Undecidability
Section 4.2
Algorithmically Unsolvable Problems

- Many problems are unsolvable by computers
  - Many tasks that seem simple, may be computationally impossible

- Previously, we have used TM to show that a problem is solvable
  - Encode a problem as a language
  - If a TM is created that can decide the language, it is solvable.

- Now we introduce techniques to show that a problem is unsolvable.
An Undecidable Problem

- **Problem:** Is it possible to determine whether a Turing Machine accepts a given input string?

- Formulate this problem as a language $A_{TM} = \{<M,w>|M \text{ is a TM and } M \text{ accepts } w\}$.

- This language is recognizable by creating a TM that simulates $M$
  - $U = \text{“On input } <M,w>, \text{ where } M \text{ is a TM and } w \text{ is a string:} $
  - 1.\text{ Simulate } M \text{ on input } w$
  - 2.\text{ If } M \text{ enters an accept state, accept. If it enters a reject state, reject.}$
  - $U$ loops if $M$ loops
    - Not guaranteed to halt, therefore is not a decider

- $U$ is a Universal Turing Machine
  - A TM that is capable of simulating any other TM
Correspondence

- In order to show that not every problem is computable
  - Assume that for every unique problem, a unique TM must be created to solve it
  - This means the set of all problems, $S_p$, must be the same size of the set of all Turing Machines, $S_{TM}$

- Both sets are infinite but one may be larger than the other
  - For sets to be the same size, there must be a correspondence (bijective) between every element in each set.
    - One-to-one (injective)
    - Onto (surjective)
Countable Set

- If a set has a correspondence to the set of natural numbers $\mathbb{N}$, that set is said to be countable.
- If we cannot find a correspondence to $\mathbb{N}$, then the set is uncountable.
  - Uncountable sets are larger than countable sets.
Example: Countable Set

- Show that the set of even numbers, \( E \), is countable
- Show correspondence between \( E \) and \( N \)
  - \( f(n) = 2n \)
- \( E \) is countable

<table>
<thead>
<tr>
<th>n</th>
<th>f(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Example: Uncountable Set

- Show that the set of real numbers, R, is uncountable

Procedure
- Systematically construct a list for R
  - The index of each element in the list corresponds to an element in N
- Find an element, x, in R that is cannot be the list
  - Diagonalization method
    - Choose the digits for x so that $x \neq f(n)$ for any $n$

For the following list, choose a number that is different from the diagonal
- Uses each digit to mismatch the corresponding element
- For $x \neq f(1)$, 1st digit must be different
- For $x \neq f(n)$, $n^{th}$ digit must be different

<table>
<thead>
<tr>
<th>n</th>
<th>f(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.14159...</td>
</tr>
<tr>
<td>2</td>
<td>55.555555...</td>
</tr>
<tr>
<td>3</td>
<td>0.12345...</td>
</tr>
<tr>
<td>4</td>
<td>0.50000...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$x = 0.4641...$
Uncountable Number of Languages

- Since there are problems related to real numbers, the set of all problems $S_p$ is uncountable.

- The set of all TMs $S_{TM}$ can be listed and is countable.

- This means $S_p$ is larger than $S_{TM}$.
  - and that there are problems without a corresponding TM.
Example: An Undecidable Language

- **Problem:** Is it possible to determine whether a TM accepts a given input string?

- **Formulate this problem as a language** $A_{TM} = \{<M,w>| M \text{ is a TM and } M \text{ accepts } w\}$.
  - Assume that $A_{TM}$ is decidable and obtain a contradiction
  - Create a decider $H$ for $A_{TM}$:

    $$H(<M,w>) = \begin{cases} 
    \text{accept} & \text{if } M \text{ accepts } w \\
    \text{reject} & \text{if } M \text{ does not accept } w
    \end{cases}$$

- **Create a TM D with H as a subroutine**
  - $D =$ “On input $<M>$, where $M$ is a TM:
    1. Run $H$ on input $<M,<M>>$.
    2. Output the opposite of what $H$ outputs. That is, if $H$ accepts, reject; and if $H$ rejects, accept.”
Example: An Undecidable Language

Is it possible to determine whether a Turing Machine accepts a given input string?

Formulate this problem as a language $A_{\text{TM}} = \{<M,w>| M$ is a TM and M accepts w\}.

- Assume that $A_{\text{TM}}$ is decidable and obtain a contradiction.
- Create a decider H for $A_{\text{TM}}$:

$$H(<M,w>) = \begin{cases} 
\text{accept} & \text{if } M \text{ accepts } w \\
\text{reject} & \text{if } M \text{ does not accept } w 
\end{cases}$$

Create a TM D to simulate diagonalization with H as a subroutine.

- D does the opposite what M does when it receives itself as an input.
- D = “On input <M>, where M is a TM:
  1. Run H on input <M,<M>>.
  2. Output the opposite of what H outputs. That is, if H accepts, reject; and if H rejects, accept.”
Example: An Undecidable Language

- D always does the opposite
  \[
  D(<M>) = \begin{cases} 
  \text{accept} & \text{if } M \text{ does not accept } <M> \\
  \text{reject} & \text{if } M \text{ accepts } <M>
  \end{cases}
  \]

- If D receives itself, then
  \[
  D(<D>) = \begin{cases} 
  \text{accept} & \text{if } D \text{ does not accept } <D> \\
  \text{reject} & \text{if } D \text{ accepts } <D>
  \end{cases}
  \]

- This is a contradiction, and we can use diagonalization to see this
Example: An Undecidable Language

- Create a table of TMs and their encode versions:

<table>
<thead>
<tr>
<th></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>( \ldots )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_1 )</td>
<td>accept</td>
<td>reject</td>
<td>accept</td>
<td>reject</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( M_2 )</td>
<td>accept</td>
<td>accept</td>
<td>accept</td>
<td>accept</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( M_3 )</td>
<td>reject</td>
<td>reject</td>
<td>reject</td>
<td>reject</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( M_4 )</td>
<td>accept</td>
<td>accept</td>
<td>reject</td>
<td>reject</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
</tbody>
</table>

- Output of \( H \):

Since \( D \) itself is a TM it will be on the list and will be the opposite of the diagonals

- When we reach \( (D, \langle D \rangle) \), we get a contradiction

- This means that such a TM does not exist
  - \( A_{TM} \) is undecidable