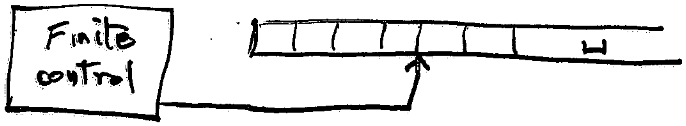
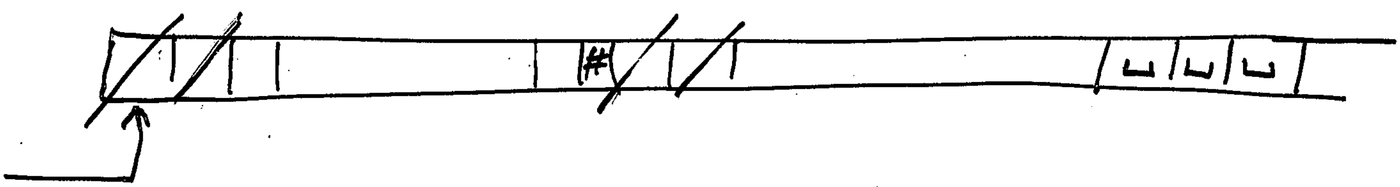
TURING MACHINES



TM can 1) read & write on tape
2) pointer (R/W head) moves L or R
3) Tape is ∞ (// in 1 direction)
4) Accept state, reject state : take effect immediately.

$B = \{ w \# w \mid w \in \{0,1\}^* \}$



TM : 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

Σ: input alphabet, cannot contain blank symbol, ⊔

Γ: tape alphabet $\sqcup \in \Gamma$, $\Sigma \subsetneq \Gamma$

δ: $Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

Config : $u q v$

$u a q_i b v \to u q_j a c v$ if $\delta(q_i, b) \to (q_j, c, L)$

$u a q_i b v \to u a c q_j v$ if $\delta(q_i, b) \to (q_j, c, R)$

start config: $q_0 w$

Defn : TM accepts w if $q_0 w \to C_1 \to C_2 \cdots \to C_k$ | C_k is a ~~halting~~ accept config. ~~(ie accept or reject config)~~

$A = \{ 0^{2^n} \mid n \geq 0 \}$ // Given, a w in unary notation, is it a power of 2?

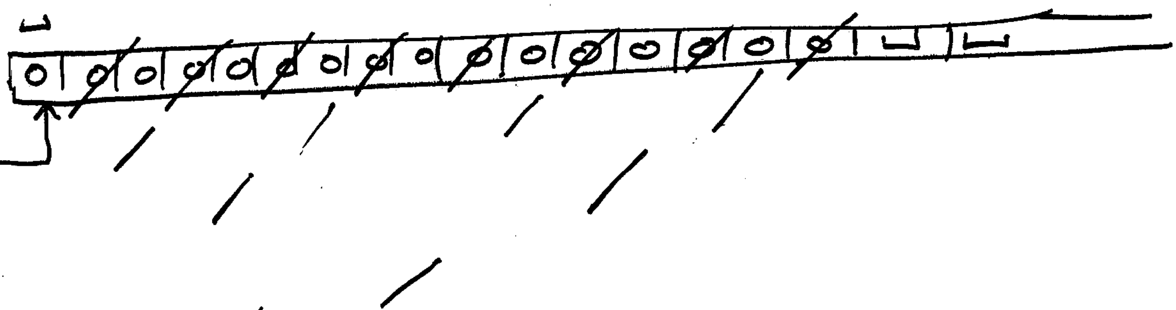
Iter 1

Iter 2

Iter 3

Iter 4

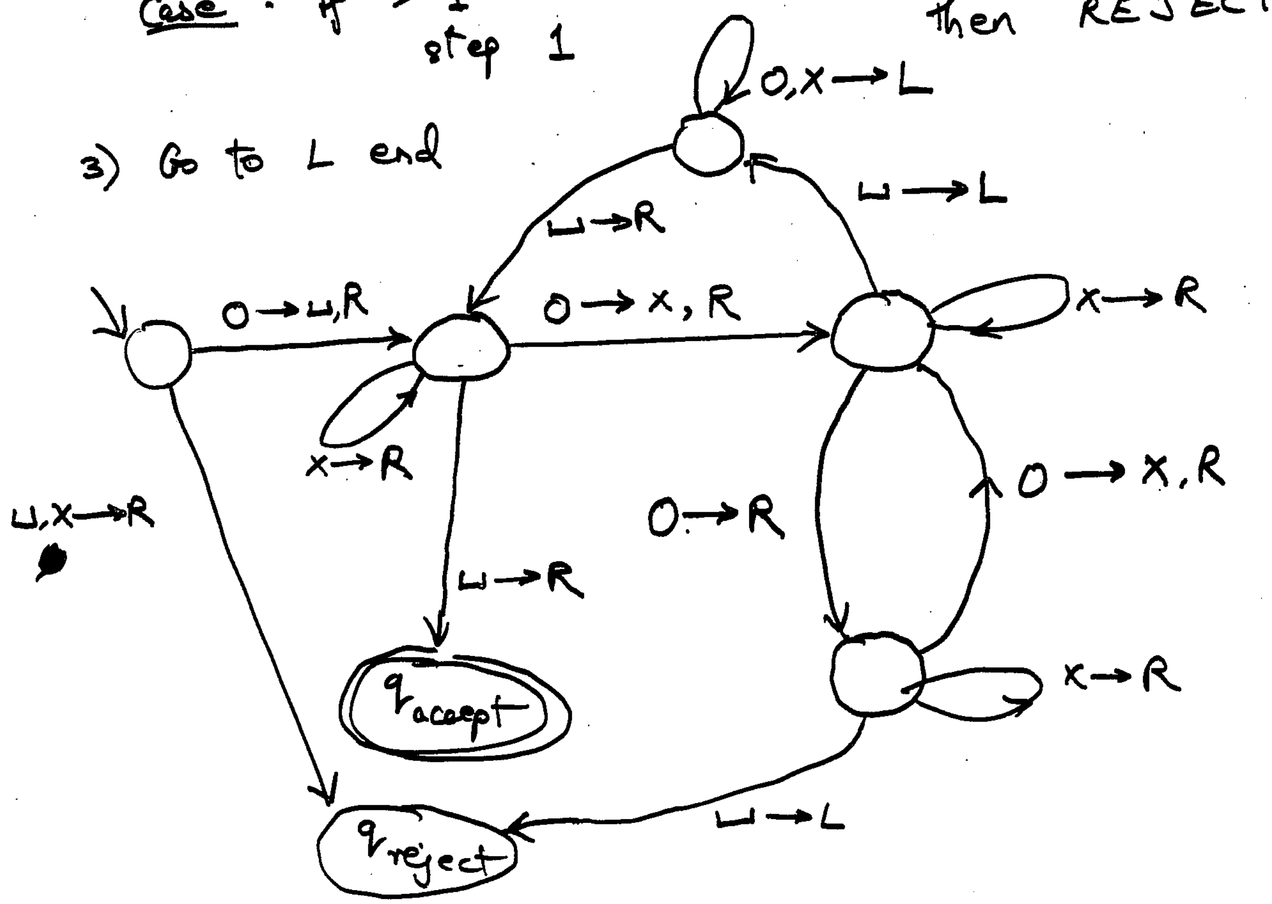
Iter 5 ⊔ & accept

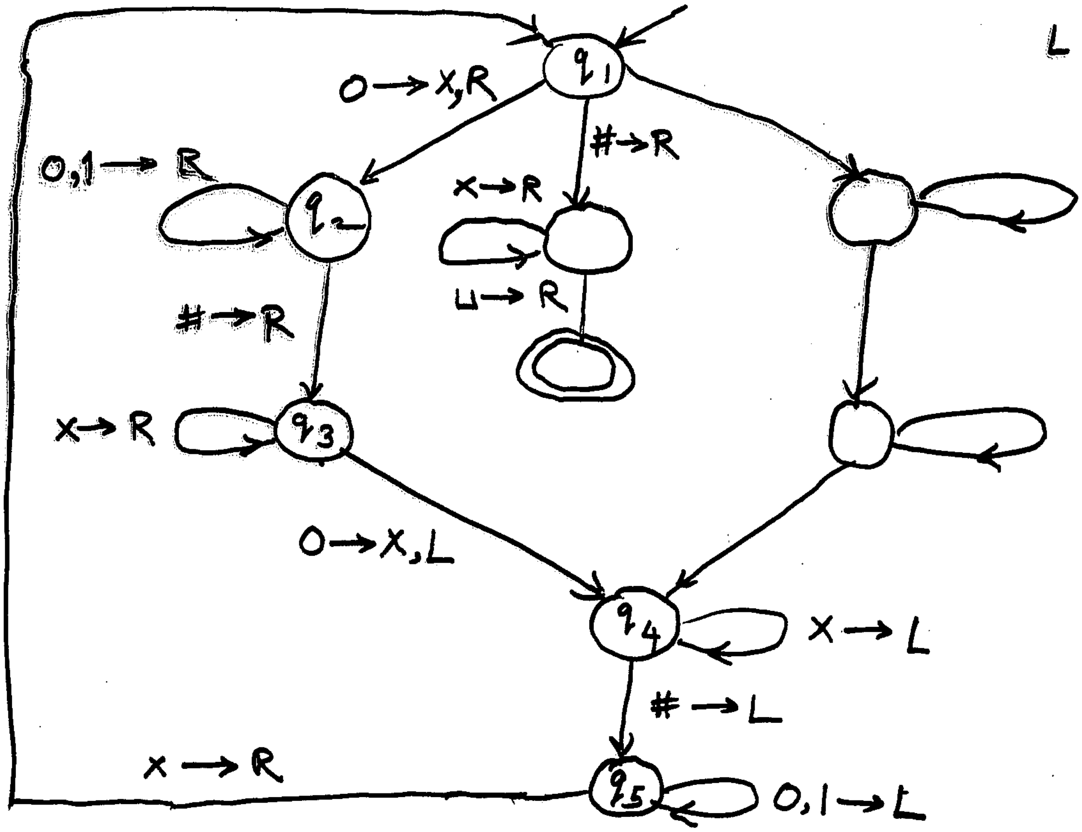


Input : w

Loop:

1) Go L → R, crossing out every 'other' 0
2) Case : if only 1 '0' in step 1 then ACCEPT
   Case : if > 1 and odd # '0's in step 1 then REJECT
3) Go to L end

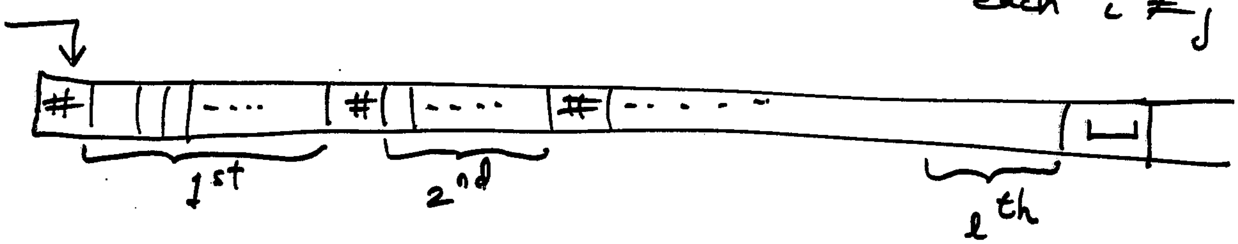




$L(M) = \{ w\#w \mid w \in \{0,1\}^* \}$

NB: Missing transition ⟷ transition to q_{reject}

$E = \{ \# x_1 \# x_2 \# \cdots x_\ell \mid$ each $x_i \in \{0,1\}^*$ & $x_i \neq x_j$ for each $i \neq j \}$



loop $i = 1$ to $\ell - 1$

  loop $j = i+1$ to ℓ

    if $x_i \neq x_j$ no-op

    else REJECT

ACCEPT

- Mark # by $\dot{\#}$ (new member of tape alphabet Γ)
- Allows you to unmark also // Simulating extra POINTERS

Defn: $L(M) = \{ w \mid M \text{ accepts } w \}$

Defn: L is <u>Turing-recognizable</u> if some TM recognizes L

<u>Observation:</u> A TM may accept, reject, or loop forever

Defn: L is <u>Turing-decidable</u> if some TM that decides L

Defn: TM <u>decides</u> L if it (is guaranteed to) enter

- q_{accept} or q_{reject}
- on each $w \in L$ and $w \notin L$

L_{174}

	$w \in L_{174}$	$w \notin L_{174}$
T-Decidable:	accept	reject
T-Recognizable:	accept	reject or keep looping
not-T-Recognizable:	accept or keep looping	reject or keep looping

VARIANTS OF TMs

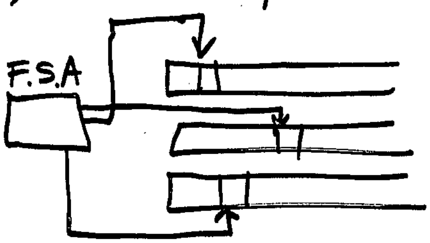
→ TM are ROBUST

eg N-TM, multi-tape TM, shift L/R by multiple positions, k-dim grid, 2-Dim ∞ tape, - - - .

Exception: doing ∞ number of actions in a single step

ORACLE

1) Multitape TM.

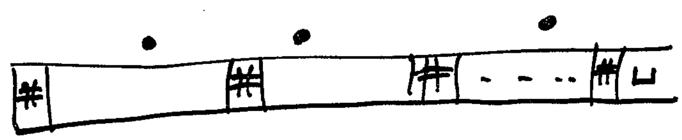


Theorem: k-tape TM $\longleftrightarrow$ 1-tape TM

$$\delta(q_i, a_1\ a_2 \cdots a_k) \longrightarrow (q_j, b_1\ b_2 \cdots b_k, L/R, L/R \cdots L/R)$$

$$\delta : Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{L, R\}^k$$

$\boxed{1}$ Initialize 1-tape TM :



$\boxed{2}$ Each Transition : 2-passes L-R:

① Read the k a_i s

② Write the k b_i s, shift L/R each & mark the new k pseudo-pointers

1-step of k-tape TM $\longleftrightarrow$ 2k steps of 1-tape TM

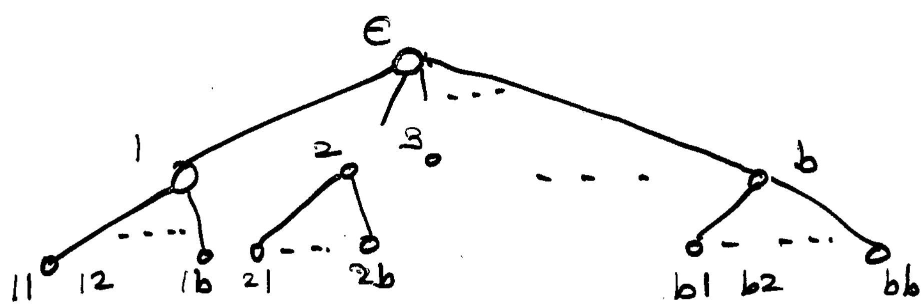
RESULT: Lang L is T-Recognizable iff some k-tape TM recognizes it.

Th: NON-DET. TM $\longleftrightarrow$ (D)-TM

Given a N-TM, simulate by D-TM

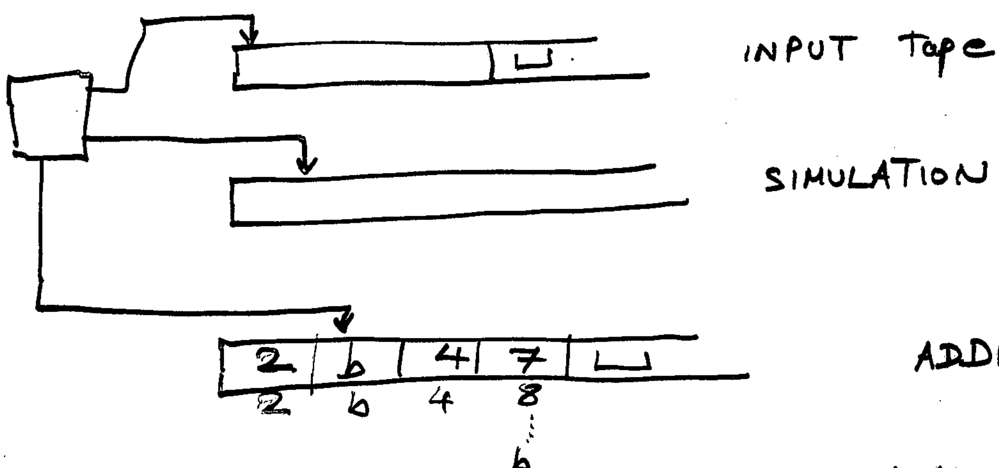
$$\delta_N : Q \times \Gamma \longrightarrow \mathcal{P}(Q \times \Gamma \times \{L,R\})$$

ACCEPT by N-TM $\equiv$ some branch (of the tree of possible computations allowed by the choice δ_N) leads to ACCEPT state



$$b = |Q| \times |\Gamma| \times |\{L,R\}|$$

D-TM uses 3-tapes



KEY: Use BFS (not DFS) of tree // obj: to search for ACCEPT state

eg Choice '2b47' $\equiv$ config resulting after choices #ed 2, b, 4, 7 in first 4 steps.

Use ADDRESS TAPE As a Counter-of-node-labels-in-BFS-traversal-.. of tree

Th : L is T-Recognizable iff some N-TM recognizes it

Defn: N-TM is a decider (for a language L) $\equiv$ $\forall w \in L$, (& $w \notin L$) all branches of (computation on) tree halts

Cor: L is decidable iff some N-TM decides it

ENUMERATER $\equiv$ TM + printer

$L(E) = \{ w \mid w \text{ printed} \}$

Theorem: Lang L is T-recognizable $\Rightarrow \Leftarrow$ some E enumerates Lang L

// L is arbitrary; cannot assume L is already "enumerated"

($\Leftarrow$) ie, show that, given E, build a TM that given any w, (w ∈ L) accepts w, ⊕ (w ∉ L) rejects w or loops

M: On input w:

1) Runs E. Each string 's' printed, compare s with w.
   if s = w, accept w
   else loop

($\Rightarrow$) Given TM ~~[illegible]~~, build E for L

E: Given w, ignore it. // Crank out all $w \in \Sigma^*$ // ie, $s_1, s_2, s_3, \ldots\ldots$

1) Loop for $i = 1, 2, 3, \ldots$
   Run TM for i steps on inputs $s_1, s_2, \ldots s_i$ only
   If any input is accepted by TM, print it (ie, that s)

ALGORITHMS.:

CHURCH-TURING THESIS

[Algorithms "seq of precise steps" $\longleftrightarrow$ Algorithm on a TM]

$x^2 - 8x + 15 = 0$

vs

$x^2 - 8x + 18 = 0$

$ax^2 + bx + c = 0$

roots $\dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

[Given a polynomial p, give an algorithm to tell if p has integral roots.

1900: Hilbert's 10^{th} problem

1970: this is UNSOLVABLE

$D_1 = \{ p \mid p$ is a polynomial over x with an integral root $\}$

D_1 is recognizable

M_1: p is input

1) Evaluate p with x set to $0, -1, +1, -2, +2, \ldots$
   If p evaluates to 0, accept.

roots of p over 1 variable $\in \left[-k \dfrac{c_{max}}{c_n}, k \dfrac{c_{max}}{c_n} \right]$

, k = # Terms in p, c_{max} = coeff with highest mod value.
c_n = coeff of x^n

$D = \{ p \mid p$ is a polynomial with an integral root$\}$

eg $27x^3y^7z^2 - 11x^5y^4z^8 \cdots = 0$

Build M (like M_1) that recognizes D.

D is T-recognizable (only)

D_1 is T-decidable

Descriptions of TMs

1) formal description (7-tuple) // assembly lang.

2) Implementation level (English prose describing TMs steps)

3) high-lvl (English prose, ignoring implementation model)

- How expressive are Langs (ie ~~all~~ sets of strings) i.t.o. describing problems?

  Very. Strings can ENCODE 'anything.' eg polynomial, graphs, grammars, automata,

  $\langle o \rangle \equiv$ encoding of object 'o'

$A = \{\langle G \rangle \mid G$ is connected undirected graph $\}$

TM decider of A:

M: Input $w = \langle G \rangle$:

0) Check for correctness of encoding

1) ~~Loop~~ Mark some node

2) Loop:

$\forall x$ unmarked, mark x if x has edge to marked node

until: no x gets marked

Implementation-lvl details

eg step 0] Encodings. $\langle G \rangle = (1,2,3)((1,3),(2,3),(1,2), \quad)$