The language $A_{TM}$

- Consider the language:
  
  $$A_{TM} = \{<M,w> \mid M \text{ is a TM and } M \text{ accepts } w\}$$

- NOTE: $<A,B,...>$ is just a string encoding the objects A, B, ...

- In particular, $<M,w>$ is a string listing the components of TM M followed by the string w

- Given input $<M,w>$, it should be easy to extract the info about M and to simulate M on w (try writing a TM to do this!)

- What can we say about $A_{TM}$?

Is $A_{TM}$ Turing-recognizable?
\( A_{TM} \) is Turing-recognizable

- \( A_{TM} \) is Turing-recognizable: Recognizer TM \( U \) for \( A_{TM} \):
  
  On input string \( <M,w> \):
  
  Simulate \( M \) on \( w \).
  
  ACCEPT \( <M,w> \) if \( M \) halts & accepts \( w \)
  
  REJECT \( <M,w> \) if \( M \) halts & rejects
  
  (Loop (& thus reject \( <M,w> \)) if \( M \) ends up looping).
  
  \( U \) accepts \( <M,w> \) iff \( M \) accepts \( w \), i.e. \( L(U) = A_{TM} \)

“Universal” TM
(can simulate any TM)

Yeah, but is it decidable?!!
Is $A_{TM}$ decidable?

- $A_{TM} = \{<M,w> \mid M \text{ is a TM and } M \text{ accepts } w\}$
- Let’s assume $A_{TM}$ is decidable and see where it leads us
- $A_{TM}$ is decidable $\Rightarrow$ there’s a decider $H$, $L(H) = A_{TM}$
  - $H$ on $<M,w> = \text{ACC}$ if $M$ accepts $w$
  - $\text{REJ}$ if $M$ rejects $w$ (by halting in $q_{REJ}$ or looping)
- Then, we can construct a new TM $D$ as follows:
  On input $<M>$:
    - Extract $M$ from $<M>$
    - Simulate $H$ on $<M,<M>>$ (here, $w = <M>$)
    - If $H$ accepts $<M,<M>>$, then REJECT input $<M>$
    - If $H$ rejects $<M,<M>>$, then ACCEPT input $<M>$
Is $A_{TM}$ decidable?

- **New TM $D$ works as follows:**
  
  On input $<M>$:
  
  Extract $M$ from $<M>$
  
  Simulate $H$ on $<M,<M>>$ (here, $w = <M>$)
  
  If $H$ accepts $<M,<M>>$, then REJECT input $<M>$
  
  If $H$ rejects $<M,<M>>$, then ACCEPT input $<M>$

- **What happens when $D$ gets $<D>$ as input?**
  
  If $D$ rejects $<D> \Rightarrow H$ accepts $<D,<D>> \Rightarrow D$ accepts $<D>$
  
  If $D$ accepts $<D> \Rightarrow H$ rejects $<D,<D>> \Rightarrow D$ rejects $<D>$

  Either way: **Contradiction!** $D$ cannot exist $\Rightarrow H$ cannot exist
  
  Therefore, $A_{TM}$ is not a decidable language.
Undecidability Proof uses Diagonalization

Input strings

<table>
<thead>
<tr>
<th>List of TMs</th>
<th>M_1</th>
<th>M_2</th>
<th>M_3</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;M_1&gt;</td>
<td>ACC</td>
<td>REJ</td>
<td>\textit{loop}</td>
<td>...</td>
</tr>
<tr>
<td>&lt;M_2&gt;</td>
<td>REJ</td>
<td>\textit{loop}</td>
<td>ACC</td>
<td>...</td>
</tr>
<tr>
<td>&lt;M_3&gt;</td>
<td>ACC</td>
<td>ACC</td>
<td>REJ</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
\text{Input strings} & <M_1> & <M_2> & <M_3> & ... & <D> \\
\hline
\text{List of TMs} & M_1 & M_2 & M_3 & ... & D \\
\hline
\text{ACC} & ACC & REJ & REJ & ... & ACC \\
\hline
\text{REJ} & REJ & ACC & ACC & ... & ACC \\
\hline
\text{ACC} & ACC & REJ & REJ & ... & REJ \\
\hline
\text{REJ} & ACC & ACC & ... & ?? \\
\end{array}\]

If H exists

D outputs opposite of diagonal

D on <M_i> accepts if and only if M_i on <M_i> rejects.
So, D on <D> will accept if and only if D on <D> rejects!
A contradiction \(\Rightarrow\) H cannot exist!
Therefore, \(A_{TM}\) is not a decidable language.
One Last Concept: Reducibility

- How do we show a new problem B is undecidable?
  - Idea: Show that a known undecidable problem (e.g., $A_{TM}$) is reducible to the new problem B
    ➔ What does this mean and how do we show this?
  - Show that if B was decidable, then you can use the decider for B as a subroutine to decide $A_{TM}$
    ➔ Contradiction, therefore B must also be undecidable
The Halting Problem is Undecidable (Turing, 1936)

- **Halting Problem**: Does TM M halt on input w?
  - Equivalent language:
    
    \[ \text{HALT} = \{ <M,w> \mid \text{TM M halts on input w} \} \]

  Need to show \( \text{HALT} \) is undecidable
  - Use the fact that \( A_{TM} = \{ <M,w> \mid \text{TM M accepts w} \} \)
    is known to be undecidable
The Halting Problem is Undecidable (cont.)

Show $A_{TM}$ is reducible to HALT (Theorem 5.1 in text)

- Suppose HALT is decidable $\Rightarrow$ there’s a decider $M_{HALT}$ for HALT
- Then, we can use $M_{HALT}$ to solve $A_{TM}$
- Define decider $D_{TM}$ as:
  
  On input $<M,w>$, first run $M_{HALT}$ on $<M,w>$.
  
  • If $M_{HALT}$ rejects, then REJ (this takes care of $M$ looping on $w$)
  
  • If $M_{HALT}$ accepts, then simulate $M$ on $w$ until $M$ halts
  
  • If $M$ accepts, then ACC input $<M,w>$; else REJ

Then, $L(D_{TM}) = A_{TM} \Rightarrow A_{TM}$ is decidable!

Contradiction. Therefore, HALT is undecidable.

- E.g. 2: Show $E_{TM} = \{<M> | M$ is a TM and $L(M) = \emptyset\}$ is undecidable
Last homework (#7) on class website today
   (due on Friday, last day of class)

   Take-Home Final on website on Friday June 4
   (due by 4:30pm Monday, June 7)

   No class this monday – UW holiday

   Enjoy the long weekend!