Some languages not Turing-recognizable

- $\Sigma^*$ is countable
  Simply count $w$ with $|w| = 0, 1, 2, 3, \ldots$
  For example, if $\Sigma = \{0, 1\}$, then
  \[
  \{\epsilon, 0, 1, 00, 01, 10, 11, \ldots\}
  \]
- The set of TMs is countable
- Each machine can be represented as a finite string
  (think about the formal definition)
- Thus the set of TMs is a subset of $\{0, 1\}^*$
- Let
Some languages not Turing-recognizable II

$L$: all languages over $\Sigma$
$B$: all infinite binary sequences

For any $A \in L$

there is a corresponding element in $B$

Example:

$A: 0\{0, 1\}^*$
$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$
$A = \{0, 00, 01, 000, 001, \ldots\}$
$\chi_A = 010110011\ldots$
Some languages not Turing-recognizable

- One-to-one correspondence between $B$ and $L$
- $B$ is uncountable (like real numbers)
  Therefore, $L$ is uncountable
- Each TM $\Rightarrow$ handles one language in $L$
  Set of TM is countable, but $L$ is not
- Thus some languages cannot be handled by TM
Recall the halting problem is

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

We prove it is undecidable by contradiction.

Assume there is an \( H \) that is a decider for \( A_{TM} \). Then \( H \) satisfies

\[ H(\langle M, w \rangle) = \begin{cases} 
\text{accept} & \text{if } M \text{ accepts } w \\
\text{reject} & \text{otherwise}
\end{cases} \]

Construct a new TM \( D \) with \( H \) as a subroutine.
Halting problem undecidable II

- For $D$, the input is $\langle M \rangle$, where $M$ is a TM.
  It runs $H$ on $\langle M, \langle M \rangle \rangle$ and outputs the opposite result of $H$.
- The machine $D$ satisfies
  
  $$D(\langle M \rangle) = \begin{cases} 
  \text{accept} & \text{if } M \text{ rejects } \langle M \rangle \\
  \text{reject} & \text{if } M \text{ accepts } \langle M \rangle 
  \end{cases}$$

- But we get a contradiction

  $$D(\langle D \rangle) = \begin{cases} 
  \text{accept} & \text{if } D \text{ rejects } \langle D \rangle \\
  \text{reject} & \text{if } D \text{ accepts } \langle D \rangle 
  \end{cases}$$
Halting problem undecidable III

- We said earlier that the diagonalization method is used for the proof. Is that the case?
- We show that indeed it is used
Diagonalization in the proof I

- Set of TMs is countable so we can have

<table>
<thead>
<tr>
<th></th>
<th>$\langle M_1 \rangle$</th>
<th>$\langle M_2 \rangle$</th>
<th>$\langle M_3 \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>$M_2$</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
</tbody>
</table>

blank entries: unknown if reject or loop

- But $H$ knows the solution as it is a decider

<table>
<thead>
<tr>
<th></th>
<th>$\langle M_1 \rangle$</th>
<th>$\langle M_2 \rangle$</th>
<th>$\langle M_3 \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>A</td>
<td>R</td>
<td>A</td>
</tr>
<tr>
<td>$M_2$</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
</tbody>
</table>
**Diagonalization in the proof II**

- $D$ outputs **opposite of diagonal entries**

<table>
<thead>
<tr>
<th></th>
<th>$\langle M_1 \rangle$</th>
<th>$\langle M_2 \rangle$</th>
<th>$\ldots$</th>
<th>$\langle D \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_2$</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D$</td>
<td></td>
<td></td>
<td>$\ldots$</td>
<td>?</td>
</tr>
</tbody>
</table>
Definition: a language is co-Turing-recognizable if its complement is Turing-recognizable

Theorem 4.22
Decidable ⇔ Turing-recognizable and co-Turing-recognizable

Why not
Turing-recognizable
⇒ complement Turing-recognizable

Note that “recognizable” means any
co-Turing-recognizable Language II

\( w \in \text{language} \)

is accepted by the machine in a finite number of steps

- That is, no infinite loop

- Example:
  \[ A_{TM} \text{ Turing-recognizable but not decidable} \]

\[ w \in \overline{A_{TM}} \]

\( \Rightarrow \) reject or loop

Thus \( \overline{A_{TM}} \) may not be Turing-recognizable
What if we swap $q_{\text{accept}}, q_{\text{reject}}$?

If

$$a \notin A \text{ and loop occurs}$$

then

$$a \in \overline{A}, \text{ but TM still loops}$$

We cannot reach the new $q_{\text{accept}}$ state.

Proof of Theorem 4.22

“$\Rightarrow$”

Decidable $\Rightarrow$ Turing-recognizable

Complement $\Rightarrow$ decidable $\Rightarrow$ Turing-recognizable
“⇐” Now $A, \overline{A}$ are Turing-recognizable by two machines $M_1, M_2$

Construct a new machine $M$: for any input $w$

1. Run $M_1, M_2$ in parallel
2. $M_1$ accept $\Rightarrow$ accept, $M_2$ accept $\Rightarrow$ reject

Never infinity loop

$M$ accepts all strings in $A$, reject all not in $A$

Thus $A$ is decidable with a decider $M$