27/2/2023

CS704

Turing machines

- Deterministic TMs
(≈ Non-deterministic TMs)
- Total TMs, accept recursive languages
- TM (need not be total), accept r.e. languages.

Examples :

(1) $\{a^n b^n c^n / n \geq 0\}$

(2) $\{1^m \# 1^n \# 1^{m+n} / m, n \geq 0\}$

(3) $\{1^m \# 1^n \# 1^{m-n} / m, n \geq 1, m > n\}$.

(4) $\{a^p / p$ is a prime$\}$.

(5) $\{ww \mid w \in \{a,b\}^*\}$

- Checks for length being odd/even. If odd, reject if even proceed
- Identify mid-point
- Go back & forth, matching one letter at a time in the 1st 'half' & in the 2nd 'half'.

(6) $\{a^{2^n} \mid n \geq 0\} = \{a, aa, aaaa, a^8, a^{16}, a^{32}, \cdots\}$

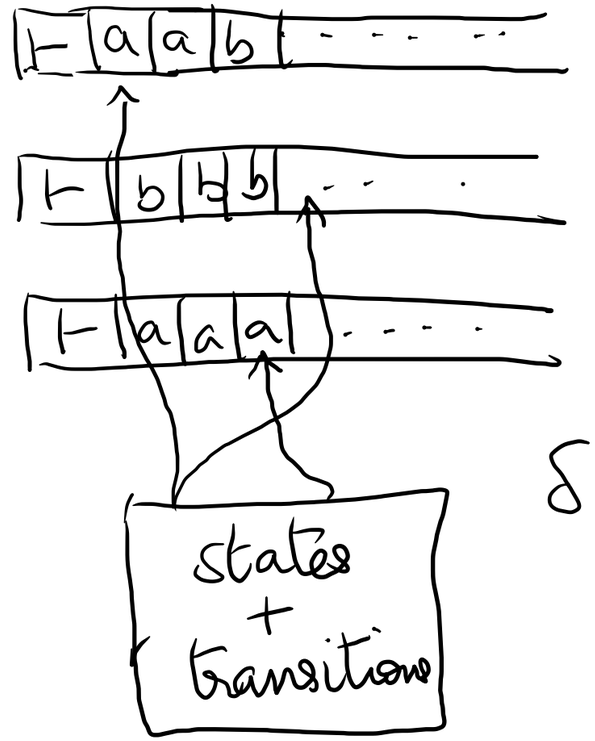
a^n, $n \geq 0$
a^{2n} — even length
a^{2n+1} — odd length
a^{2^n} — powers of 2.

All these are recursive languages.

Our model : Single, one-way infinite tape deterministic TM.

Equivalent models of TMs:

- Multi-tape TM: k-tapes



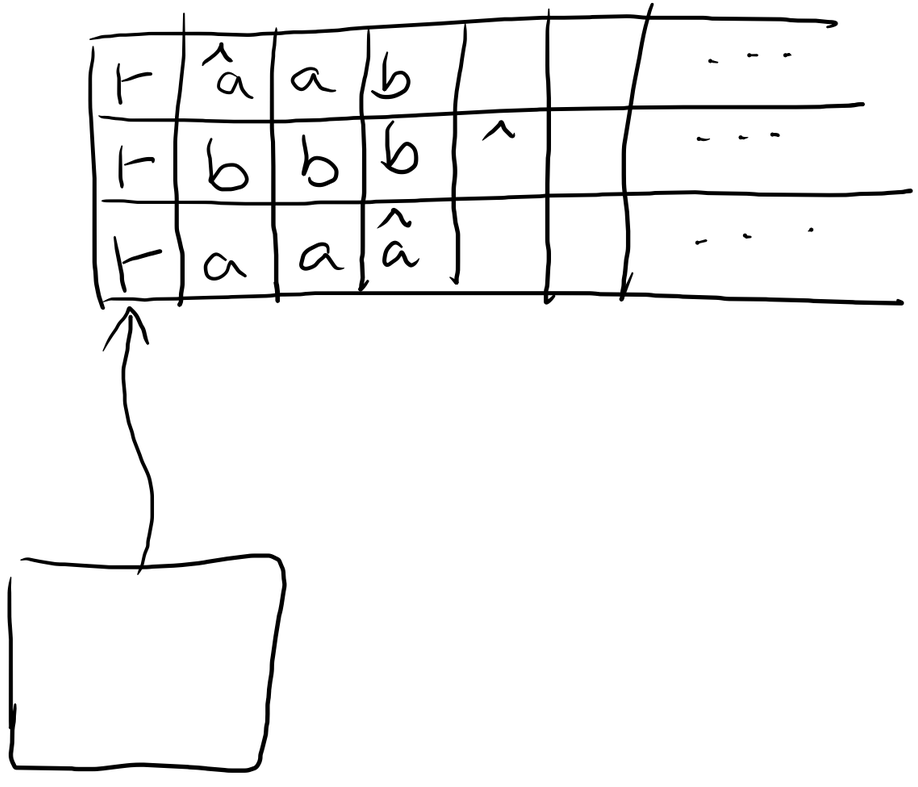
$(k=3)$

$(p, a_1, a_2, a_3) \rightarrow (p, X, Y, Z, L/R, L/R, L/R)$

$\delta : (Q \times \Gamma_1 \times \Gamma_2 \times \Gamma_3) \rightarrow (Q \times \Gamma_1 \times \Gamma_2 \times \Gamma_3 \times \{L/R\}^3)$

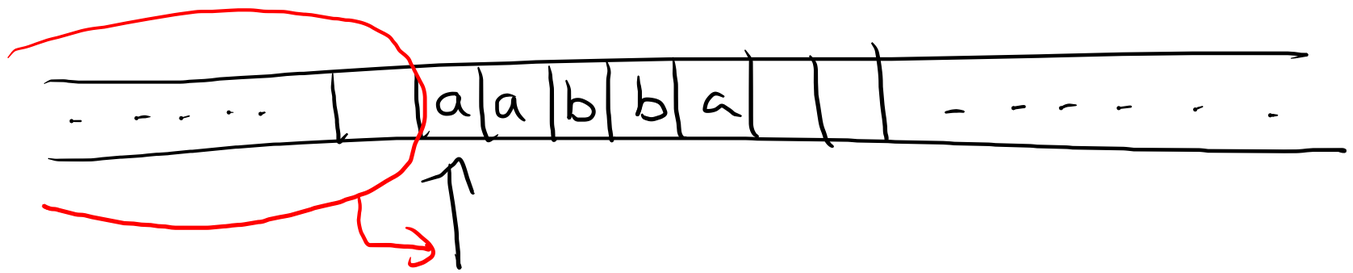
Claim: Multi-tape TM can be simulated by a single tape TM.

Proof idea: 'Paste' the k tapes together & 'simulate' the k read/write heads with just one head.



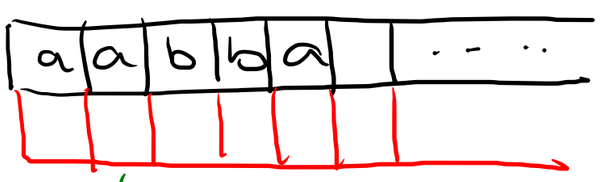
Mark read/write head positions of the k heads using a symbol say ^, & move back & forth, once for each tape to simulate one move of the multi-tape TM.

- Two-way infinite tape:



Two-way infinite tape is equivalent to a one-way infinite tape.

→ move in the same direction



↳ move in the reverse direction.

- Non-deterministic Turing machines (NDTM):

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

(deterministic)

$$\delta : (Q \times \Gamma) \to 2^{(Q \times \Gamma \times \{L,R\})}$$

$$\delta \subseteq ((Q \times \Gamma) \times (Q \times \Gamma \times \{L,R\}))$$

(non-deterministic TM)

FSA :

$$\delta : Q \times \Sigma \to Q$$

(deterministic)

run (sequence)

$$\delta : Q \times \Sigma \to 2^{Q}$$

or

$$\delta \subseteq (Q \times \Sigma \times Q)$$

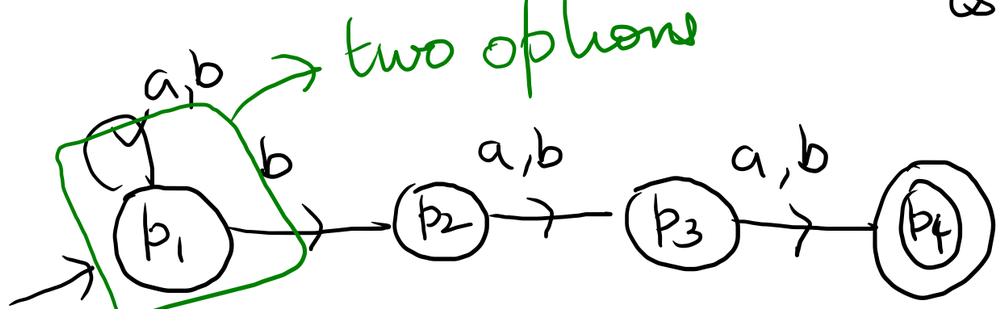
(non-deterministic)

A run of a NDTM on an input w is a tree. If one of the paths lead to an accept state, accept the i/p. If one of the paths lead to a reject state, reject i/p. Otherwise, I don't know.

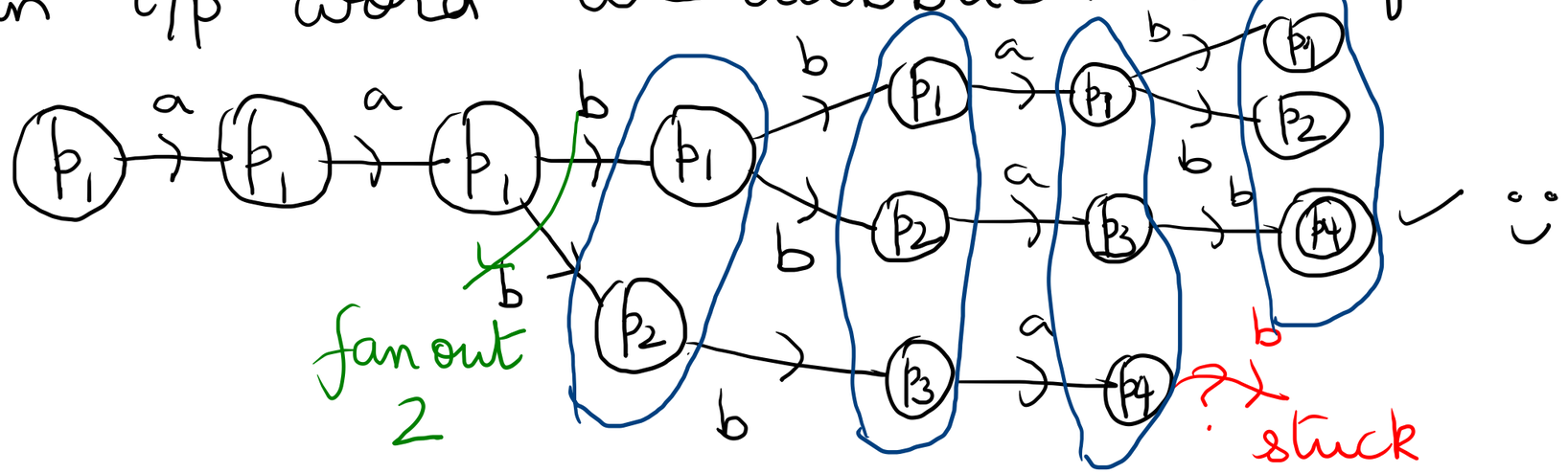
Example (NFA): $\Sigma = \{a, b\}$

$L = \{\omega \in \Sigma^* \mid$ third last letter from the right is a 'b'$\}$.

M:



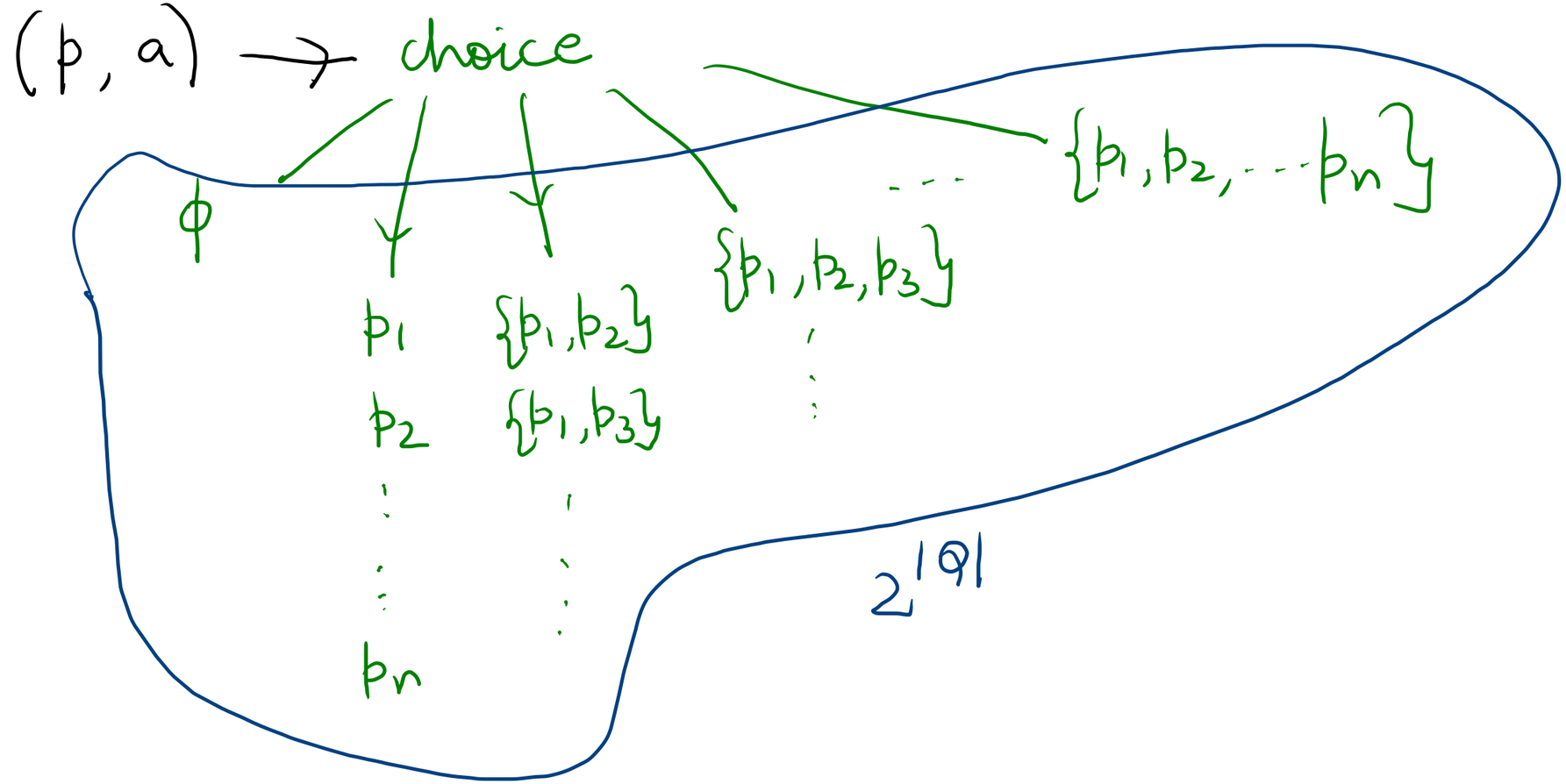
Take an i/p word ω = aabbbab. Run of M on ω is



Subset construction

Transition relation $\delta : Q \times \Sigma \to 2^Q$ $\quad Q = \{p_1, p_2 \cdots, p_n\}$

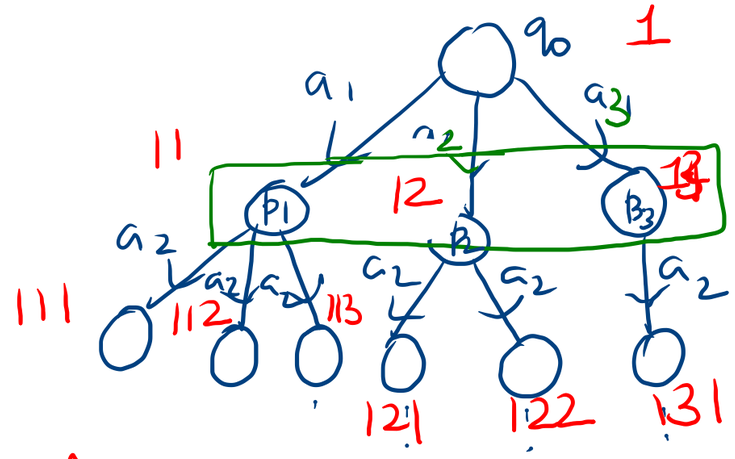
(NFA)



Theorem: Given a NDTM M, we define a DTM M' that 'simulates' a run of the m/c M on i/p w by doing BFS on the run (tree).

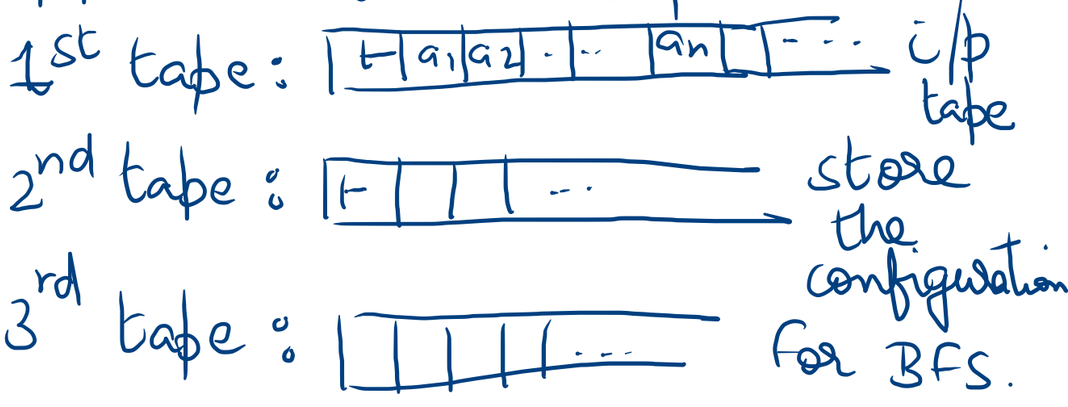
Proof idea: Given i/p $w = a_1 a_2 \cdots a_n$.

Run of M on w:



M' will 'catchup' with the run of M on w by doing a BFS on T_w.

M' will use 3 tapes:



The third tape (or any no. of finite tapes) is used for BFS. If we 'name' each node using a string over $\{1, 2, \ldots, b\}$, where 'b' is the maximum number of options that δ has, we can use lexicographic ordering to identify the successor node at each step of BFS.

Exercise : Given i/p $w \in \{1, 2, \ldots, b\}^*$, design a TM that write back $f(w)$ on the tape & halts where $f(w)$ is the lexicographic successor of w.