

Undecidability of the Halting Problem

Deepak D'Souza

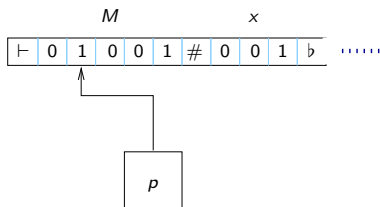
Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

16 November 2015

Outline

- 1 Universal Turing machine
- 2 Halting Problem
- 3 Some corollaries

Universal Turing machine



- 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

$0^n 10^m 10^k 10^s 10^t 10^r 10^u 10^v 1 0^p 10^a 10^q 10^b 10 1 0^{p'} 10^{a'} 10^{q'} 10^{b'} 100 \dots 1 0^{p''} 10^{a''} 10^{q''} 10^{b''} 10$.

represents a TM M with

- states $\{1, 2, \dots, n\}$.
- Tape alphabet $\{1, 2, \dots, m\}$.
- Input alphabet $\{1, 2, \dots, k\}$ (with $k < m$).
- Start state $s \in \{1, 2, \dots, n\}$.
- Accept state $t \in \{1, 2, \dots, n\}$.
- Reject state $r \in \{1, 2, \dots, n\}$.
- Left-end marker symbol $u \in \{k + 1, \dots, m\}$.
- Blank symbol $v \in \{k + 1, \dots, m\}$.
- Each string $0^p 10^a 10^q 10^b 10$ represents the transition $(p, a) \rightarrow (q, b, L)$.

Example encoding of TM and its input

Input is encoded as $0^a10^b10^c$ etc.

Exercise: What does the following TM do on input 001010?

Example encoding of a TM

```
00010000100101001000100010000 1 01000101000100 1 0100100100100 1 010101010.
```

[Assume accept and reject states are sink states]

Example encoding of TM and its input

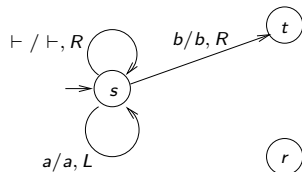
Input is encoded as $0^a 10^b 10^c$ etc.

Exercise: What does the following TM do on input 001010?

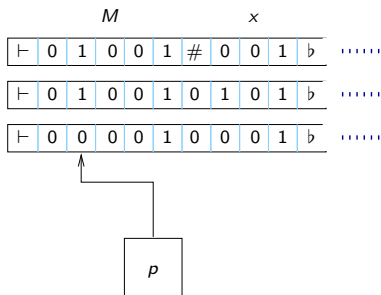
Example encoding of a TM

00010000100101001000100010000 1 01000101000100 1 0100100100100 1 010101010.

[Assume accept and reject states are sink states]



How the universal Turing machine works



- 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.

Halting Problem for Turing machines

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

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

Undecidability of HP

Theorem (Turing 1936)

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	...
M_ϵ	L	H	L	L	L	H	H	L	L	L	L	L	...
M_0	L	L	L	L	L	L	L	L	L	L	L	L	...
M_1	H	H	L	H	L	H	H	L	L	H	L	H	...
M_{00}	L	L	L	L	L	L	L	L	L	L	L	L	...
M_{01}	L	H	L	L	L	H	H	L	L	L	L	L	...
M_{10}	H	H	L	H	L	H	H	L	L	H	L	H	...
M_{11}	L	H	L	L	L	H	H	L	L	L	L	L	...
M_{000}	L	L	L	L	L	L	H	L	L	L	H	L	...
⋮													
⋮													

- 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 a one-state Turing machine that loops on all its inputs.
- Table entry (x, y) tells whether TM M_x halts on the input y . Note that y is an (unencoded) input in $\{0, 1\}^*$.

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 $x\#enc(x)$.
 - If M accepts (i.e. M_x halts on x), goes to a new “looping” state l and loops there.
 - If M rejects (i.e. M_x loops on x), goes to the accept state t' .
- 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.

How N behaves

	ϵ	0	1	00	01	10	11	000	001	010	011	111	...
M_ϵ	L	H	L	L	L	H	H	L	L	L	L	L	...
M_0	L	L	L	L	L	L	L	L	L	L	L	L	...
M_1	H	H	L	H	L	H	H	L	L	H	L	H	...
M_{00}	L	L	L	L	L	L	L	L	L	L	L	L	...
M_{01}	L	H	L	L	L	H	H	L	L	L	L	L	...
M_{10}	H	H	L	H	L	H	H	L	L	H	L	H	...
M_{11}	L	H	L	L	L	H	H	L	L	L	L	L	...
M_{000}	L	L	L	L	L	L	H	L	L	L	H	L	...
⋮													
N	H	H	H	H	H	L	L	H	...				
⋮													

The constructed TM N **complements** the diagonal of the table, and hence does not occur as any of the TM's listed. This is not possible!

Complement of HP is not r.e.

Fact 1: If L and \bar{L} are both r.e. then L (and \bar{L}) must be recursive.

- Let M accept L and M' accept \bar{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.

Fact 2: HP is recursively enumerable.

- Just run the universal TM U on input $M\#x$; accept iff U halts (i.e. M accepts or rejects x).

Corollary

The language \neg HP is not even recursively enumerable.

Where HP lies

