Lecture 10 CSE 431
Intro to Theory of Computation

So far: A_{TM} undecidable, T-rec
$\overline{A_{TM}}$ not T-rec
Other undecidable problems:
$HALT_{TM}$
E_{TM}
EQ_{TM}
Computable functions

<u>Mapping reductions</u>:

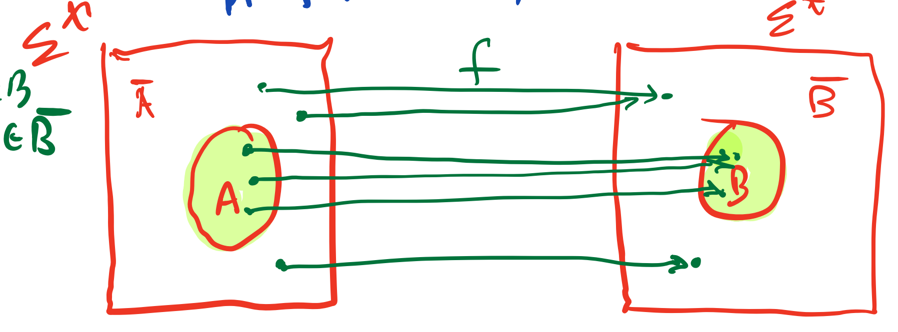
reduction

Def[n] $A \le_m B$ iff $\exists$ computable function $f: \Sigma^* \to \Sigma^*$
s.t. $\forall w \in \Sigma^*.\ w \in A \Leftrightarrow f(w) \in B$

<u>Thm</u> Suppose that $A \le_m B$:
- if B is decidable then A is decidable
- if A is <u>undecidable</u> then so is B
- if B is T-rec then A is T-rec
- if A is not T-rec then B is not T-rec

Correctness of Mapping Reduction f

$w \in A \Rightarrow f(w) \in B$
$w \in \overline{A} \Rightarrow f(w) \in \overline{B}$

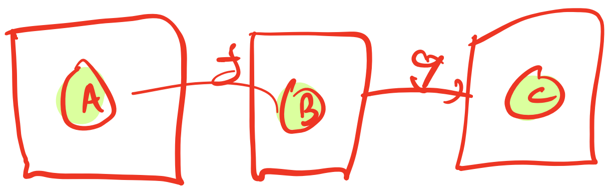


<u>Thm</u> $A \le_m B \Leftrightarrow \overline{A} \le_m \overline{B}$
Pf see picture

Thm $A \le_m B$ and $B \le_m C \Rightarrow A \le_m C$

Proof Let f be reduction given by TM M_f showing that $w \in A \Leftrightarrow f(w) \in B$

Let g be reduction given by TM M_g showing that $w' \in B \Leftrightarrow g(w') \in C$.

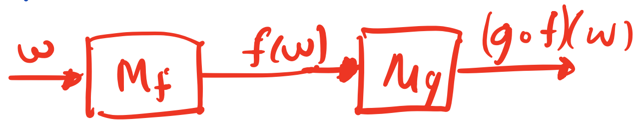


Let $w \in A$. $w \in A \Leftrightarrow f(w) \in B \Leftrightarrow g(f(w)) \in C$

$$g(f(w)) = g \circ f\,(w)$$

$\therefore\ w \in A \Leftrightarrow (g \circ f)(w) \in C$

$g \circ f$ is computable:



$\therefore\ g \circ f$ is a reduction showing $A \le_m C$ ∎

Thm Neither EQ_{TM} nor $\overline{EQ_{TM}}$ is T-rec

Proof (a) EQ_{TM} is not T-rec:

Claim: $\overline{A_{TM}} \le_m EQ_{TM}$

Want f s.t. $\langle M, w\rangle \overset{f}{\longmapsto} \langle M_1, M_2\rangle$

and $\langle M, w\rangle \in \overline{A_{TM}} \Leftrightarrow L(M_1) = L(M_2)$

i.e. M does not accept w $\iff L(M_1) = L(M_2)$

Idea: $\langle M, w \rangle \longmapsto \langle M_w, M_\phi \rangle$

Clearly f is computable

where (from reduction for E_{TM})
- M_w is the TM that ignores its input and runs M on input w
- M_ϕ is a simple TM that always rejects

Now $L(M_w) = \begin{cases} \Sigma^* & \text{if } M \text{ accepts } w \\ \phi & \text{if } M \text{ does not accept } w \end{cases}$

$L(M_\phi) = \phi$

Correctness: $L(M_\phi) = L(M_w) \iff M$ does not accept w.

(b) $\overline{EQ_{TM}}$ is not T-rec

Claim: $\overline{A_{TM}} \leq_m \overline{EQ_{TM}}$

ie $A_{TM} \leq_m EQ_{TM}$

Want f with $\langle M, w \rangle \xmapsto{f} \langle M_1, M_2 \rangle$

s.t. M accepts w $\iff L(M_1) = L(M_2)$

Similar idea: $\langle M, w \rangle \xmapsto{f} \langle M_w, M_{\Sigma^*} \rangle$

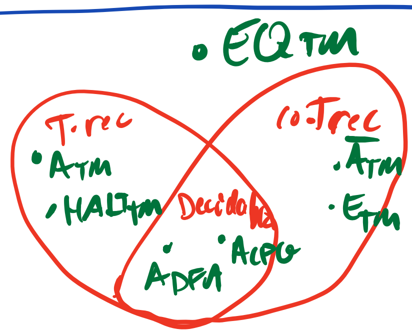
Clearly f is computable

where M_{Σ^*} is a TM with input alphabet Σ that always accepts

$\therefore L(M_{\Sigma^*}) = \Sigma^*$

and $L(M_w) = \Sigma^* = L(M_{\Sigma^*})$
iff M accepts w.

$\therefore$ Reduction is correct ∎



A much broader class of properties that are undecidable

$P_{TM} = \{\langle M \rangle : M$ is a TM s.t. $L(M)$ has property $P\}$

Examples of properties P: "$=\emptyset$" empty, "regular"

Rice's Theorem
Unless P is "trivial"
P_{TM} is undecidable

We prove this in a special case first.

$REGULAR_{TM} = \{\langle M \rangle : L(M)$ is regular$\}$

Thm $REGULAR_{TM}$ is undecidable

Proof We prove this by showing

Claim $A_{TM} \le_m REGULAR_{TM}$

Want $\langle M, w\rangle \xmapsto{f} \langle M'_w\rangle$

Goal: M accepts $w \iff L(M'_w)$ is regular

More specific goal: M accepts $w \Rightarrow L(M'_w) = \Sigma^*$

regular

$\{0,1\} \subseteq \Sigma$

M doesn't accept $w \Rightarrow L(M'_w) = \{0^n 1^n : n \ge 0\}$

not regular

Construction of M'_w: On input x
if x is of form $0^n 1^n$ for some n then accept
otherwise
Run M on input w
& accept iff M does

Clearly f is computable

Correctness: If M accepts w then M'_w accepts every string
M doesn't accept w then M'_w accepts strings of the form $0^n 1^n$.

$\therefore$ $L(M'_w)$ is regular $\iff$ M accepts w ✓ P

Rice's Theorem :

Def^n Property $\mathcal{P}$ is non-trivial iff

- There is some TM M_1 s.t. $L(M_1)$ has property $\mathcal{P}$

and

- There is some TM M_0 s.t. $L(M_0)$ does not have property $\mathcal{P}$

Examples of trivial properties:

$L(M)$ is T-rec (always has $\mathcal{P}$)

$L(M)$ is not T-rec (never has $\mathcal{P}$)

Proof of Rice's Theorem

Case 1 : Σ^* has property $\mathcal{P}$

We use TM M_0 s.t. $L(M_0)$ does not have property $\mathcal{P}$

Claim $A_{TM} \leq_m \mathcal{P}_{TM}$

Want f: $\langle M, w \rangle \xmapsto{f} \langle M_{\mathcal{P}} \rangle$

s.t. M accepts $w \Longleftrightarrow L(M_{\mathcal{P}})$ has property $\mathcal{P}$

Design goal: $L(M_{\mathcal{P}}) = \begin{cases} \Sigma^* & \text{if } M \text{ accepts } w \\ L(M_0) & \text{if } M \text{ doesn't accept } w \end{cases}$

Want to use similar idea as for $REGULAR_{TM}$

Design of M_P (Attempt 1)
On input x
Run M_0 on input x
If M_0 accepts then accept
else run M on input w
and accepts iff M does

M_0 take place of the test for 0^n1^n

problem: M_0 may not halt
(unlike test for 0^n1^n)

We want x to be accepted if
(M_0 accepts x
or M accepts w.)

Actual Design for M_P:
On input x:
Run M_0 on input x
and M on input w
in parallel one step at a time
If either accepts then accept

Function mapping $\langle M, w\rangle$ to $\langle M_P\rangle$ is easily computable

If M accepts w then $L(M_P) = \Sigma^*$ has P
and if not, then $L(M_P) = L(M_0)$ doesn't have P

$\therefore A_{TM} \leq P_{TM}$ so P_{TM} is undecidable.

Case 2 Σ^* does not have property P

We prove that $A_{TM} \leq_m \overline{P}_{TM}$.

$\therefore \overline{P}_{TM}$ is undecidable which means that P_{TM} is undecidable

We use $\overline{P}$, the complement of property P

$L(M_1)$ has property P

$\therefore$ $L(M_1)$ does not have property $\overline{P}$.

on the other hand, Σ^* has property $\overline{P}$

To show $A_{TM} \leq_m \overline{P}_{TM}$

we use M_1 in place of M_0 and the same proof idea to get $M_{\overline{P}}$.

$\therefore$ $\overline{P}_{TM}$ is undecidable and so P_{TM} is undecidable as required $\blacksquare$

We give an alternative proof in the notes