Recall

An infinite set A is countable $\Leftrightarrow$ There is a one-to-one map from $\mathbb{N}$ to A

e.g. $\mathbb{Z}$, $\mathbb{N} \times \mathbb{N}$, $\mathbb{Q}$ are countable

$\mathbb{R}$ is not countable

① Σ^* where Σ is a finite set is countable

We can order them by length (and then some arbitrary order)

② The set of all TM is countable

$TM = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

Note that M can be written as a finite string;

We can order them by length of their descriptions.

③ The set of all languages over $\Sigma = \{0, 1\}$ is NOT countable

We saw that Σ^* is countable and $w_1, w_2, w_3 \cdots$ is an ordering

Suppose $\cdots$ countable $L_1, L_2, L_3 \cdots$

	ε w_1	0 w_2	1 w_3	00 w_4	01 w_5	$\cdots$
L_1	✓	✓	✓	✓	✓	
L_2	✗	✗	✗	✗	✗	
L_3	✓	✗	✓	✗	✓	
L_4	✓	✓	✗	✓	✗	
L_5	✓	✓	✓	✓	✗	
$\vdots$						

$L = \{ w_i \mid w_i \notin L_i \} = \{ w_2, w_5 \cdots \}$

$\Rightarrow L \neq L_i$ for $\forall i$ (just look at w_i)

$\Rightarrow$ There are languages that are not Turing recognizable

An example of a non Turing recognizable language

	ε	0	1	00	01	10	11	...
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	Acc	
M_2	Rej	Rej	Rej	Rej	Rej	Rej	Rej	
M_3	Rej	Acc	Rej	Rej	Acc	Acc	Rej	
M_4	Rej	∞	Acc	∞	Rej	∞	Rej	
M_5	Rej	Acc	∞	Acc	Rej	Acc	Acc	
M_6	Rej	Acc	∞	Rej	∞	Acc	Rej	
M_7	Rej	Rej	Acc	Rej	∞	Acc	Rej	
$\vdots$								

Let

$= \{w_2, w_3, w_4, w_5, w_7, \dots\}$

$A_{DIAG} = \{ w_i \mid i \in \mathbb{N} \text{ and } M_i \text{ does not accept } w_i \}$

Can some TM M_r recognize A_{diag}?

Note if M_r accepts $w_r \Rightarrow w_r \notin A_{diag}$

if M_r does not accept $w_r \Rightarrow w_r \in A_{diag}$

$$\left.\begin{array}{l} w_r \in L(M_r) \Rightarrow w_r \notin A_{diag} \\ w_r \notin L(M_r) \Rightarrow w_r \in A_{diag} \end{array}\right\} \Rightarrow L(M_r) \neq A_{diag}$$

Thm: A_{DIAG} is not Turing Recognizable.

• A_{DIAG}

An example of non decidable

$A_{TM} = \{ \langle M, w \rangle \mid M \text{ accepts } w \}$

Suppose there is a decider D with $L(D) = A_{TM}$. Then the following TM will recognize A_{diag}, which is impossible, and hence a contradiction.

- On input w_i
- use the Universal U to run D on $\langle M_i, w_i \rangle$

  If it tells us $\langle M_i, w_i \rangle \in A_{TM} \Rightarrow$ reject w_i

  If it tells us $\langle M_i, w_i \rangle \notin A_{TM} \Rightarrow$ accept w_i

Ex. $L = \{ \langle M, w_1, w_2 \rangle \mid M \text{ is a TM that accepts both } w_1 \text{ and } w_2 \}$

If it is decidable, then some Turing machine R can decide L
Then the following TM will decide A_{TM}

- On input $\langle M, w \rangle$ of A_{TM}
- use the universal U run R on $\langle M, w, w \rangle$. If R says accept (meaning M accepts w), then accept $\langle M, w \rangle$

  If R says reject (meaning M doesn't accept w), then reject

Ex. $Halt_{TM} = \{ \langle M, w \rangle \mid M$ is a TM that terminates on $w \}$

- On input $\langle M, w \rangle$ of A_{TM}
- Use the universal U run R on $\langle M, w \rangle$. If R says accept (meaning M halts on w), using U run M on w and

  if M accepts $\Rightarrow$ accept

  if M rejects $\Rightarrow$ reject

  If R says reject (meaning M loop on w), then reject

Ex. $E_{TM} = \{ \langle M \rangle \mid L(M) = \phi \}$

- On input $\langle M, w \rangle$ of A_{TM}
- Construct the following TM N

  On input x

  Use the universal U run M on w

  if accept $\Rightarrow$ accept

  else $\Rightarrow$ reject

Note $L(N) = \begin{cases} \Sigma^* & M \text{ accepts } w \\ \phi & M \text{ doesn't accept } w \end{cases}$

- Run R on $\langle N \rangle$

  if R accepts $(L(N) = \phi)$ reject $\langle M, w \rangle$

  if R rejects $(L(N) \neq \phi)$ accept $\langle M, w \rangle$

Ex. $EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid L(M_1) = L(M_2) \}$

- On input $\langle M \rangle$ to E_{TM}
- Construct a TM N that rejects everything
- Run R on $\langle M, N \rangle$

  if R accepts $(L(M) = L(N) = \phi)$, accept $\langle M \rangle$

  if R rejects $(L(M) \neq L(N) = \phi)$, reject $\langle M \rangle$.

Ex. $DECIDERS = \{ \langle M \rangle \mid M$ halts on every $w \}$

Undecidable language

- $A_{DIAG} = \{\langle M_i, w_i\rangle \mid M_i \text{ doesn't accept } w_i\}$
- $A_{TM} = \{\langle M, w\rangle \mid M \text{ accepts } w\}$
- $HALT_{TM} = \{\langle M, w\rangle \mid M \text{ terminates on } w\}$
- $DECIDER_{TM} = \{\langle M\rangle \mid M \text{ is a decider}\}$
- $E_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) = \phi\}$
- $EQ_{TM} = \{\langle M_1, M_2\rangle \mid L(M_1) = L(M_2)\}$

Notation: $L^c = \Sigma^* \setminus L$

THM If L is decidable $\Rightarrow$ L^c is also decidable

Remark L is Turing recognizable $\not\Rightarrow$ L^c Turing recognizable.

DEF A language L is called co-Turing recognizable if L^c is Turing recognizable

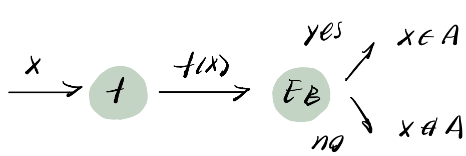
THM If L and L^c are both Turing recognizable
$\Rightarrow$ L and L^c are decidable.

$\Rightarrow$ Cor : A_{TM}^c is NOT Turing recognizable.

Reduction

Let A and B be two languages. We say A (easier) is mapping reducible to B if there is an algorithm that takes a string x and return $f(x)$ s.t.

$$x \in A \Leftrightarrow f(x) \in B$$



THM Suppose $A \leq_m B$, then

① B is decidable $\Rightarrow$ A is decidable

② B is Turing recognizable $\Rightarrow$ A is Turing recognizable

Ex. $A_{TM} = \{ \langle M, w \rangle \mid M \text{ accepts } w \} \leq_m ALL_{TM} = \{ \langle N \rangle \mid L(N) = \Sigma^* \}$

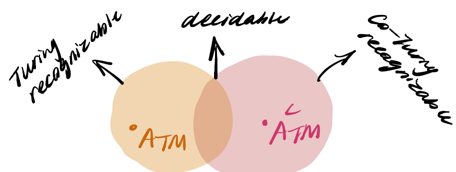
$f(\langle M, w \rangle) = \langle N \rangle$ = "On input x, run M on w:
if accept $\Rightarrow$ ACCEPT
if reject $\Rightarrow$ REJECT"

Note $\langle M, w \rangle \in A_{TM} \Leftrightarrow f(\langle M, w \rangle) \in ALL_{TM}$

Since A_{TM} is undecidable $\Rightarrow$ ALL_{TM} is undecidable

Recall that A_{TM} and $\overline{A_{TM}}$ are both undecidable.

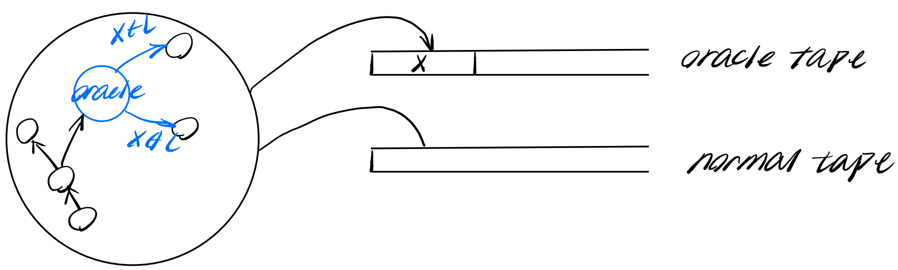
It's NOT true that $\overline{A_{TM}} \leq_m A_{TM}$, this will imply $\overline{A_{TM}}$ is Turing recognizable



Turing reduction

DEF An oracle for a language A is an imaginary machine that takes an input x and immediately ACCEPT or REJECT correctly depending on $x \in A$ or $x \notin A$.

DEF Let L be a language. An L-oracle TM M^L is similar to TM except that it has an oracle tape and an oracle state that work as below



DEF A is Turing reducible to B ($A \leq_T B$) if there is a B-oracle TM M^B that decides A.

① $E_{TM} \le_T A_{TM}$ $\quad (E_{TM} = \{\langle M\rangle \mid L(M) = \emptyset\})$

The following oracle TM decides E_{TM}.

Let $w_1, w_2, \dots$ be all the possible inputs

$T^{A_{TM}}$ = "On input $\langle M\rangle$,

① construct the following TM N:

N = "On input x, for $i = 0, 1, 2, \dots$

Run M on each of $w_1, w_2, \dots w_i$ for i steps

if it accepts any string ⇒ accept

N accept x ⇔ $L(M) \ne \emptyset$

② use the oracle

if $\langle N, \varepsilon\rangle \in A_{TM}$ ⇒ reject $\langle M\rangle$ $\quad L(M) \ne \emptyset$

if $\langle N, \varepsilon\rangle \notin A_{TM}$ ⇒ accept $\langle M\rangle$ " $\quad L(M) = \emptyset$

We showed $E_{TM} \le_T A_{TM}$

Note: this $T^{A_{TM}}$ is stronger than TM (can decide a undecidable language), but it is an imaginary machine (cannot use in practice)

THM $A \le_T B$ and B is decidable ⇒ A is decidable

THM If $A \le_m B$ ⇒ $A \le_T B$

① $A \le_m B$ and B is decidable ⇒ A decidable ✓

② $A \le_m B$ and B is Turing recognizable ⇒ A Turing recognizable ✓

③ $A \le_T B$ and B is Turing recognizable ⇒ A Turing recognizable ✗

$\overline{A_{TM}} \le_T A_{TM}$ $\left\{ \begin{array}{l} \text{On input } \langle M, w\rangle \\ \text{if } \langle M, w\rangle \in A_{TM} \text{ reject} \\ \text{else accept} \end{array} \right.$

$\overline{A_{TM}}$: NOT TR $\qquad A_{TM}$: TR

④ $A \le_T B$ ⇒ $A \le_m B$ ✗

⑤ $A \le_m B$ ⇒ $A \le_T B$ ✓

Rice's Theorem

$L = \{\langle M\rangle \mid M \text{ is a TM and } L(M) \text{ has some non-trivial property}\}$

$\exists \langle M\rangle \notin L$ and $\exists \langle M\rangle \in L$

$L(M_1) = L(M_2)$ ⇒ either $M_1, M_2 \in L$ or $M_1, M_2 \notin L$

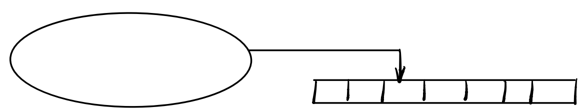
Example ①

If M halts on w, then the computational history of M on w is

$$C_1 \Rightarrow C_2 \Rightarrow C_3 \Rightarrow \cdots \Rightarrow C_\ell$$

· $q_0 w$ · accept or reject

DEF A **linear bounded automaton** is a TM whose tape-head is not allowed to go beyond the input part. However, it is still allowed to alter the input.



$A_{LBA} = \{ \langle M, w \rangle \mid M \text{ is an LBA that accepts } w \}$

$\Rightarrow$ decidable.

Note that given M and w the set of possible configurations is finite

$$\underset{\text{state}}{|Q|} \times \underset{\text{position of tape head}}{|w|} \times \underset{\text{content of the tape}}{|\Gamma|^{|w|}}$$

We can simulate the computation for < that many steps, either same configuration will be repeated and then we'll know that M loops on w

$$C_1 \Rightarrow \cdots \Rightarrow C_K \Rightarrow \cdots C_K \Rightarrow \cdots \Rightarrow C_K \cdots$$

Or we'll arrive at accept or reject

Claim Let M be a TM and w a string. There is a LBA that decides
$H_{M,w} = \{ \langle C \rangle \mid C = C_1 \# C_2 \# C_3 \ldots \# C_\ell$ is an accepting computational history for M on $w \}$

On input $C = C_1 \# C_2 \# \cdots \# C_\ell$

- We check to see if C_1 is the correct initial configuration for M and w
- We go back and forth between C_1 and C_2 to see if $C_1 \Rightarrow C_2$ is valid

  ...
- Check if C_ℓ is an accept configuration.

$E_{LBA} = \{ \langle N \rangle \mid N$ is a LBA and $L(N) = \phi \}$ <u>undecidable</u>

We'll reduce A_{TM} to this language

Given $\langle M, w \rangle$ where M is a TM, let $D_{M,w}$ be the LBA that decides

$$H_{M,w} = \{ \langle C \rangle \mid C = C_1 \# C_2 \# C_3 \ldots \# C_l \text{ is an accepting computational history for } M \text{ on } w \}$$

Note that

$$\langle M, w \rangle \in A_{TM} \iff L(D_{M,w}) \neq \phi$$

On input $\langle M, w \rangle$ to A_{TM}

- if $L(D_{M,w}) \neq \phi \Rightarrow$ accept $\langle M, w \rangle$
- if $L(D_{M,w}) = \phi \Rightarrow$ reject $\langle M, w \rangle$

THM $EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid L(M_1) = L(M_2) \}$ is NOT Turing recognizable and it's not co-Turing recognizable either.

(i) Suppose it's Turing recognizable by TM R

On input $\langle M, w \rangle$ to $\overline{A_{TM}}$

Let M_1 be the TM that rejects everything $L(M_1) = \phi$

M_2 "On input x, run M on w
- if it accepts $\Rightarrow$ accept
- if it rejects $\Rightarrow$ reject"

Note that
- if $\langle M, w \rangle \notin A_{TM} \Rightarrow L(M_2) = \phi$
- if $\langle M, w \rangle \in A_{TM} \Rightarrow L(M_2) \neq \phi$

$$\langle M, w \rangle \in \overline{A_{TM}} \iff \langle M_1, M_2 \rangle \in EQ_{TM}.$$

$\Rightarrow$ A TM R that recognizes EQ_{TM} then can be used to recognize $\overline{A_{TM}}$
NOT T.R.

(ii) $\overline{EQ_{TM}}$ NOT Turing recognizable

$$\langle M, w \rangle \in \overline{A_{TM}} \iff \langle M_1, M_2 \rangle \in \overline{EQ_{TM}}$$

- M_1 accepts everything
- M_2 same

The Post Corresponse Problem

Fix an alphabet Σ

We're given a set of dominos each containing two strings (top and bottom)

e.g.

$$\text{input} = \frac{b}{ca}, \frac{a}{ab}, \frac{ca}{a}$$

We want to make a finite sequence of dominos (repetition is allowed) such that the top string matches the bottom string.

e.g.

$$\frac{a}{ab} \frac{b}{ca} \frac{ca}{a} \frac{a}{ab} \frac{abc}{c} = \frac{abcaaabc}{abcaaabc}$$

PCP = { <P> | P is an instance of a PCP where a match exists }

MPCP = { <P> | Same as PCP except that the match has to start by using the first domino }

Both Turing recognizable but NOT decidable [Lec 19-20]