# Universal Turing Machines: Diagonalization and Undecidability

U ↳ Given the description of any TM M and an input x for M, runs M on x, and decides whatever M does.

- M accepts x by going to state t, and halts.
- M explicitly rejects x by going to state r, and halts.
- M loops on x. M never enters t or r.

$(Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$ ← a description of M

Given this description of M and the input x for M, U simulates M on x.

Encoding of Turing machines
↳ should be understood by U

How to encode $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$

finite binary encoding

$|Q| = n, \quad |\Gamma| = m, \quad |\Sigma| = k$

$Q = \{0, 1, 2, \ldots, n-1\}$

$\Gamma = \{0, 1, 2, \ldots, m-1\}, \quad \Sigma = \{0, 1, 2, \ldots, k-1\}$

$\vdash \in \{0, \ldots, m-1\}, \quad \sqcup \in \{0, \ldots, m-1\}$

$s, t, r \in \{0, 1, 2, \ldots, n-1\}$

Binary encoding

$0^n 1 0^m 1 0^k 1 0^\ell 1 0^b 1 0^s 1 0^t 1 0^r 1$ ← start

$\delta(p, a) = (q, b, d)$ $\quad d \in \{L, R\}$, $L \to 0$, $R \to 1$

$0^p 1 0^a 1 0^q 1 0^b 1 0$ ← write the transitions one after another

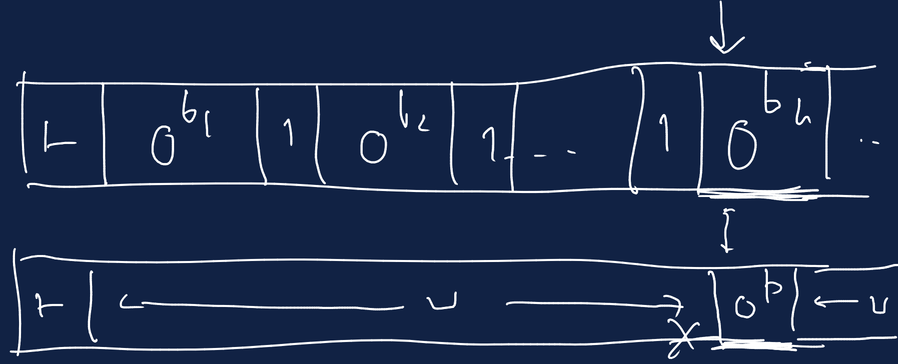
$0 \to L$, $1 \to R$

M can be encoded in binary

$x = a_1 a_2 \cdots a_n$ (input for M)

$a_i \in \Sigma = \{0, 1, 2, \ldots, k-1\}$

$0^{a_1} 1 0^{a_2} 1 0^{a_3} 1 \cdots 1 0^{a_n} 1$




$M, x \in \{0, 1\}$

$\# \notin \{0, 1\}$

Input for U : M # x

Simulate M on x

$0, 1, \#$

00 01 10 11

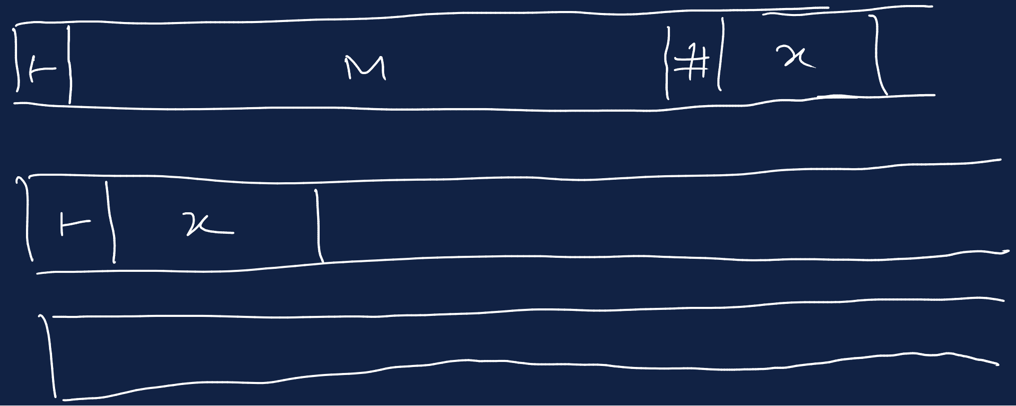
$\delta(p, a)$

U → 3-tape machine

Tape 1

Tape 2

Tape 3




← simulate the tape of M

← state of M, head position

M goes to $t$ $\longrightarrow$ U detects this $\rightarrow$ accept and halt

M goes to $r$ $\longrightarrow$ U detects this $\rightarrow$ rejects and halt

M loops $\longrightarrow$ Simulation by U also loops

Blind simulation

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

membership problem

$$\boxed{MP \text{ is r.e.}}$$

$$\boxed{MP \text{ is not recursive.}}$$

Blind simulation loops if M loops.

Is there a more intelligent way than blind simulation to decide whether M halts on $x$?

$HP = \{ M \# x \mid M \text{ halts on } x \}$

(halting problem)

If HP has a decider, then MP also has a decider.

(not if and only if)

$\boxed{\text{HP is undecidable.}}$

This does not immediately imply MP is undecidable.

Proof: Use diagonalization.

$\Sigma^* \subseteq \{0,1\}^*$ is countable

$M \in \{0,1\}^*$

Set of all encodings of TMs is $\subseteq \{0,1\}^*$ and so is also countable

Invalid encoding of M and/or $x$

$\underline{M}$ is a TM that immediately rejects and halts.

$\underline{x}$ is invalid $\longrightarrow$ $x = \epsilon$

Every string is the encoding of a TM.
Every string is the encoding of an input.
(binary)

$\{0,1\}^*$ is countable.

$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ldots$ an exhaustive enumeration of $\{0,1\}^*$

$x_1 \quad x_2 \ldots x_n \ldots$

$M_1$
$M_2$
$\vdots$
$M_n$
$\vdots$

$\alpha_i \longrightarrow M_i$

$\alpha_j \longrightarrow x_j$

$\cancel{H}$P has a decider D

Given $i$ and $j$,

D can decide whether the $(i,j)$-th entry is H or L.

---

Given D, construct $D'$ as follows.

$D'$ has only one input $i$

$D'$ runs $M_i$ on $x_i$.

If D says H, $D'$ enters a loop.

If D says L, $D'$ accepts and halts


$\boxed{D' = M_n}$

$D'$ halts on $x_n$

$\Leftrightarrow$ $D'$ loops on $x_n$

CONTRADICTION

D cannot exist.

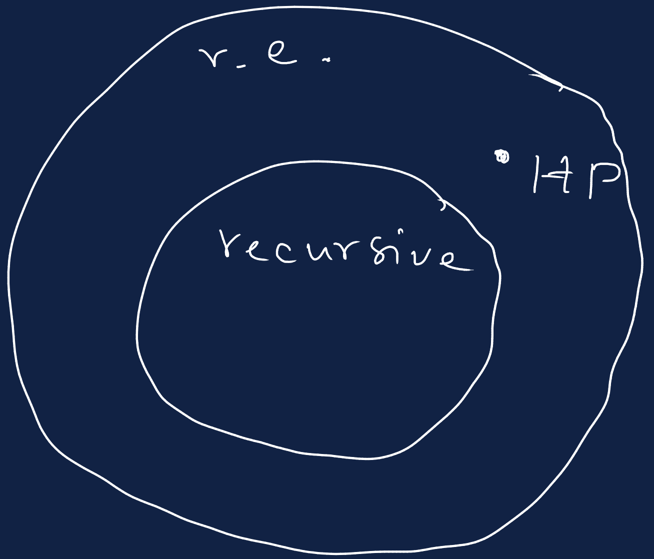
MP is r.e.

HP is r.e. → Blindly simulate M on $x$

— if M goes to $t$ or $r$, accept

All rejections are implicit.

HP is r.e. but not recursive.




MP is undecidable too.

non-recursive

→ Use a similar diagonalization argument (Exercise)

D

M#$x$ — Accept / Not accept