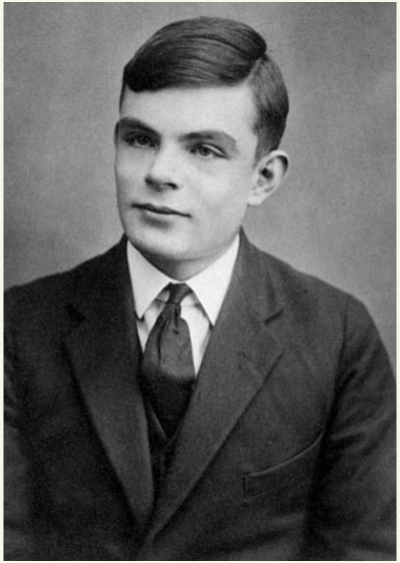
Lecture 14

HW3 : out Nov 6, DUE NOV 20

Today : Computability and Turing Machines

Turing Machines

"On computable Numbers, with an application to the Entscheidungsproblem"
1936

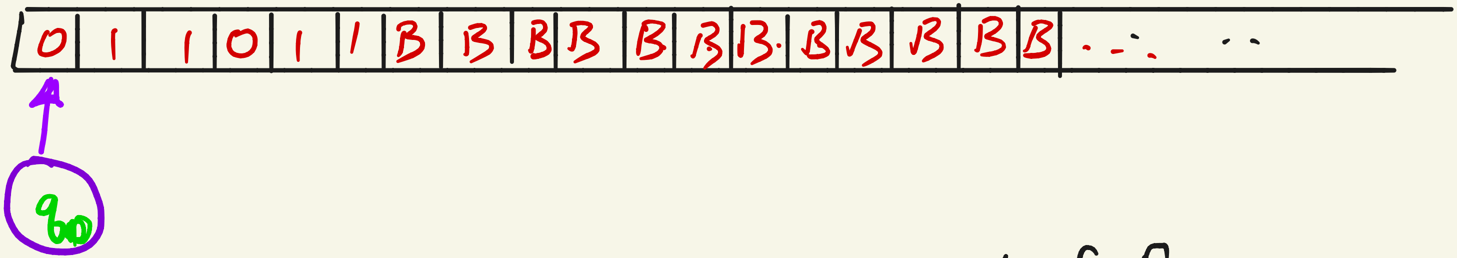


1912 - 1954

- Concept of 1st general model of computation.
- Proved there is no algorithm for deciding truth in mathematics
- code breaking of Nazi ciphers WWII
- also worked in mathematical biology
- prosecuted in '52 for homosexuality

Turing Machines

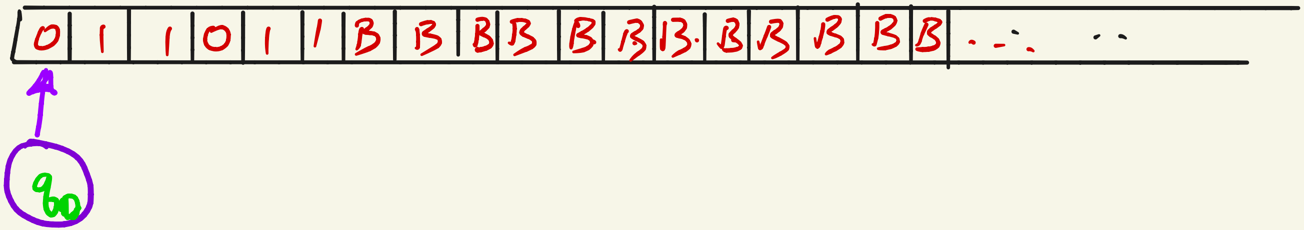
Input $x = 011011$



- Every cell of tape contains an element of Γ
- at every point in time, tape head points to some tape cell
- Initally head points to left most cell of tape, and start state is q_0
- Initially input $x = x_1 x_2 \cdots x_n$ in 1^{st} n cells; rest of cells contain 'B'

Turing Machines

Input $x = 011011$



- at every time step, M makes one transition according to δ

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

if at state q_i and head reads 'a',
replace 'a' by 'b', move head one cell
Left or Rt, and move to state q_j

- If we reach state q_{acc}, Halt and accept x
- If we reach q_{rej}, Halt and reject x
- otherwise (if we never reach q_{acc} or q_{rej}) loop forever

Turing Machines

$M = \{Q, \Sigma, \Gamma, \delta, q_0 \in Q, q_{acc} \in Q, q_{rej} \in Q\}$ $q_{acc} \neq q_{rej}$

$Q = \{q_0, \ldots, q_k\}$ states, $k \geq 2$

Σ = finite input alphabet (does not include 'B')

Γ = finite tape alphabet, $\Sigma \subseteq \Gamma$, includes 'B' (blank symbol)

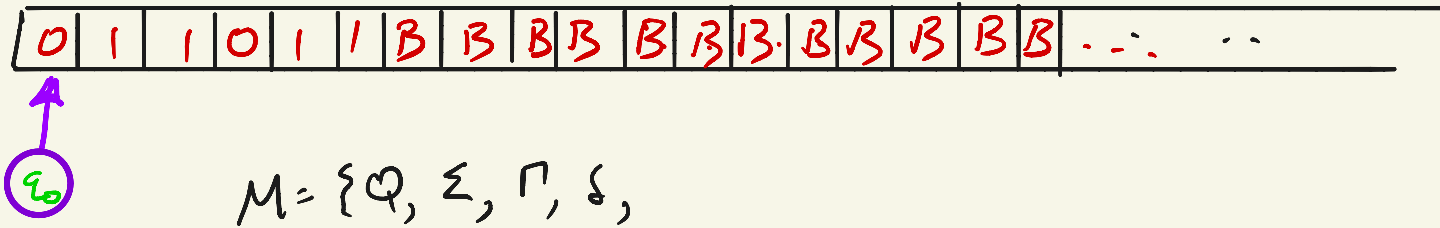
q_0 : start state

q_{acc}, q_{rej} : halt states (q_{acc} = accept state, q_{rej} = reject state)

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L \times R\}$

Turing Machines

Input $x = 011011$



$M = \{Q, \Sigma, \Gamma, \delta,$

$Q = \{q_0, q_1, q_2, q_3\}$ $\quad \Sigma = \{0, 1\}, \quad , \quad \Gamma = \{0, 1, B\}$

$\delta:$

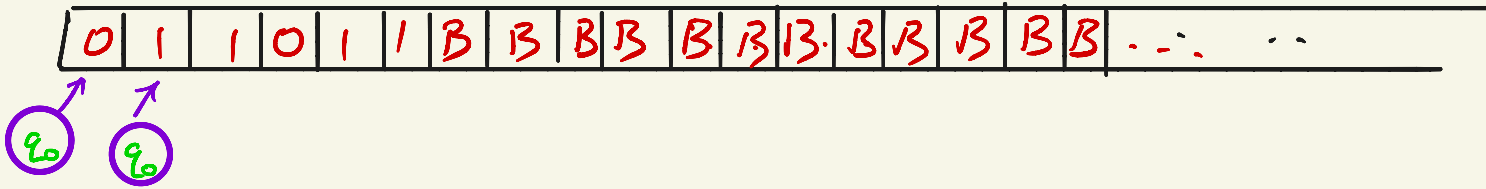
$$(0, q_0) \rightarrow (0, q_0, R)$$
$$(1, q_0) \rightarrow (1, q_1, R)$$
$$(0, q_1) \rightarrow (0, q_1, R)$$
$$(1, q_1) \rightarrow (1, q_0, R)$$
$$(B, q_0) \rightarrow (B, q_2, R)$$
$$(B, q_1) \rightarrow (B, q_3, R)$$

here q_2 = accept state

q_3 = reject state

Turing Machines

Input $x = 011011$

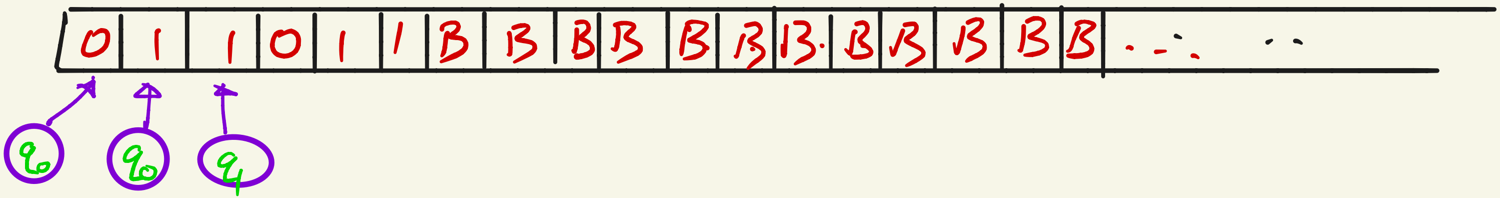


δ:

$$(0, q_0) \rightarrow (0, q_0, R)$$
$$(1, q_0) \rightarrow (1, q_1, R)$$
$$(0, q_1) \rightarrow (0, q_1, R)$$
$$(1, q_1) \rightarrow (1, q_0, R)$$
$$(B, q_0) \rightarrow (B, q_2, R)$$
$$(B, q_1) \rightarrow (B, q_3, R)$$

Turing Machines

Input $x = 011011$

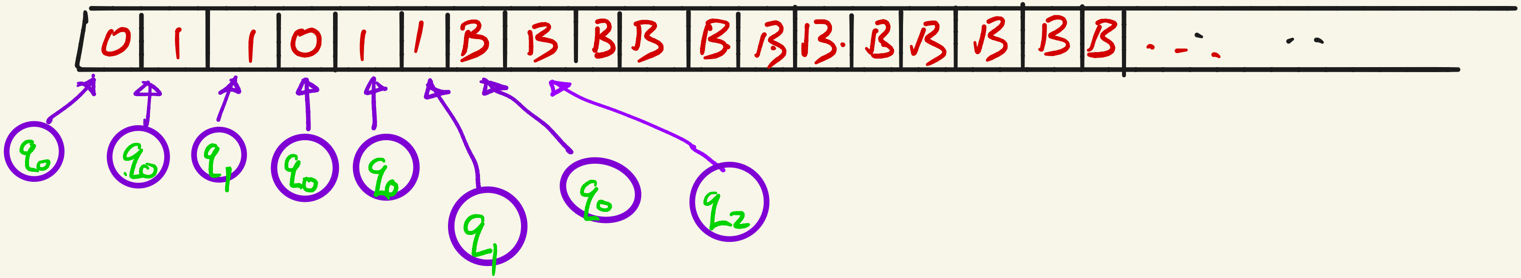


δ:

$$(0, q_0) \to (0, q_0, R)$$
$$(1, q_0) \to (1, q_1, R)$$
$$(0, q_1) \to (0, q_1, R)$$
$$(1, q_1) \to (1, q_0, R)$$
$$(B, q_0) \to (B, q_2, R)$$
$$(B, q_1) \to (B, q_3, R)$$

Turing Machines

Input $x = 011011$



δ:

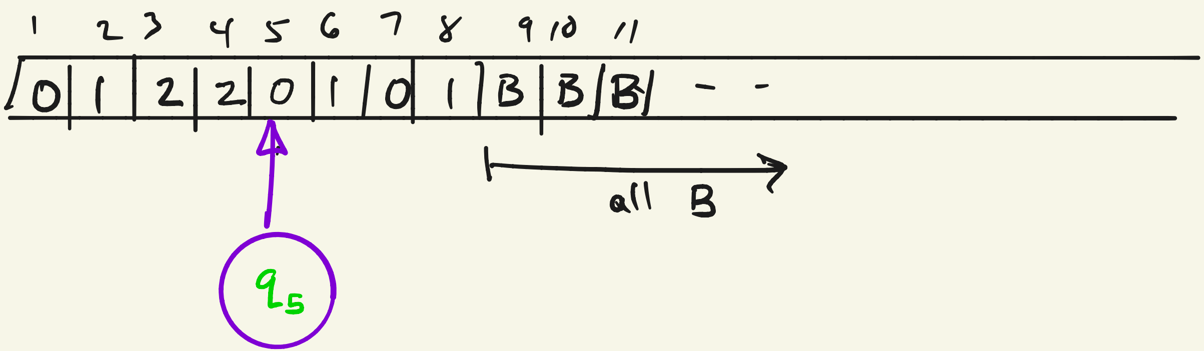
$$
\begin{aligned}
(0, q_0) &\to (0, q_0, R)\\
(1, q_0) &\to (1, q_1, R)\\
(0, q_1) &\to (0, q_1, R)\\
(1, q_1) &\to (1, q_0, R)\\
(B, q_0) &\to (B, q_2, R)\\
(B, q_1) &\to (B, q_3, R)
\end{aligned}
$$

Since M on x halts in state q_2, M accepts x

Turing Machine Configurations

- A configuration describes entire memory content of a TM at some point in time

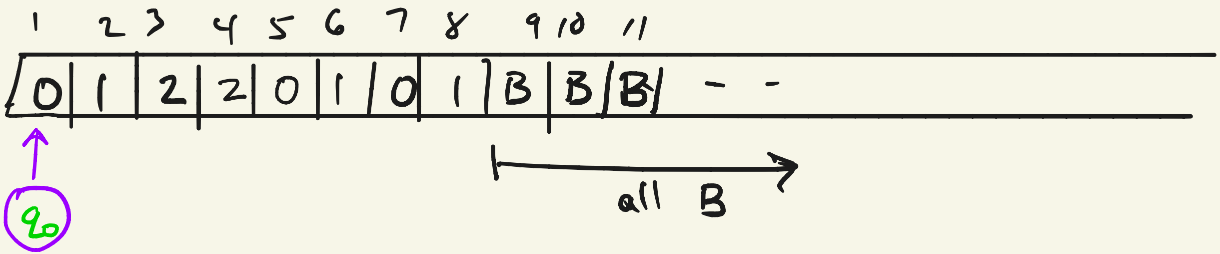
Example: Here $\Sigma = \{0,1,2\}$, $\Gamma = \{0,1,2,B\}$, $Q = \{q_0, q_1, \dots, q_5\}$



Configuration : $0122\,q_5\,0101$

Turing Machine Configurations

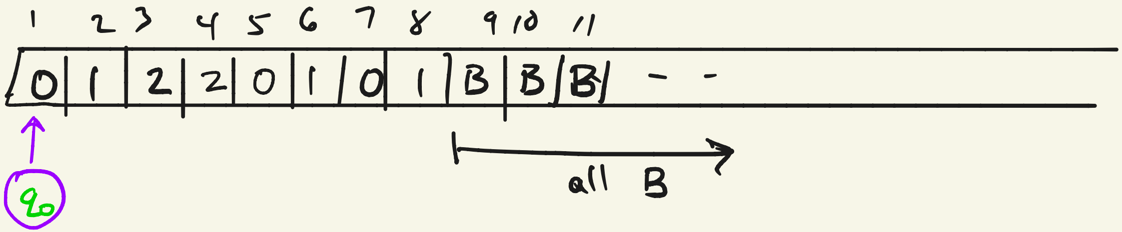
Start configuration on input $w = 01220101$



Start configuration : $q_0 01220101 = q_0 w$

Turing Machine Configurations

Start configuration on input $w = 01220101$



Start Configuration : $q_0 01220101 = q_0 w$

accept configuration : $u\, q_{acc}\, v$ (the state is q_{acc})

Reject Configuration : $u\, q_{rej}\, v$ (the state is q_{rej})

Turing Machine Configurations

Definition Configuration C_1 yields C_2 $\quad (C_1 \Rightarrow C_2)$

if the TM can go from C_1 to C_2 in one step

More formally: Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

Let $a, b \in \Gamma$, $u, v \in \Gamma^*$, $q_i, q_j \in Q$

Then $\underbrace{u a q_i b v}_{C_1} \Rightarrow \underbrace{u a c q_j v}_{C_2}$ if $\delta(q_i, b) = (q_j, c, R)$

and $\underbrace{u a q_i b v}_{C_1} \Rightarrow \underbrace{u q_j a c v}_{C_2}$ if $\delta(q_i, b) = (q_j, c, L)$

Turing Machine Configurations

Special end cases (when head is pointing to left or rt end of configuration):

If head at Left end:

$$q_i b v \Longrightarrow q_j c v \quad \text{if} \quad \delta(q_i, b) = (q_j, c, L)$$

$$q_i b v \Longrightarrow c q_j v \quad \text{if} \quad \delta(q_i, b) = (q_j, c, R)$$

if tape head tries to go Left from leftmost cell, head just stays in place

Language $L \subseteq \Sigma^*$ accepted by a TM

Defn A TM M accepts an input $w \in \Sigma^*$ iff

$\exists$ a sequence of configurations $C_1, C_2, \ldots, C_K$ such that:

1. C_1 is the start config of M on input w
2. Each C_i yields C_{i+1}
3. C_K is an accept configuration

$$L(M) = \{ w \in \Sigma^* \mid w \text{ is accepted by } M \}$$

Recursive / Turing Decidable Languages

Recursively Enumerable / Turing Recognizable Languages
(RE)

Note that when we run M on w either:

(i) M halts and accepts w

(ii) M halts and rejects w

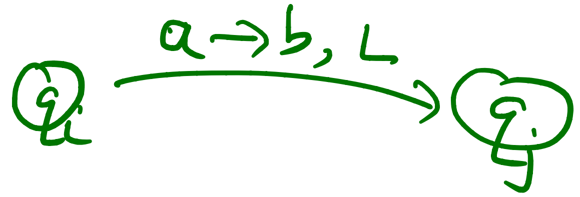
(iii) M never halts on w (goes into infinite loop, never reaching the accept or reject state)

L is RE (Turing Recognizable) if $\exists$ TM M st $L(M) = L$

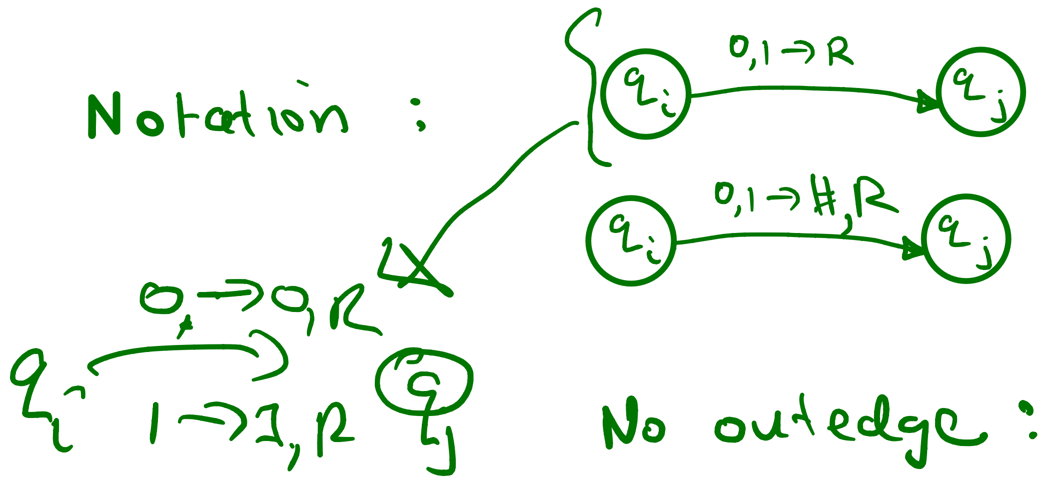
RE ← recursively enumerable

L is Recursive (Turing Decidable) if $\exists$ TM M st $L(M) = L$ and M halts on every input $w \in \Sigma^*$

Example : $L = \{0^n 1^n 2^n \mid n \geq 1\}$



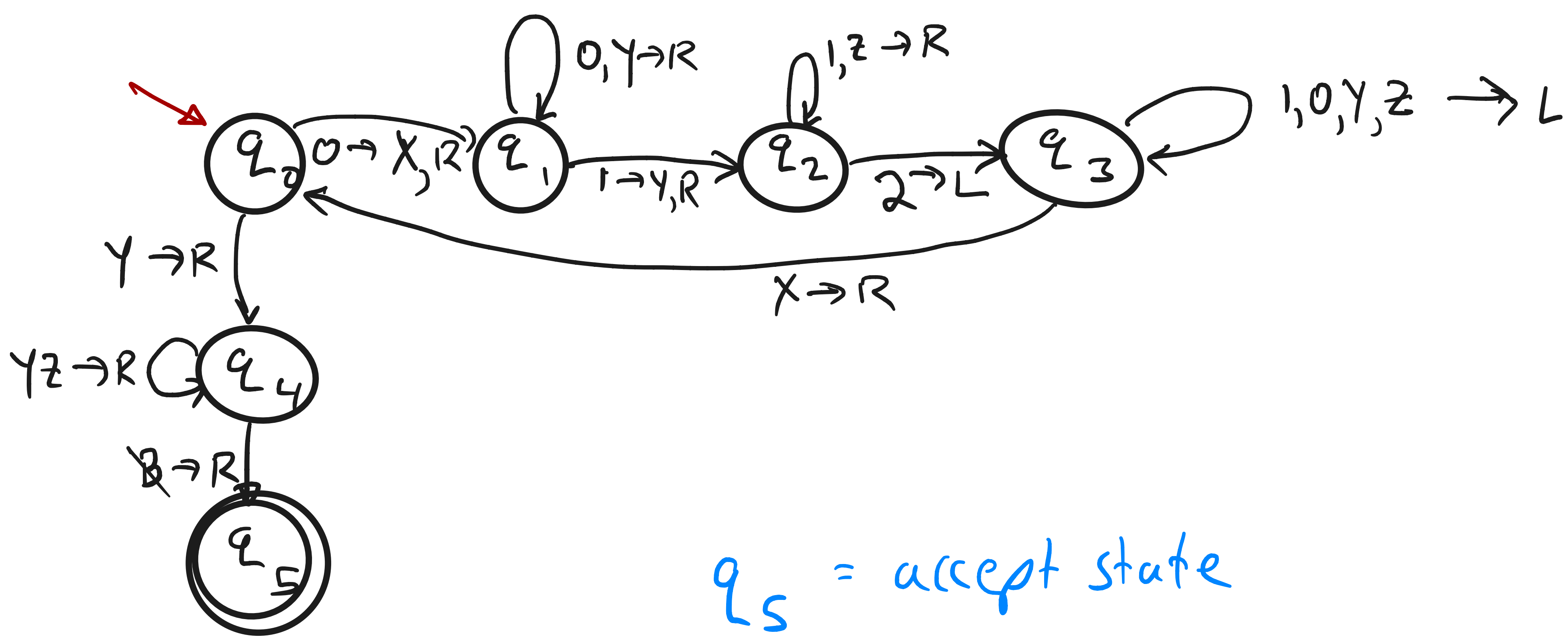
Notation :



If head reads 0 or 1 go R one cell and move to q_j (dont change symbol on tape)

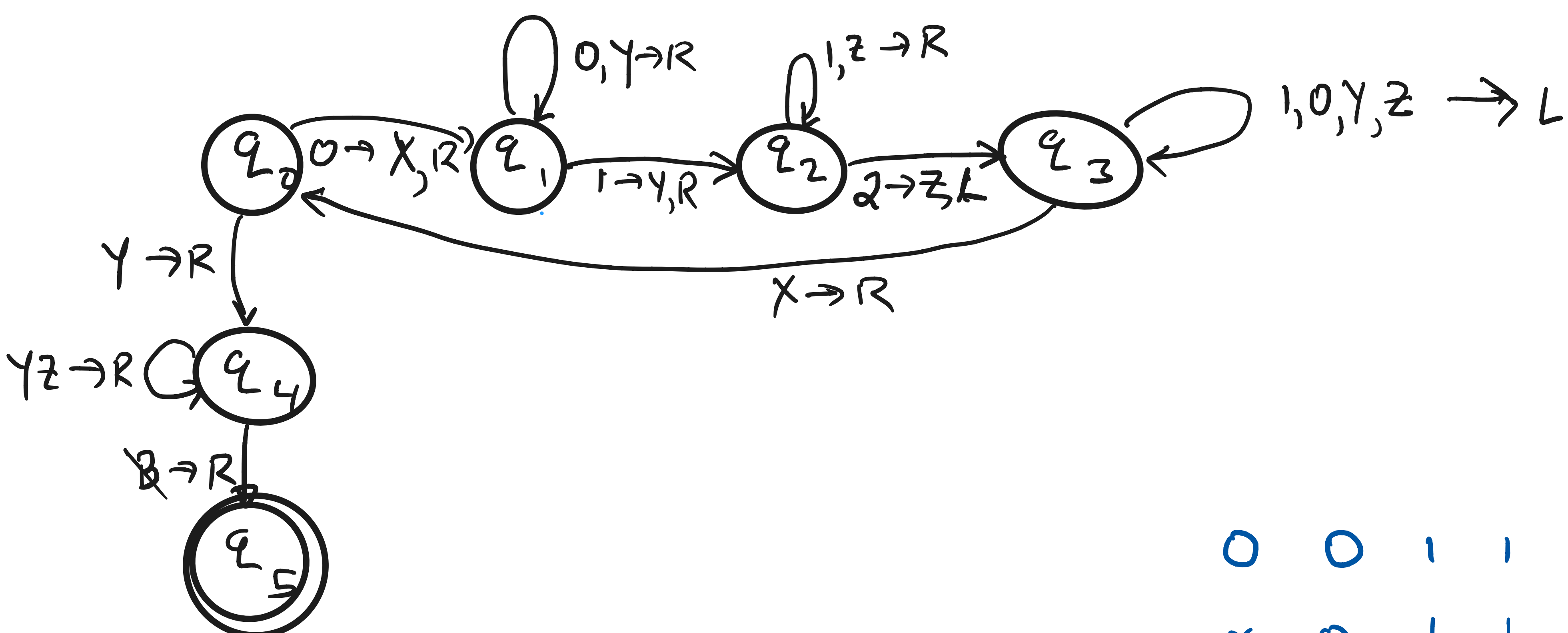
If head reads 0 or 1, change symbol to #, move R one cell and move to state q_j

No outedge : transition to q_{rej}



q_5 = accept state

Example : $L = \{0^n 1^n 2^n \mid n \geq 1\}$



0	0	1	1	2	2	B
x	0	1	1	2	2	B
X	0	Y	1	2	2	B
x	0	Y	1	Z	2	B
x	x	Y	1	Z	2	B
x	x	Y	Y	Z	2	B
x	x	Y	Y	Z	Z	B

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

Pseudocode:

On input w:

(1) Scan input to make sure it has one # symbol. If not reject

(2) go back and forth across tape to corresponding pair of positions on either side of "#" symbol

If they dont match reject.

OW, cross off matched symbols

(3) If all symbols to left/rt of '#' are crossed off, check if any symbols to rt of #

If any symbols remain → reject

OW accept

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

0 1 1 0 0 1 # 0 1 1 0 0 1

(remember 0)

B 1 1 0 0 1 # 0 1 1 0 0 1

(matches!)

B 1 1 0 0 1 # 0 1 1 0 0 1

B 1 1 0 0 1 # X 1 1 0 0 1

B 1 1 0 0 1 # X 1 1 0 0 1

(remember 1)

B X 1 0 0 1 # X 1 1 0 0 1

(matches)

B X 1 0 0 1 # X 1 1 0 0 1

(matches)

B X 1 0 0 1 # X X 1 0 0 1

remember 1)

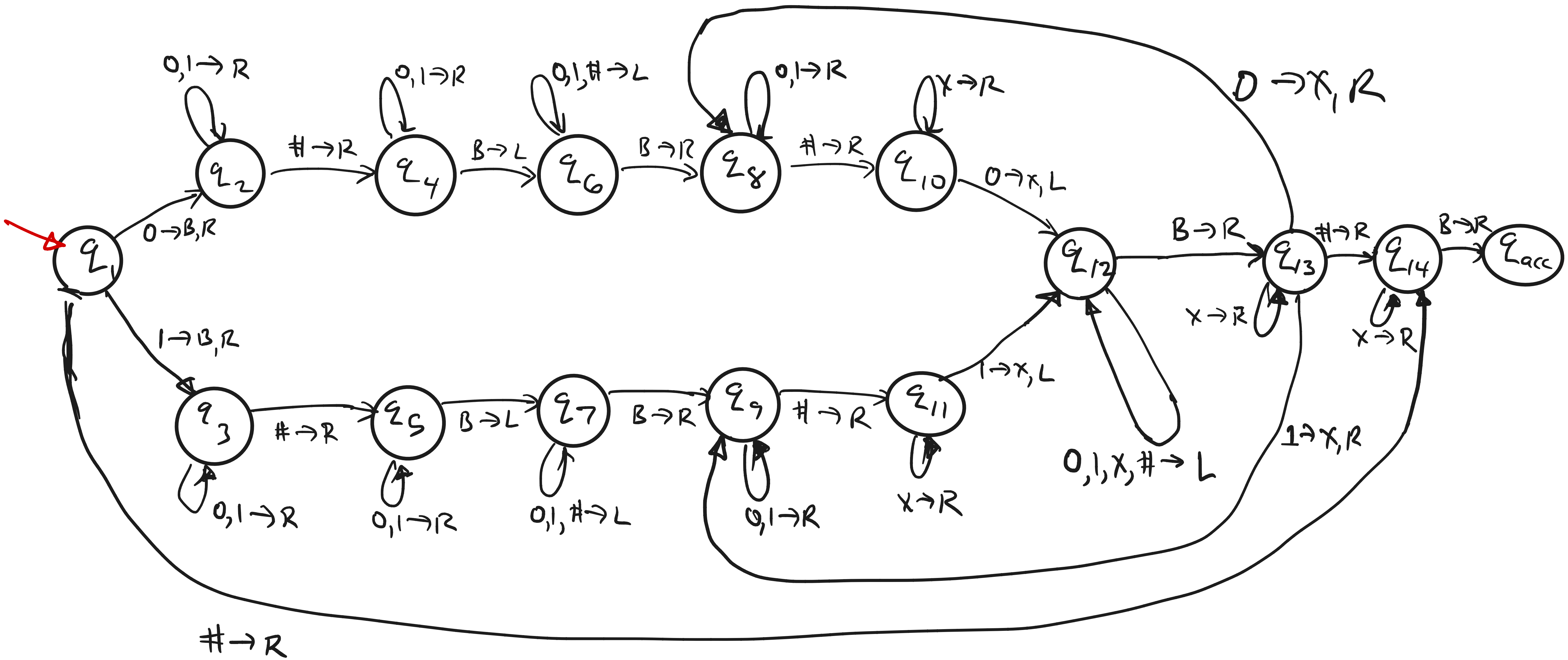
B X X 0 0 1 # X X 1 0 0 1

matches

B X X 0 0 1 # X X X 0 0 1

. . .

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



① $q_1 - q_7$: scan input replace 1st cell by B and remember 0 (q_6) or 1 (q_7). Scan to find one #, then go back left.

$q_8\ q_{10}\ q_{12}$: scan to # then to 1st 0/1 after #. If it matches (=0) continue and overwrite with X

$q_9\ q_{11}\ q_{12}$: same as $q_8 - q_{10}$ but remember '1' and match # a 1, overwrite w/ X

$q_{12}\ q_{13}$: scan Left and repeat

②,③