Undecidability and Reductions

CSCI 3130 Formal Languages and Automata Theory

Siu On CHAN
Fall 2019

Chinese University of Hong Kong
Undecidability

\[ A_{\text{TM}} = \{ \langle M, w \rangle \mid \text{Turing machine } M \text{ accepts input } w \} \]

Turing’s Theorem

The language \( A_{\text{TM}} \) is undecidable

Note: 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 $A_{TM}$:

$\langle M, w \rangle \rightarrow H$

- accept if $M$ accepts $w$
- 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 $A_{TM}$:

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

Construct a new TM $D$ (that uses $H$ as a subroutine):

On input $\langle M \rangle$ (i.e. the description of a Turing machine $M$),
1. Run $H$ on input $\langle M, \langle M \rangle \rangle$
2. Output the opposite of $H$: If $H$ accepts, $D$ rejects; if $H$ rejects, $D$ accepts
Proof of Turing’s theorem

What happens when $M = D$?

Contradiction! $D$ cannot exist! $H$ cannot exist!
Proof of Turing’s theorem

\[\langle M \rangle \rightarrow D \]
- accept if \( M \) rejects or loops on \( \langle M \rangle \)
- reject if \( M \) accepts \( \langle M \rangle \)

What happens when \( M = D \)?

\[\langle D \rangle \rightarrow D \]
- accept if \( D \) rejects or loops on \( \langle D \rangle \)
- 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

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

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

#### inputs $w$

<table>
<thead>
<tr>
<th>Turing machines</th>
<th>$\langle M_1 \rangle$</th>
<th>$\langle M_2 \rangle$</th>
<th>$\langle M_3 \rangle$</th>
<th>$\langle M_4 \rangle$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>acc</td>
<td>loop</td>
<td>rej</td>
<td>rej</td>
<td></td>
</tr>
<tr>
<td>$M_2$</td>
<td>rej</td>
<td>rej</td>
<td>acc</td>
<td>rej</td>
<td>...</td>
</tr>
<tr>
<td>$M_3$</td>
<td>loop</td>
<td>acc</td>
<td>acc</td>
<td>acc</td>
<td></td>
</tr>
<tr>
<td>$M_4$</td>
<td>acc</td>
<td>acc</td>
<td>loop</td>
<td>acc</td>
<td></td>
</tr>
</tbody>
</table>

Only look at those $w$ that **describe** Turing machines.
## Diagonalization

If $A_{TM}$ is decidable, then TM $D$ is in the table.
### Diagonalization

#### Inputs $w$

<table>
<thead>
<tr>
<th>( \langle M_1 \rangle )</th>
<th>( \langle M_2 \rangle )</th>
<th>( \langle M_3 \rangle )</th>
<th>( \langle M_4 \rangle )</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>acc</td>
<td>loop</td>
<td>rej</td>
<td>rej</td>
</tr>
<tr>
<td>$M_2$</td>
<td>rej</td>
<td>[rej]</td>
<td>acc</td>
<td>rej</td>
</tr>
<tr>
<td>$M_3$</td>
<td>loop</td>
<td>acc</td>
<td>[acc]</td>
<td>acc</td>
</tr>
<tr>
<td>$D$</td>
<td>rej</td>
<td>acc</td>
<td>rej</td>
<td>rej</td>
</tr>
</tbody>
</table>

$D$ does the opposite of the diagonal entries

$D$ on \( \langle M_i \rangle \) = opposite of $M_i$ on \( \langle M_i \rangle \)

\[
\begin{align*}
\langle D \rangle & \rightarrow D \\
& \quad \text{accept if $D$ rejects or loops on } \langle D \rangle \\
& \quad \text{reject if $D$ accepts } \langle D \rangle 
\end{align*}
\]
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?

$\overline{A_{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM that does not accept } w \}$

$= \{ \langle M, w \rangle \mid M \text{ rejects or loops on input } w \}$

Claim

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

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

Proof of Claim from Theorem:

We know \( A_{TM} \) is recognizable
if \( A_{TM} \) were also, then \( A_{TM} \) would be decidable

But Turing’s Theorem says \( A_{TM} \) is not decidable
Theorem

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

Proof idea:

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

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.
Theorem

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

Proof idea:

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

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
Theorem

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

Proof idea (2nd attempt):

Let $M = \text{TM}$ recognizing $L, M' = \text{TM}$ 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
Suppose you have a program $R$ that solves problem $A$

Now you want to solve problem $B$, if you can reduce $B$ to $A$

Then you can solve problem $B$

Using $R$ as a subroutine

Example from Lecture 16

$A_{\text{DFA}} = \{ \langle D, w \rangle \mid D \text{ is a DFA that accepts input } w \}$

$A_{\text{NFA}} = \{ \langle N, w \rangle \mid N \text{ is an NFA that accepts input } w \}$

$A_{\text{NFA}}$ reduces to $A_{\text{DFA}}$ (by converting NFA into DFA)
If language $A$ is decidable, and language $B$ reduces to language $A$ then $B$ is also decidable.

If language $B$ reduces to language $A$, and $B$ is undecidable then $A$ is also undecidable.
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:

\[ \text{HALT}_{\text{TM}} \text{ is an undecidable language} \]

We will argue that

If \( \text{HALT}_{\text{TM}} \) is decidable, then so is \( A_{\text{TM}} \)

...but by Turing’s theorem, \( A_{\text{TM}} \) is not
Undecidability of halting

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

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

$$\langle M, w \rangle \rightarrow H$$

- accept if $M$ halts on $w$
- reject if $M$ loops on $w$

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

$$\langle M, w \rangle \rightarrow ?$$

- accept if $M$ accepts $w$
- reject if $M$ rejects or loops on $w$
Undecidability of halting

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

Suppose \( \text{HALT}_{TM} \) is decidable

Let \( H \) be a TM that decides \( \text{HALT}_{TM} \)

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 universal TM \( U \) on input \( \langle M, w \rangle \)

If \( U \) accepts, accept; else reject
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

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

Is \( A'_{TM} \) decidable? Why?
Example 1

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

Is \( A'_{TM} \) decidable? Why?

Undecidable!

Intuitive reason:
To know whether \( M \) accepts \( \varepsilon \) 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 $R$

$\langle M' \rangle \rightarrow R$ accept if $M'$ accepts $\varepsilon$

reject otherwise

We want to build a TM $S$

$\langle M, w \rangle \rightarrow ?$ $\langle M' \rangle \rightarrow R$ accept if $M$ accepts $w$

reject otherwise

$M'$ should be a Turing machine such that

outcome of $M'$ on input $\varepsilon = $ outcome of $M$ on input $w$
Example 1: Implementing the reduction

\[ \langle M, w \rangle \rightarrow ? \rightarrow \langle M' \rangle \]

\( M' \) should be a Turing machine such that

\( M' \) on input \( \varepsilon = M \) 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 $M'$:

   $M' = \text{a TM such that on input } z,$
   
   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

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

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 \( M' \):

\[ M' = \text{a TM such that on input } z, \]

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

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

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.
Example 2

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

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
Example 2: Figuring out the reduction

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

$\langle M' \rangle \xrightarrow{R} \text{accept if } M' \text{ accepts some strings}$

$\text{reject otherwise}$

We want to build a TM $S$

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

$\text{reject otherwise}$

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

\[ M' = \text{a TM such that on input } z, \]

1. Simulate \( M \) on input \( w \)
2. If \( M \) accepts, accept
3. Otherwise, reject
Example 2: The formal proof

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

Suppose \( A''_{\text{TM}} \) 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 \( M' \):

\[ M' = \text{a TM such that on input } z, \]

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 \( A_{\text{TM}} \), which is impossible
Example 3

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

Is \( E_{\text{TM}} \) decidable?
Example 3

\[ E_{TM} = \{ \langle M \rangle | M \text{ is a TM that accepts no input} \} \]

Is \( E_{TM} \) decidable?

Undecidable! We will show:

If \( E_{TM} \) can be decided by some TM \( R \)

Then \( A''_{TM} \) can be decided by another TM \( S \)

\[ A''_{TM} = \{ \langle M \rangle | M \text{ is a TM that accepts some input strings} \} \]
Example 3

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

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

\[ \text{EQ}_{\text{TM}} = \{ \langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs such that } L(M_1) = L(M_2) \} \]

Is \( \text{EQ}_{\text{TM}} \) decidable?
Example 4

\[ \text{EQ}_{\text{TM}} = \{ \langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs such that } L(M_1) = L(M_2) \} \]

Is \( \text{EQ}_{\text{TM}} \) decidable?

Undecidable!

We will show that \( \text{EQ}_{\text{TM}} \) can be decided by some TM \( R \) then \( E_{\text{TM}} \) can be decided by another TM \( S \)
Example 4: Setting up the reduction

\[
\text{EQ}_{\text{TM}} = \{\langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs such that } L(M_1) = L(M_2)\} \\
\text{E}_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM that accepts no input}\}
\]

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

\[ \text{EQ}_{\text{TM}} = \{ \langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs such that } L(M_1) = L(M_2) \} \]

\[ \text{E}_{\text{TM}} = \{ \langle M \rangle \mid M \text{ is a TM that accepts no input} \} \]

Suppose \( \text{EQ}_{\text{TM}} \) 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 \( \text{E}_{\text{TM}} \) which is impossible.