Gödel Incompleteness

Plan
We saw last time

- "COMPUTABLE" = computable by a TM
- \( L_{\text{Accept}}, L_{\text{Halt}} \) are UNDECIDABLE.
  i.e. \( \exists \text{TM to decide if } (\langle M \rangle, x) \in L \)

- Strings are COUNTABLE
  Languages are UNCOUNTABLE

The 1. Reals in \([0, 1]\) are UNCOUNTABLE
Pf. Suppose not. I an ordering

\[ 0 \]
\[ 0.00 \]
\[ \vdots \]
\[ 0.45 \ldots \]
Consider \( i \)th position of \( i \)th number is 0

\[
\begin{array}{llll}
Y & 0 & 1 & 0.5 \\
0 & 0 & 0 & 0.5
\end{array}
\]

Let the index for \( Y \) be \( j \).

What is \( Y \) in \( j \)th position?

- If \( Y_j = 0 \) then it is 1
- \( \neq 0 \) 

\( \exists \) exist an ordering.

One more example of an UNDECIDABLE language for any string \( x \) from \( \Sigma^* \),

KOLMOGOROV COMPLEXITY

\[ K(x) = \min \{ |M| + |y| : \text{TM } M \text{ on input } y \text{ outputs } x \} \]
\( L_k = \{ x \mid k(x) \geq \log n \} \)

**Lemma.** \( L_k \) is an infinite language.

**Pf.** For distinct \( x_1, x_2 \), we must have distinct \( \langle M, y_1 \rangle, \langle M, y_2 \rangle \).

**Theorem 2.** \( L_k \) is UNDECIDABLE.

**Pf.** Suppose \( \exists TM T \) that decides \( L_k \).

Consider \( TM M \):

- **on input integer** \( n \),
- **enumerate all strings** \( x \) of length \( n \)
- **check if** \( T \) **accepts** \( x \) (i.e. \( x \in L_k \))
- **output the first** \( x \) **that** \( T \) **accepts**.

Then, \( \forall n \), \( M \) outputs \( x \) s.t. \( k(x) \geq |x| \)

and \( |x| = n \).

But by definition, \( k(x) \leq |\langle M \rangle| + \log n \).
\[ K(X) \leq 1^{|M|} + x \operatorname{log} n \]

i.e. \( n \leq C + \log n \)

which is a contradiction for \( n \) larger than a fixed number. So \( T \) does not exist.

\( L_K \) is not decidable.

---

**Incompleteness**

In [GÖDEL] any sufficiently powerful axiomatic system is either inconsistent or incomplete.

**Sufficiently powerful**: can encode Turing Machines

**Inconsistent**: a statement that can both be proved and disproved

**Incomplete**: a statement that can neither be disproved nor proved.

(a "true" statement that cannot be proved)

**Pf.** \( \text{HALT}(M, x) : \text{"TM } M \text{ halts on input } x" \)
Pf. \( \text{HALT}(M,x) \): "TM \( M \) halts on input \( x \)."

Assume that \( \text{HALT}(M,x) \) can be expressed in a given axiomatic system.

Suppose that:
- if TM \( M \) halts on \( x \) then there is a proof of \( \text{HALT}(M,x) \).
- if TM \( M \) does not halt on \( x \), there is a proof of \( \overline{\text{HALT}(M,x)} \).

Consider the following TM \( D \):

On input \( <M,x> \):

- Start at \( k=1 \)
  - Enumerate candidate proofs \( P \) of length \( k \)
    - If \( P \) is a proof of \( \text{HALT}(M,x) \), accept.
    - If \( P \) is a proof of \( \overline{\text{HALT}(M,x)} \), reject.
  - Set \( k=k+1 \).

Run assumption that there exists a proof for
By assumption that there exits a proof for either \( \text{HALT}(M,x) \) or \( \overline{\text{HALT}}(M,x) \), \( D \) will decide \( \text{HALT} \). Contradiction.

Another proof, using Kolmogorov complexity.

Th. Any consistent axiomatic system that can express "\( K(x) \geq n \)" is incomplete.

Pf. \( L_k = \{ x : K(x) \geq 1x1 \} \) is undecidable.

Suppose \( \Delta \) a consistent axiomatic system that can express "\( K(x) \geq n \)" and \( \Delta \) is consistent and complete. I.e., whenever \( K(x) \geq n \), there is a proof.

Assume that such a proof can be verified in the system.

Consider a TM \( M \) that on input integer \( n \),

- Enumerate integers \( \langle A, \phi \rangle \)

\[ \vdots \]
- Enumerate integers \((x, y)\)
  - Enumerate strings \(x\) of length \(s\) and proofs \(p\) of length \(p\)
  - Check if \(p\) is a proof that \(K(x) \geq n\)
  - If so, output \(x\) and stop.

For any \(n\), \(M\) outputs \(x\) s.t. \(K(x) \geq n\)

i.e. \(n \leq K(x) \leq |M| + \log n \leq C + \log n\)

\(\exists n_0\) s.t. \(M\) fails for all \(n \geq n_0\).

(e.g. \(n \geq 2C\))

i.e. \(\exists x\) s.t. \(K(x) \geq n\) has no proof.

This is surprising since for any \(n\), there are many strings of length \(N + 1\) that have \(K(x) \geq n\). Just pick \(x\) at random from strings of length \(N + 1\).!!
from strings of length