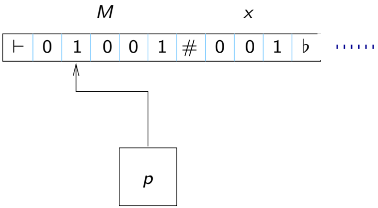
Undecidability of the Halting Problem

Meenakshi D'Souza

International Institute of Information Technology
Bangalore.

Term I 2022-23

Universal Turing machine



- A Universal Turing Machine is a TM that can simulate another TM whose description is presented as a part of its input.
- We can construct a TM U that takes the encoding of a TM M and its input x, and "interprets" M on the input x.
- U accepts if M accepts x, rejects if M rejects x, and loops if M loops on x.

Encoding a TM as a $\{0, 1\}$-string

$$0^n10^m10^k10^s10^t10^r10^u10^v\ 1\ 0^p10^a10^q10^b10\ 1\ 0^{p'}10^{a'}10^{q'}10^{b'}100\ \cdots 1\ 0^{p''}10^{a''}10^{q''}10^{b''}10.$$

represents a TM M with

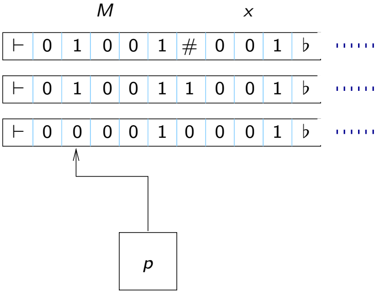
- states $\{1, 2, \ldots, n\}$.
- Tape alphabet $\{1, 2, \ldots, m\}$.
- Input alphabet $\{1, 2, \ldots, k\}$, $k < m$.
- Start state $s \in \{1, 2, \ldots, n\}$.
- Accept state $t \in \{1, 2, \ldots, n\}$.
- Reject state $r \in \{1, 2, \ldots, n\}$.
- Left-end marker symbol $u \in \{k + 1, \ldots, m\}$.
- Blank symbol $v \in \{k + 1, \ldots, m\}$.
- Each string $0^p10^a10^q10^b10$ represents the transition $(p, a) \to (q, b, L)$.

Example encoding of TM and its input

Input is encoded as $0^a10^b10^c$ etc.
What does the following TM do on input 001010?

000100001001010010001000010000 1 01000101000100 1 0100100100100 1 010101010.

Working of a universal Turing machine



- Use 3 tapes: for input $M\#x$, for current configuration, and for current state and position of head.
- Repeat:
  - Execute the transition of M applicable in the current config.
- Accept if M gets into t state, Reject if M gets into r state.
- $L(U) = \{M\#x \mid x \in L(M)\}$.

Diagonalization

- A technique used by Cantor 1874!
- Cantor's Theorem: There **does not** exist a one-to-one correspondence between the natural numbers $\mathbb{N}$ and its power set $2^{\mathbb{N}} = \{A \mid A \subseteq \mathbb{N}\}$.
- In fact, we can show that there **does not** even exist a function $f : \mathbb{N} \to 2^{\mathbb{N}}$ that is onto.

Diagonalization

- Proof is by contradiction: Suppose such an onto function f did exist.
- Define an infinite two-dimensional matrix indexed along the top by natural numbers $0, 1, 2, \ldots$ and down the left by the sets $f(0), f(1), \ldots$.
- The matrix is filled by placing a 1 in position i, j if j is in the set $f(i)$ and 0 if $j \notin f(i)$.

Cantor's Proof

Example:

	0	1	2	3	4	5	6	7	8	$\cdots$
$f(0)$	1	0	0	1	1	0	1	0	1	$\cdots$
$f(1)$	1	1	0	1	0	0	0	1	0	$\cdots$
$f(2)$	0	0	0	1	1	1	1	1	1	$\cdots$
$f(3)$	1	0	1	0	0	0	0	0	0	$\cdots$
$f(4)$	0	0	1	1	1	1	1	0	1	$\cdots$
$f(5)$	1	1	0	1	1	1	0	0	0	$\cdots$
$f(6)$	0	0	1	1	1	1	1	1	1	$\cdots$
$f(7)$	0	0	1	0	0	0	1	1	1	$\cdots$
$\vdots$										

- The ith row of the matrix is a bit string describing the set $f(i)$.
- Since f is onto (by our assumption), every subset of $\mathbb{N}$ appears as a row of this matrix.

Cantor's Proof

- Look at the infinite bit string down the main diagonal of the matrix and take its Boolean complement.
- This new bit string represents a set B that does not appear anywhere in the list down the left side of the matrix, since it differs from every $f(i)$ on the element i.
  - In this example, it will be the string $00110000\ldots$ representing the set $B = \{2, 3, \ldots\}$.
- This is a contradiction, since every subset of $\mathbb{N}$ was supposed to occur as a row of the matrix, by our assumption that f was onto.

Diagonalization: Generalized

- Cantor's theorem can be generalized to any set A.
- Suppose (for a contradiction) there existed an onto function $f : A \to 2^A$. Let

$$B = \{x \in A \mid x \notin f(x)\}$$

  Then $B \subseteq A$.
  Since f is onto, there must exist $y \in A$ such that $f(y) = B$.
  Now,

$$y \in f(y) \Leftrightarrow y \in B \Leftrightarrow y \notin f(y).$$

  Thus, no such f can exist.

Halting Problem for Turing machines

- Fix an encoding *enc* of TM's as above.
- Define the language

$$\mathrm{HP} = \{enc(M)\#enc(x) \mid M \text{ halts on } x\}.$$

Undecidability of HP

Theorem (Turing)

The language HP *is not recursive.*

Proving undecidability of HP

Assume that we have a Turing machine M which decides HP. Then we can compute the entries of the table below:

	ϵ	0	1	00	01	10	11	000	001	010	011	111	$\cdots$
M_ϵ	L	H	L	L	L	H	H	L	L	L	L	L	$\cdots$
M_0	L	L	L	L	L	L	L	L	L	L	L	L	$\cdots$
M_1	H	H	L	H	L	H	H	L	L	H	L	H	$\cdots$
M_{00}	L	L	L	L	L	L	L	L	L	L	L	L	$\cdots$
M_{01}	L	H	L	L	L	H	H	L	L	L	L	L	$\cdots$
M_{10}	H	H	L	H	L	H	H	L	L	H	L	H	$\cdots$
M_{11}	L	H	L	L	L	H	H	L	L	L	L	L	$\cdots$
M_{000}	L	L	L	L	L	L	H	L	L	L	H	L	$\cdots$
$\vdots$													

- For each $x \in \{0,1\}^*$ let M_x denote the TM
  - M, if x is the encoding of TM M with input alphabet $0, 1$.
  - M_{loop} otherwise, where M_{loop} is 1-state turing machine that loops on all its inputs.

A TM N that behaves differently from all TM's

Let us assume we have a TM M that decides HP. Then we can define a TM N as follows: Given input $x \in \{0,1\}^*$, it

- runs as M on $M_x\#x$.
- If M accepts (i.e. M_x halts on x), goes to a new "looping" state and loops there.
- If M rejects (i.e. M_x loops on x), goes to the accept state.

N essentially "complements the diagonal" of the table: Given input $x \in \{0,1\}^*$ it halts iff M_x loops on x.
Consider $y = enc(N)$. Then y cannot occur as any row of the table since the behaviour of N differs from all rows in the table. This is a contradiction.

Complement of HP is not r.e.

Fact: If L and $\overline{L}$ are both r.e. then L (and $\overline{L}$) must be recursive.

- Let M accept L and M' accept $\overline{L}$.
- We can construct a total TM that simulates M and M' on given input, one step at a time.
- Accept if M accepts, Reject if M' accepts.

Corollary

The language $\neg$HP *is not even recursively enumerable.*

Where HP lies