Limits of Computation II

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Is every set $L \subseteq \{0,1\}^*$ semi-decidable?

Again: countably many programs
but uncountably many $L \subseteq \{0,1\}^*$.

So there exist (lots of) algorithmically unsolvable problems.

But can one write down one explicitly?

Diagonal Language

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

$D := \{ \langle P \rangle \mid$ program P does not terminate on input $\langle P \rangle \}$

Suppose D semi-decidable, by program P.
How does P behave on input $\langle P \rangle$?
Case: P terminates on $\langle P \rangle$.
  Then $\langle P \rangle \notin D$ by definition of D.
  But P semi-deciding D means,
  P does not terminate on $\langle P \rangle \notin D$.
Case: P does not terminate on $\langle P \rangle$.
  Then $\langle P \rangle \in D$ by definition of D.
  Similar contradiction.

Diagonal Language

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

$D := \{ \langle P \rangle \mid$ program P does not terminate on input $\langle P \rangle \}$

	P1	P2	P3	P4	P5	
P1	⊥	⊥	↑	↑	⊥	…
P2	⊥	⊥	⊥	⊥	↑	…
P3	⊥	⊥	↑	⊥	↑	…
P4	↑	↑	↑	⊥	↑	…
P5	⊥	⊥	⊥	↑	↑	…
…	…	…	…	…	…	…

Diagonal Language

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

$D := \{ \langle P \rangle \mid \text{program } P \text{ does } \underline{\text{not}} \text{ terminate on input } \langle P \rangle \}$

	P1	P2	P3	P4	P5	
P1	↑	⊥	↑	↑	⊥	…
P2	⊥	↑	⊥	⊥	↑	…
P3	⊥	⊥	⊥	⊥	↑	…
P4	↑	↑	↑	↑	↑	…
P5	⊥	⊥	⊥	↑	⊥	…
…	…	…	…	…	…	…

***D* constructed s.t. each putative program *P* semi-deciding *D* makes an error!**

Very artificial computational problem!

Undecidable Halting Problem

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

$H_z := \{ \langle P \rangle \mid$ program P terminates on fixed input $z\}$

- Very practical problem in algorith. software verification:
  - minimum requirement on an algorithm to be correct!
- Semi-decidable:
  - UTM can simulate given P and terminate iff P does
- but not decidable: complement not semi-decidable:
  - Else semi-decide (non-semidecidable) D as follows:
  - SMN: Given P, construct program $Q=Q(P)$ that, on any input, behaves like P on $\langle P \rangle$.
  - By hypothesis, semi-decide "$\langle Q \rangle \notin H$"; $\Leftrightarrow P \in D$.

$D := \{ \langle P \rangle \mid$ program P does not terminate on input $\langle P \rangle\}$

Rice's Theorem

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Examples: a) syntactical correctness ✓

b) { ⟨P⟩ : ⟨P⟩ is ≤1000 characters long } ✓

c) { ⟨P⟩ : P makes ≤1000 steps on empty input } ✓

d) { ⟨P⟩ : P terminates on the empty input } = *H* ↯

e) { ⟨P⟩ : P semi-decides a non-empty set }

Theorem (Rice-Myhill-Shapiro): Let $\mathbf{S}$ denote a set of subsets of {0,1}* and suppose some $L^+ \in \mathbf{S}$ and some $L^- \notin \mathbf{S}$ are semi-decidable.

Then, given *P*, it is undecidable whether the language semi-decided by *P* belongs to $\mathbf{S}$.

Rice's Theorem

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

d) $H = \{ \langle P \rangle : P \text{ terminates on the empty input} \}$

$\rightarrow \mathbf{S} := \{ L \subseteq \{0,1\}^* : L \text{ contains the empty string} \}$

e) $\{ \langle P \rangle : P \text{ semi-decides a non-empty set} \}$

$\rightarrow \mathbf{S} := \{ L \subseteq \{0,1\}^* : L \text{ is non-empty} \}$

„*Any non-trivial property of the language semidecided by a program is undecidable*“

Theorem (Rice-Myhill-Shapiro): Let $\mathbf{S}$ denote a set of subsets of $\{0,1\}^*$ and suppose some $L^+ \in \mathbf{S}$ and some $L^- \notin \mathbf{S}$ are semi-decidable.

Then, given P, it is undecidable whether the language semi-decided by P belongs to $\mathbf{S}$.