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:02.01.24

Background

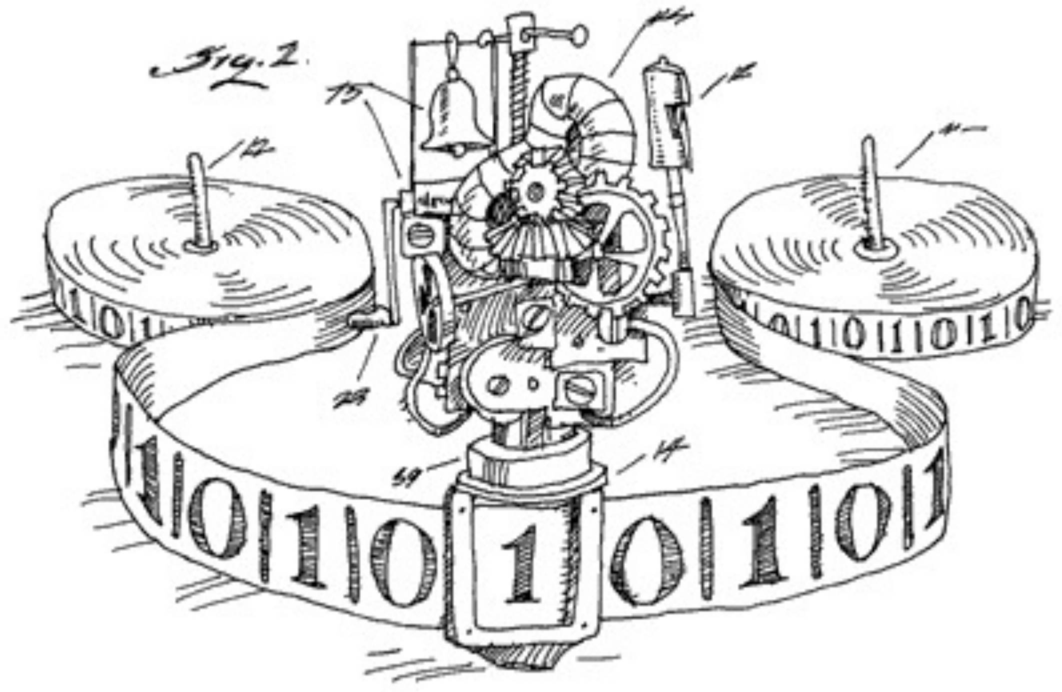
Synthetic Computability

Recap

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

What is computable ?

Turing machine



λ-Calculus

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

Synthetic Computability

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 K

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

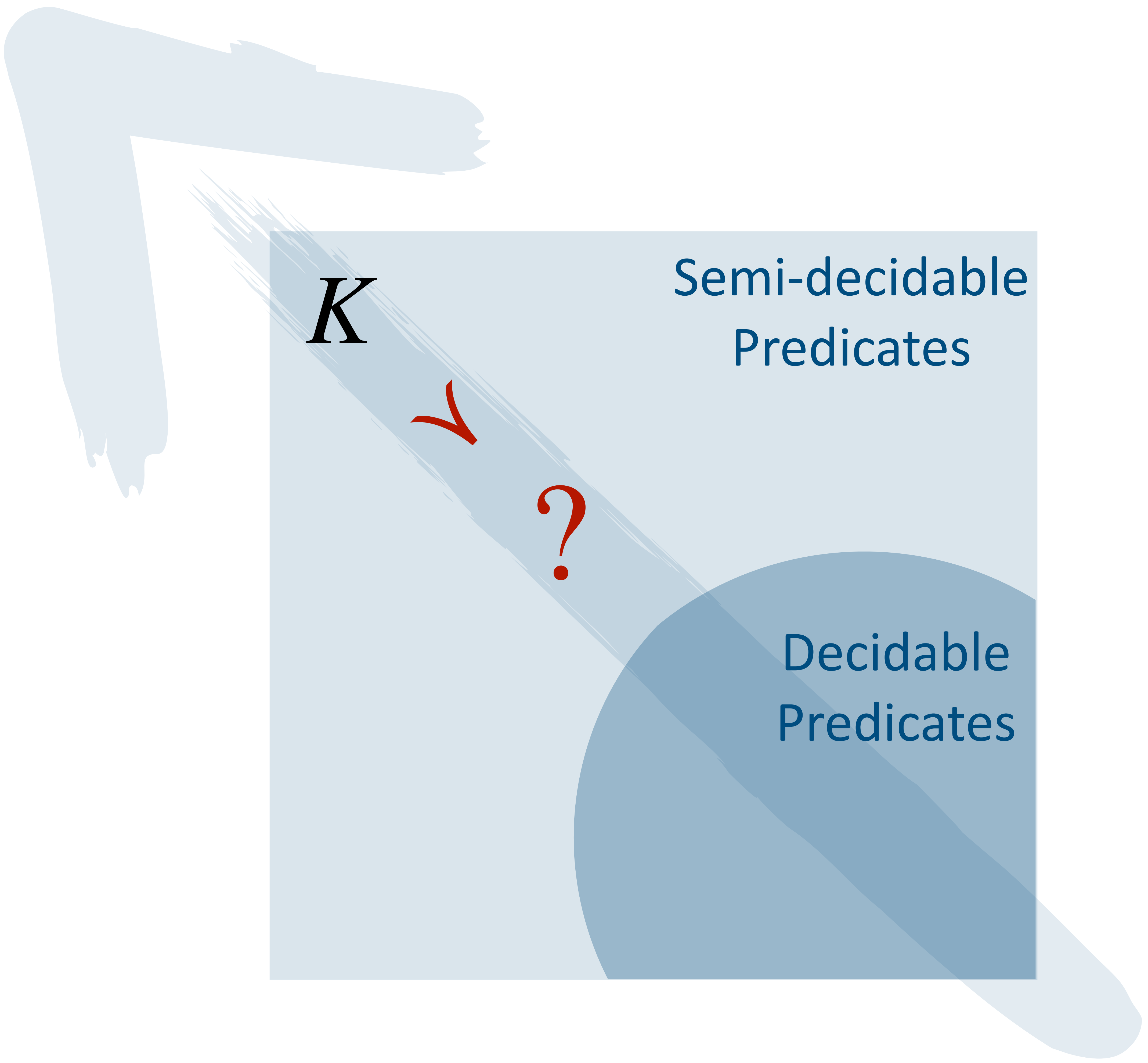
Post’s Problem

“Is there an undecidable,

semi-decidable predicate

that is strictly easier than the Halting problem?”

- Post, 1944

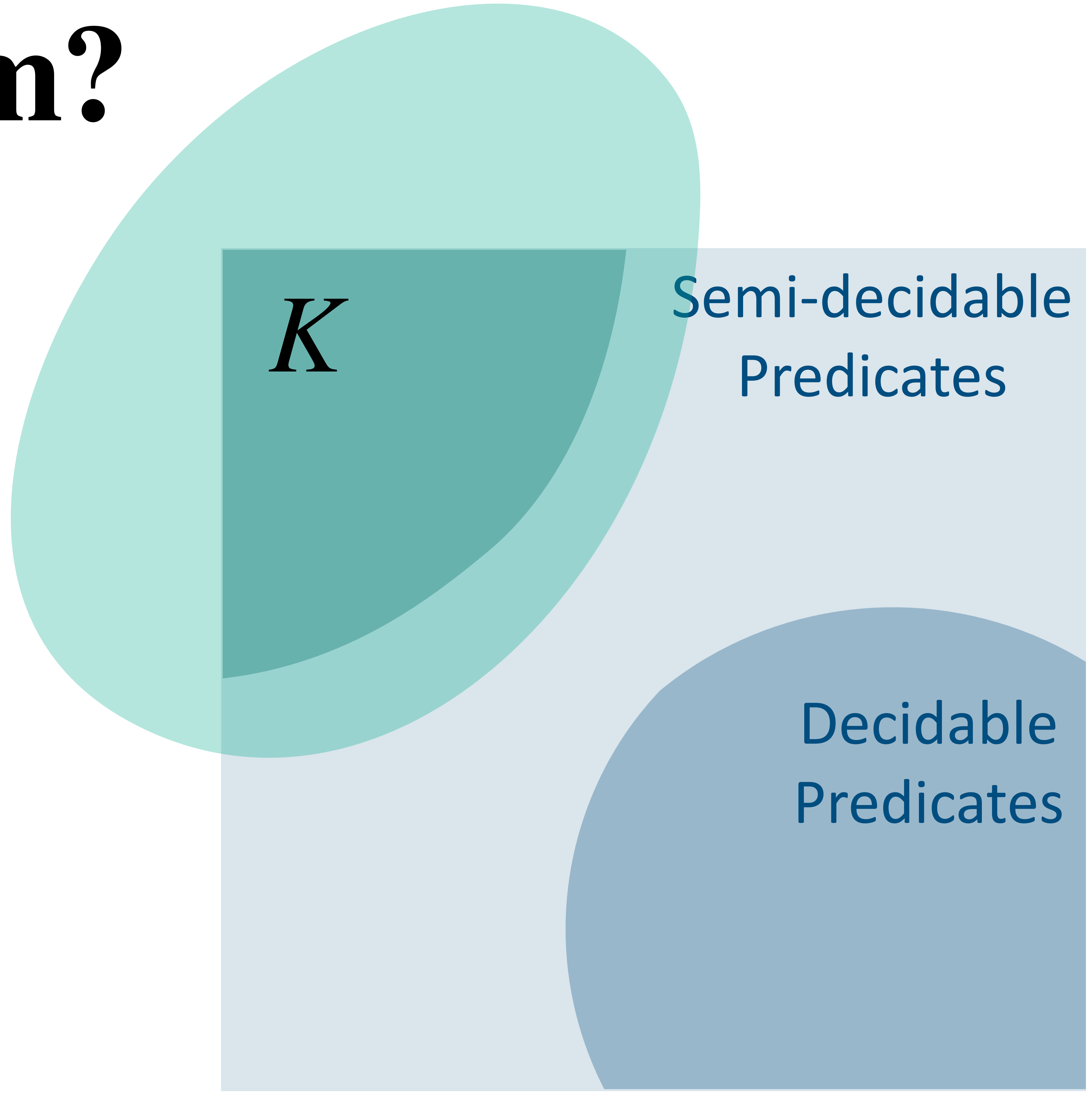


Easier than Halting Problem?

K is reducible to P

Many-one reduction: $K \preceq_m P$

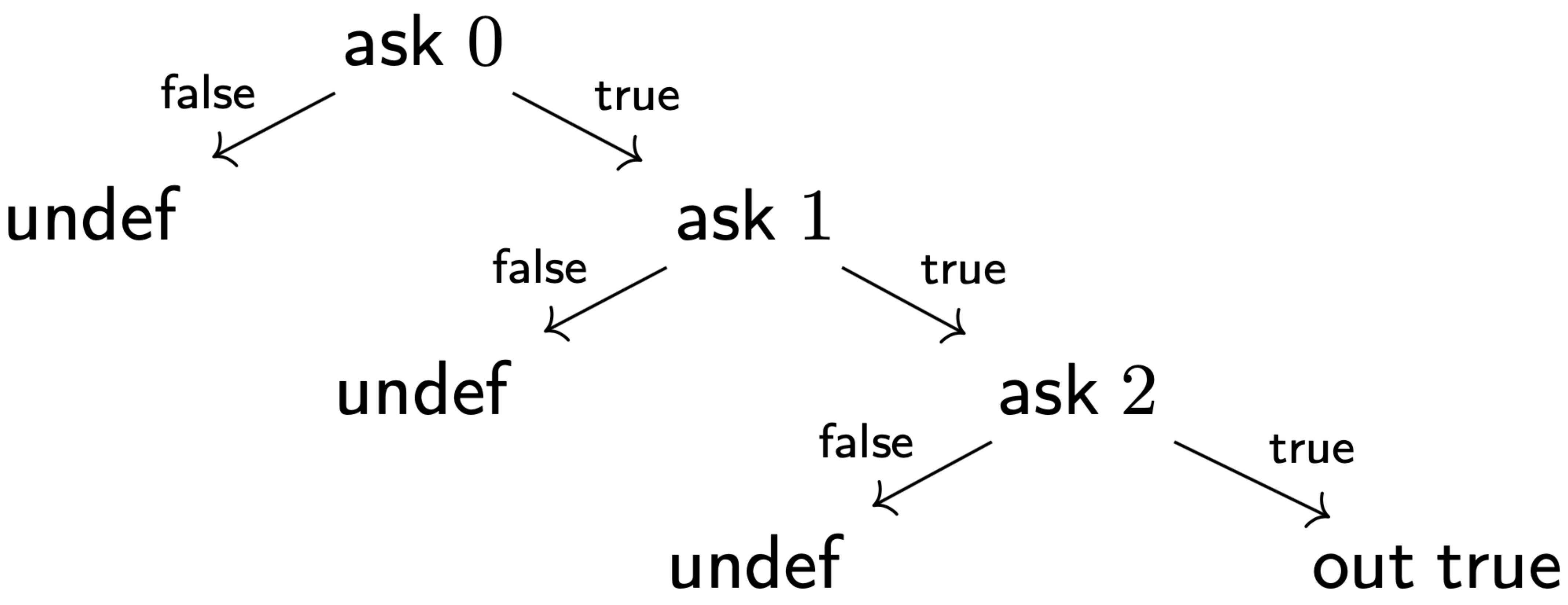
Truth-table reduction: $K \preceq_{tt} P$



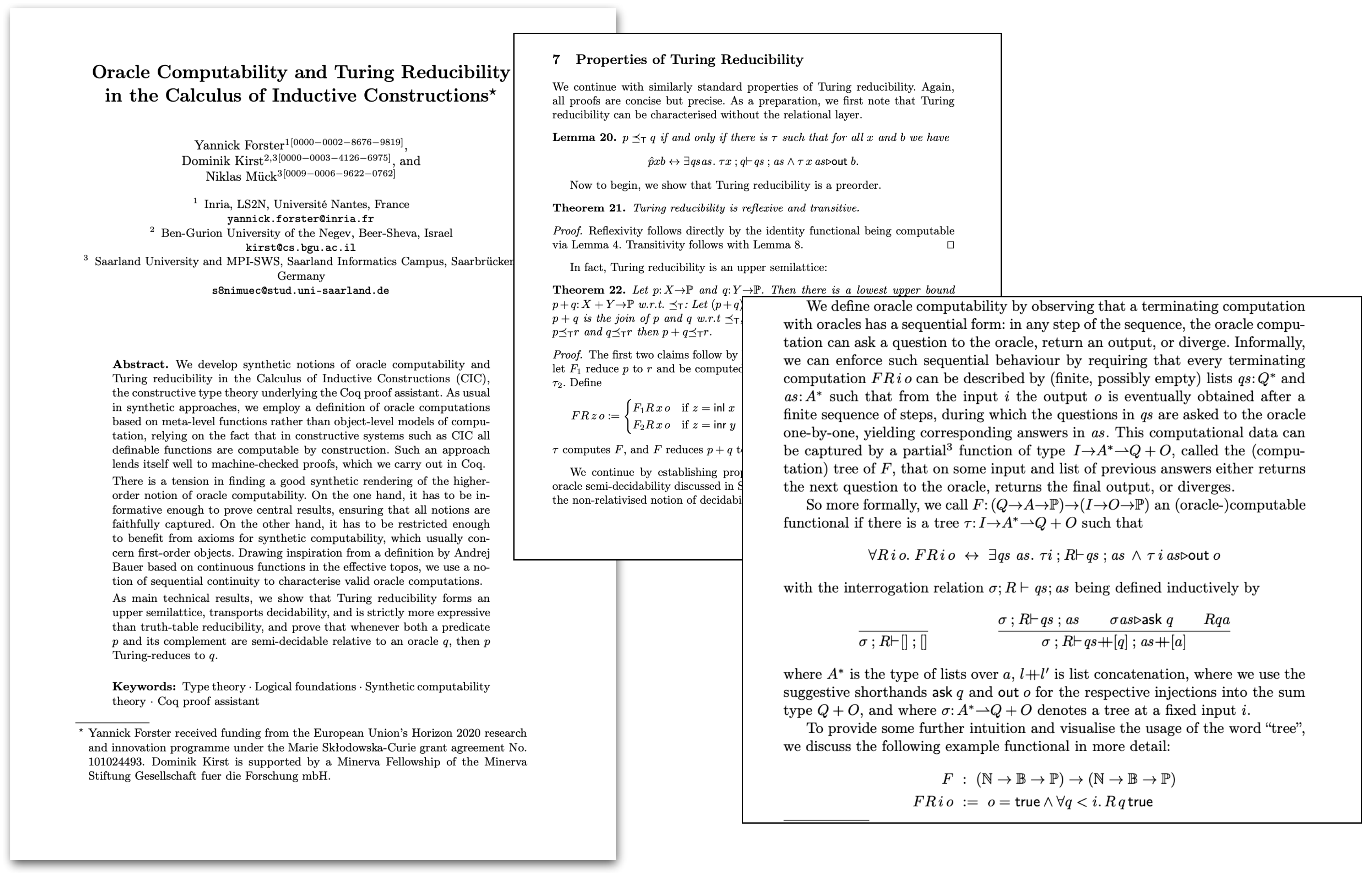
Consider reductions in the most general sense, i.e., Turing reduction, which is also the problem Post left open in his paper.

Turing reducible in synthetic computability

Modelling Oracle Computable (O. C.):



F is O.C. if F described by such computable tree [Forster, Kirst & Mück 2023]

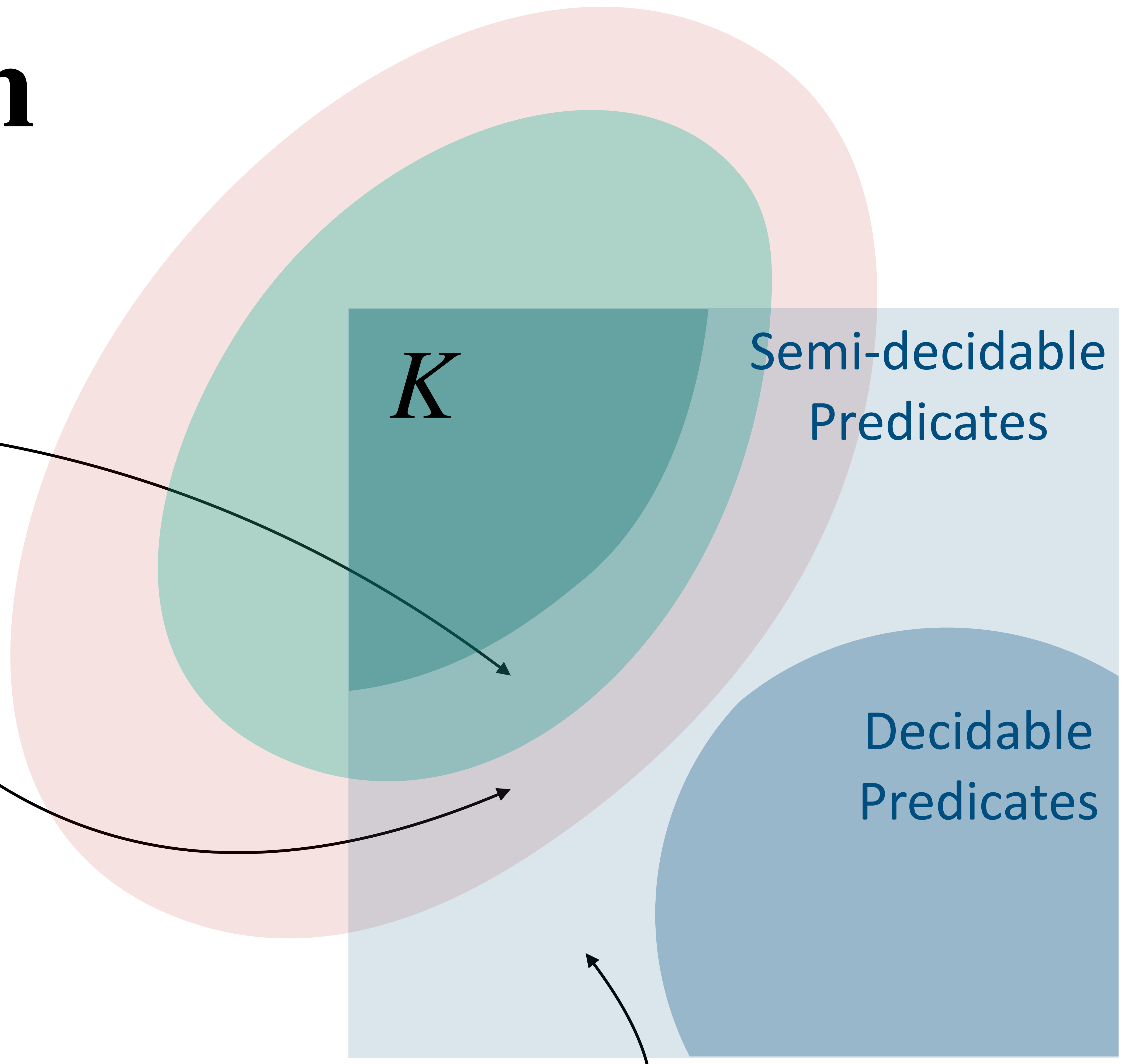


Turing reduction $P \leq Q := \exists F \,.\, F$ is O.C. $\land$ $\forall x \,.\, \begin{array}{l} P\,x \leftrightarrow F\,\hat{Q}\,x\ \mathrm{tt} \\ \neg P\,x \leftrightarrow F\,\hat{Q}\,x\ \mathrm{ff} \end{array}$

Solutions to Post’s Problem

Finite extension method [Post 1944] { Simple Set; Hyper Simple Set

in synthetic computability
[Forster & Jahn 2023]



Priority Method { Friedberg–Muchnik Theorem [Mučnik 1956] [Friedberg 1957]; Low Simple Set [Lerman & Soare 1980] [Soare 1999]

Lowness

“Showing P is reducible to K is” Limit computable difficult!"

Lowness Turing jump of P is reducible to halting problem: $P' \preceq K$

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

Low Simple Set [Lerman & Soare 1980] [Soare 1999]

Current State

Limit Computable in synthetic computability

[Shoenfield 1959] [Gold 1965]

A uniform sequence $f : \mathbb{N} \to Y$ is convergent to some value b if :

$$\lim_{n\to\infty} f(n) = b \quad \text{iff} \quad \exists n : \mathbb{N} \,.\, \forall m \geq N \,.\, f(m) = b$$

A given predicate P is termed **limit computable** when there is a decider $f : X \to \mathbb{N} \to \mathbb{B}$ s.t.

$$\forall x \,.\, \begin{array}{l} P\ x \leftrightarrow \lim\limits_{n\to\infty} f(x,n) = \mathrm{tt} \\ \neg P\ x \leftrightarrow \lim\limits_{n\to\infty} f(x,n) = \mathrm{ff} \end{array}$$

Example

Fix an input x, test whether x is in a limit computable predicate P by executing the function f:

$$n$$

$$f(x,0) \quad f(x,1) \quad f(x,2) \quad f(x,3) \quad \cdots \quad \vdots \quad f(x,n) \quad f(x,n+1) \quad f(x,n+2) \quad f(x,n+3) \quad \cdots$$

It doesn't matter what any of the runs turn out to be, we need to observe the limits

Limit Lemma 1

Lemma 1: If a predicate P is limit computable, then both P and $\bar{P}$ are Σ_2 predicates.

Proof. Rewrite the definition:

$$P\ x \iff \exists n\,.\ \forall m \geq n\,.\ f(x,m) = \mathsf{tt}$$

$$\bar{P}\ x \iff \neg P\ x \iff \exists n\,.\ \forall m \geq n\,.\ f(x,m) = \mathsf{ff}$$

Lemma 2: If both P and $\bar{P}$ are Σ_2 predicates, then P is reducible to K.

Proof. By Post's theorem. [Forster, Kirst & Mück 2024]

Limit Lemma 2

Lemma 3: A predicate P is limit computable, if P is reducible to K.

Proof. Define a step-indexing function: $\Phi_e^K(x)[n] = \Phi_{e,\, n}^{K_n}(x)$,

since $K := \cup_{n \in \mathbb{N}} K_n$ can be approximated by an accumulative sequence.

$P\ x \iff \lim_{n \to \infty} \Phi_e^K(x)[n] = \text{tt}$ $\qquad$ $\neg P\ x \iff \lim_{n \to \infty} \Phi_e^K(x)[n] = \text{ff}$

Corollary: A predicate P is limit computable iff P is reducible to K.

Outlook

Priority Method

Positive Requirements $P_e := W_e$ is infinite $\rightarrow W_e \cap A \neq \emptyset$

Negative Requirements $N_e := (\exists^\infty s\,.\,\Phi^A_{\xi\ e}(e)[s]\downarrow) \rightarrow \Phi^A_{\xi\ e}(e)\downarrow$

Construct a predicate $A := \cup_{n\in\mathbb{N}} A_n$ stage by stage such that:

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

Priority Method

Positive Requirements $P_e := W_e$ is infinite $\rightarrow W_e \cap A \neq \varnothing$

Negative Requirements $N_e := (\exists^\infty s\,.\,\Phi^{A}_{\xi\ e}(e)[s]\downarrow) \rightarrow \Phi^{A}_{\xi\ e}(e)\downarrow$

Construct a predicate $A := \cup_{n\in\mathbb{N}} A_n$ stage by stage such that:

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

Priority Method

Positive Requirements $P_e := W_e$ is infinite $\rightarrow W_e \cap A \neq \varnothing$

Negative Requirements $N_e := (\exists^\infty s\,.\ \Phi^A_{\xi\ e}(e)[s] \downarrow) \rightarrow \Phi^A_{\xi\ e}(e) \downarrow$

Construct a predicate $A := \cup_{n \in \mathbb{N}} A_n$ stage by stage such that:

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

Priority Method

Positive Requirements $P_e := W_e$ is infinite $\rightarrow W_e \cap A \neq \varnothing$

Negative Requirements $N_e := (\exists^\infty s\,.\,\Phi^A_{\xi\ e}(e)[s]\downarrow) \rightarrow \Phi^A_{\xi\ e}(e)\downarrow$

Construct a predicate $A := \cup_{n\in\mathbb{N}} A_n$ stage by stage such that:

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

Priority Method

Positive Requirements $P_e := W_e$ is infinite $\rightarrow W_e \cap A \neq \emptyset$

Negative Requirements $N_e := (\exists^\infty s\,.\,\Phi^{A}_{\xi\ e}(e)[s]\downarrow) \rightarrow \Phi^{A}_{\xi\ e}(e)\downarrow$

Construct a predicate $A := \cup_{n\in\mathbb{N}} A_n$ stage by stage such that:

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

Priority Method

Positive Requirements $P_e := W_e$ is infinite $\rightarrow W_e \cap A \neq \varnothing$

Negative Requirements $N_e := (\exists^\infty s\,.\,\Phi^A_{\xi\,e}(e)[s]\downarrow) \rightarrow \Phi^A_{\xi\,e}(e)\downarrow$

Construct a predicate $A := \cup_{n\in\mathbb{N}} A_n$ stage by stage such that:

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

Priority Method

Positive Requirements $P_e := W_e$ is infinite $\rightarrow W_e \cap A \neq \varnothing$

Negative Requirements $N_e := (\exists^\infty s\,.\,\Phi^A_{\xi\ e}(e)[s]\downarrow) \rightarrow \Phi^A_{\xi\ e}(e)\downarrow$

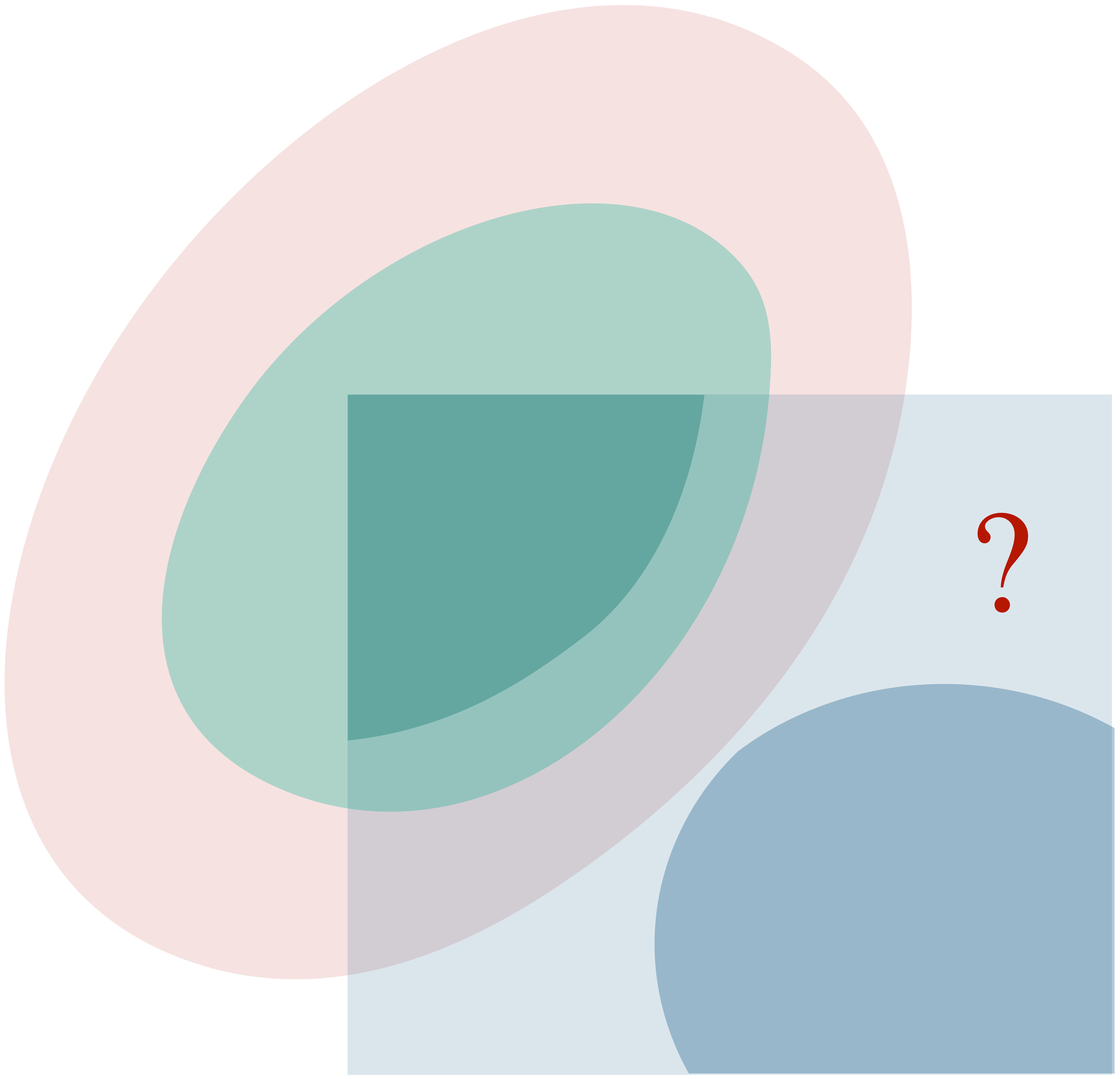
Construct a predicate $A := \cup_{n \in \mathbb{N}} A_n$ stage by stage such that:

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

Goals

We aim to show the following theorem and construction 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 $\tau : I \to A^* \rightharpoonup (Q + O)$:

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

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

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

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 $\emptyset$, 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!