Lecture 15

HW3 : out Nov 6, DUE NOV 20

Today : Turing Machines (cont'd)

(1) Variants of TMs: Multitape TMs
Nondet TMs
TMs that compute functions

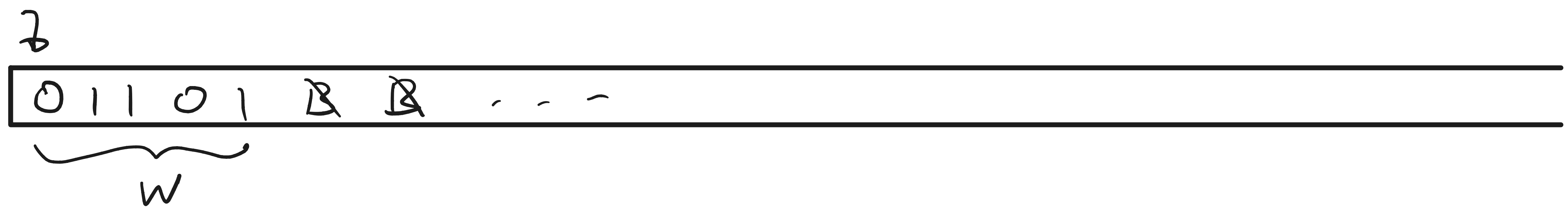
(2) Church-Turing Thesis

(3) Universal Turing Machines

$\mathcal{U}$: take as input $\langle \underline{M}, \underline{w} \rangle$

Recap from Last Week

TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ $\quad \delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$



When we run M on w either: (i) M halts and accepts w
(ii) M halts and rejects w
(iii) M never halts on w

$\mathcal{L}(M) = \{ w \in \Sigma^* \mid M \text{ on } w \text{ halts and accepts} \}$

Recursively Enumerable (RE) / Recognizable Languages

A Language $L \subseteq \Sigma^*$ is **RE** or **recognizable** if there exists a TM M such that $\mathcal{L}(M) = L$.

That is: $\forall w \in L$ M on w halts and accepts, and

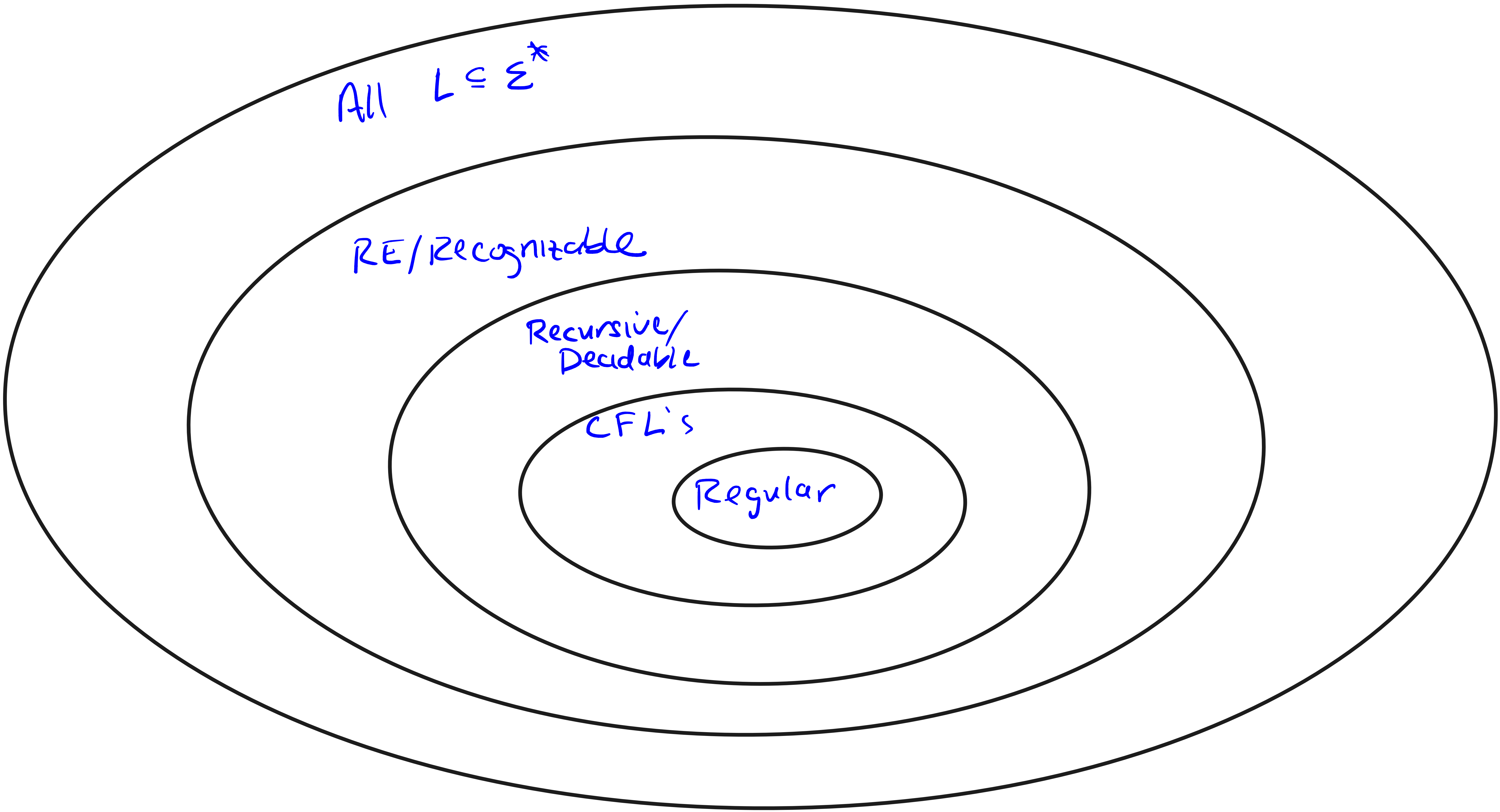
$\forall w \notin L$ M on w *either* halts & rejects or never halts

Recursive / Decidable Languages

A Language $L \subseteq \Sigma^*$ is recursive or decidable if there exists a TM M such that $\mathcal{L}(M) = L$ and M halts on all inputs

That is: $\forall w \in L$ M on w halts and accepts
and $\forall w \notin L$ M on w halts and rejects

RE/Recognizable vs Recursive/Decidable

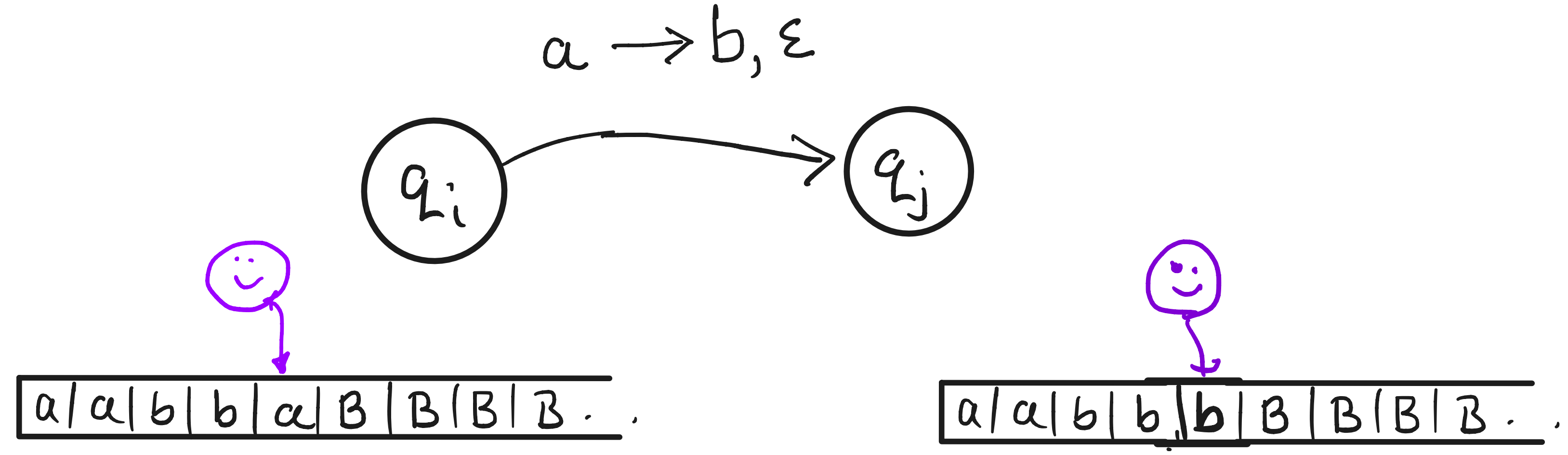


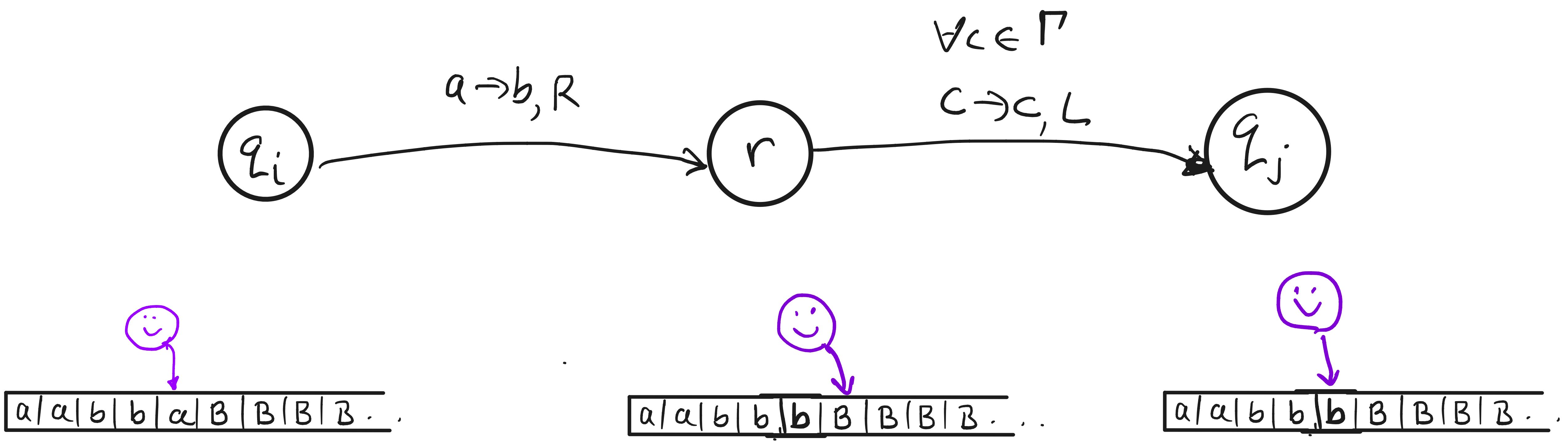
Equivalent TM Models

① Head stays in place:

Now

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

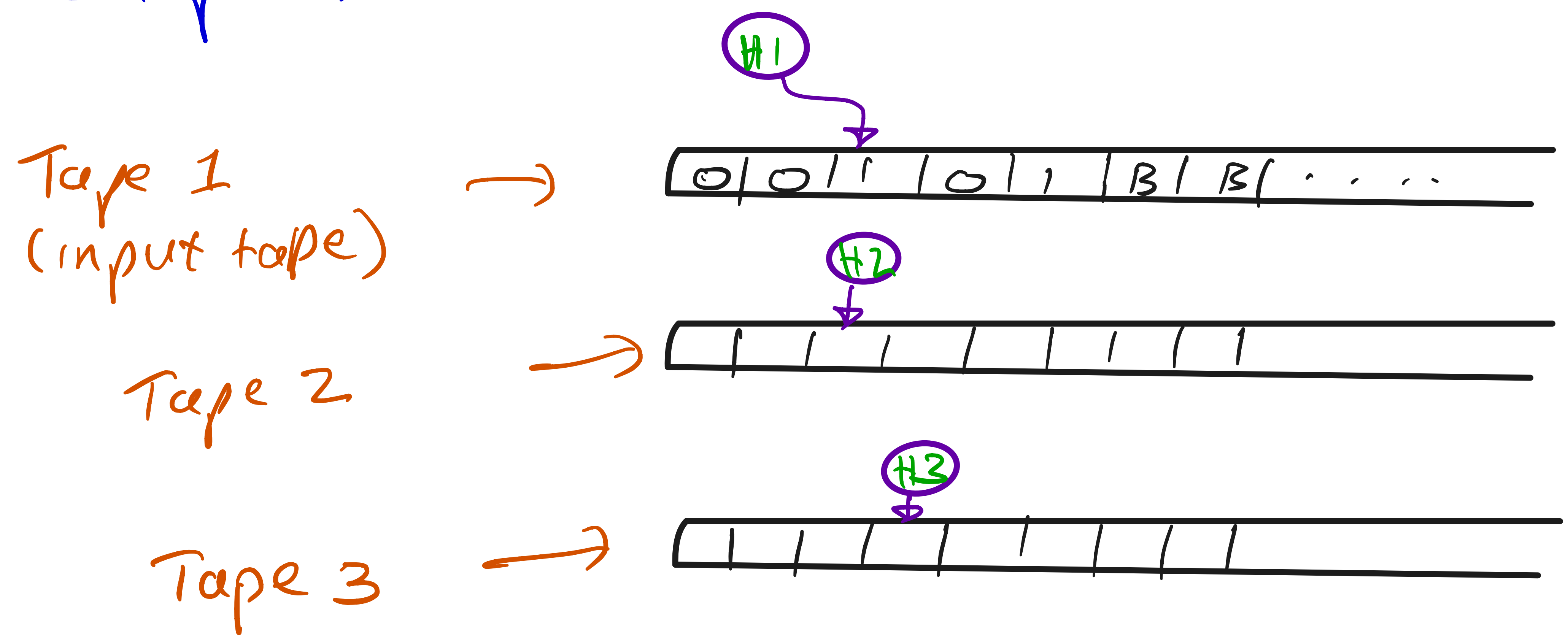




Equivalent TM Models

② Multitape TMs

Example: 3 tapes:



3 Tape TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

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

② Multitape TMs

Example: $L = \{ w \# w \mid w \in \{0,1\}^* \}$

<u>High level:</u>

① copy string to left of "#" on 2nd tape

② Compare string to rt of "#" to string on 2nd tape.

accept if they match.

② Multitape TMs

We enlarged tape alphabet

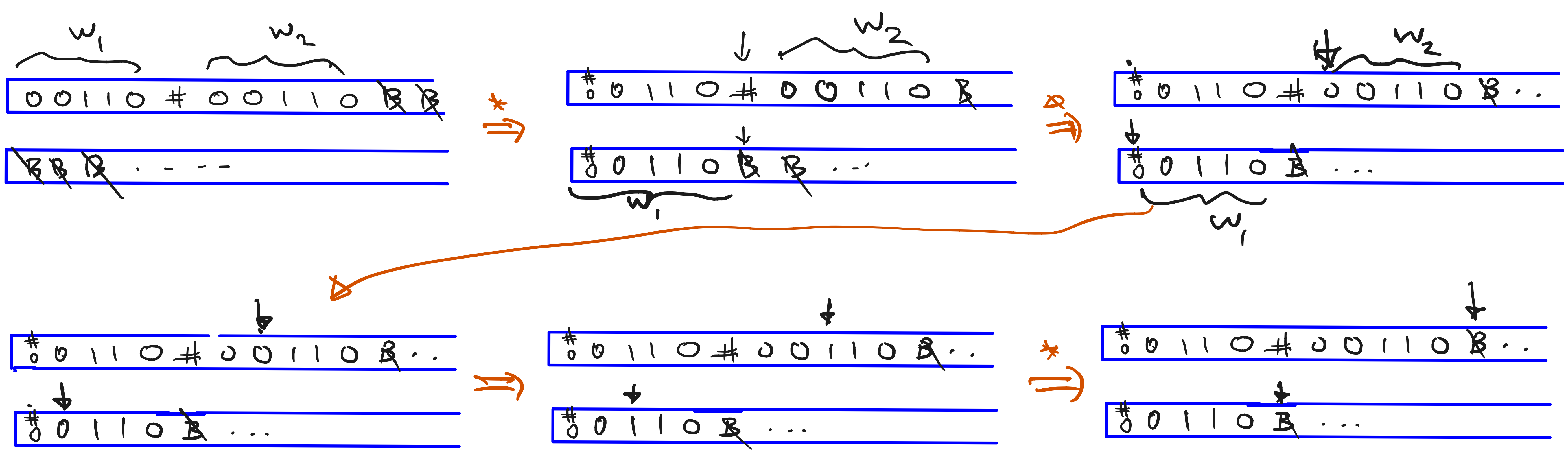
$\Gamma = \{0, 1, \#, B, \#0, \#1\}$

Example: $L = \{w_1 \# w_2 \mid w_1 = w_2\}$

<u>High level:</u>

① Copy string to left of "#" on 2^{nd} tape

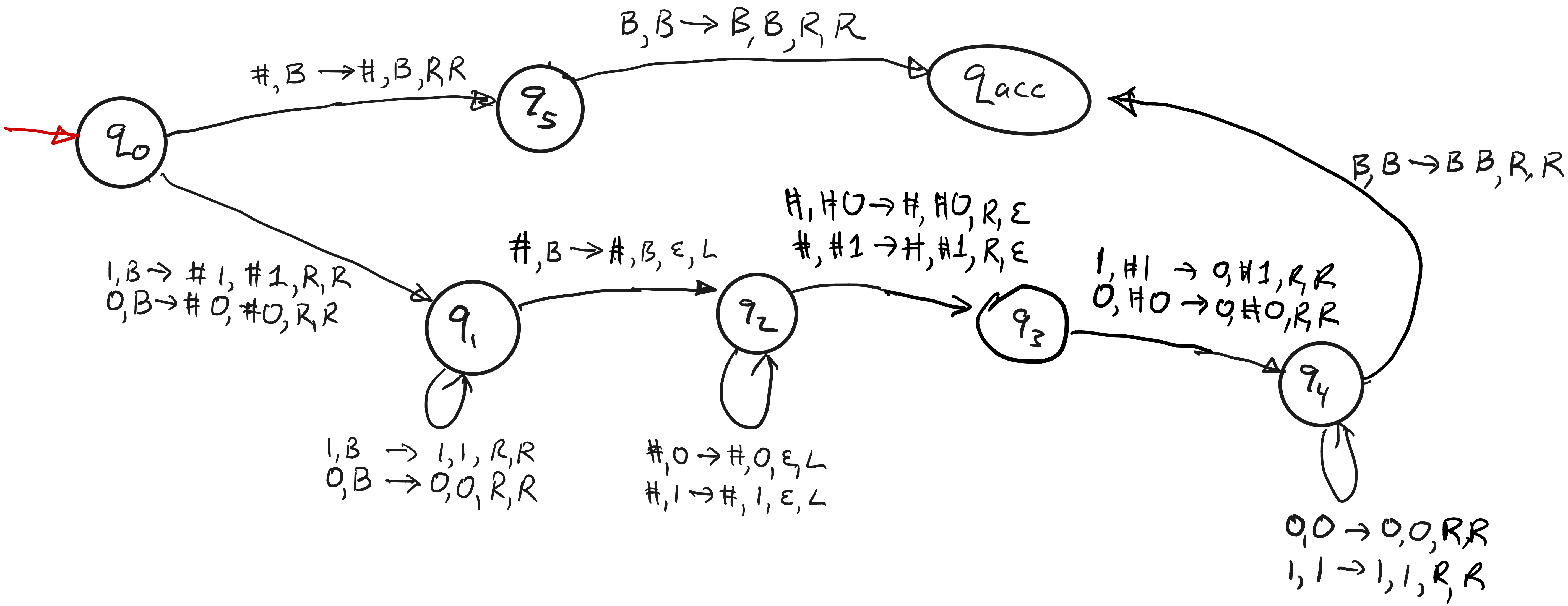
② Compare string to rt of "#" to string on 2^{nd} tape. accept if they match.



② Multitape TMs

Example : $L = \{ w\#w \mid w \in \{0,1\}^* \}$

* any missing transition goes to q_{rej}

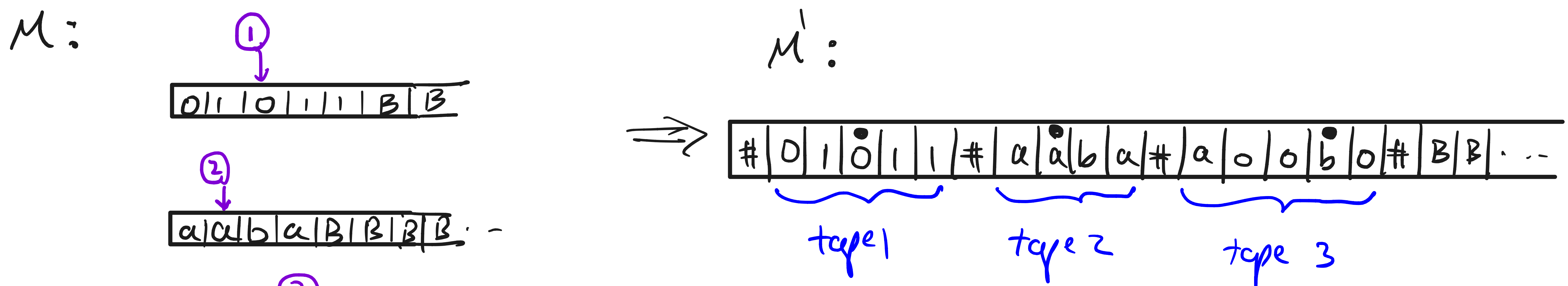


② Multitape TMs

Theorem Let $L \subseteq \Sigma^*$ be accepted by a k-tape TM, M. Then L is also accepted by a 1-tape TM, M. And if M always halts, then M' always halts

Proof sketch

Suppose M is a 3-tape TM (general case is similar.) We will represent contents of all 3 tapes by a single tape as follows:



• on top of symbol denotes the head position

② Multitape TMs

Theorem Let $L \subseteq \Sigma^*$ be accepted by k-tape TM, $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$
Then L is also accepted by a 1-tape TM, M'

Simulation of M by M' on $w = w_1 w_2 \cdots w_n$:

① Put M' in this form: $\# \mid \dot{w}_1 \mid w_2 \mid w_3 \mid \cdots \mid w_n \mid \# \mid \dot{B} \mid \# \mid \dot{B} \mid \# \mid B \mid B \mid \cdots$

② To simulate one transition of M: M' scan tape from first # to 4^{th} # and remembers symbols under each tape head (by state we are in)
Then make a second pass over tape to update tapes according to M's transition

Example

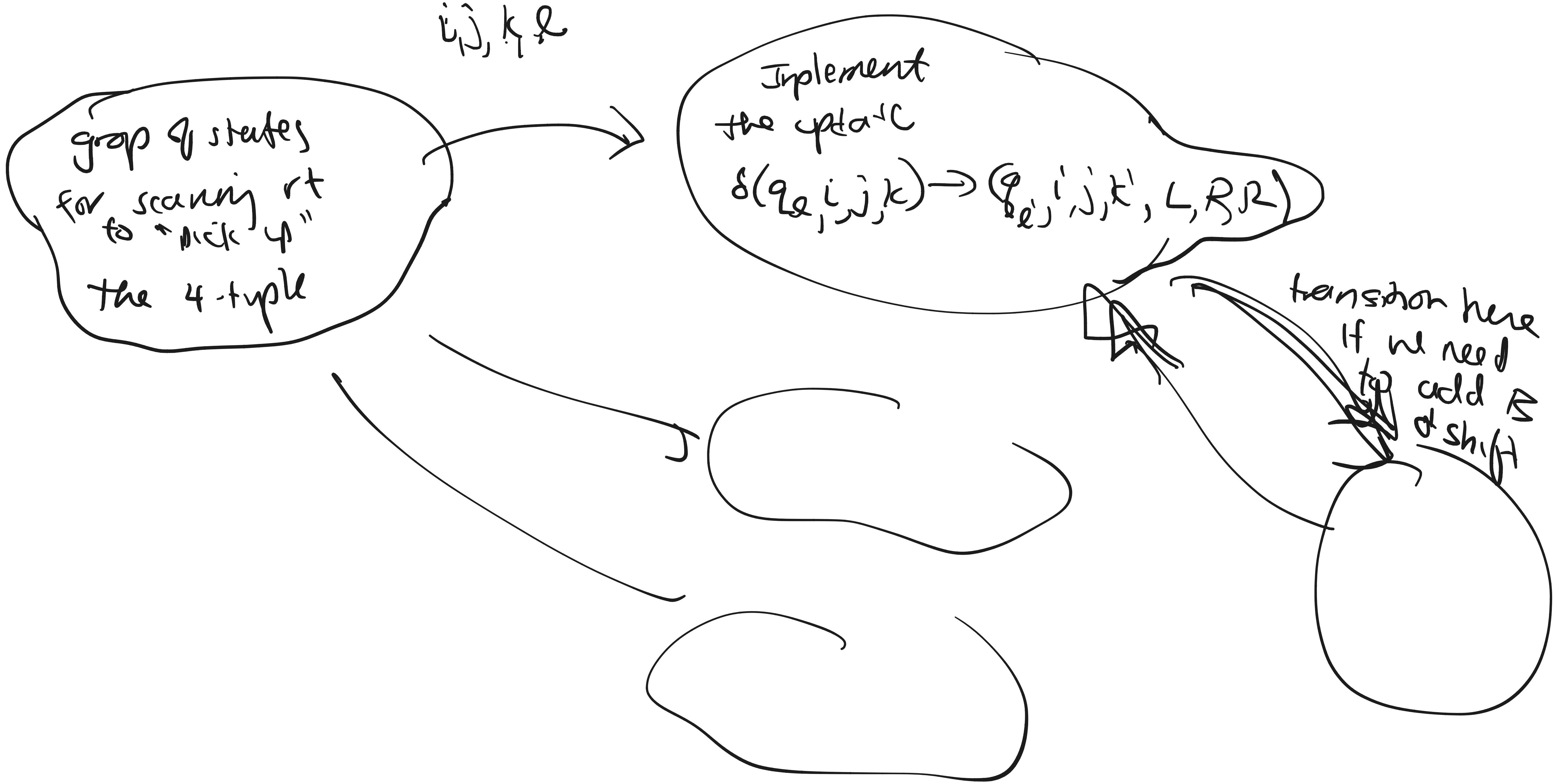
#	0	$\dot{1}$	0	#	0	0	$\dot{1}$	1	#	1	1	$\dot{0}$	0	#

$\rightsquigarrow$

#	$\dot{0}$	1	0	#	0	0	0	$\dot{1}$	#	1	1	0	$\dot{0}$	#

$$\delta(q_i, 1, 1, 0) \rightarrow (q_j, 1, 0, 0, L, R, R)$$

corresponding states of M': $\underline{q_{i,j,k,\ell}}$ $\quad i, j, k \in \Gamma \cup \dot{\Gamma}$
$\ell \in Q$


i, j, k, l
group of states for scanning rt to "pick up" the 4-tuple
Implement the update
δ(q_l, i, j, k) → (q_l', i', j', k', L, R, R)
transition here if we need to add B & shift

② Multitape TMs

Theorem Let $L \subseteq \Sigma^*$ be accepted by a k-tape TM, M. Then L is also accepted by a 1-tape TM.

Simulation of M by M' on $w = w_1 w_2 \cdots w_n$:

① Put M' in this form:

#	$\dot{w}_1$	w_2	w_3	⋯	w_n	#	$\dot{B}$	#	$\dot{B}$	#	B	B	⋯

② To simulate one transition of M: M' scan tape from first # to 4^{th} # and remembers symbols under each tape head (by state we are in)
Then make a second pass over tape to update tapes according to M's transition

③ If at any point M' moves one of its 'virtual heads' to the right onto a '#' then M' writes a 'B' symbol here and shift tape contents from this cell one unit to right (and then continue simulation as before)

Example

shift over

#	a	$\dot{a}$	b	#	a	b	b	$\dot{b}$	#	$\dot{a}$	a	#

$\Rightarrow$

#	a	a	$\dot{b}$	#	a	b	b	b	$\dot{B}$	#	a	$\dot{a}$	#

$$\delta(q_i, a, b, a) \to q_j, a, b, a, R, R, L$$

Equivalent TM Models

③ Nondeterministic TMs

A nondeterministic TM M is described by: $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

but Now $\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$

powerset

So given a pair (q_i, a) we can take 0, 1, or any number of transitions. (Note the total number of possible transitions from (q_i, a) is $|\mathcal{P}(Q \times \Gamma \times \{L, R\})| = 2^{|Q| \cdot |\Gamma| \cdot 2}$

Theorem Let L be accepted by some Nondet TM, M

Then L is accepted by a deterministic TM, M'

(and if M always halts, then M' always halts)

Turing machines computing functions

Defn. An input-output TM M is a tuple $(Q, \Sigma, \Gamma, \delta, q_{start}, q_{halt})$.

On an input $w \in \Sigma^*$, M runs on w (using the transition function δ)

- M halts if it reaches the state q_{halt} and the output is the string written on the tape up to the place where tape is all $\cancel{B}$'s

$$\underbrace{0\ 1\ 1\ 0\ 1}_{\text{output}}\ \cancel{B}\ \cancel{B}\ \cdots$$

- alternatively output is the string after '\$' and up to the place where tape is all $\cancel{B}$'s

$$\$\ \underbrace{0\ 1\ 1\ 0\ 1}_{\text{output}}\ \cancel{B}\ \cancel{B}\ \cdots$$

Turing machines computing functions

Defn. An input-output TM M is a tuple $(Q, \Sigma, \Gamma, \delta, q_{start}, q_{halt})$.

On an input $w \in \Sigma^*$, M runs on w (using the transition function δ)

M halts if it reaches the state q_{halt} and the output is the string written on the tape (up to the place where tape is all B's)

Defn $f : \Sigma^* \to \Sigma^*$ is TM-computable if $\exists$ an input-output TM M such that $\forall w \in \Sigma^*$ M on input w halts and outputs $f(w)$

Examples of Computable Functions (practice problems)

1. On input n (represented in unary) output $n+1$ in unary
   → give a 1-tape TM. ex: $w = 1111$ output 11111

2. On input n in binary notation output $n+1$ in binary
   → give a 1-tape TM ex. $w = 0111$ output 1000

3. Insert # in beginning of input: $w \longrightarrow \#w$
   → give a 1-tape TM

4. unary addition: $1^n \# 1^m \longrightarrow 1^{n+m}$
   → give a 1-tape TM

5. binary addition: $u \# v \longrightarrow w$ such that $u+v=w$
   → give a 3-tape or 1-tape TM Ex. $\underbrace{0100}_{u}\#\underbrace{0101}_{v}$ output $1001 = w$

6. binary multiplication: $u \# v \longrightarrow w$ such that $u \cdot v = w$
   → HARDER. give high level description of a k-tape TM

Church-Turing Thesis

Every reasonable model of computation can be simulated by a TM.

In other words, TMs can compute any function that can be computed by any current/future computational device