Assignment Project Exam Help

Computational Complexity and Computability

Lecture 8 - Undecidable Problems

https://powcoder.com

Koushik Pal

University of Toronto

Add WeChat powcoder

February 3, 2021

Reducibility

Definition

Let A, B be two languages. We say A is mapping reducible to B, written $A \leq_m B$, if there is a computable function $f : \Sigma^* \to \Sigma^*$ such that for every $w \in \Sigma^*$,

$$w \in A \iff f(w) \in B.$$

The function f is called the reduction from A to B.

The Halting Problem

Definition
$HALT_{TM} := \{\langle M, w\rangle \mid M$ is a TM which halts on $w\}$.

Theorem
$HALT_{TM} \in SD \setminus D$.

Proof.
We will prove the theorem by showing that

- $HALT_{TM} \leq_m A_{TM}$ (hence, $HALT_{TM} \in SD$); and
- $A_{TM} \leq_m HALT_{TM}$ (hence, $HALT_{TM} \notin D$).

This is equivalent to saying $HALT_{TM} \equiv_m A_{TM}$.

The Halting Problem

$HALT_{TM} \leq_m A_{TM}$: Define $f : \Sigma^* \to \Sigma^*$ as follows:

$$\begin{aligned} \langle M, w \rangle &\mapsto \langle M', w \rangle \\ x &\mapsto \langle M_{loop}, \varepsilon \rangle, \end{aligned}$$

where M' accepts w iff M halts on w, and M_{loop} always loops.

Clearly, f is computable and $y \in HALT_{TM} \iff f(y) \in A_{TM}$.

$A_{TM} \leq_m HALT_{TM}$: Define $f : \Sigma^* \to \Sigma^*$ as follows:

$$\begin{aligned} \langle M, w \rangle &\mapsto \langle M', w \rangle \\ x &\mapsto \langle M_{loop}, \varepsilon \rangle, \end{aligned}$$

where M' halts on w iff M accepts w.

Clearly, f is computable and $y \in A_{TM} \iff f(y) \in HALT_{TM}$. $\square$

The Equality Problem

Definition

$EQ_{TM} := \{\langle M_1, M_2\rangle \mid M_1, M_2 \text{ are TMs with } \mathcal{L}(M_1) = \mathcal{L}(M_2)\}.$

Theorem

EQ_{TM} is neither in SD nor in coSD.

Proof.

We will prove the theorem by showing that

- $A_{TM} \leq_m EQ_{TM}$ (hence, $EQ_{TM} \notin coSD$); and
- $A_{TM} \leq_m EQ_{TM}^c$ (hence, $EQ_{TM} \notin SD$).

The Equality Problem

$A_{TM} \leq_m EQ_{TM}$: Define $f : \Sigma^* \to \Sigma^*$ as follows:

$$\begin{aligned} \langle M, w \rangle &\mapsto \langle M_{accept}, M' \rangle \\ x &\mapsto \langle M_{accept}, M_{reject} \rangle \end{aligned}$$

where M_{accept} is a TM that accepts everything, M_{reject} is a TM that rejects everything, and M' is a TM that, on any input, runs M on w and accepts iff M accepts. In other words,

$$\mathcal{L}(M') = \begin{cases} \Sigma^* & \text{if } M \text{ accepts } w \\ \emptyset & \text{otherwise.} \end{cases}$$

Thus,

$$\begin{aligned} \langle M, w \rangle \in A_{TM} &\iff M \text{ accepts } w \\ &\iff \mathcal{L}(M_{accept}) = \mathcal{L}(M') \\ &\iff \langle M_{accept}, M' \rangle \in EQ_{TM}. \end{aligned}$$

The Equality Problem

$A_{TM} \leq_m EQ_{TM}^c$: Define $f : \Sigma^* \to \Sigma^*$ as follows:

$$\begin{aligned} \langle M, w \rangle &\mapsto \langle M_{reject}, M' \rangle \\ x &\mapsto \langle M_{accept}, M_{accept} \rangle, \end{aligned}$$

where M_{accept}, M_{reject} and M' are as before.

Thus,

$$\begin{aligned} \langle M, w \rangle \in A_{TM} &\iff M \text{ accepts } w \\ &\iff \mathcal{L}(M_{reject}) \neq \mathcal{L}(M') \\ &\iff \langle M_{reject}, M' \rangle \notin EQ_{TM} \\ &\iff \langle M_{reject}, M' \rangle \in EQ_{TM}^c. \end{aligned}$$

□

Rice's Theorem

Theorem

Let P be any language consisting of Turing machine descriptions such that

1. *P is nontrivial, i.e., P contains some, but not all, Turing machine descriptions.*
2. *P is a property of the TM's languages, i.e., whenever $\mathcal{L}(M_1) = \mathcal{L}(M_2)$, then $\langle M_1 \rangle \in P \iff \langle M_2 \rangle \in P$.*

Then P is undecidable.

Proof of Rice's Theorem

Proof.

We prove this by showing $A_{TM} \leq_M P$.

Let M_{reject} be a TM that always rejects. We may assume without loss of generality that $M_{reject} \notin P$ (otherwise replace P by P^c).

Since P is nontrivial, there exists a TM T such that $\langle T \rangle \in P$.

Define $f : \Sigma^* \to \Sigma^*$ as follows:

$$\begin{aligned} \langle M, w \rangle &\mapsto M_w \\ x &\mapsto M_{reject}, \end{aligned}$$

where M_w, on input y, rejects if M rejects w, else simulates T on y and outputs whatever T outputs.

Proof of Rice's Theorem

If M accepts w, then $\mathcal{L}(M_w) = \mathcal{L}(T)$. Hence, $M_w \in P$.

If M rejects w, then $\mathcal{L}(M_w) = \emptyset = \mathcal{L}(M_{reject})$. Hence, $M_w \notin P$.

Finally, if M loops on w, then M_w loops on all inputs y. In this case too, $\mathcal{L}(M_w) = \emptyset = \mathcal{L}(M_{reject})$. Hence, $M_w \notin P$.

Consequently,

$$\langle M, w \rangle \in A_{TM} \iff M_w \in P,$$

i.e., $A_{TM} \leq_M P$. $\square$

Applications of Rice's Theorem

Applications

1. $EMPTY_{TM} := \{\langle M \rangle \mid M$ is a TM and $\mathcal{L}(M) = \emptyset\}$
2. $FINITE_{TM} := \{\langle M \rangle \mid M$ is a TM and $\mathcal{L}(M)$ is finite$\}$
3. $REGULAR_{TM} := \{\langle M \rangle \mid M$ is a TM and $\mathcal{L}(M)$ is regular$\}$
4. $DECIDABLE_{TM} := \{\langle M \rangle \mid M$ is a TM and $\mathcal{L}(M)$ is decidable$\}$

By Rice's Theorem, all the above languages are undecidable.

Venn diagram of different classes