Undecidability

April 6, 2020
First, let us take a stock…

   - Mon 3/23
   - Wed 3/25
   - Fri 3/27: HW6 out ✔

2. Week 3/30 – 4/3 ✔
   - Mon 3/30
   - Wed 4/1
   - Fri 4/3: HW6 in ✔

3. Week 4/6 – 4/10
   - Mon 4/6 TODAY
   - Wed 4/8
   - Fri 4/10: Mid Term 2 (take-home)

4. Week 4/13 – 4/17
   - Mon 4/13
   - Wed 4/15
   - Fri 4/17: HW7 out

5. Week 4/20 – 4/24
   - Mon 4/20
   - Wed 4/22
   - Fri 4/24: HW 7 in, HW8 out

6. Week 4/27 – 5/1
   - Mon 4/27
   - Wed 4/29
   - Fri 5/1: HW 8 in, Wrap-up

7. Week 5/4 – 5/8
   - Thurs 5/7: Final due (take-home)
Today (this week) we will prove

One of the most philosophically important theorems of the theory of computation:

There is a specific problem that is algorithmically unsolvable
Consider software verification

- **Computer Program** $P$
- **Specification of what** $P$ **is supposed to do** $\text{spec}(P)$
  - Program $Q$ that verifies $P$ performs as in $\text{spec}(P)$

No $Q$ exists! That is, software verification is not solvable by computer
Turing Machine: Acceptance

Let:

\[ \mathcal{A}_{TM} = \{<M,w> \mid M \text{ is a TM and } M \text{ accepts } w\} \]

Theorem:

\[ \mathcal{A}_{TM} \text{ is undecidable} \]

We will prove this theorem shortly, but first a “smaller” theorem:

\[ \mathcal{A}_{TM} \text{ is Turing-recognizable} \]
The following Turing machine $U$ recognizes $A_{TM}$

$U =$ “On input $\langle M, w \rangle$, where $M$ is a TM and $w$ is a string:
1. Simulate $M$ on input $w$.
2. If $M$ ever enters its accept state, accept; if $M$ ever enters its reject state, reject.”

- This machine loops on input $\langle M, w \rangle$ if $M$ loops on $w$
- $U$ is called *universal TM* because it is capable of simulating any other TM from the description of that machine
The diagonalization method

• The proof of the undecidability of $A_{TM}$ uses a technique called diagonalization.

• Diagonalization was discovered by mathematician Georg Cantor in 1873.

• Cantor was concerned with the problem of measuring the sizes of infinite sets.

• If we have two infinite sets, how can we tell whether one is larger than the other or whether they are of the same size?
Counting elements of an infinite set…

• Cantor observed that two finite sets have the same size if the elements of one set can be paired with elements of the other set.

• The idea can extended to infinite sets
Pairing: some technical definitions

Suppose we have sets $A$ and $B$ and a function $f$ from $A$ to $B$.

- $f$ is **one-to-one** if it never maps two different elements to the same place -- i.e., $f(a) \neq f(b)$ whenever $a \neq b$

- $f$ is **on to** if it hits every element of $B$ -- i.e., for every $b \in B$ there is an $a$ in $A$ such that $f(a) = b$

- $f$ is a **correspondence** if it is both one-to-one and on to
Pairing: some technical definitions

- We say that sets $A$ and $B$ have the same size if there is a correspondence $f: A \rightarrow B$.

- In a correspondence, every element of $A$ maps to a unique element of $B$ and each element of $B$ has a unique element of $A$ mapping to it.

- A correspondence is simply a way of pairing the elements of $A$ with the elements of $B$. 
Example

- Let:
  - $N$ be the set of natural numbers $\{1, 2, 3, \ldots\}$
  - $E$ be the set of even natural numbers $\{2, 4, 6, \ldots\}$

- Using Cantor’s definition of size,
  - $N$ and $E$ have the same size

- The correspondence $f$ mapping $N$ to $E$ is simply $f(n) = 2n$

<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>

- Intuitively, $E$ seems smaller than $N$ because $E$ is a proper subset of $N$
- Yet, pairing each member of $N$ with its own member of $E$ is possible
  Strange, but true!
Countable sets

- **Definition:**
  A set $A$ is **countable** if either it is **finite** or it has the **same size** as the set of natural numbers $\mathbb{N}$.

- **An even stranger example:**
  - Let $Q = \{m/n \mid m, n \in \mathbb{N}\}$ be the set of positive rational numbers.
  - $Q$ seems to be much larger than $\mathbb{N}$, yet $Q$ and $\mathbb{N}$ are of the **same size**!
  - Need to give a correspondence with $\mathbb{N}$ to show that $Q$ is countable.
Counting elements of $\mathbb{Q}$

we make an infinite matrix containing all the positive rational numbers ($\mathbb{Q}$)

row $i$ has all numbers with numerator $i$

column $j$ has all numbers with denominator $j$

we turn this matrix into a list by going along the diagonals as shown in the picture
Uncountable sets

- For some infinite sets, no correspondence with \( \mathbb{N} \) exists.
- Such sets are called **uncountable**.
- The set of **real numbers** \( \mathbb{R} \) is an example of an uncountable set.
- Cantor proved that \( \mathbb{R} \) is uncountable.
- In doing so, he introduced the diagonalization method.
To show that \( \mathbb{R} \) is uncountable, we show that no correspondence exists between \( \mathbb{N} \) and \( \mathbb{R} \).

The proof is by contradiction.

Suppose a correspondence \( f \) existed between \( \mathbb{N} \) and \( \mathbb{R} \).

Our job is to show that \( f \) fails to work as it should.

For it to be a correspondence, \( f \) must pair all the members of \( \mathbb{N} \) with all the members of \( \mathbb{R} \).

But we still find an \( x \) in \( \mathbb{R} \) that is not paired with anything in \( \mathbb{N} \), which will be a contradiction.
Finding $x$

- We find this $x$ by actually *constructing* it.

- We choose each digit of $x$ to make $x$ *different from one* of the real numbers that is paired with an element of $\mathbb{N}$.

- In the end, we are sure that $x$ is different from *any* real number that is paired.
We construct the desired $x$ by giving its decimal representation.

Our objective is to ensure that $x \neq f(n)$ for any $n$.

To ensure that $x \neq f(1)$, we let the first digit of $x$ be anything different from the first fractional digit of $f(1) = 3.14159\ldots$ Arbitrarily, let it be 4.

To ensure that $x \neq f(2)$, we let the second digit of $x$ be anything different from the second digit of $f(2) = 5.5555\ldots$ Arbitrarily, let it be 6.

Continue in this fashion…
The theorem that \( \mathbb{R} \) is uncountable – and its proof using the diagonalization method – has an important application to the theory of computation.

It shows that some languages are not decidable or even Turing recognizable because there are uncountably many languages yet countably many Turing machines.

Because each TM can recognize a single language and there are more languages than TMs, some languages are not recognized by any TM.

Next lecture, we will see a proof for the statement that some languages are not Turing-recognizable.