CS 483 - Introduction to the Theory of Computation

Lecture 17

Problems Outside **RE**

For $L \in$ **RE**, $L \leq_{\mathrm{m}} \textsc{Halt}$ holds, meaning that L is no harder than $\textsc{Halt}$.

For $L \notin$ **RE**, $L \not\leq_{\mathrm{m}} \textsc{Halt}$ holds. Does it mean that L is harder than $\textsc{Halt}$?

Problems Outside **RE**

For $L \in$ **RE**, $L \leq_{\mathrm{m}} \textsc{Halt}$ holds, meaning that L is no harder than $\textsc{Halt}$.

For $L \notin$ **RE**, $L \not\leq_{\mathrm{m}} \textsc{Halt}$ holds. Does it mean that L is harder than $\textsc{Halt}$?

Example 1:

$\overline{\textsc{Halt}} \notin$ **RE**. But deciding $\overline{\textsc{Halt}}$ is equivalent to deciding $\textsc{Halt}$.

Problems Outside **RE**

For $L \in$ **RE**, $L \leq_{\mathrm{m}} \textsc{Halt}$ holds, meaning that L is no harder than $\textsc{Halt}$.

For $L \notin$ **RE**, $L \not\leq_{\mathrm{m}} \textsc{Halt}$ holds. Does it mean that L is harder than $\textsc{Halt}$?

Example 1:

$\overline{\textsc{Halt}} \notin$ **RE**. But deciding $\overline{\textsc{Halt}}$ is equivalent to deciding $\textsc{Halt}$.

Example 2:

$\textsc{Halt} \times \overline{\textsc{Halt}} \notin$ **RE** $\cup$ **coRE**. But if you can decide $\textsc{Halt}$,

you can also decide $\textsc{Halt} \times \overline{\textsc{Halt}}$.

Turing Reductions

We call this more general type of reductions Turing reductions.

Problem A can be Turing reduced to problem B, if we can use the solutions to problem B as subroutines to solve problem A.

To formally define Turing reductions, we need to define what are subroutines.

Oracle Turing Machine

Let O be a language. An oracle Turing machine M^O with oracle O is a Turing machine augmented with:

- A special state $q_{\mathbf{query}} \in Q$;
- Two special tapes: query tape and oracle answer tape.
- Whenever M^O enters $q_{\mathbf{query}}$ state, in the next transition it writes

$$\begin{cases} 1 & x \in O \\ 0 & x \notin O \end{cases}$$

on the oracle answer tape, where x is the content on the query tape.

Turing Reductions

A Turing reduction from language A to B is an oracle Turing machine M^B that decides A. In this case we write $A \leq_{\mathrm{T}} B$.

Turing Reductions

A Turing reduction from language A to B is an oracle Turing machine M^B that decides A. In this case we write $A \leq_{\mathrm{T}} B$.

Example. $\overline{\mathrm{HALT}} \leq_{\mathrm{T}} \mathrm{HALT}$

Description of the oracle machine $M^{\mathrm{HALT}}(M, w)$:

- Query the oracle if $(M, w) \in \mathrm{HALT}$;
  - If so, reject; otherwise accept.

Turing Reductions

A Turing reduction from language A to B is an oracle Turing machine M^B that decides A. In this case we write $A \leq_{\mathrm{T}} B$.

Example. $\mathrm{HALT} \times \overline{\mathrm{HALT}} \leq_{\mathrm{T}} \mathrm{HALT}$

Description of the oracle machine $M^{\mathrm{HALT}}(M_1, w_1, M_2, w_2)$:

- Query the oracle if $(M_1, w_1) \in \mathrm{HALT}$, obtain answer b_1
- Query the oracle if $(M_2, w_2) \in \mathrm{HALT}$, obtain answer b_2
- Accept if $b_1 = 1$ and $b_2 = 0$; otherwise reject.

Turing Reductions

A Turing reduction from language A to B is an oracle Turing machine M^B that decides A. In this case we write $A \leq_{\mathrm{T}} B$.

Example. $A_{\mathrm{TM}} \leq_{\mathrm{T}} \mathrm{HALT}$

Description of the oracle machine $M^{\mathrm{HALT}}(M, w)$:

- Query the oracle if $(M, w) \in \mathrm{HALT}$;
  - If not then reject;
  - Otherwise, run $U(M, w)$ and accept or reject accordingly.

Turing Reductions

Theorem. If $A \leq_T B$ and B is decidable, then A is also decidable.

(I.e., **R** is closed under Turing reductions)

Turing Reductions

Theorem. If $A \leq_T B$ and B is decidable, then A is also decidable.

(I.e., **R** is closed under Turing reductions)

(**RE** is NOT closed under Turing reductions, as seen in $\overline{\text{HALT}} \leq_T \text{HALT}$)

Turing Reductions

Theorem. If $A \leq_T B$ and B is decidable, then A is also decidable.

(I.e., **R** is closed under Turing reductions)

(**RE** is NOT closed under Turing reductions, as seen in $\overline{\text{HALT}} \leq_T \text{HALT}$)

Let M^B be a Turing reduction from A to B.

We can decide A by running M^B on the input. Every time it makes a query to B, run the decider for B to answer the query.

Turing Reductions

Theorem. If $A \leq_{\mathrm{m}} B$ then $A \leq_{\mathrm{T}} B$.

(I.e., Turing reduction is more general than mapping reduction)

Turing Reductions

Theorem. If $A \leq_{\mathrm{m}} B$ then $A \leq_{\mathrm{T}} B$.

(I.e., Turing reduction is more general than mapping reduction)

Let f be the mapping reduction from A to B.

A Turing reduction $M^B(x)$ could be:

- Compute $y = f(x)$;
- Query the oracle if $y \in B$. If so accept, otherwise reject.

Relativization

Given an oracle O, we define

$$\mathbf{R}^O = \{ L \subseteq \Sigma^* \mid L \text{ can be decided by some oracle TM } M^O \}$$

$$\mathbf{RE}^O = \{ L \subseteq \Sigma^* \mid L \text{ can be recognized by some oracle TM } M^O \}$$

Relativization

Given an oracle O, we define

$$\mathbf{R}^O = \{ L \subseteq \Sigma^* \mid L \text{ can be decided by some oracle TM } M^O \}$$

$$\mathbf{RE}^O = \{ L \subseteq \Sigma^* \mid L \text{ can be recognized by some oracle TM } M^O \}$$

In general, but NOT formally:

For a class of languages $\mathcal{C}$ (defined using Turing machines) and oracle O, we use $\mathcal{C}^O$ to denote the class defined similarly but using oracle Turing machines with oracle O.

Relativization

Given an oracle O, we define

$$\mathbf{R}^O = \{ L \subseteq \Sigma^* \mid L \text{ can be decided by some oracle TM } M^O \}$$

$$\mathbf{RE}^O = \{ L \subseteq \Sigma^* \mid L \text{ can be recognized by some oracle TM } M^O \}$$

In other words, $\mathbf{R}^O$ is the class of decidable languages relative to O, and $\mathbf{RE}^O$ is the class of recognizable languages relative to O.

Relativization

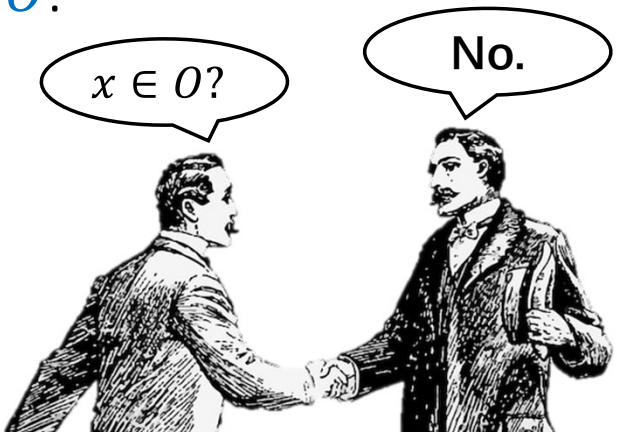
Given an oracle O, we define

$$\mathbf{R}^O = \{ L \subseteq \Sigma^* \mid L \text{ can be decided by some oracle TM } M^O \}$$

$$\mathbf{RE}^O = \{ L \subseteq \Sigma^* \mid L \text{ can be recognized by some oracle TM } M^O \}$$

In other words, $\mathbf{R}^O$ is the class of decidable languages relative to O, and $\mathbf{RE}^O$ is the class of recognizable languages relative to O.

In another world where deciding O is trivial, $\mathbf{R}^O$ and $\mathbf{RE}^O$ are their **R** and **RE**.



Relativization

Is it possible that $\mathbf{R}^{O} = \mathbf{ALL}$ for some oracle O?

Relativization

Is it possible that $\mathbf{R}^O = \mathbf{ALL}$ for some oracle O?

No, because $\mathbf{R}^O$ is still countable:

Every oracle Turing machine can be described as

$$(Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{acc}}, q_{\mathrm{rej}}, q_{\mathrm{query}})$$

and therefore has a finite-length binary encoding.
So the set of all oracle Turing machines is countable.

Relativization

Examples of undecidable languages relative to O:

$$L^O = \{ M^O \mid M^O \text{ is an oracle Turing machine}, M^O(M^O) \neq 1 \}$$

Relativization

Examples of undecidable languages relative to O:

$$L^O = \{ M^O \mid M^O \text{ is an oracle Turing machine}, M^O(M^O) \neq 1 \}$$

List of all oracle Turing machines

	$\langle M_0^O \rangle$	$\langle M_1^O \rangle$	$\langle M_2^O \rangle$	$\langle M_3^O \rangle$	$\langle M_4^O \rangle$	$\cdots$
M_0^O	0	1	1	1	0	
M_1^O	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	
M_2^O	0	$\perp$	1	$\perp$	1	
M_3^O	0	0	$\perp$	$\perp$	$\perp$	
M_4^O	$\perp$	0	0	1	0	
$\vdots$						$\ddots$
L	1	1	0	1	1	$\cdots$

Relativization

Examples of undecidable languages relative to O:

$$L'^{O} = \{ M^{O} \mid M^{O} \text{ is an oracle Turing machine}, M^{O}(M^{O}) = 1 \}$$

Relativization

Examples of undecidable languages relative to O:

$$L'^{O} = \{ M^{O} \mid M^{O} \text{ is an oracle Turing machine}, M^{O}(M^{O}) = 1 \}$$

By closure properties:

$\mathbf{R}^{O}$ is closed under complement and intersection (why?), and

$$L^{O} = \{\text{oracle Turing machine } M^{O}\} \cap \overline{L'^{O}}$$

Relativization

Examples of undecidable languages relative to O:

$$A_{\mathrm{TM}}^{O} = \{\, (M^O, w) \mid M^O \text{ is an oracle Turing machine}, M^O(w) = 1 \,\}$$

By reduction: $\mathbf{R}^O$ is closed under mapping reductions (why?), and we can show that $L'^O \leq_{\mathrm{m}} A_{\mathrm{TM}}^{O}$:

Let $f(M^O) = (M^O, M^O)$. Then f is computable and

$$M^O \in L'^O \Leftrightarrow (M^O, M^O) \in A_{\mathrm{TM}}^{O}$$

Relativization

We can also show $A_{\mathrm{TM}}^{O} \leq_{\mathrm{m}} \textsc{Halt}^{O}$ for

$$\textsc{Halt}^{O} = \{ (M^{O}, w) \mid M^{O} \text{ is an oracle Turing machine}, M^{O}(w) \neq \perp \}$$

therefore $\textsc{Halt}^{O} \notin \mathbf{R}^{O}$.

Relativization

We can also show $A_{\mathrm{TM}}^{O} \leq_{\mathrm{m}} \textsc{Halt}^{O}$ for

$$\textsc{Halt}^{O} = \{ (M^{O}, w) \mid M^{O} \text{ is an oracle Turing machine}, M^{O}(w) \neq \perp \}$$

therefore $\textsc{Halt}^{O} \notin \mathbf{R}^{O}$.

The proofs are essentially the same as the ones without oracles, and these results are true no matter what the oracle O is.

So we say that the statements, such as “halting problem is undecidable” (and their proofs) relativize.

Relativization

Many of the results we proved and will prove in this course relativizes.

However, relativization is NOT automatically true, and you should never assume any statement relativizes without a proof.

Instead, relativization is helpful for understanding why some proof techniques work and some do not (we will see examples in future lectures).