CSE322 Theory of Computation (L~~18~~ 16)

$M \equiv M'$ $\quad L(M) = L(M')$

M accepts w iff M' accepts w

Variants of TM

TM on input w may · halt & accept
halt & reject
not halt (loop)

- Two-way infinite tape
- 2, 3, ..., K - finite # tapes
  → infinite # types · X
- non-deterministic moves
- multi-step TMs

TM with stationary move

$=\langle Q, \dots, \delta, \dots \rangle$

Given a TM M whose head movement belongs to {L,R,S} construct another TM N whose head movement belongs to {L,R} (usual type), such that...

$=\langle Q', \dots, \delta', \dots \rangle$

for any input x,

$Q' = Q \cup Q \times \{1\}$

M(x) halts iff N(x) halts and

M(x) accepts iff N(x) accepts.

$\delta'(q,a) = \begin{cases} \delta(q,a) & \text{if } \delta(q,a) = (\dots, L/R) \\ ((r,1), b, R) & \text{if } \delta(q,a) = (r, b, S) \end{cases}$

$\delta'((r,1),a) = (r,a,L)$

Show how to simulate one move of M by a sequence of moves of N.

Multitape TM $= \langle Q, \Sigma, \Gamma, \delta, q_0, q_a, q_{rej} \rangle$

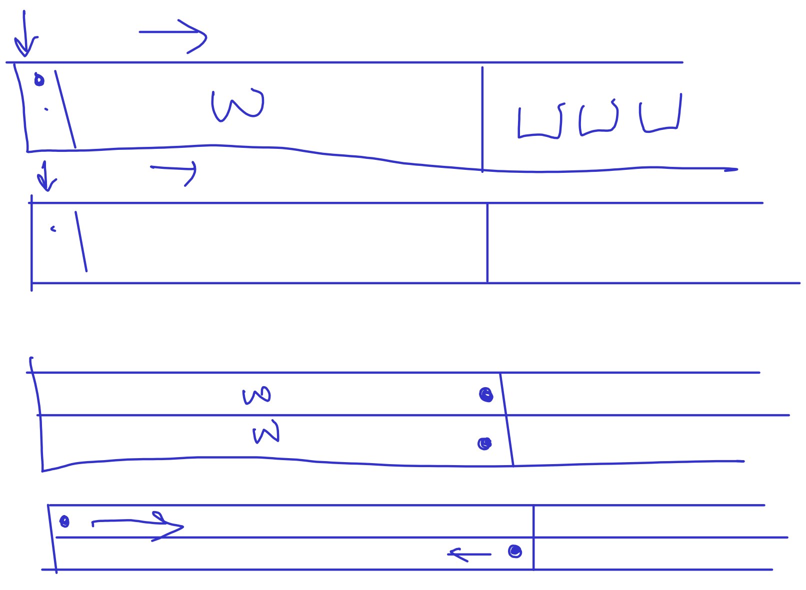
k = Number of tapes, each with independent read/write head

Input on first tape, other tapes start empty

d : $(Q \times \Gamma^K)$ -> $Q \times \Gamma^K \times \{L, R\}^K$

$= \{ w : w = rev(w) \}$

Deciding PALINDROME using 2-tape TM



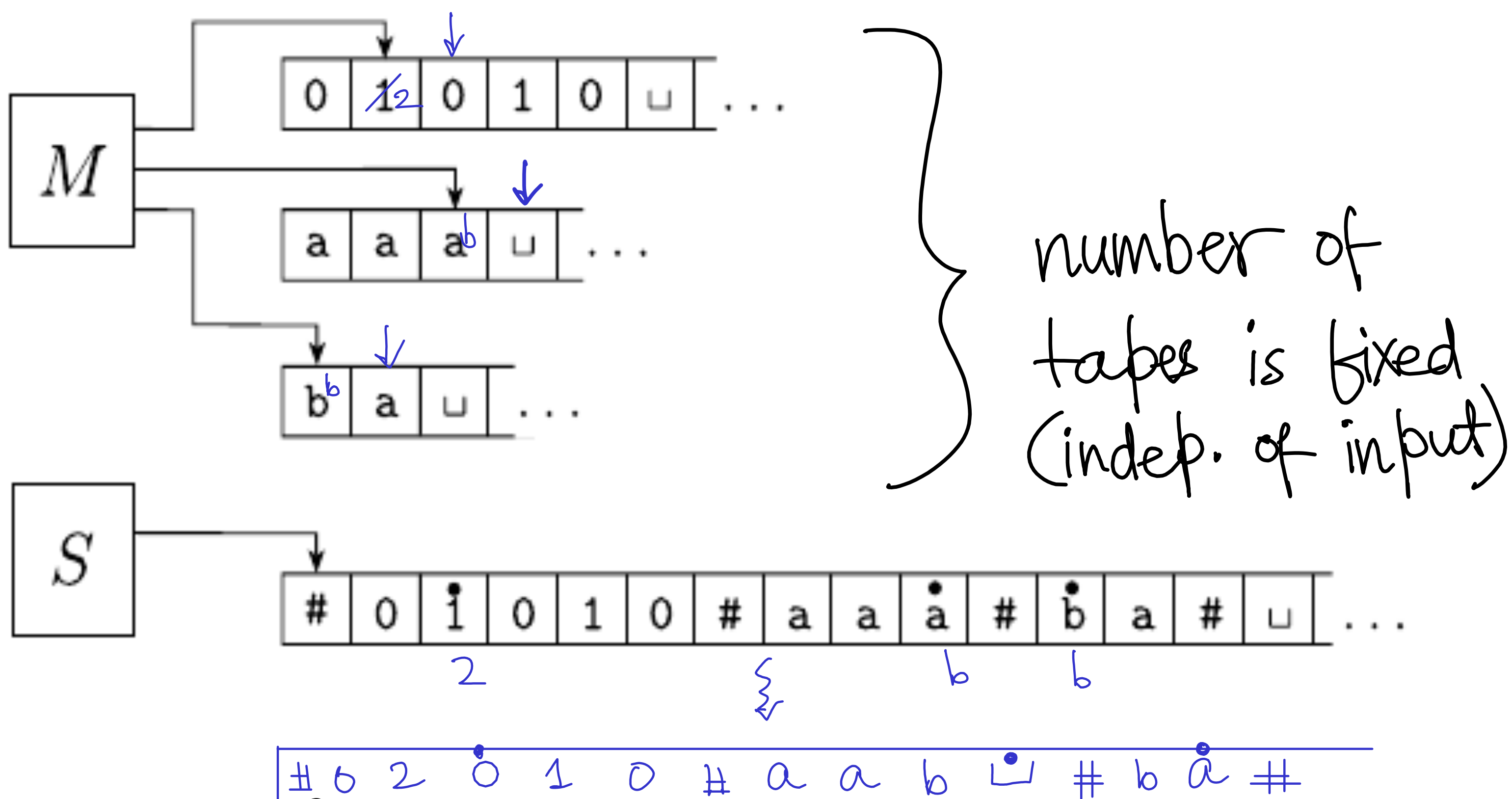
⓪ Mark the 1st cell of 1st tape

① Move both head to right, copying from 1st to 2nd tape

② Move 1st head to the leftmost cell

③ Move 1st head to right, 2nd head to left, comparing symbols in each steps

Multitape TM equivalent to 1-tape TM



First move of S:

Simulation of one move of M:

(1) One L→R scan to read all symbols under 3 heads, remembers them in its state

(2) S tries to implement $\delta_M(q,(t_1,t_2,t_3)) = (r, (t_1', t_2', t_3'), (R,R,R))$ by going to a special state

(3) One L→R scan to write new symbols

(3) One L→R scan to place head marks
Require making space

Non-deterministic 1-tape TM (NDTM)

Non-deterministically choose actions at every state.

d : $(Q \times \Gamma)$ -> $\mathcal{P}(Q \times \Gamma \times \{L, R\})$

NDTM accepts w if ? there is some non-deterministic branch leading to q_{acc}.

Exercise: Formally define acceptance by an NDTM.

SUBSET-SUM NP-complete

NDTM to accept

#100#01 #101 # 110 $\in L$.

4 1 5 6

{ #w1#w2...#wk#n : n, wi are binary integers,

there exists some wi's whose sum is n }

1. L→R move and non-deterministically mark some of wi's.
2. L-R " " add the marked wi's and write the total T after the input, separated by a #
3. goto q_a if ~~T~~ = n

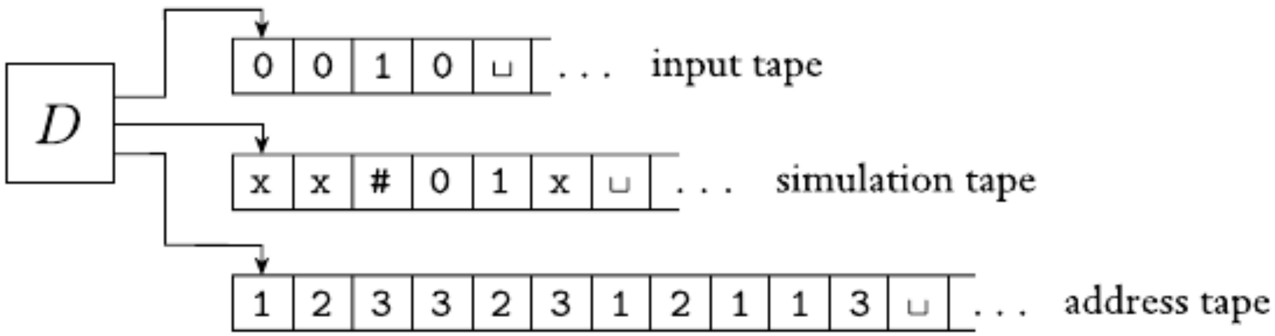
1-tape NDTM to 3-tape DTM

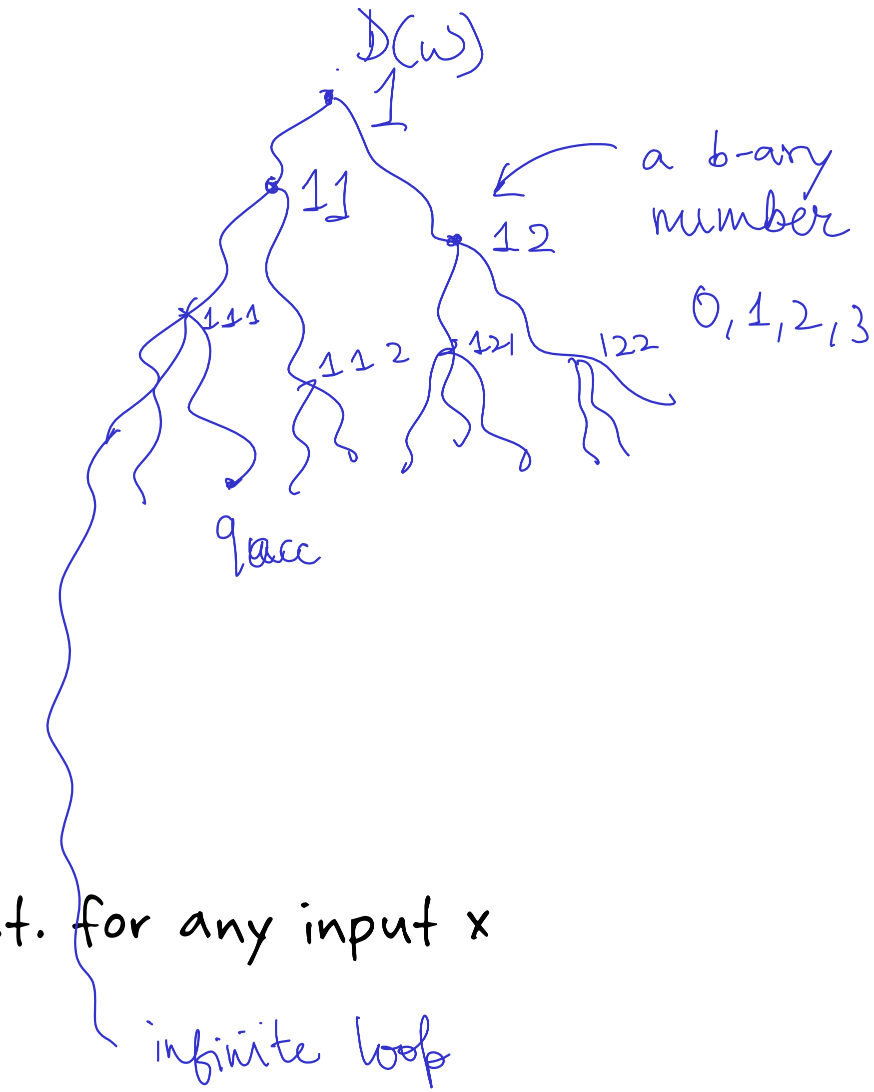
M

D

D accepts w iff M accepts w.

∃ non-deterministic branch ⇝ q_{acc}.





$0 \le |\delta(q,a)| \le b-1$ (Ex. Can you upper bound b?)

$b = 3$

1 2 2 3 1 2 3 2

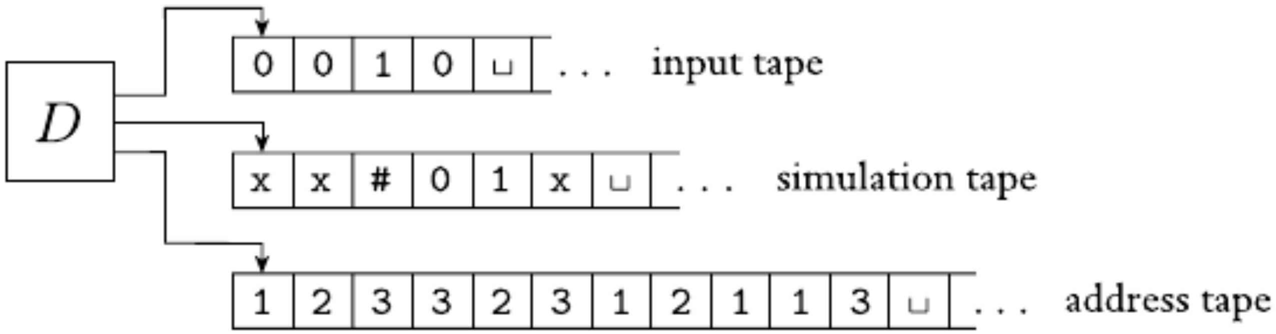
For 1st move, use 2nd nondet choice

For 2nd move, " " " " "

3rd " 3rd " " "

4th 1st " " . . .

Thm: For any NDTM N, there is a DTM M s.t. for any input x N(x) accepts if and only if M(x) accepts.



Do a BFS traversal of NDTM evaluation tree.
(Why not DFS traversal?)

Given an NDTM N=<Q,...,dN...>, show that a DTM M=<QM, ..., dM, ...> can be constructed with same L.

1. Order (q,a,{L,R}) tuples in dN(...) and construct d'(q,a,i)=i-th tuple. Let b=max {i} for all dN().

2. Add {1,2,3...b} to tape alphabet of M.

3. Construct M that runs in 3 stages.

4a. Stage a [Prepare address]: Increment value on address-tape in base-b. Reset head to left.

4b. Stage b [Prepare input]: Copy from input tape to simulation tape. Reset head to left.

4c. Stage c [Simulate]: Make transitions of the form

dM(q,a,b,i) = d'(q,b,i) where a:input head (unused during simulation), b: simulation head, i:address head

Always move address head to right. When reaches blank, move to (4a).