Lecture 4 CSE 431 Intro to Theory of Computation

Turing Machines

Recall $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

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

configurations, $\vdash_M$ (yields in one step), $\vdash_M^*$ (yields)

M accepts w iff $q_0 w \vdash_M^* u q_{acc} v$ $\quad u, v \in \Gamma^*$

rejects w iff $q_0 w \vdash_M^* u q_{rej} v$. $\quad u, v \in \Gamma^*$

M is a decider iff $\forall w \in \Sigma^*$
M accepts w or M rejects w

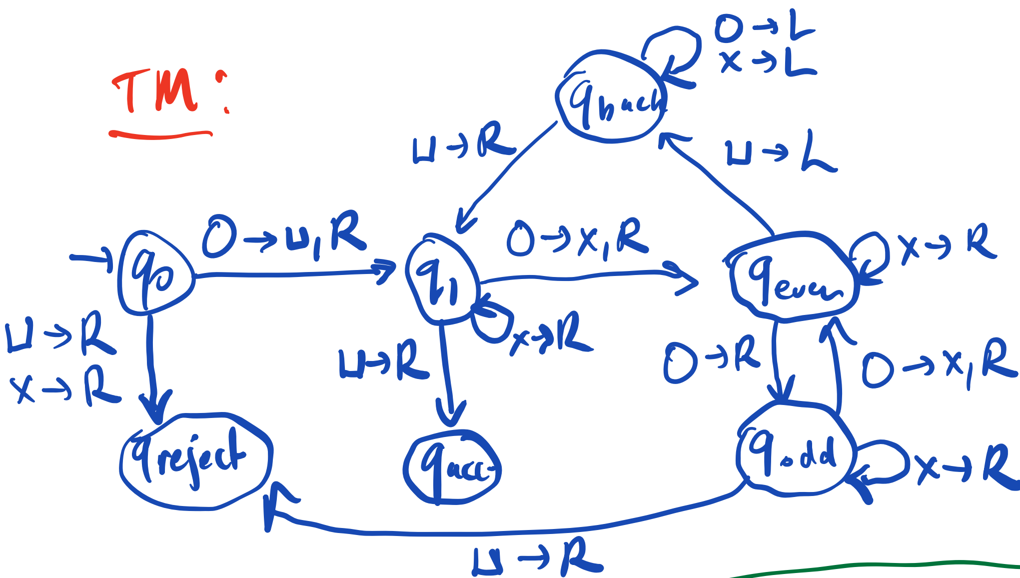
Example: $\{0^{2^n} : n \geq 0\}$ $\quad \Sigma = \{0\}$ input alphabet

Plan:
1. Check if one 0, if yes accept
2. If more than one 0, cross off every second 0 (if odd reject)
3. Repeat above with remaining 0's

eg $0\ \cancel{0}\ 0\ \cancel{0}\ 0\ \cancel{0}\ 0\ \sqcup$ reject

$0\ \cancel{0}\ 0\ \cancel{0}\ 0\ \cancel{0}\ 0\ \cancel{0}\ \sqcup$ need to mark start of the string to get back to mark start of string we could use a new tape symbol but to keep TM simple we use a blank ⊔ representing a 0 and the start.

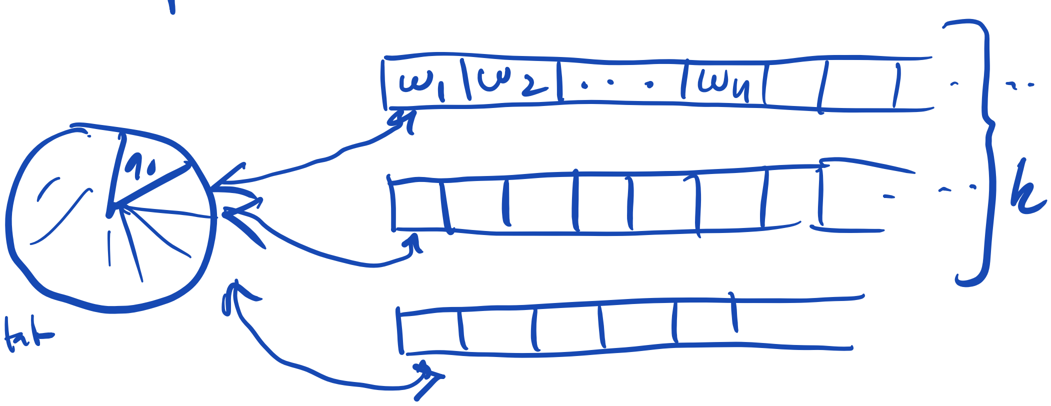
TM:



Notation: $p \xrightarrow{a \to R} q$ means $p \xrightarrow{a \to a, R} q$

Generalizations of TMs:

k-tape TM



Transitions based on all symbols scanned

Input on 1^{st} tape, rest start blank

Head movement independent.

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

Theorem Every k-tape TM M is equivalent to some 1-tape TM M'

equivalent:
- M' accepts w iff M accepts w
- M' rejects w iff M rejects w
- same input alphabet Σ.

Proof Basic idea:

represent all k tapes contents on a single tape.

Suppose $M = (Q, \Sigma, \Gamma, \delta, \ldots)$

we create $M' = (Q', \Sigma, \Gamma', \delta', \ldots)$

Let $\# \notin \Gamma$ be a new symbol

We represent all k tapes' contents in M' by

$\# \boxed{\text{tape } 1} \# \boxed{\text{tape } 2} \# \cdots \# \boxed{\text{tape } k} \#$

Since each tape is infinite we only represent the portion that is used. The start

We will represent the start tape contents

$\# w_1 w_2 \cdots w_n \# \sqcup \# \sqcup \# \quad \# \sqcup \#$

but we also need to store head position for each tape

We put a $\bullet$ over each char if it also has the head on on it.

$\Gamma' = \Gamma \cup \dot{\Gamma} \cup \#$?

(*) $\# \dot{w}_1 w_2 \cdots w_n \# \dot{\sqcup} \# \quad \cdots \# \dot{\sqcup} \#$

So.. first convert $w_1 \cdots w_n$ input to above string. (*)

To figure out what move to make, need to store scanned symbols in state.

- Do L to R sweep recording the symbols under the dots.

- To execute the moves for this step of M

  - sweep R to L and execute all the left moves

    (rewrite the symbol and dot the symbol just to the left
    - if that symbol was a #
    put that dot back on the first symbol of that tape)

  - sweep L to R and execute all the right moves most right moves are simple moving the dot one to the right, except when that is # (reached end of used portion of tape) and need to insert a blank symbol and shift the rest of the tape to the right during the sweep.

(to shift by c characters keep track of queue of c most recent chars (in state) reading at one end and writing from the other.)

- Return to the left end.

Note: If original machine ran for T steps (suppose $T \geq n$) new machine may take $O(kT^2)$ steps:

Original step costs = # cells in tapes

$\leq kT + k + 1$

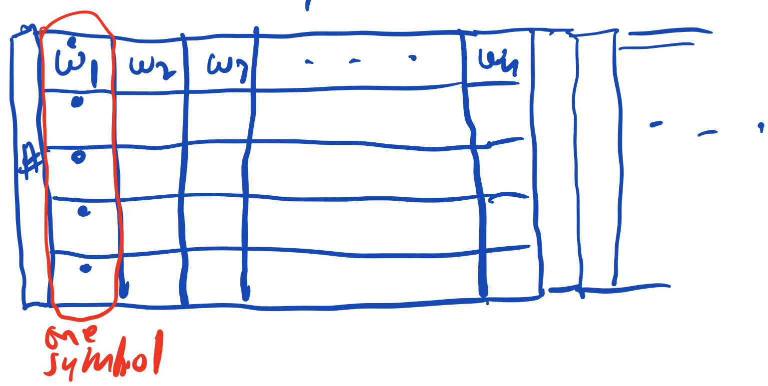
Total $O(kT^2)$

This simulation is step by step and the machine accepts

$\square$

Alternative simulation: Multitracked.

New set of symbols $\Gamma' = \Gamma \cup (\Gamma \cup \dot{\Gamma})^k$



each tape represented as a "track"

Behaviour is same as before:

Sweep L to R to collect scanned symbols

Sweep R to L doing L moves

Sweep L to R doing R moves (maybe make multitracked multi-dotted blank to replace blank)

return to L end.

Time: $O(T^2)$ no shifting required

Fact: twin converting 2-tapes to 1-tape best possible simulation is $O(T^2)$

Next time: Nondeterministic TMs.