Undecidability and Reductions
CSCI 3130 Formal Languages and Automata Theory

Siu On CHAN

Chinese University of Hong Kong

Fall 2016

Undecidability

$$A_{\mathsf{TM}} = \{\langle M, w \rangle \mid \text{Turing machine } M \text{ accepts input } w\}$$
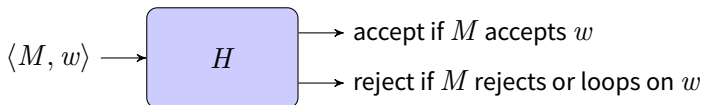
Turing's Theorem

The language A_{TM} is undecidable

Note that a Turing machine M may take as input its own description $\langle M \rangle$

Proof of Turing's Theorem
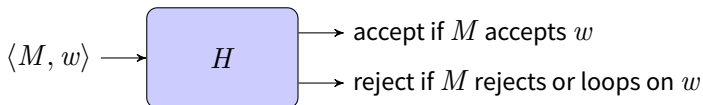
Proof by contradiction:

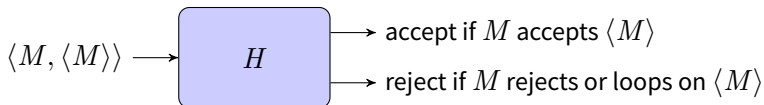Suppose A_{TM} is decidable, then some TM H decides A_{TM}:

$\langle M, w \rangle \longrightarrow$ [H] $\longrightarrow$ accept if M accepts w

$\longrightarrow$ reject if M rejects or loops on w

Proof of Turing's Theorem

Proof by contradiction:

Suppose A_{TM} is decidable, then some TM H decides A_{TM}:

$\langle M, w \rangle \longrightarrow$ | H | $\longrightarrow$ accept if M accepts w
$\longrightarrow$ reject if M rejects or loops on w

If $w = \langle M \rangle$,

$\langle M, \langle M \rangle \rangle \longrightarrow$ | H | $\longrightarrow$ accept if M accepts $\langle M \rangle$
$\longrightarrow$ reject if M rejects or loops on $\langle M \rangle$

Proof of Turing's theorem

$$\langle M, \langle M \rangle \rangle \longrightarrow \boxed{H} \longrightarrow \text{accept if } M \text{ accepts } \langle M \rangle$$
$$\longrightarrow \text{reject if } M \text{ rejects or loops on } \langle M \rangle$$
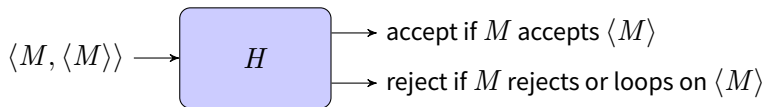
Let H' be a TM that does the opposite of H
accept states in H becomes reject states in H', and vice versa
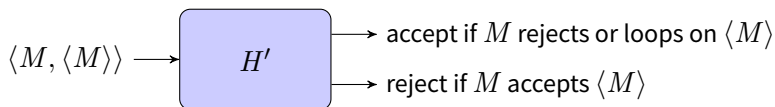
$$\langle M, \langle M \rangle \rangle \longrightarrow \boxed{H'} \longrightarrow \text{accept if } M \text{ rejects or loops on } \langle M \rangle$$
$$\longrightarrow \text{reject if } M \text{ accepts } \langle M \rangle$$

Proof of Turing's theorem

$\langle M, \langle M \rangle \rangle \longrightarrow$ **H′** $\longrightarrow$ accept if M rejects or loops on $\langle M \rangle$
$\longrightarrow$ reject if M accepts $\langle M \rangle$

Let D be the following TM:
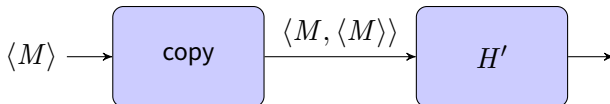
$\langle M \rangle \longrightarrow$ **copy** $\xrightarrow{\langle M, \langle M \rangle \rangle}$ **H′** $\longrightarrow$

Proof of Turing's theorem

$\langle M \rangle \longrightarrow$ [D] $\longrightarrow$ accept if M rejects or loops on $\langle M \rangle$
$\longrightarrow$ reject if M accepts $\langle M \rangle$

What happens when $M = D$?

$\langle D \rangle \longrightarrow$ [D] $\longrightarrow$ accept if D rejects or loops on $\langle D \rangle$
$\longrightarrow$ reject if D accepts $\langle D \rangle$

Proof of Turing's theorem

$\langle M \rangle \longrightarrow$ [D] $\longrightarrow$ accept if M rejects or loops on $\langle M \rangle$
$\longrightarrow$ reject if M accepts $\langle M \rangle$
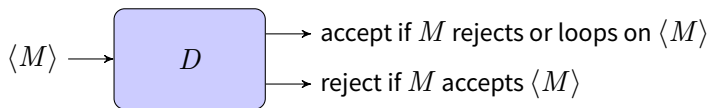
What happens when $M = D$?
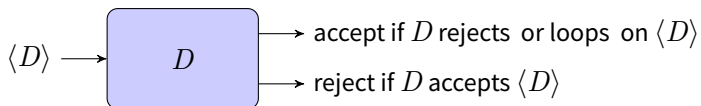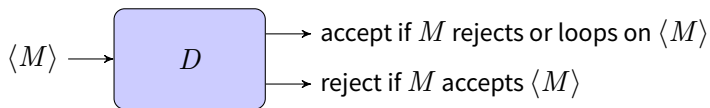
$\langle D \rangle \longrightarrow$ [D] $\longrightarrow$ accept if D rejects ~~or loops~~ on $\langle D \rangle$
$\longrightarrow$ reject if D accepts $\langle D \rangle$
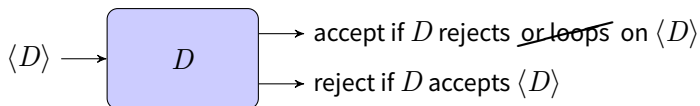
H never loops indefinitely, neither does D

If D rejects $\langle D \rangle$, then D accepts $\langle D \rangle$
If D accepts $\langle D \rangle$, then D rejects $\langle D \rangle$

Contradiction! D cannot exist! H cannot exist!

Proof of Turing's theorem: conclusion

Proof by contradiction

Assume A_{TM} is decidable
Then there are TM H, H' and D
But D cannot exist!

Conclusion

The language A_{TM} is undecidable

Diagonalization

		all possible inputs w				
		ε	0	1	00	...
all possible Turing machines	M_1	acc	rej	rej	acc	
	M_2	rej	acc	loop	rej	...
	M_3	rej	loop	rej	rej	
	M_4	acc	rej	acc	loop	
			$\vdots$			

Write an infinite table for the pairs (M, w)

(Entries in this table are all made up for illustration)

Diagonalization

		inputs w				
		$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	…
	M_1	acc	loop	rej	rej	
	M_2	rej	rej	acc	rej	…
all possible Turing machines	M_3	loop	acc	acc	acc	
	M_4	acc	acc	loop	acc	
			$\vdots$			

Only look at those w that describe Turing machines

Diagonalization

		inputs w			
		$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\ldots$
all possible Turing machines	M_1	acc	loop	rej	rej	
	M_2	rej	rej	acc	rej	$\ldots$
	M_3	loop	acc	acc	acc	
	$\vdots$		$\vdots$		
	D	rej	acc	rej	rej	
	$\vdots$		$\vdots$		

If A_{TM} is decidable, then TM D is in the table

Diagonalization

		inputs w			
		$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\ldots$
	M_1	acc	loop	rej	rej	
	M_2	rej	rej	acc	rej	$\ldots$
	M_3	loop	acc	acc	acc	
	$\vdots$		$\vdots$			
	D	rej	acc	rej	rej	
	$\vdots$		$\vdots$			

all possible Turing machines

D does the opposite of the diagonal entries
D on $\langle M_i \rangle$ = opposite of M_i on $\langle M_i \rangle$

$$\langle D \rangle \longrightarrow \boxed{D} \longrightarrow \text{accept if } D \text{ rejects or loops on } \langle D \rangle$$
$$\longrightarrow \text{reject if } D \text{ accepts } \langle D \rangle$$

Diagonalization

		inputs w				
		$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	...	$\langle D \rangle$
	M_1	acc	loop	rej	rej		loop
	M_2	rej	rej	acc	rej	...	acc
	M_3	loop	acc	acc	acc		rej
	⋮		⋮				
	D	rej	acc	rej	rej		?
	⋮		⋮				

all possible Turing machines

We run into trouble when we look at $(D, \langle D \rangle)$

Unrecognizable languages

> The language A_{TM} is recognizable but not decidable

How about languages that are not recognizable?

$$\overline{A_{\mathsf{TM}}} = \{\langle M, w\rangle \mid M \text{ is a TM that does not accept } w\}$$
$$= \{\langle M, w\rangle \mid M \text{ rejects or loops on input } w\}$$

Claim

> The language $\overline{A_{\mathsf{TM}}}$ is not recognizable

Unrecognizable languages

Theorem

If L and $\overline{L}$ are both recognizable, then L is decidable

Proof of Claim from Theorem:

We know $\overline{A_{\mathsf{TM}}}$ is recognizable

if $\overline{A_{\mathsf{TM}}}$ were also, then A_{TM} would be decidable

But Turing's Theorem says A_{TM} is not decidable

Unrecognizable languages

Theorem

If L and $\overline{L}$ are both recognizable, then L is decidable

Proof idea:

Let $M =$ TM recognizing L, $M' =$ TM recognizing $\overline{L}$

The following Turing machine N decides L:
On input w,

1. Simulate M on input w. If M accepts, N accepts.
2. Simulate M' on input w. If M' accepts, N rejects.

Unrecognizable languages

Theorem

If L and $\overline{L}$ are both recognizable, then L is decidable

Proof idea:

Let $M =$ TM recognizing L, $M' =$ TM recognizing $\overline{L}$

The following Turing machine N decides L:
On input w,

1. Simulate M on input w. If M accepts, N accepts.
2. Simulate M' on input w. If M' accepts, N rejects.

Problem: If M loops on w, we will never go to step 2

Unrecognizable languages

Theorem

> If L and $\overline{L}$ are both recognizable, then L is decidable

Proof idea (2nd attempt):

Let $M =$ TM recognizing L, $M' =$ TM recognizing $\overline{L}$

The following Turing machine N decides L:
On input w,

For $t = 0, 1, 2, 3, \ldots$
  Simulate first t transitions of M on input w.
  If M accepts, N accepts.
  Simulate first t transitions of M' on input w.
  If M' accepts, N rejects.

Reductions

Another undecidable language

$$\text{HALT}_{\text{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM that halts on input } w\}$$

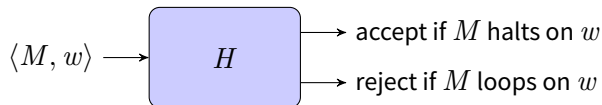We'll show:

HALT$_{\text{TM}}$ is an undecidable language

We will argue that
If HALT$_{\text{TM}}$ is decidable, then so is A_{TM}
…but by Turing's theorem, A_{TM} is not

Undecidability of halting

If HALT$_{\text{TM}}$ can be decided, so can A_{TM}

Suppose H decides HALT$_{\text{TM}}$

$\langle M, w \rangle \longrightarrow$ H $\longrightarrow$ accept if M halts on w

$\longrightarrow$ reject if M loops on w

We want to construct a TM S that decides A_{TM}

$\langle M, w \rangle \longrightarrow$? $\longrightarrow$ accept if M accepts w

$\longrightarrow$ reject if M rejects or loops on w
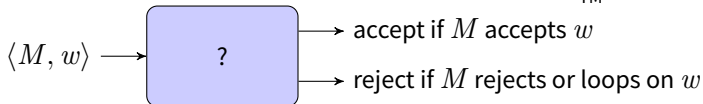
Undecidability of halting

$$\text{HALT}_{\text{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM that halts on input } w\}$$
$$A_{\text{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM that accepts input } w\}$$

Suppose HALT_{TM} is decidable
Let H be a TM that decides HALT_{TM}
The following TM S decides A_{TM}
On input $\langle M, w\rangle$:

Run H on input $\langle M, w\rangle$
If H rejects, reject
If H accepts, run U on input $\langle M, w\rangle$
  If U accepts, accept; else reject

Reductions

Steps for showing that a language L is undecidable:

1. If some TM R decides L
2. Using R, build another TM S that decides A_{TM}

But A_{TM} is undecidable, so R cannot exist

Example 1

$$A'_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts input } \varepsilon\}$$

Is A'_{TM} decidable? Why?

Example 1

$$A'_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts input } \varepsilon\}$$

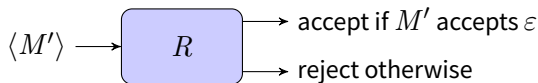Is A'_{TM} decidable? Why?

Undecidable!

Intuitive reason:
To know whether M accepts ε seems to require simulating M
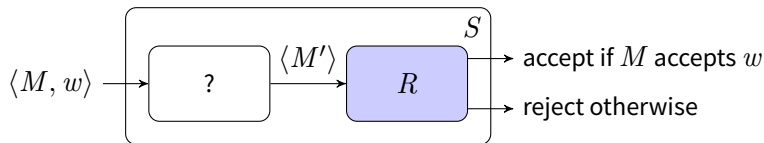But then we need to know whether M halts

Let's justify this intuition

Example 1: Figuring out the reduction

Suppose A'_{TM} can be decided by a TM R

$\langle M' \rangle \longrightarrow$ [R] $\longrightarrow$ accept if M' accepts ε

$\longrightarrow$ reject otherwise

We want to build a TM S

$\langle M, w \rangle \longrightarrow$ [?] $\xrightarrow{\langle M' \rangle}$ [R] $\longrightarrow$ accept if M accepts w

$\longrightarrow$ reject otherwise

M' should be a Turing machine such that
M' on input $\varepsilon = M$ on input w

Example 1: Implementing the reduction

$$\langle M, w \rangle \longrightarrow \boxed{\ ?\ } \longrightarrow \langle M' \rangle$$

M' should be a Turing machine such that
M' on input ε = M on input w

Description of the machine M':
On input z

1. Simulate M on input w
2. If M accepts w, accept
3. If M rejects w, reject
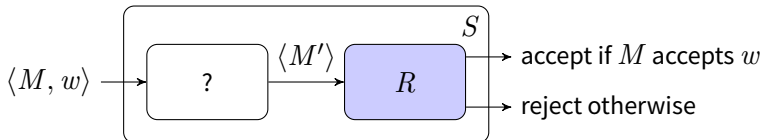
Description of S:
On input $\langle M, w \rangle$ where M is a TM

1. Construct the following TM M':

   $M' = $ a TM such that on input z,

   Simulate M on input w and accept/reject according to M

2. Run R on input $\langle M' \rangle$ and accept/reject according to R

Example 1: The formal proof

$$A'_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts input } \varepsilon\}$$
$$A_{\mathsf{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM that accepts input } w\}$$

Suppose A'_{TM} is decidable by a TM R.
Consider the TM S: On input $\langle M, w \rangle$ where M is a TM

1. Construct the following TM M':

$M' =$ a TM such that on input z,

Simulate M on input w and accept/reject according to M

2. Run R on input $\langle M' \rangle$ and accept/reject according to R

Then S accepts $\langle M, w \rangle$ if and only if M accepts w
So S decides A_{TM}, which is impossible

Example 2

$$A_{\mathsf{TM}}'' = \{\langle M \rangle \mid M \text{ is a TM that accepts some input strings}\}$$
Is A_{TM}'' decidable? Why?

Example 2

$$A''_{\mathsf{TM}} = \{\langle M\rangle \mid M \text{ is a TM that accepts some input strings}\}$$
Is A''_{TM} decidable? Why?
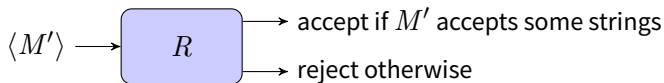
Undecidable!

Intuitive reason:
To know whether M accepts some strings seems to require simulating M
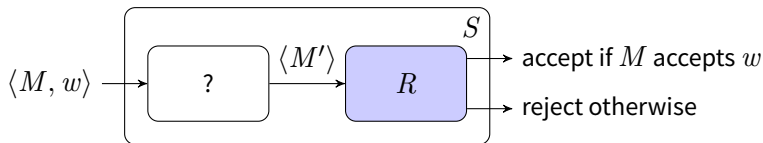But then we need to know whether M halts

Let's justify this intuition

Eample 2: Figuring out the reduction

Suppose A''_{TM} can be decided by a TM R

$\langle M' \rangle \longrightarrow$ [R] $\longrightarrow$ accept if M' accepts some strings

$\longrightarrow$ reject otherwise

We want to build a TM S

$\langle M, w \rangle \longrightarrow$ [?] $\xrightarrow{\langle M' \rangle}$ [R] $\longrightarrow$ accept if M accepts w

$\longrightarrow$ reject otherwise

M' should be a Turing machine such that
M' accepts some strings if and only if M accepts input w

Implementing the reduction

Task: Given $\langle M, w \rangle$, construct M' so that
If M accepts w, then M' accepts some input
If M does not accept w, then M' accepts no inputs

$M' =$ a TM such that on input z,

1. Simulate M on input w
2. If M accepts, accept
3. Otherwise, reject

Example 2: The formal proof

$$A''_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts some input}\}$$
$$A_{\mathsf{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM that accepts input } w\}$$

Suppose A''_{TM} is decidable by a TM R.
Consider the TM S: On input $\langle M, w \rangle$ where M is a TM

1. Construct the following TM M':

$M' = $ a TM such that on input z,

Simulate M on input w and accept/reject according to M

2. Run R on input $\langle M' \rangle$ and accept/reject according to R

Then S accepts $\langle M, w \rangle$ if and only if M accepts w
So S decides A_{TM}, which is impossible

Example 3

$$E_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts no input}\}$$
$$\text{Is } E_{\text{TM}} \text{ decidable?}$$

Example 3

$$E_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts no input}\}$$
$$\text{Is } E_{\mathsf{TM}} \text{ decidable?}$$

Undecidable! We will show:

If E_{TM} can be decided by some TM R
Then A''_{TM} can be decided by another TM S

$$A''_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts some input strings}\}$$

Example 3

$$E_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts no input}\}$$
$$A''_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts some input}\}$$

Note that E_{TM} and A''_{TM} are complement of each other
(except ill-formatted strings, which we will ignore)

Suppose E_{TM} can be decided by some TM R
Consider the following TM S:
On input $\langle M \rangle$ where M is a TM

1. Run R on input $\langle M \rangle$
2. If R accepts, reject
3. If R rejects, accept

Then S decides A''_{TM}, a contradiction

Example 4

$$EQ_{\mathsf{TM}} = \{\langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs such that } L(M_1) = L(M_2)\}$$
Is EQ_{TM} decidable?

Example 4

$\mathsf{EQ_{TM}} = \{\langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs such that } L(M_1) = L(M_2)\}$
Is $\mathsf{EQ_{TM}}$ decidable?

Undecidable!
We will show that $\mathsf{EQ_{TM}}$ can be decided by some TM R
then E_{TM} can be decided by another TM S

Example 4: Setting up the reduction

$\mathsf{EQ_{TM}} = \{\langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs such that } L(M_1) = L(M_2)\}$

$E_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts no input}\}$

Given $\langle M \rangle$, we need to construct $\langle M_1, M_2 \rangle$ so that

If M accepts no input, then M_1 and M_2 accept same set of inputs

If M accepts some input, then M_1 and M_2 do not accept same set of inputs

Idea: Make $M_1 = M$

Make M_2 accept nothing

Example 4: The formal proof

$\mathsf{EQ_{TM}} = \{\langle M_1, M_2\rangle \mid M_1 \text{ and } M_2 \text{ are TMs such that } L(M_1) = L(M_2)\}$

$E_{\mathsf{TM}} = \{\langle M\rangle \mid M \text{ is a TM that accepts no input}\}$

Suppose $\mathsf{EQ_{TM}}$ is decidable and R decides it
Consider the following TM S:
On input $\langle M\rangle$ where M is a TM

1. Construct a TM M_2 that rejects every input z
2. Run R on input $\langle M, M_2\rangle$ and accept/reject according to R

Then S accepts $\langle M\rangle$ if and only if M accepts no input
So S decides E_{TM} which is impossible