COMP2210: Theory of Computing

Lecture 16

Diagonalisation & Halting Problem

Corina Cirstea
Objectives for Today

- The Halting Problem
- Diagonalisation
- Undecidability of the Halting Problem
The UTM

- The UTM reads an (encoding of) a TM $M$, an (encoding of) input $x$ and simulates $M$ on $x$

$$L(U) = \{ M \# x \mid x \in L(M) \}$$

- UTM will loop on $M \# x$ if $M$ loops on $x$;
- Can we come up with a smarter TM?
- For example, consider the following: is it recursive?

$$\text{HP} \overset{\text{def}}{=} \{ M \# x \mid M \text{ halts on } x \}$$
The Halting Problem

• In other words, can we come up with the following total TM $K$. On $M \neq x$ it:
  ○ halts and accepts if $M$ halts on $x$
    ✴ i.e. $M$ either accepts or rejects $x$;
  ○ halts and rejects if $M$ loops on $x$.

• In this lecture we will prove that no such $K$ is possible! So there are real theoretical limitations to the power of algorithms.
Diagonalisation, 1

• A proof technique due to Cantor. How to show that $2^\mathbb{N}$ is a ‘larger’ infinity than $\mathbb{N}$?

• Recall:
  
  o $2^\mathbb{N}$ is the set of subsets of $\mathbb{N}$;
  
  o alternatively, it is the set of functions $\mathbb{N} \rightarrow \{0, 1\}$.

  $A \subseteq \mathbb{N} \quad \mapsto \quad f_A(n) = \begin{cases} 
  1 & \text{if } n \in A \\
  0 & \text{if } n \notin A
\end{cases}$

  $f : \mathbb{N} \rightarrow \{0, 1\} \quad \mapsto \quad \{ \ n \mid f(n) = 1 \ \}$

• **Claim**: there is no onto function $\varphi : \mathbb{N} \rightarrow 2^\mathbb{N}$.

• We cannot number all of the subsets of $\mathbb{N}$, there are too many!
Proof: Suppose \( \varphi : \mathbb{N} \rightarrow 2^\mathbb{N} \) is any function whatsoever. Then we can draw a table:

\[
\begin{array}{l|cccc}
   & 0 & 1 & 2 & \ldots \\
\hline
\varphi(0) & 0 & 1 & 0 & \ldots \\
\varphi(1) & 1 & 1 & 0 & \ldots \\
\varphi(2) & 0 & 0 & 1 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \ddots \\
\end{array}
\]

Consider the subset of \( \mathbb{N} \):

\[\psi(i) = 1 - (\varphi(i))(i)\]

— i.e. complement the diagonal in the table. \( \psi \) does not appear in the table — it differs with \( \varphi(k) \) at \( k \)! So \( \varphi \) is not onto.

we made no assumptions about \( \varphi \), so there are no onto functions!
Halting Problem

- A similar argument can be used to show that:

There does not exist a total TM $K$ that decides whether, given some TM $M$ and input $x$, $M$ halts on $x$.

- Can enumerate all strings over $\{0, 1\}$:
  - $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots$

- Can enumerate all TMs with input alphabet $\{0, 1\}$
  - $M_\epsilon, M_0, M_1, M_{00}, M_{01}, M_{10}, M_{11}, \ldots$
    - where $M_b$ is the TM with binary encoding $b$
Undecidability of the Halting Problem

• Suppose that a total TM $K$ exists that decides whether a given TM halts or not on a given input. We can construct a table:

<table>
<thead>
<tr>
<th></th>
<th>$\epsilon$</th>
<th>0</th>
<th>1</th>
<th>00</th>
<th>01</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_\epsilon$</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>...</td>
</tr>
<tr>
<td>$M_0$</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>...</td>
</tr>
<tr>
<td>$M_1$</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>...</td>
</tr>
<tr>
<td>$M_{00}$</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>...</td>
</tr>
<tr>
<td>$M_{01}$</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Consider the TM $N$ that on $x$:

1. constructs $M_x$ from $x$ and writes $M_x\#x$ on tape;
2. runs $K$ on $M_x\#x$;
3. if $K$ accepts then $N$ goes into a trivial loop. If $K$ rejects then $N$ accepts.

Then $N$ is not in the table – its behaviour differs from $M_b$ at $b$!
Undecidability ctd.

- Since we took the table on the previous slide so that it contains *all the* TM’s with input alphabet \( \{0, 1\} \), we can derive a contradiction – we have constructed a TM that is *not* in the table!

- Any finite portion of the table can be constructed assuming only that \( K \), the total TM for the halting problem, exists, so:

  \[ K \text{ exists} \quad \not\Rightarrow \quad \text{table with all TMs} \quad \not\Rightarrow \quad \text{contradiction} \]

- Hence \( K \) cannot exist.
Halting Problem in C

- simplify to C programs that take binary as input
- is it possible to write a C program $Q$ that takes a C program + input and decides whether the program terminates on the input?

We can write a program $R$ that on input $k$:

<table>
<thead>
<tr>
<th>$Q$</th>
<th>0</th>
<th>1</th>
<th>00</th>
<th>01</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>...</td>
</tr>
<tr>
<td>$P_1$</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>...</td>
</tr>
<tr>
<td>$P_{00}$</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>...</td>
</tr>
<tr>
<td>$P_{01}$</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

1. runs $Q$ on $(P_k, k)$ (we have access to $Q$’s source!)
2. if $Q$ says H, go into an infinite loop;
3. if $Q$ says L, stop.

- $R$ does not appear in the table! (why?) – so $Q$ does not exist!
Halting Problem

\[ \text{HP} \overset{\text{def}}{=} \{ M \# x \mid M \text{ halts on } x \} \]

- We have shown that the set HP is not recursive.
Halting Problem

\[ HP \overset{\text{def}}{=} \{ M\#x \mid M \text{ halts on } x \} \]

- We have shown that the set \( HP \) is not recursive.
- It is recursively enumerable (why?):
  - if it were then \( HP \) would be recursive by construction from last lecture;
- So \( \neg HP \) cannot even be recursively enumerable!
- It follows that

\[ \{ M\#x \mid M \text{ loops on } x \} \]

is not a r.e. set.
The big picture

\[ 2^{\Sigma^*} \]

recursively enumerable

recursive

context-free

regular

\[ \{ a^n b^n \mid n \geq 0 \} \]

\[ \{ a^n b^n c^n \mid n \geq 0 \} \]

HP

\neg HP