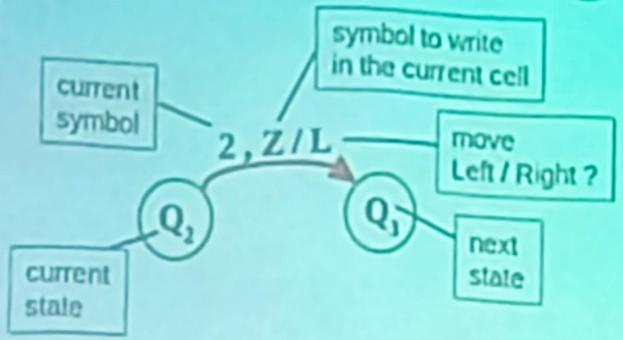
Transition Diagram of Turing Machine

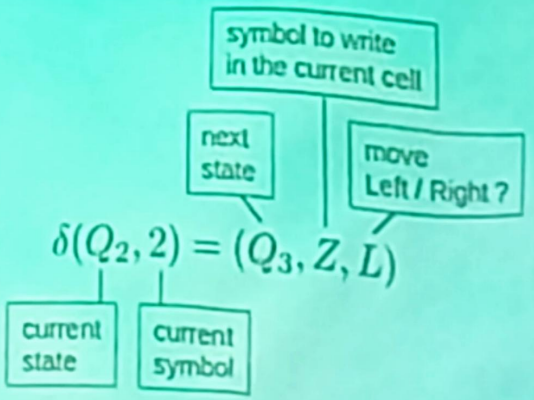
Circles are states.

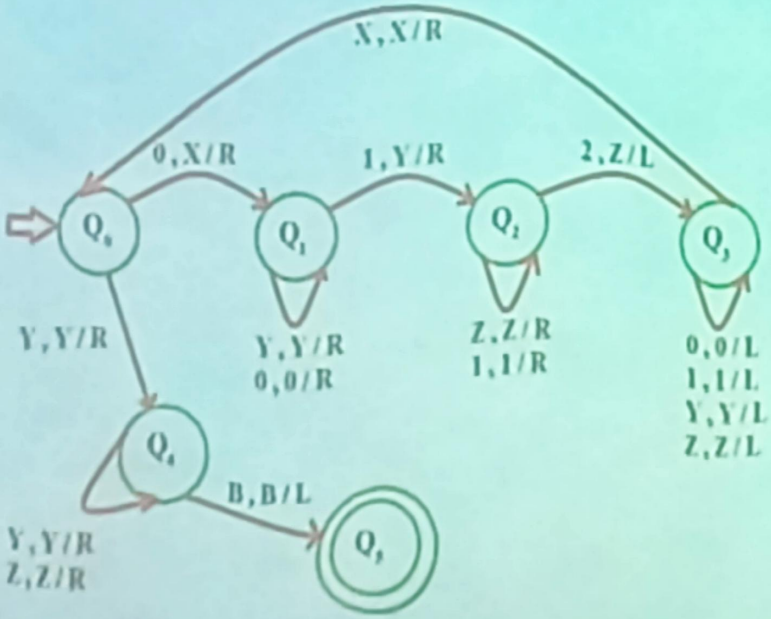
Start state is marked with an extra incoming arrow. 

Each final state is marked with double circle. 

Transition







In our example :

Set of states = $\{Q_0, \ldots, Q_5\}$

Input Alphabet = $\{0, 1, 2\}$

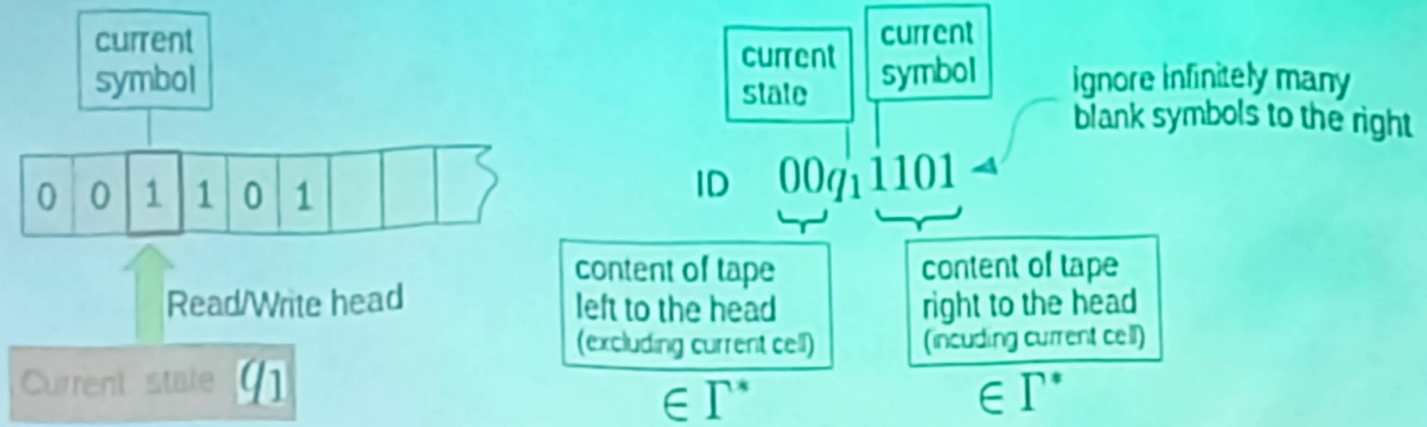
Tape Alphabet = $\{0, 1, 2, X, Y, Z, B\}$

Start state = Q_0

Set of final states = $\{Q_5\}$

Instantaneous Description (ID)

A textual way to express the '*configuration*' of TM (or NTM) at any moment during the computation



ID is a string of the form $\alpha q \beta$ where, $\alpha, \beta \in \Gamma^*$ and $q \in Q$

It shows
- Content of the whole tape
- Current position of head on the tape
- Current state

Any string in $\Gamma^* \times Q \times \Gamma^*$ is an ID. It does matter whether our TM can ever reach that configuration.

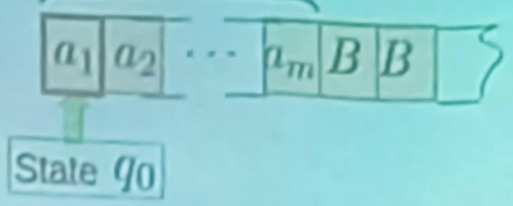
$\Gamma^* \times Q \times \Gamma^*$ = Set of all IDs.

Computational Path of TM and NTM

Consider NTM M with start state q_0 and input alphabet Σ

(NTM is a generalization of DTM. Whatever we say about NTM is also applicable to DTM as special case)

With input $x = a_1 \cdots a_m \in \Sigma^*$ the TM starts with ID $q_0 x$



then, it steps through subsequent IDs

$$ID_0 = q_0 x \vdash ID_1 \vdash ID_2 \vdash \cdots$$

For **Deterministic** TM, ID_i uniquely determines ID_{i+1} for all i

x uniquely determines the chain $ID_0 \vdash ID_1 \vdash ID_2 \cdots$

DTM has a *unique computational path* (for a given input)

For **Nondeterministic** TM, we may have $ID_i \vdash ID_{i+1}$, $ID_i \vdash ID'_{i+1}$, $ID_i \vdash ID''_{i+1}$, ... etc

NTM can proceed along *any one of the many possible computational paths* (for the same input)

$$\begin{array}{c} \nearrow ID'_{i+1} \vdash ID'_{i+2} \vdash \cdots \\ ID_0 \vdash \cdots \vdash ID_i \vdash ID_{i+1} \vdash ID_{i+2} \vdash \cdots \\ \searrow ID'_{i+1} \vdash ID'_{i+2} \vdash \cdots \end{array}$$

Computational Path of TM and NTM

Consider NTM M with start state q_0 and input alphabet Σ

With input $x \in \Sigma^*$ the TM starts with ID q_0x and steps through $ID_0 = q_0x \vdash ID_1 \vdash ID_2 \vdash \ldots$

Let us depict a chain of moves $ID_i \vdash \cdots \vdash ID_j \vdash \cdots$ as  (circles denote IDs)

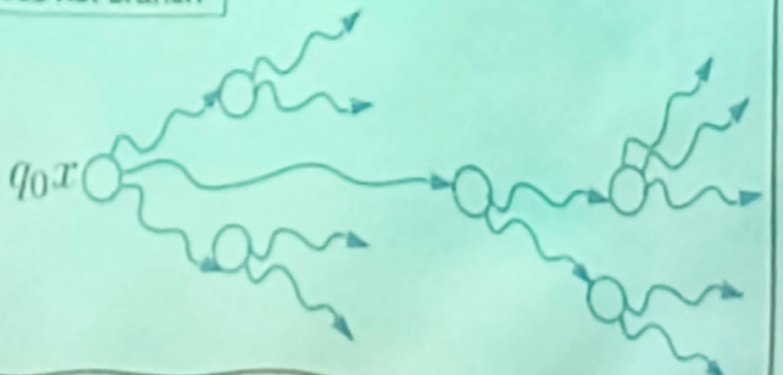
For (**Deterministic**) TM, input x uniquely determines the chain $q_0x = ID_0 \vdash ID_1 \vdash ID_2 \cdots$

DTM has a *unique computational path* (for a given input)

For **Nondeterministic** TM, All possible *computational paths* for given input x can be represented as a tree with starting ID q_0x as root

For DTM the tree does not branch

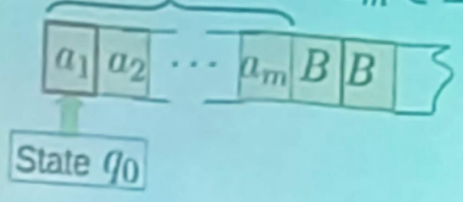
$$ID_0 \vdash \cdots \vdash ID_i \vdash \begin{cases} ID_{i+1} \vdash ID'_{i+2} \vdash \cdots \\ ID_{i+1} \vdash ID_{i+2} \vdash \cdots \\ ID'_{i+1} \vdash ID'_{i+2} \vdash \cdots \end{cases}$$



Acceptance Criteria (for TM and NTM)

Consider a TM or NTM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

With input $x = a_1 \cdots a_m \in \Sigma^*$ the TM or NTM starts with ID $q_0 x$



Same criteria for both TM and NTM

x is accepted if and only if $q_0 x \vdash^* \alpha q \beta$ for some $q \in F$

i.e. iff it is **possible to reach a final state** from starting configuration

The **language** accepted by M is $L(M) = \{x \in \Sigma^* \mid q_0 x \vdash^* \alpha q \beta \text{ for some } q \in F\}$

i.e. the set of all strings that are accepted

Example of acceptance / non-acceptance (for NTM)

Consider NTM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ where $Q = \{q_0, \ldots, q_5\}$ and $\boxed{F = \{q_2\}}$

	0	1
q_0	$(q_1, 1, R), (q_4, 0, R)$	$(q_4, 0, R), (q_5, 1, R)$
q_1	$(q_2, 1, R), (q_3, 0, R)$	– – –
q_2	$(q_2, 0, L)$	$(q_3, 0, R)$
q_3	$(q_0, 1, R)$	– – –
q_4	$(q_5, 0, L)$	– – –
q_5	$(q_4, 0, R)$	– – –

reached a final state
Halts and decides $000 \in L(M)$

Unnecessary to compute

Input 000: $q_0000 \vdash 0q_400 \vdash q_5000 \vdash 0q_400 \vdash \cdots$ **Does NOT halt !!**

$1q_100 \vdash 10q_30 \vdash 101q_3$ no valid move! Halts in a nonfinal state

$11q_20 \vdash 1q_210 \vdash 10q_30 \vdash 101q_0$

Input 10: $q_010 \vdash 1q_50 \vdash 10q_4$ no valid move! Halts in a nonfinal state

$0q_40 \vdash q_500 \vdash 0q_40 \cdots$ **Does NOT halt !!**

Acceptance / Non-acceptance for NTM

Take a NTM M. For any one particular computational path of M one of the following 3 cases happens -

Case 1: A final state is reached *(for the first time on that path)*.
NTM halts on the final state and it is *decided that input is in L(M)*.

Case 2: A nonfinal state is reached *(without passing through any final state)* from where there is *no valid move*. *NTM halts on the nonfial state*.

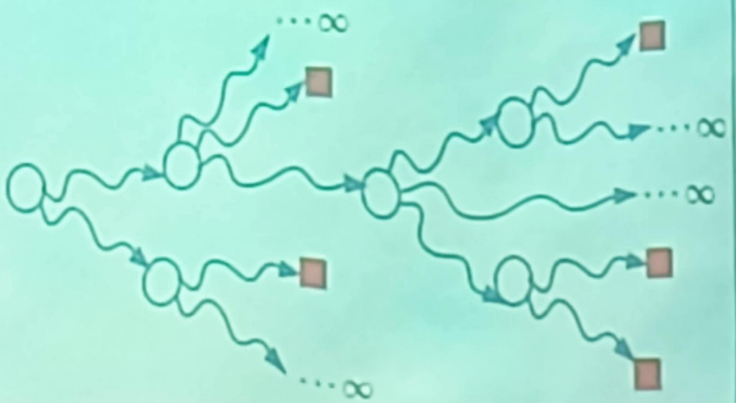
Case 3: *NTM **does not halt** and never reaches any final state*.

when input $\in$ L(M)

There is **at least one** path as in **Case 1**
There are **zero or more** paths as in **Case 2** and **3**

when input $\in$ L(M)

All the paths are either as in **Case 2**
or as in **Case 3**



Acceptance / Non-acceptance for (Deterministic) TM

A TM M, given input x, has *only one computational path uniquely determined by input x*

For *that unique path of M (on input x)* one of the following 3 cases happens -

when input $\in L(M)$

Case 1: A final state is reached *(for the first time on that path).*
TM halts on the final state and it is *decided that input $x \in L(M)$.*

when input $\notin L(M)$

We have either **Case 2** or **Case 3**
Only one of them for a given input (not both at the same time because there is *just one path*)

Case 2: A nonfinal state is reached *(without passing through any final state)* from where there is *no valid move*. *TM halts on the nonfinal state* and it is *decided that input $x \in L(M)$*

Case 3: *TM does not halt and never reaches any final state.*

Nondeterministic TM

Acceptance of string:

Consider NTM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$. M accepts x if and only if $q_0 x \vdash^* \alpha q \beta$ for some $q \in F$

i.e. iff it is **possible to reach a final state** from starting configuration

NTM halts on a final state (the first final state during computation)

Does it imply that input ∈ L(M) ? **Yes**

NTM halts on a non-final state (without passing through any final state). There is no further valid move.

Does it imply that input ∈ L(M) ? **No**

NTM does not halt. It never reaches any final state.

Does it imply that input ∈ L(M) ? **No**

Other `accepting paths' may exist

Nondeterministic Automata (NFA, PDA or NTM etc.)

- Not meant to be implementable as *"effective computers"*.
- Mathematical tools to help classify computational problems according to hardness.

Decidability of a string

Acceptance of string:

Consider TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$. M accepts x if and only if $q_0 x \vdash^* \alpha q \beta$ for some $q \in F$

Deciding the membership of a string: *We say that the TM decides* $x \in L(M)$ *or* $x \notin L(M)$ *if, given* x *as input, it halts on a final or a non-final state respectively.*
(we interpret it as the TM actually producing the output "input is accepted" or "input is rejected")

TM halts on a final state (the first final state during computation)
It implies that input $\in$ L(M) and the **TM can decide that.**

TM halts on a non-final state (without passing through any final state). There is no further valid move.
It implies that input $\in$ L(M) and the **TM can decide that.**

TM does not halt. It never reaches any final state.
Does it imply that input $\in$ L(M) ? Yes
It implies that input $\in$ *L(M) but The TM cannot decide that.*

Notice that, If $x \in L(M)$ the TM always decides that.
Only when $x \notin L(M)$ the TM may or may not be able to decide

Recursively Enumerable and Recursive Language

Consider TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

Acceptance of string: M accepts x if and only if $q_0 x \vdash^* \alpha q \beta$ for some $q \in F$

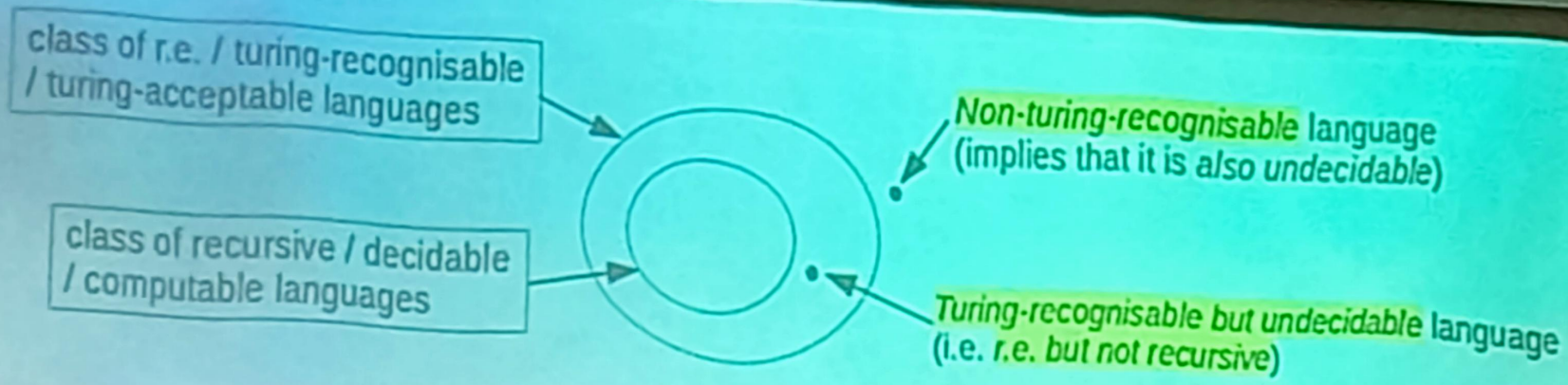
Deciding the membership of a string: *We say that the TM decides* $x \in L(M)$ *or* $x \notin L(M)$ *if, given* x *as input, it halts on a final or a non-final state respectively.*
(we interpret it as the TM actually producing the output "input is accepted" or "input is rejected")

Acceptance of language: The language accepted by M is $L(M) = \{x \in \Sigma^* | M \text{ accepts } x\}$

Definition (r.e. language) : *A language is recursively enumerable (r.e.) if it is accepted by some TM.*
(people also use "Turing acceptable", "Turing Recognisable" or simply "Recognisable" to mean "r.e.")

Definition (recursive) : *A language is recursive if it is accepted by some TM which halts on all input.*
(people also use "Decidable" or "Computable" to mean "recursive")

An equivalent definition : $L \subset \Sigma^*$ is *decidable* if there exists a TM M which $\forall x \in \Sigma^*$, can decide whether $x \in L$ or $x \notin L$



Does there exist any non-r.e. / non-turing-recognisable language ? Yes

Some languages cannot be accepted by any TM

Does there exist any r.e. but non-recursive language ? Yes

There are languages which can be accepted by some TM

but any TM accepting the language is bound to fall into "infine loop / infinite recursion"

(on some input which are not in the language)

Simulating Certain "High Level Instructions" with transition function

Move right or left (without changing symbol) $\delta(p, a) = (q, a, L \text{ or } R) \quad \forall a \in \Gamma$

Don't move (just change symbol) $\delta(p, a) = (q, b, R)$
then "move left" without changing symbol

Go to the end (from where the blank cells begin) of the tape $\forall a \in \Gamma \setminus \{B\},\ \delta(q, a) = (q, a, R)$
$\delta(q, B) = (q', B, L)$

Search for a particular symbol $a \in \Gamma \setminus \{B\}$
(start from the end and scan in the left direction)

First, "Go to the end". Then,
$\forall b \in \Gamma \setminus \{B, a\},\ \delta(q', b) = (q', b, L)$
$\delta(q', a) = \text{whatever we want}$

Suppose $a, b, a', b' \in \Gamma$ and we want to replace symbol a or b with its "marked" version

$$\text{For } A = a, b \quad \delta(p, A) = (q, A', L \text{ or } R)$$

Move the mark (to the right) $\text{For } A = a, b \quad \delta(p, A') = (q, A, R) \text{ and } \delta(q, A) = (r, A', L)$
$\Rightarrow \alpha p a' b \beta \vdash \alpha a q b \beta \vdash \alpha r a b' \beta$

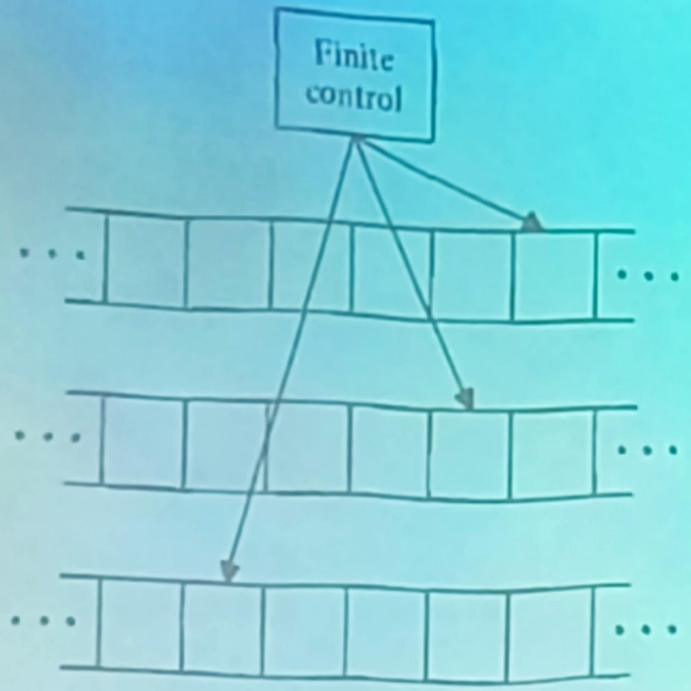


Fig. 7.8 Multitape Turing machine.

Multi-tape Turing Machine

Similar to ordinary TM but

- it has multiple tapes
- each tape has its own corresponding head
- A "single move" is of the form

$$\left(p, \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix}\right) = \left(q, \begin{bmatrix} (X_1, L \text{ or } R) \\ \vdots \\ (X_k, L \text{ or } R) \end{bmatrix}\right)$$

- p: current state
- q: next state
- $a_1, \ldots, a_k$: current symbols in tape-1 to tape-k
- $X_1, \ldots, X_k$: symbols to write on tape-1 to tape-k
- L or R: move Left / Right ? for head-1 to head-k

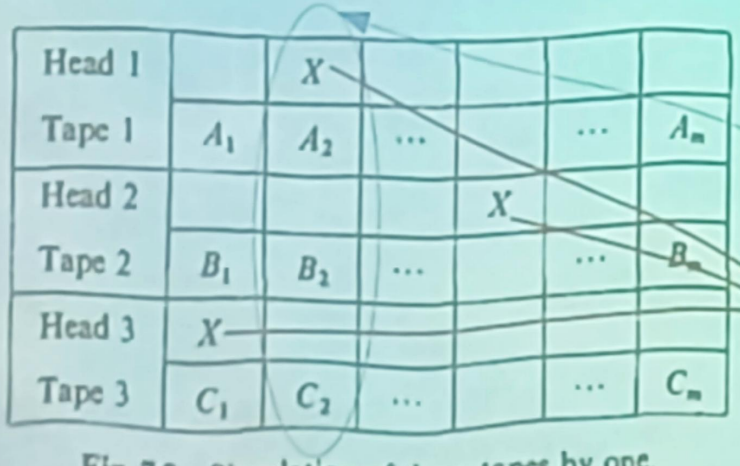


Fig. 7.9 Simulation of three tapes by one.

Theorem: *Multi-tape TM can be simulated by ordinary TM*

Proof Idea

- In the ordinary TM a single symbol is a 2k-tuple. (for example, this is a single cell containing a single symbol $[X, A_2, \diamond\text{blank}\diamond, B_2, \diamond\text{blank}\diamond, C_2]$)
- Xs mark the position of k many heads
- A single move of MT-TM is simulated by k "passes"
- Each "pass" consists of multiple moves. In i^{th} pass, the TM searches for the i^{th} X, rewrites the symbol below and "moves" the X.

A language is accepted by a multi-tape TM if and only if it is accepted by a (1-tape) TM

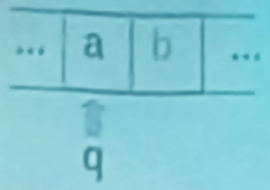
The (1-tape) TM does not halt on some input
if and only if the multitape TM does not halt on the input

A language is decided by a multi-tape TM if and only if it is decided by a (1-tape) TM

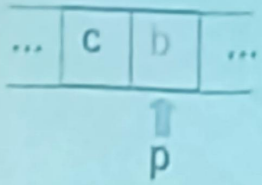
If the multitape TM halts on the input in N many steps
then the (1-tape) TM halts on the input in $O(N^2)$ many steps

TM with alphabet {0,1,B} can simulate arbitrary TM

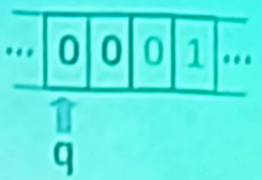
TM with alphabet {a,b,c,B}



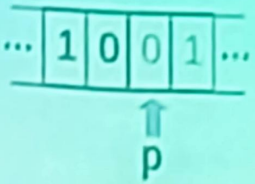
$\delta(q, a) = (p, c, R)$



a,b,c are encoded as 00,01,10



$\delta(q,0) = (r_0, 0, R), \delta(r_0, 0) = (r_{0'0}, 0, L),$
$\delta(r_{0'0}, 0) = (r_{00'}, 1, R), \delta(r_{00'}, 0) = (p, 0, R)$



TM with alphabet {0,1,B} is as powerful as Multitape TM with arbitrary alphabet.

if and only if The (arbitray) TM does not halt on some input
the 0/1 TM does not halt on the input

If the (arbitray) TM halts on the input in N many steps
then the 0/1 TM halts on the input in O(N) many steps