf{none} & \text{otherwise} \end{cases}$$

Given that P is Turing reducible to $\varnothing$, we obtain the computable tree τ. Building upon the step-index function described above, we define the following function:

$$\chi_P(s, x)\ :=\ \begin{cases} b & \text{if } \Phi_{\tau}^{K}(x)[s] = \ulcorner b \urcorner \\ \mathsf{tt} & \text{otherwise} \end{cases}$$

Low Simple Predicate 1

undecidable predicate

Simple predicate:

If a predicate P is simple, then P is semi-decidable and $\neg(P \preceq \varnothing)$

Turing jump of P:

$P' \ x \leftrightarrow x$-th oracle machine with oracle P halts on x

Low Simple Predicate 2

Low predicate: A predicate P is low, if the Turing jump of P is reducible to K

$$P' \preceq K \Rightarrow \neg(K \preceq P)$$

Low Simple predicate: $\varnothing \prec P \prec K$, where $P \prec Q := P \preceq Q \wedge \neg(P \preceq Q)$

A positive solution to Post's Problem

Showing a predicate is reducible to K is difficult!

Use Function

$$\text{Let } k = \varphi_e^A(e)[n]$$

$$\Phi_e^p(e)[n] = \ulcorner\star\urcorner \to \forall q.\ q \equiv_k p[n] \to \Xi_e\ \hat{q}\ e\ \star$$

$$\varphi_e^p(x)[n] = k \to p[n] \equiv_k p[n+1] \to \varphi_e^p(x)[n+1] = k$$