Sipser Section 3.2 (first part)

Different Turing machine models

Model 1: Several tapes

tape 1 $\triangleright\ \underbrace{w_1 w_2 \cdots w_n}_{w,\ |w| = n}$

tape 2 $\triangleright$

$\vdots$

tape k $\triangleright$

NB: # tapes is fixed (so k for a k-tape TM)

Transition function:

$$\delta(q, a_1, a_2, \ldots, a_k) = (p, b_1, b_2, \ldots, b_k, \gamma_1, \gamma_2, \ldots, \gamma_k)$$

b_i replaces a_i when head moves

$\gamma_j \in \{R, L, S\}$

Why useful with several tapes?

ex copy on a 2-tape TM:

$$\underline{\triangleright} w_1 w_2 \cdots w_n \qquad \leadsto \qquad \triangleright w_1 w_2 \cdots w_n \underline{\sqcup}$$
$$\underline{\triangleright} \qquad\qquad\qquad\qquad\qquad \underline{\triangleright} w_1 w_2 \cdots w_n$$

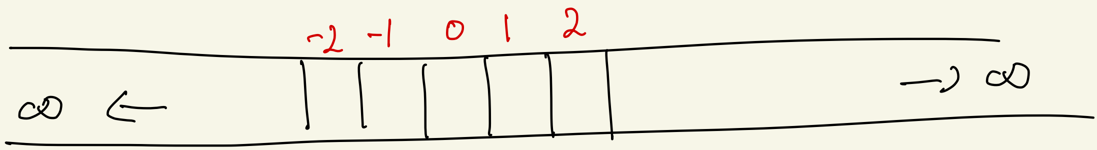
$$\leadsto \qquad \triangleright w_1 w_2 \cdots w_n w_1 \cdots \underline{w_n}$$
$$\qquad \triangleright w_1 \cdots \underline{w_n}$$

$$\leadsto \qquad \underline{\triangleright} w_1 w_2 \cdots w_n w_1 w_2 \cdots w_n$$
$$\qquad \underline{\triangleright}$$

$O(|w|)$ steps

We saw how to do this in $O(|w|^2)$ steps on a 1-tape TM

Model 2   2-way ∞-tape



Model 3   Non-deterministic TM (NDTM)

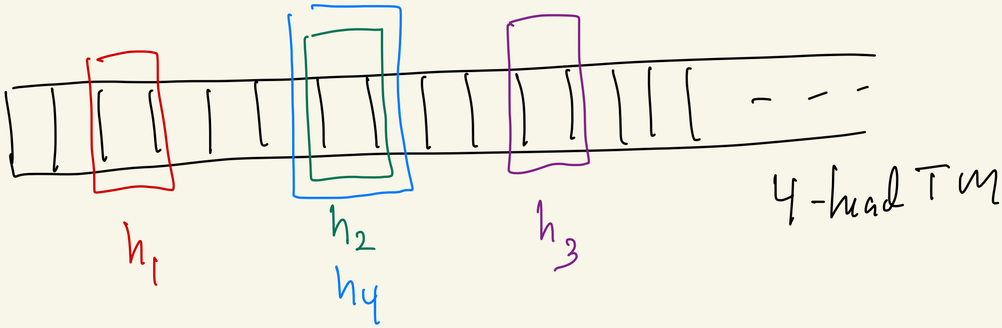
loosely : NDTM's can guess
from a state q on a symbol $a \in \Sigma$ we may
have up to $B = |Q| \cdot |\Gamma| \cdot 3$ different transitions

Example where non-determinism is useful:

Given integer $n > 1$, does there exist integers $n_1, n_2 > 1$
such that $n = n_1 \cdot n_2$ (is n composite?)

A non-deterministic TM M can 'guess'
two integers $n_1, n_2 > 1$, calculate $m = n_1 \cdot n_2$
and then check whether $m = n$

Model 4 : One tape with several heads



Initially all heads in leftmost position

$L = \{a^n b^n c^n \mid n \geq 0\}$

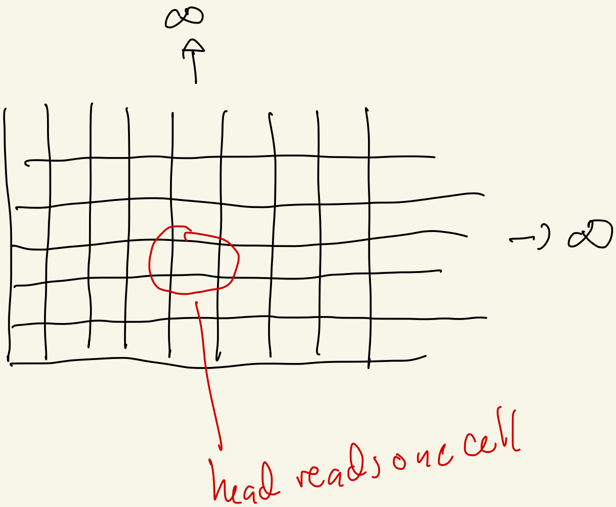
1. check that input is of the form $a^* b^* c^*$ (using one head) reject if not the case

2. $\triangleright \boxed{a}\, a \cdots a \boxed{b} \cdots b \boxed{c} \cdots c$

   $\quad h_1 \qquad\qquad h_2 \qquad h_3$

   move each head 1 step forward until h_3 reaches '$\sqcup$'
   and check whether $\#a = \#b = \#c$

Model 5   2-dimensional tape

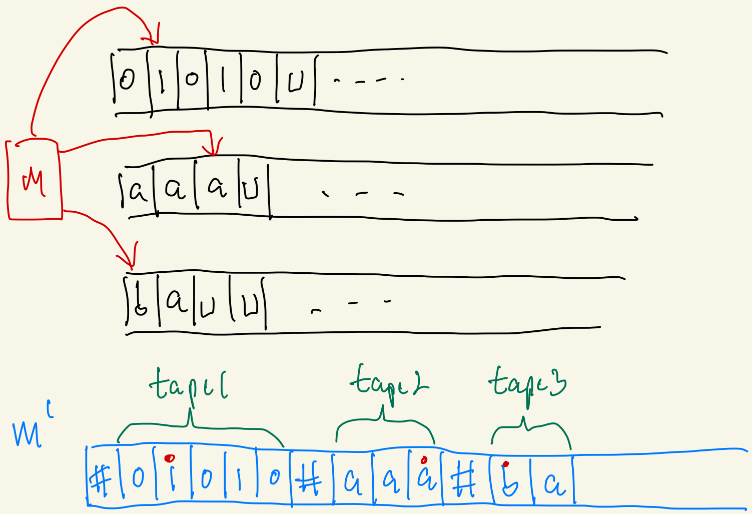


Many more models possible, including combinations of the above (e.g. NDTM with several tapes and several heads on each tape)

Our goal: prove that none of these models are stronger than a standard deterministic one-tape TM

In particular, if $L = L(M)$ for M being e.g. a k-tape TM then $\exists$ standard TM M' such that $L = L(M')$

Simulating a 3-tape TM on a 1-tape deterministic TM



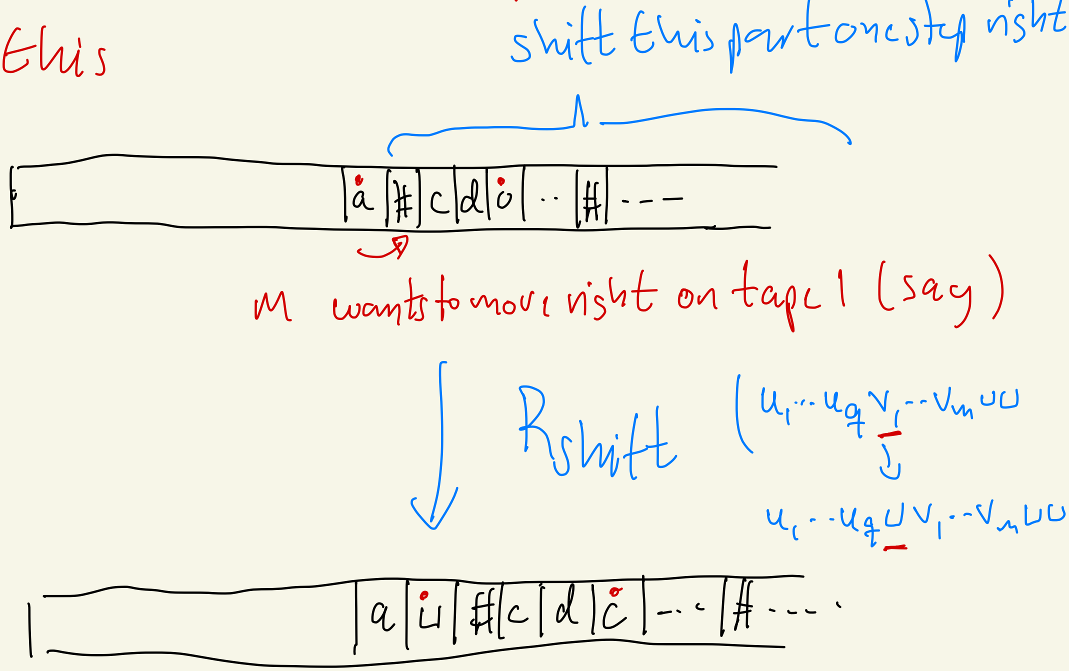
red dot indicates the positions of the 3 heads of M.

We remember them via special characters

Simulation idea: to simulate 1 step of M, M'

1. starts with its head in leftmost position
2. Then moves forward until it has found content under each reading head
3. Move head to left most position
4. move forward and modify the tape cell under each head and move this R, L or S
5. Go to 1. (to start simulating next step of M)

Note that if M is moving into blank area ('␣'s) on one of its tapes, we need to make space on M's tape for this



one execution of Rshift take $O(|w'|)$ when w' is string from head position to empty tape.

This is at most # steps taken so far by M

How do we implement this simulation idea?

assume M is a k-tape TM

For each state q_i of M we have

- states $q^{i}_{(\alpha_1, \alpha_2, \dots, \alpha_k)}$ when $\alpha_i \in \Gamma \cup \{-\}$

- states $p^{i}_{(\delta_1, \delta_2, \dots, \delta_k, b_1, b_2, \dots, b_k, \gamma_1, \gamma_2, \dots, \gamma_k)}$

where $\delta_i \in \Gamma \cup \{-\}$, $b_i \in \Gamma$, $\gamma_i \in \{R, L, S\}$

Meaning of these states:

- when we are in state $q^{i}_{(\beta_1, \beta_2, \dots, \beta_r, -, -, \dots, -)}$ we have collected the symbols currently under the heads on the first r tapes

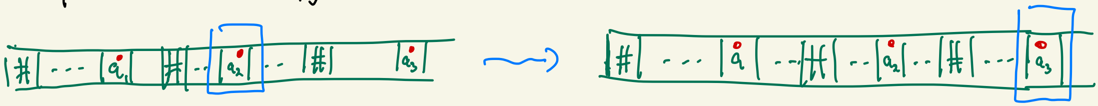
- In state $p^{i}_{(b_1, \dots, b_q, -, \dots, -, b_1, b_2, \dots, b_k, \gamma_1, \gamma_2, \dots, \gamma_k)}$ we have modified the tape cells under the first q heads and move the i'th head $i \leq q$ according to γ_i (possibly shifting part of the tape)

__Note__ M has a HUGE # of states, BUT it is at most some finite function $f(k, \Gamma, Q)$

Implementing one step $\delta(q_i, a_1, a_2, \dots a_k) = (q_j, b_1, b_2, \dots, b_k, \gamma_1, \gamma_2, \dots \gamma_k)$ (*)

1. M' starts in state $q^i_{(-,-,\dots-)}$ at leftmost part of tape

2. In state $q^i_{(a_1,\dots,a_r,-\dots-)}$ $r<k$ M' moves its head forward to copy content under M's $(r+1)$-head



$q^i_{(a_1,a_2,-,\dots,-)}$

shift to state $q^i_{(a_1,a_2,a_3,-,\dots-)}$

3. When we reach a state $q^i_{(a_1,a_2,\dots,a_k)}$ $\beta_i \in \Gamma$ we have collected all characters under M's heads

Move to state $p^j_{(-,\dots,-,b_1,\dots,b_k,\gamma_1,\dots,\gamma_k)}$ because of (*)

4. In state $p^j_{(b_1,\dots,b_s,-,\dots-,b_1,b_2,\dots,b_k,\gamma_1,\gamma_2,\dots,\gamma_k)}$ $s<k$

- Move to position of the $(s+1)$'st head
- replace a_{s+1} by b_{s+1} (according to (*))
  and move 'head' $s+1$ according to γ_{s+1}
- If $s+1<k$ go to state $p^j_{(b_1,\dots,b_{s+1},-\dots-,b_1,b_2\dots b_k,\gamma_1\dots\gamma_k)}$ and go to 4.
  else move head to leftmost position and go to state $q^j_{(-,\dots,-)}$

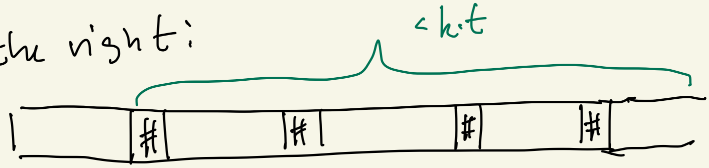
Running time for simulation

Suppose M takes t steps on input w, $|w| = n$

Then the input tape can contain at most $n+t$ symbols $\neq$ '␣'
and all other tapes at most t symbols $\neq$ '␣'
(M writes at most one symbol on each tape in each step)

- Simulating one step where no shift right is needed takes $O(\sum \text{length of tapes}) = O(n+kt)$

- Possible extra time to shift parts of tape to the right:



$$O((k-1)t + \dots + t) = O(k^2 t)$$

we ignore the k-1 #'s

- Total work for simulating one step

$$O(n+kt) + O(k^2 t) = O(n + k^2 t)$$

- Total work for simulating t steps

$$O(n \cdot t + k^2 t^2)$$

polynomial in n and t (k is fixed)