TURING MACHINE

Recursively Innumerable Language

Final April 29

Last assignment is 9

OVERVIEW

- Turing Machines
- Turing-recognizable
- Turing-decidable
- Variants of Turing Machines
- Algorithms
- Encoding input for TM

REVIEW

- DFA
  - Reads input from left to right
  - Finite control (i.e., transition function) based on
    - current state, ✓
    - current input symbol read. ✓

regular DFA, NFA

- PDA
  - Has stack for extra memory
  - Reads input from left to right
  - Can read/write to memory (stack) by popping/pushing
  - Finite control based on
    - current state, ✓
    - what's read from input, ✓
    - what's popped from stack ✓

non regular
extra memory, stack

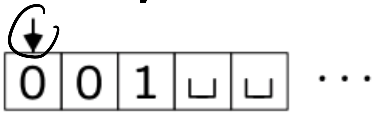
memory is limited

Memory is infinite

infinite like array

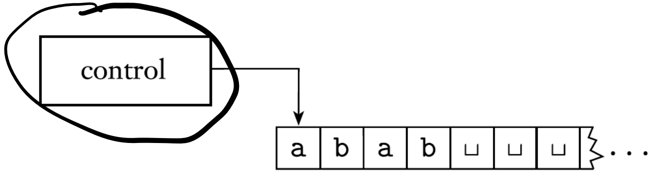
TURING MACHINE (TM)

- Infinitely long tape, divided into cells, for memory
- Tape initially contains input string followed by all blanks



- Tape head (↓) can move both right and left
- Can read from and write to tape , delete, update
- Finite control based on
  - current state,
  - current symbol that head reads from tape.

multiple



- Machine has one accept state and one reject state, infinite loop state
- Machine can run forever: infinite loop.

For loop , not halting , undesirable machines

- If it doesn't enter an accepting or a rejecting state, it will go on forever, never halting.

KEY DIFFERENCE BETWEEN TMS AND PREVIOUS MACHINES

- Turing machine can both read from tape and write on it.
- Tape head can move both right and left. , one cell at a time
- Tape is infinite and can be used for storage. storage is also infinite
- Accept and reject states take immediate effect. , infinite (not halting)

Example: Machine for recognizing language

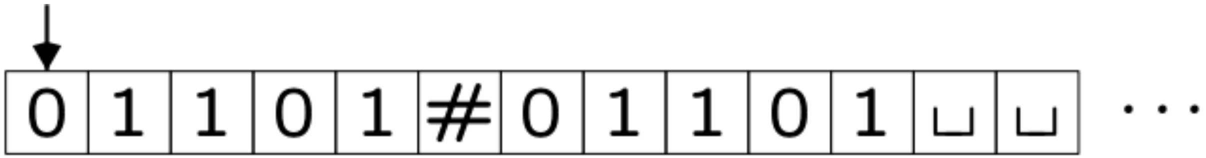
string hash string

$$A = \{ s\#s \mid s \in \{0, 1\}^* \}$$

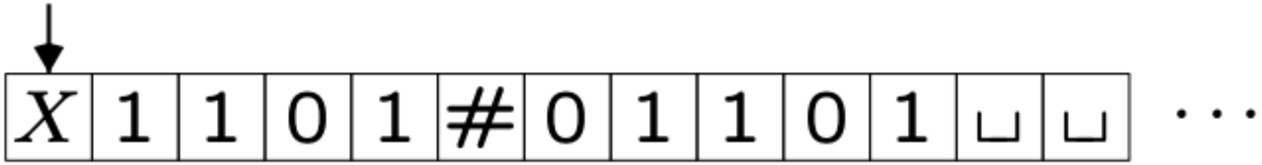
$w\#w$

Idea: Zig-zag across tape, crossing off matching symbols.

- Consider string 01101#01101 ∈ A.
- Tape head starts over leftmost symbol

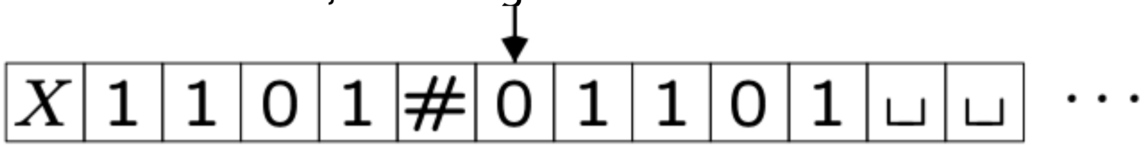


- Record symbol in control and overwrite it with X

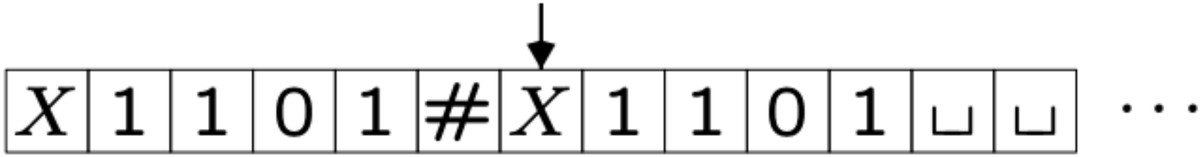


- Scan right: reject if blank "⊔" encountered before #

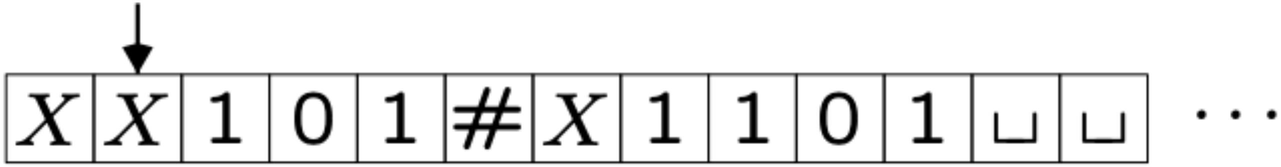
- When # encountered, move right one cell.



- If current symbol doesn't match previously recorded symbol, reject.
- Overwrite current symbol with X

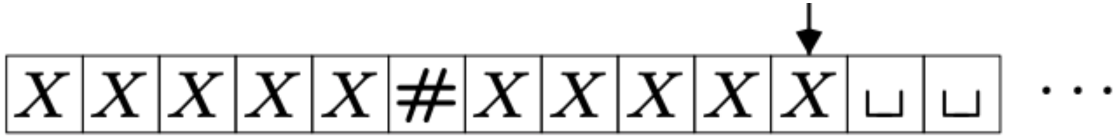


- Scan left, past # to X
- Move one cell right
- Record symbol and overwrite it with X

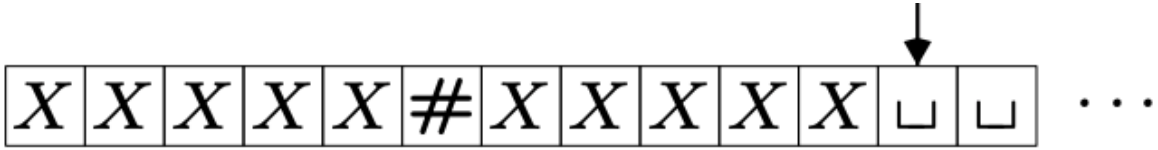


- Scan right past # to (last) X and move one cell to right . . .

- After several more iterations of zigzagging, we have



- After all symbols left of # have been matched to symbols right of #, check for any remaining symbols to the right of #.
  - If blank $\sqcup$ encountered, accept.
  - If 0 or 1 encountered, reject.



- The string that is accepted or not by our machine is the original input string 01101#01101.

Description of TM M_1 for $\{ s\#s \mid s \in \{0, 1\}^* \}$

M_1 = "On input string w:

1. Scan input to be sure that it contains a single #. If not, reject.
2. Zig-zag across tape to corresponding positions on either side of the # to check whether these positions contain the same symbol. If they do not, reject. Cross off symbols as they are checked off to keep track of which symbols correspond.
3. When all symbols to the left of # have been crossed off along with the corresponding symbols to the right of #, check for any remaining symbols to the right of the #. If any symbols remain, reject; otherwise, accept."

7-tuple

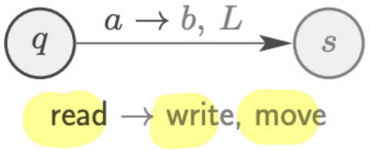
FORMAL DEFINITION OF TURING MACHINE

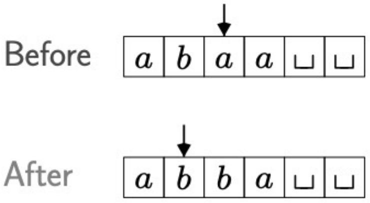
Definition: A Turing machine (TM) is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where

1. Q is a finite set of states
2. Σ is the input alphabet not containing blank symbol $\sqcup$
3. Γ is tape alphabet with blank $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function, where L means move tape head one cell to left R means move tape head one cell to right
5. $q_0 \in Q$ is the start state
6. $q_{accept} \in Q$ is the accept state
7. $q_{reject} \in Q$ is the reject state, with $q_{reject} \neq q_{accept}$

TRANSITION FUNCTION OF TM

- Transition function $\boxed{\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}}$
- $\delta(q, a)=(s, b, L)$ means
  - if TM
    - in state $q \in Q$, and
    - tape head reads tape symbol $a \in \Gamma$,
  - then TM
    - moves to state $s \in Q$
    - overwrites a with $b \in \Gamma$
    - moves head left (i.e., $L \in \{L, R\}$)

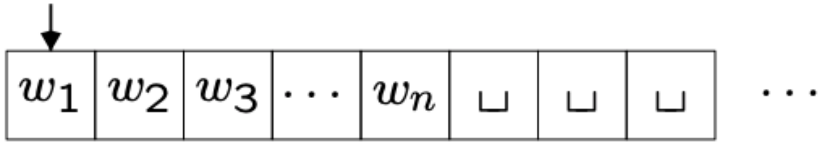




$a \rightarrow a, L = a \rightarrow L$

START OF TM COMPUTATION

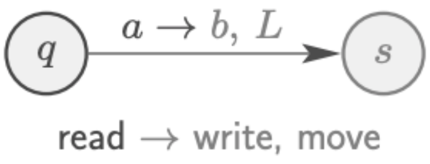
- A TM, $M = (Q, \Sigma, \Gamma, \delta, q0, q_{accept}, q_{reject})$ begins computation as follows:
- Given input string $w = w_1 w_2 \cdots w_n \in \Sigma^*$ with each $w_i \in \Sigma$, i.e., w is a string of length n for some $n \geq 0$.
- TM begins in start state q_0
- Input string is on n leftmost tape cells



- Rest of tape contains blanks $\sqcup$
- Head starts on leftmost cell of tape
- Because $\sqcup \notin \Sigma$, first blank denotes end of input string.
- The computation continues until it enters either the accept or reject states, at which point it halts. If neither occurs, M goes on forever.

- When computation on TM M = (Q, Σ, Γ, δ, q0, q_{accept}, q_{reject}) starts,
- TM M proceeds according to transition function

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



- If M tries to move head off left end of tape, then head remains on first cell.
- Computation continues until q_{accept} or q_{reject} is entered.
- Otherwise, M runs forever: infinite loop.
- In this case, input string is neither accepted nor rejected.
- As TM computes changes occurs in current state, the current tape contents, and the current head location, these items altogether is called configuration.

OPERATIONS ON THE TAPE

- Read or scan symbol below the tape head
- Update/write symbol below the tape head
- Move tape head one step Left ‘L’
- Move tape head one step Right ‘R’

UNDERSTANDING THE WAY, A TM COMPUTES

Configuration C_1 yields configuration C_2 if the TM can legally go from C_1 to C_2 in a single step.

- Definition:
  - Suppose that we have a, b, and c in Γ, as well as u and v in Γ* and states q_i and q_j.
    - In that case, ua q_i bv and u q_j acv are two configurations.

      ua $q_i$bv *yields* u q_j acv
  - if in the transition function $\delta(q_i, b) = (q_j, c, L)$. That handles the case where the Turing machine moves leftward.
  - For a rightward move, say that ua q_i bv *yields* uac q_j v
  - if δ(qi, b) = (qj, c,R). Special cases occur when the head is at one of the ends of the configuration.
  - For the left-hand end, the configuration q_i bv *yields* q_j cv if the transition is left moving. It yields c q_j v for the right-moving transition.
  - For the right-hand end, the configuration uaq_i is equivalent to uaq_i ⊔ because we assume that blanks follow the part of the tape represented in the configuration.

Configuration C_1 yields configuration C_2 if the Turing machine can legally go from C1 to C2 in a single step.

Specifically, for TM M = (Σ, Γ, δ, q0, q_{accept}, q_{reject}), suppose

no need for blanks

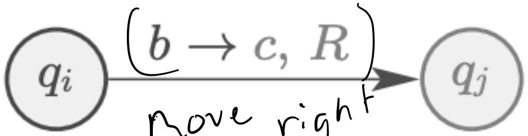
- u, v $\in \Gamma^*$
- a, b, c $\in \Gamma$
- q_i, $q_j \in Q$
- transition function $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$.

$u a q_i b v$

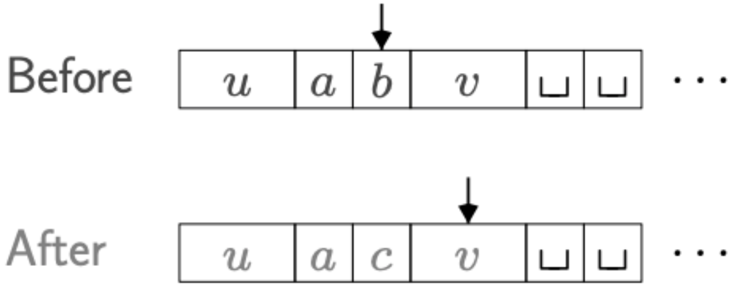
- Then configuration uaq_ibv yields configuration $uacq_jv$ if

$$\delta(q_i, b) = (q_j, c, R).$$

Read b
write c



move right

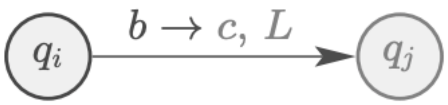


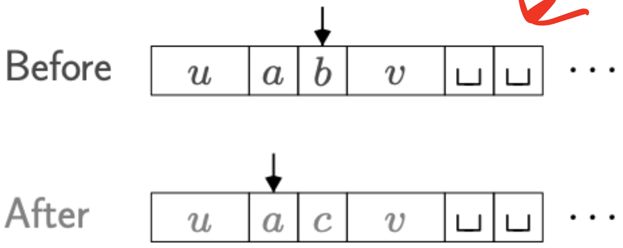
$u a c q_j v$

- Similarly, configuration uaq_ibv yields configuration uq_jacv if

$$\delta(q_i, b) = (q_j, c, L).$$

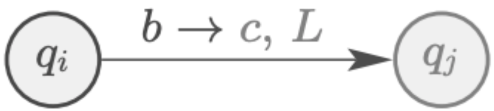
uaq_ibv

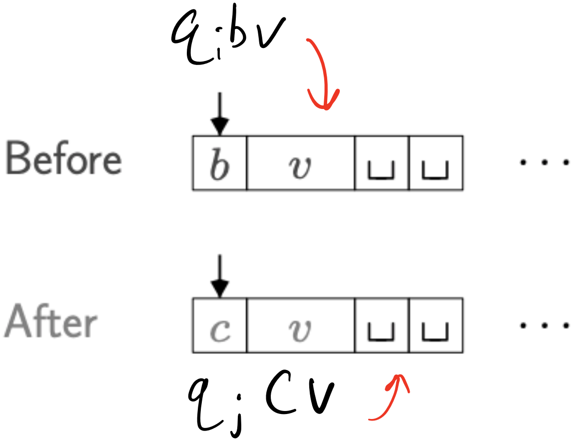




uq_jacv

- Special case: $q_i bv$ yields $q_j cv$ if $\delta(q_i, b)=(q_j, c, L)$
- If head is on leftmost cell of tape and tries to move left, then it stays in same place.

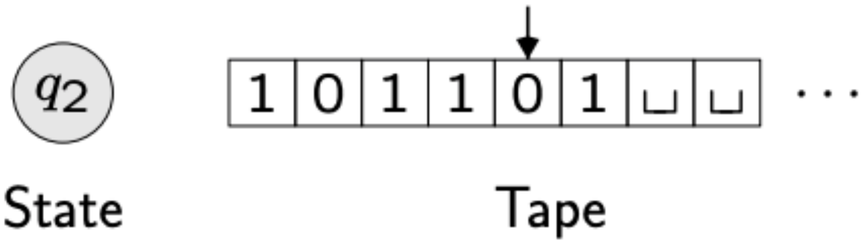




TM CONFIGURATIONS

- Computation changes
  - current state
  - current head position
  - tape contents State

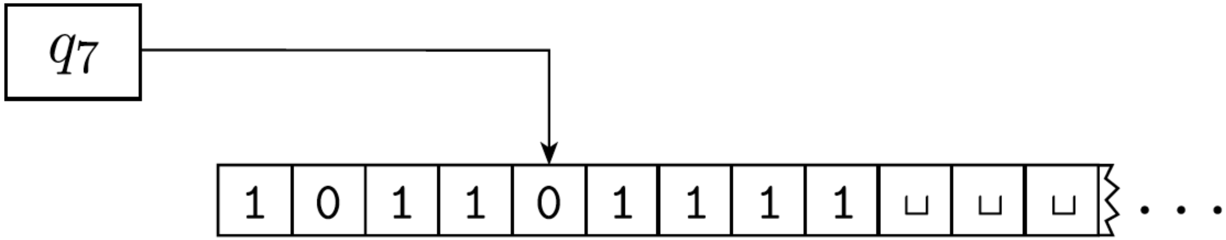
$1011q_2 01$



- Configuration provides "snapshot" of TM at any point during computation:
  - current state q ∈ Q
  - current tape contents ∈ Γ*
  - current head location

CONFIGURATIONS

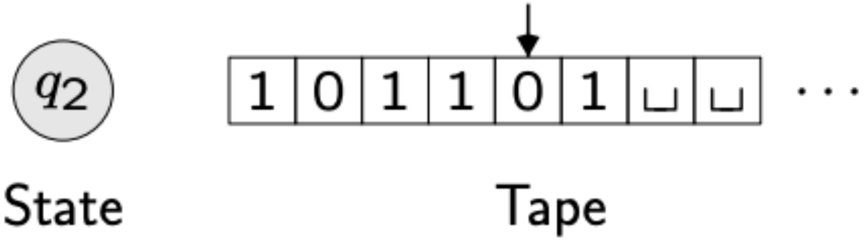
- For a state q and two strings u and v over the tape alphabet Γ, we write u q v for the configuration where the current state is q, the current tape contents is uv, and the current head location is the first symbol of v.
- The tape contains only blanks following the last symbol of v.
- For example, 1011q701111 represents the configuration when the tape is 101101111, the current state is q7, and the head is currently on the second 0.



$1011q_7 01111$

- Configuration 1011q201 means
  - current state is q2
  - LHS of tape is 1011
  - RHS of tape is 01
  - head is on RHS 0

$1011q_2 01$



- Definition: a configuration of a TM M = (Q, Σ, Γ, δ, q_0, q_{accept}, q_{reject}) is a string uqv with u, v ∈ Γ∗ and q ∈ Q, and specifies that currently
  - M is in state q
  - tape contains uv
  - tape head is pointing to the cell containing the first symbol in v

CONFIGURATIONS

Consider TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.

- Starting configuration on input $w \in \Sigma^*$ is q_0w, starting state
- An accepting configuration is $uq_{accept}v$ for some $u, v \in \Gamma^*$, accept state
- A rejecting configuration is $uq_{reject}v$ for some $u, v \in \Gamma^*$, rejecting state
- Accepting and rejecting configurations are halting configurations.
- Configuration wq_i is the same as $wq_i\sqcup$

TERMINOLOGIES RECALL

- The collection of strings that M accepts is the language of M, or the language recognized by M, denoted L(M)
- Call a language Turing-recognizable if some TM recognizes it.
- When we start a TM on an input, 3 outcomes are possible. The machine may accept, reject, or loop.
  - loop - the machine does not halt.
- A TM M can fail to accept an input by entering the q_{reject} state and rejecting, or by looping.
- TMs that halt on all inputs, such machines never loop. These machines are called deciders because they always make a decision to accept or reject.
- A decider that recognizes some language also is said to decide that language.
- Call a language Turing-decidable or decidable if some TM decides it.

TM DESIGN

TURING MACHINES

- Finite automata accept precisely the regular languages.
- We may need unbounded memory to recognize context-free languages.
  - e.g. $\{ 0^n 1^n \mid n \in \mathbb{N} \}$ requires unbounded counting.
- How do we build an automaton with finitely many states but unbounded memory?

TM AT EACH STEP

- writes a symbol to the tape cell under the tape head,
- changes state, and
- moves the tape head to the left or to the right.

0 x 0 x
0 x 0 x 0

- Consider a TM, M_2 that recognizes, $A = \{0^{2^n} \mid n \geq 0\}$,

  which consists of strings of 0s whose length is a power of 2.

- Basic idea: The number k of zeros is a power of 2 iff successively halving k always results in a power of 2 (i.e., each result > 1 is never odd)

- M2 = "On input string w:
  - Sweep left to right across the tape, crossing off every other 0.
  - If in stage 1 the tape contained a single 0, accept.
  - If in stage 1 the tape contained more than a single 0 and the number of 0s was odd, reject .
  - Return the head to the left-hand end of the tape.
  - Go to stage 1."

$n=0$ 0
even # of 0's

$n=1$ 00

$n=2$ 0000

$0^{2^1} = 0^2 = 00$

$0^{2^0} = 0^1 = 0$

Basic steps

- Each iteration of stage 1 cuts the number of 0s in half.
- In stage 1, it keeps track of whether the number of 0s seen is even or odd.
- If that number is odd and greater than 1, the original number of 0s in the input could not have been a power of 2.
- Therefore, the machine rejects in this instance.
- However, if the number of 0s seen is 1, the original number must have been a power of 2.

Running TM with input 0000

- Tape initially contains input 0000.

↓					
0	0	0	0	␣	...

- Run stage 1: Sweep left to right across tape, crossing off every other 0.

				↓	
␣	x	0	x	␣	...

(Put ␣ in first cell to mark beginning of tape.)

- Run stage 4: Return head to left end of tape (marked by ␣).

↓					
␣	x	0	x	␣	...

- Run stage 1: Sweep left to right across tape, crossing off every other 0.

				↓	
␣	x	x	x	␣	...

- Run stages 4 and 1: Return head to left end and scan tape.
- Run stage 2: If in stage 1 the tape contained a single 0, *accept*.

NOTATIONS

$a \to b, R$

- $\delta(q_i, a) = (q_j, b, R)$ is denoted by an arrow that starts at q_i, ends at q_j, and is labeled by $a \to b, R$

$a \to b, L$

- $\delta(q_i, a) = (q_j, b, L)$ is denoted by an arrow that starts at q_i, ends at q_j, and is labeled by $a \to b, L$

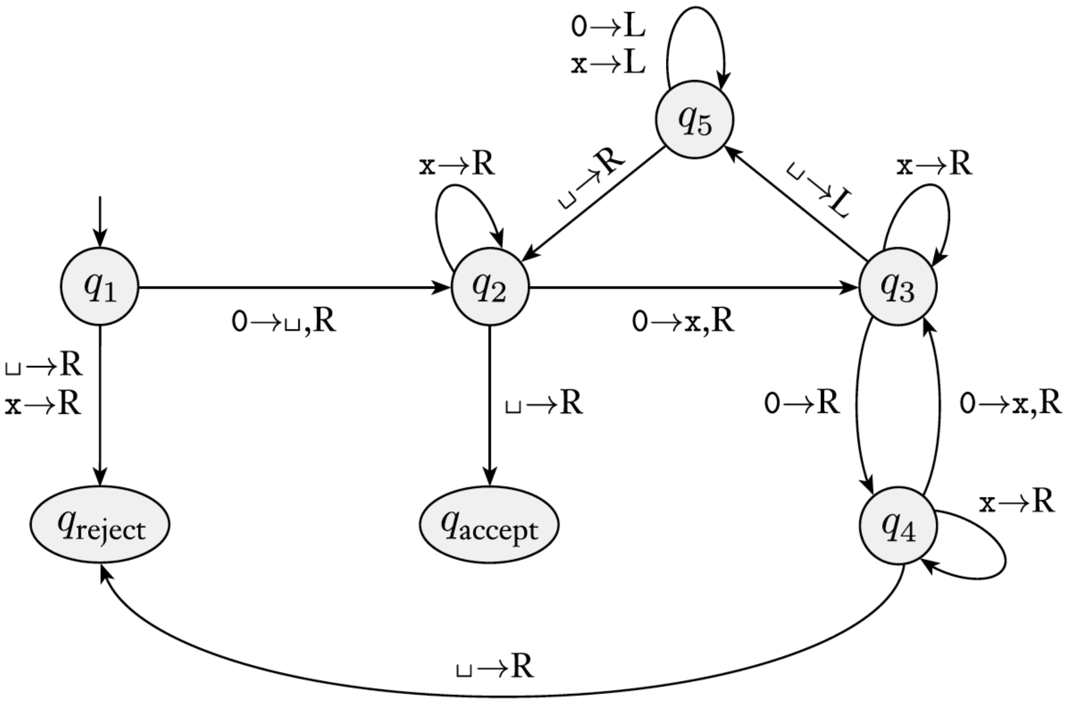
$a \to R$

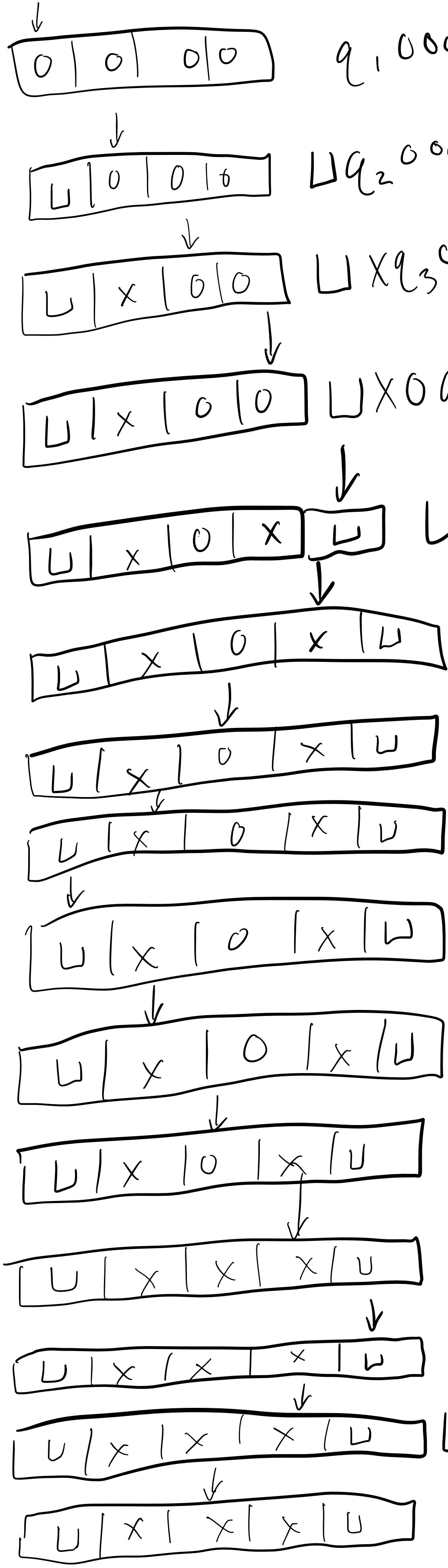
- $\delta(q_i, a) = (q_j, a, R)$ is denoted by an arrow that starts at q_i, ends at q_j, and is labeled by $a \to R$

$a \to L$

- $\delta(q_i, a) = (q_j, a, L)$ is denoted by an arrow that starts at q_i, ends at q_j, and is labeled by $a \to L$

TM FOR $A = \{0^{2^n} \mid n \geq 0\}$





$q_1 0000$

$\sqcup q_2 000$

$\sqcup x q_3 00$

$\sqcup x 0 q_4 0$

$\sqcup x 0 x q_3 \sqcup$

$\sqcup x 0 q_5 x \sqcup$

$\sqcup x q_5 0 x \sqcup$

$\sqcup q_5 x 0 x \sqcup$

$q_5 \sqcup x 0 x \sqcup$

$\sqcup q_2 x 0 x \sqcup$

$\sqcup x q_2 0 x \sqcup$

$\sqcup x x q_3 x \sqcup$

$\sqcup x x x q_3 \sqcup$

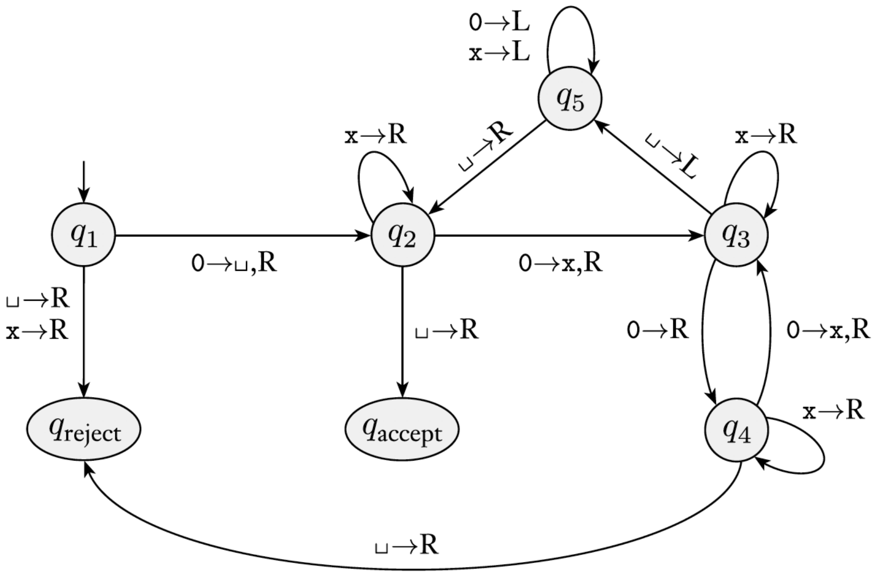
$\sqcup x x q_5 x \sqcup$

$\sqcup x q_5 x x \sqcup$

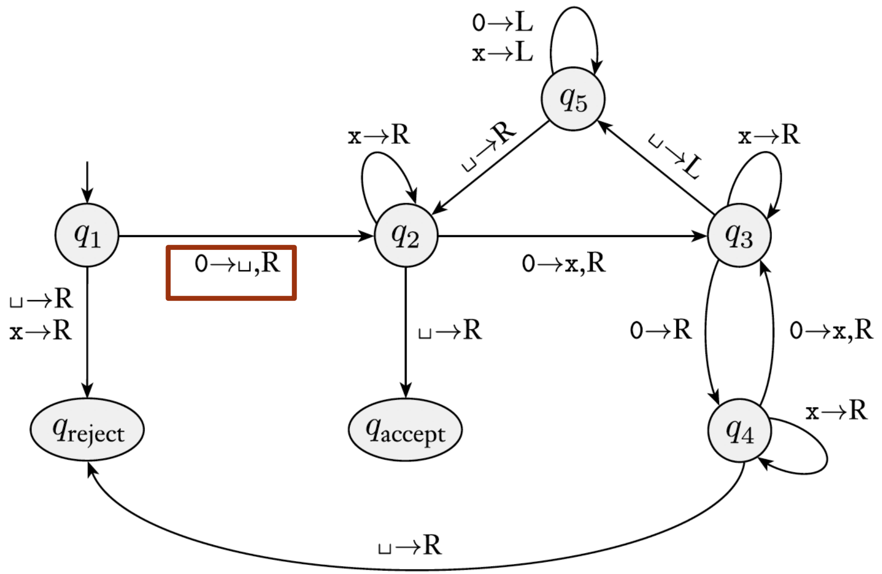
.... more

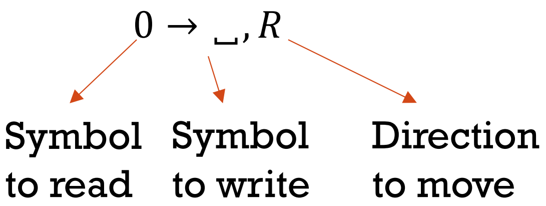
FORMAL DESCRIPTION

- $Q = \{q1, q2, q3, q4, q5, q_{accept}, q_{reject}\}$,
- $\Sigma = \{0\}$, and
- $\Gamma = \{0, x, \sqcup\}$.
- The start, accept, and reject states are $q1$, q_{accept}, and q_{reject}, respectively.

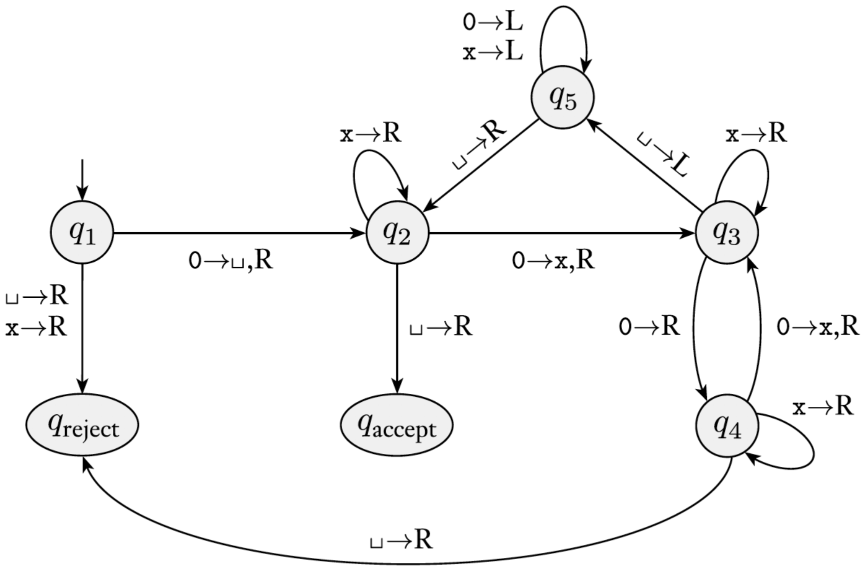


- The label 0→␣,R appears on the transition fromq1 to q2.
  - This label signifies that when in state q1 with the head reading 0, the machine goes to state q2, writes ␣, and moves the head to the right.
  - In other words, δ(q1,0) = (q2,␣,R).
- We use the shorthand 0→R in the transition from q3 to q4, to mean that the machine moves to the right when reading 0 in state q3 but doesn't alter the tape,
  - so δ(q3,0) = (q4,0,R).





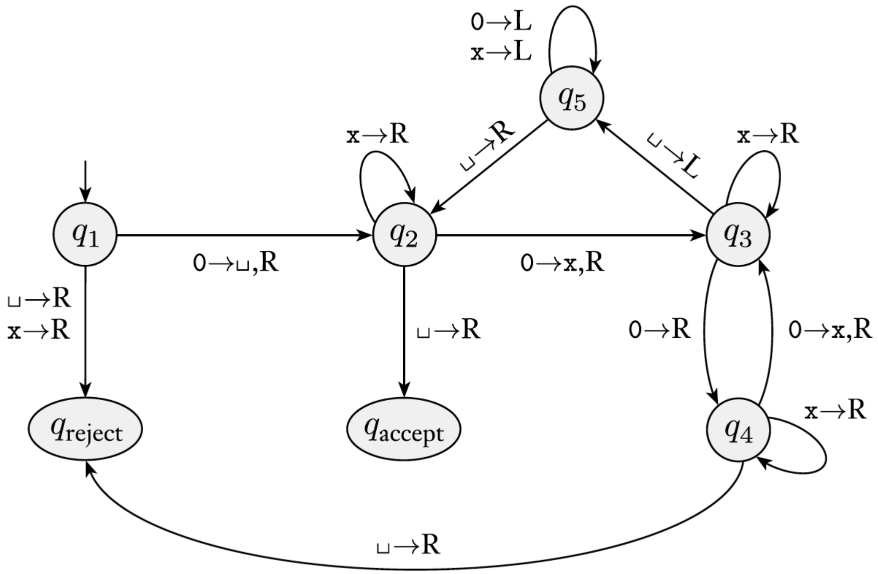
- We give a sample run of this machine on input 0000.
- The starting configuration is q10000.
- On input 0000, get following sequence of configurations:



$$q_1 0000$$
$$\sqcup q_2 000$$
$$\sqcup \mathrm{x} q_3 00$$
$$\sqcup \mathrm{x} 0 q_4 0$$
$$\sqcup \mathrm{x} 0 \mathrm{x} q_3 \sqcup$$
$$\sqcup \mathrm{x} 0 q_5 \mathrm{x} \sqcup$$
$$\sqcup \mathrm{x} q_5 0 \mathrm{x} \sqcup$$

$$\sqcup q_5 \mathrm{x} 0 \mathrm{x} \sqcup$$
$$q_5 \sqcup \mathrm{x} 0 \mathrm{x} \sqcup$$
$$\sqcup q_2 \mathrm{x} 0 \mathrm{x} \sqcup$$
$$\sqcup \mathrm{x} q_2 0 \mathrm{x} \sqcup$$
$$\sqcup \mathrm{x} \mathrm{x} q_3 \mathrm{x} \sqcup$$
$$\sqcup \mathrm{x} \mathrm{x} \mathrm{x} q_3 \sqcup$$
$$\sqcup \mathrm{x} \mathrm{x} q_5 \mathrm{x} \sqcup$$

$$\sqcup \mathrm{x} q_5 \mathrm{x} \mathrm{x} \sqcup$$
$$\sqcup q_5 \mathrm{x} \mathrm{x} \mathrm{x} \sqcup$$
$$q_5 \sqcup \mathrm{x} \mathrm{x} \mathrm{x} \sqcup$$
$$\sqcup q_2 \mathrm{x} \mathrm{x} \mathrm{x} \sqcup$$
$$\sqcup \mathrm{x} q_2 \mathrm{x} \mathrm{x} \sqcup$$
$$\sqcup \mathrm{x} \mathrm{x} q_2 \mathrm{x} \sqcup$$
$$\sqcup \mathrm{x} \mathrm{x} \mathrm{x} q_2 \sqcup$$
$$\sqcup \mathrm{x} \mathrm{x} \mathrm{x} \sqcup q_{\mathrm{accept}}$$



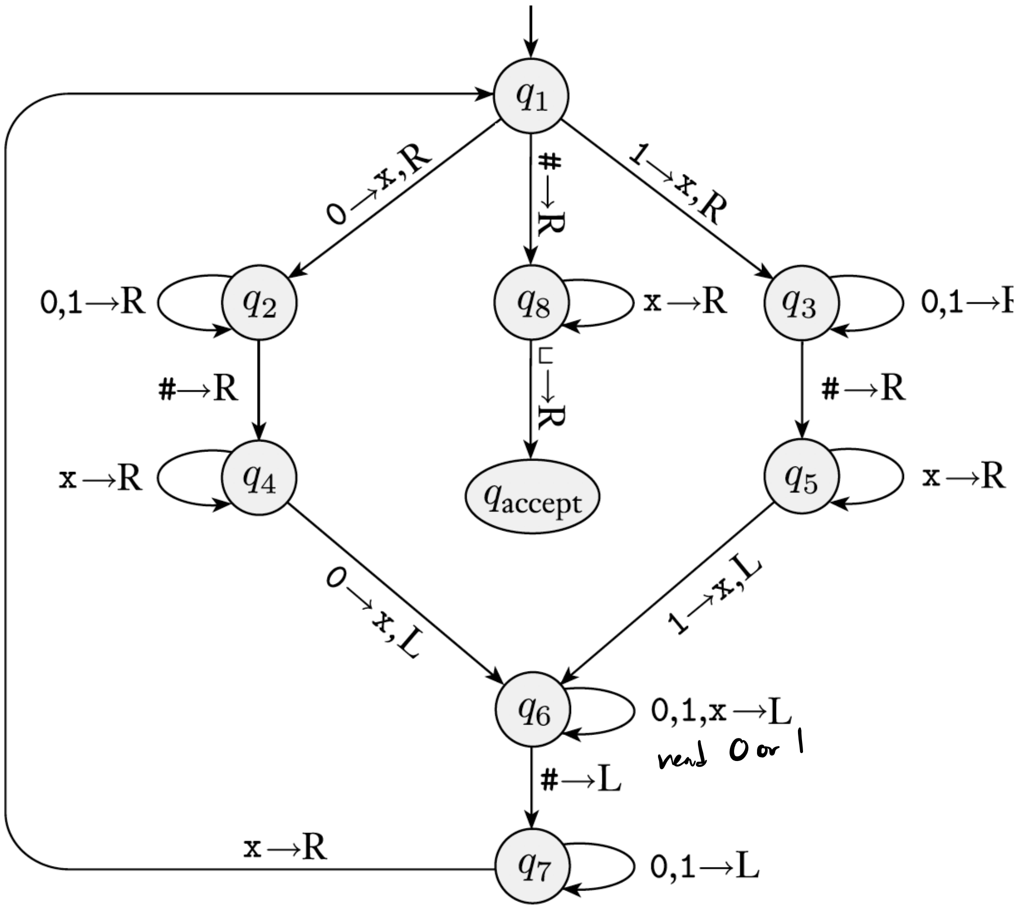
TM for input ‘0000’

Step	State	Tape	Step	State	Tape
0	q_1	0 0 0 0 $\sqcup$ …	4	q_3	$\sqcup$ x 0 x $\sqcup$ …
1	q_2	$\sqcup$ 0 0 0 $\sqcup$ …	5	q_5	$\sqcup$ x 0 x $\sqcup$ …
2	q_3	$\sqcup$ x 0 0 $\sqcup$ …	6	q_5	$\sqcup$ x 0 x $\sqcup$ …
3	q_4	$\sqcup$ x 0 0 $\sqcup$ …	⋮	⋮	⋮

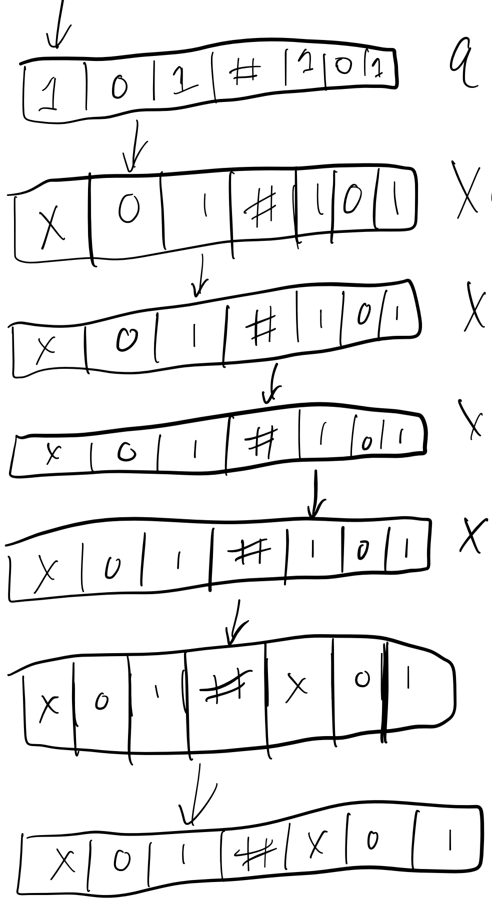
Step	State	Tape
0	q_1	0 0 0 0 ⊔ . . . .
1	q_2	⊔ 0 0 0 ⊔ . . . .
2	q_3	⊔ x 0 0 ⊔ . . . .
3	q_4	⊔ x 0 0 ⊔ . . .
4	q_3	⊔ x 0 x ⊔ . . . .
5	q_5	⊔ x 0 x ⊔ . . . .
6	q_5	⊔ x 0 x ⊔ . . . .

Turing machine for deciding the language $B = \{w\#w \mid w \in \{0,1\}^*\}$.

- $Q = \{q_1, \ldots, q_8, q_{accept}, q_{reject}\}$,
- $\Sigma = \{0,1,\#\}$, and
- $\Gamma = \{0,1,\#,x,\sqcup\}$
- The start, accept, and reject states are q_1, q_{accept}, and q_{reject}, respectively.



101#101



$q_1 101\#101$

$xq_3 01\#101$

$x0q_3 1\#101$

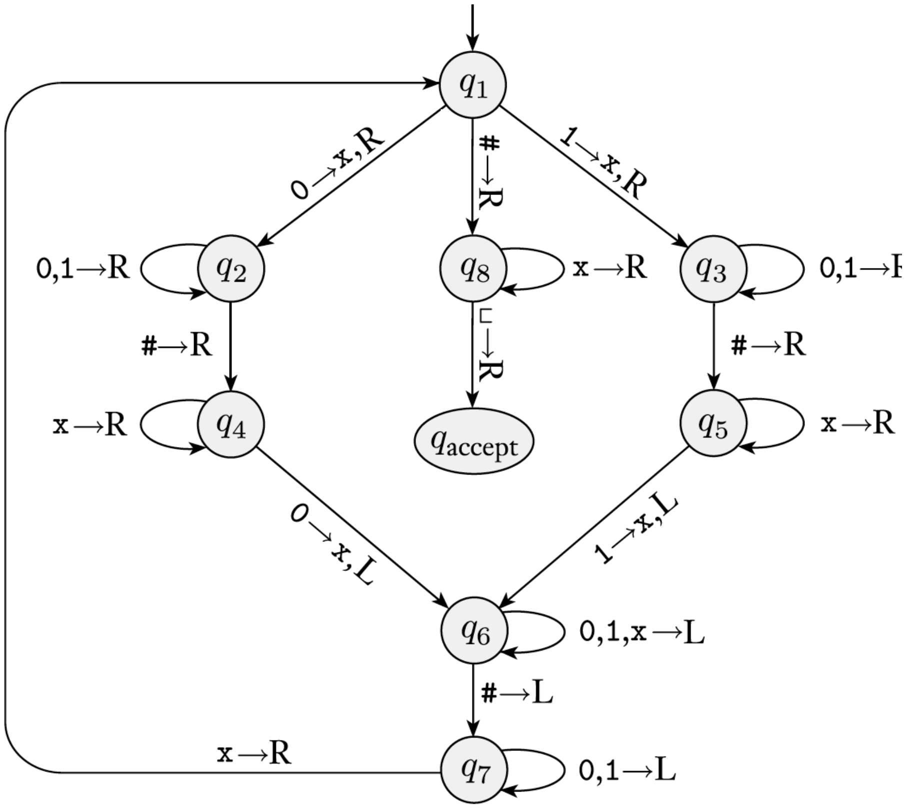
$x01q_3\#101$

$x01\#q_5 101$

$x01q_6\#x01$

$x0q_7 1\#x01$

- The label 0,1→R on the transition going from q_3 to itself.
  - That label means that the machine stays in q3 and moves to the right when it reads a 0 or a 1 in state q_3.
  - It doesn't change the symbol on the tape.
- Stage 1 is implemented by states q_1 through q_7, and stage 2 by the remaining states.
- In state q_5 no outgoing arrow with a # is present, if a # occurs under the head when the machine is in state q_5, it goes to state q_{reject}.



Design a TM to accept the language L = $\{a^n b^n, n \geq 1\}$. Show an ID for the string 'aaabbb' with tape symbols.

When the leftmost 'a' is traversed, that 'a' is replaced by X and the head moves to one right.

$$\delta(q0, a) \rightarrow (q1, X, R)$$

Then, the machine needs to search for the leftmost 'b'. Before that 'b', there exist (n – 1) numbers of 'a'. Those 'a' are traversed by

$$\delta(q1, a) \rightarrow (q1, a, R)$$

When the leftmost 'b' is traversed, the state is q1. That 'b' is replaced by Y, the state is changed to q2, and the head is moved to one left. The transitional functional is

$$\delta(q1, b) \rightarrow (q2, Y, L)$$

Then, it needs to search for the second 'a' starting from the left. The first 'a' is replaced by X, which means the second 'a' exists after X. So, it needs to search for the rightmost 'X'. After traversing the leftmost 'b', the head moves to the left to find the rightmost X. Before that, it has to traverse 'a'. The transitional function is

$$\delta(q2, a) \rightarrow (q2, a, L)$$

After traversing all the ‘a’ we get the rightmost ‘X’. Traversing the X the machine changes its state from q2 to q0 and the head moves to one right. The transitional function is

$$\delta(q2, X) \rightarrow (q0, X, R)$$

Similarly, after traversing ‘b’, the machine has to traverse some Y to get the rightmost ‘X’.

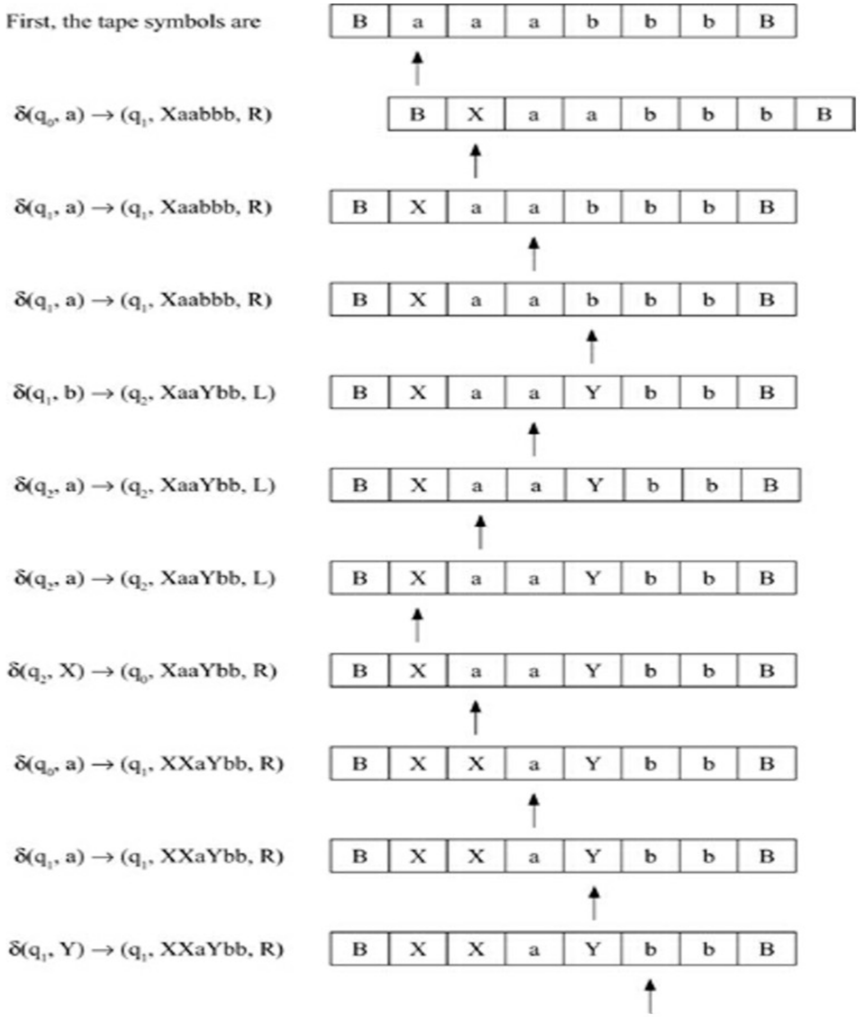
$$\delta(q2, Y) \rightarrow (q2, Y, L)$$

When all the ‘a’s are traversed, the state is q0, because before that state was q2 and the input was X. The head moves to one right and gets a Y. Getting a Y means that all the ‘a’s are traversed and the same number of ‘b’s are traversed. Traversing right, if at last the machine gets no ‘b’ but a blank ‘B’, then the machine halts. The transitional functions are

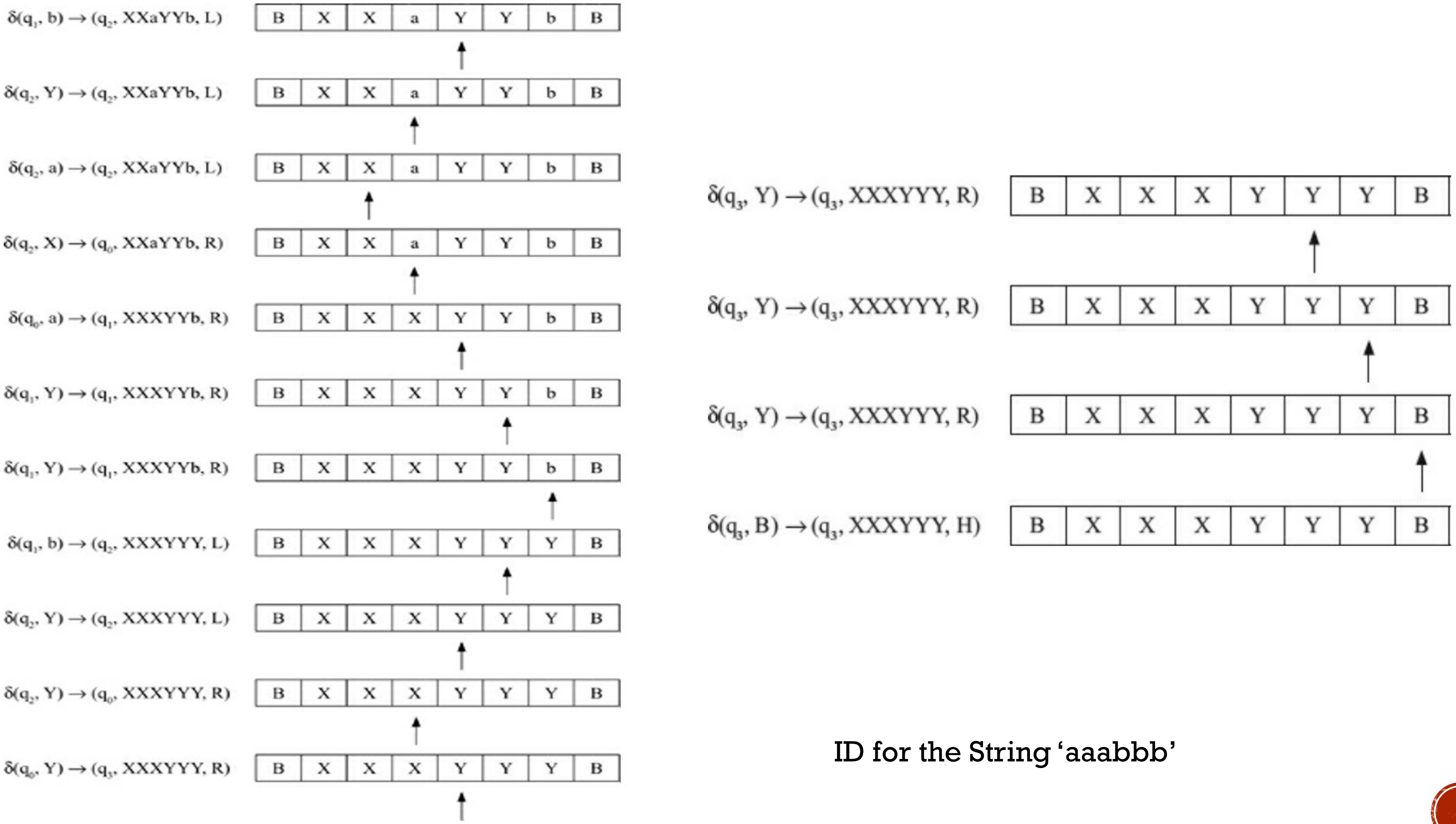
$$\delta(q0, Y) \rightarrow (q3, Y, R)$$

$$\delta(q3, Y) \rightarrow (q3, Y, R)$$

$$\delta(q3, B) \rightarrow (q4, B, H)$$



ID for the String ‘aaabbb’



ID for the String ‘aaabbb’

Turing machine M_3 to decide language $C = \{ a^i b^j c^k \mid i \times j = k \text{ and } i, j, k \geq 1 \}$.

Idea: If i collections of j things each, then $i \times j$ things total.

TM: for each ‘a’, cross off j c’s by matching each b with a ‘c’.

M_3 = “On input string w:

- Scan the input from left to right to make sure that it is a member of $L(a{*}b{*}c{*})$ and reject if it isn’t.
- Return the head to the left-hand end of the tape
- Cross off an a and scan to the right until a b occurs. Shuttle between the b’s and the c’s, crossing off each until all b’s are gone. If all c’s have been crossed off and some b’s remain, reject.
- Restore the crossed off b’s and repeat stage 3 if there is another a to cross off. If all a’s are crossed off, check whether all c’s also are crossed off. If yes, accept; otherwise, reject.”

Clarity above all: high-level description of TMs is allowed

M = “On input string w:

1. Scan input ...”

but it should not be used as a trick to hide the important details of the program.

- Standard tools:
  - Expanding tape alphabet Γ with
    - separator “#”
    - dotted symbols $\dot{0}$, $\dot{a}$, to indicate “activity,”
    - Typical example: $\Gamma = \{0, 1, \#, \sqcup, \dot{0}, \dot{1}\}$

Running TM M_3 on Input $a^3b^2c^6 \in C$

- Tape head starts over leftmost symbol

↓													
a	a	a	b	b	c	c	c	c	c	c	$\sqcup$	$\sqcup$	$\cdots$

- Stage 1: Mark leftmost symbol and scan to see if input $\in L(a^*b^*c^*)$

↓													
A	a	a	b	b	c	c	c	c	c	c	$\sqcup$	$\sqcup$	$\cdots$

- Stage 3: Cross off one a and cross off matching b's and c's

						↓							
•			•	•	•	•							
A	a	a	b	b	c	c	c	c	c	c	$\sqcup$	$\sqcup$	$\cdots$

- Stage 4: Restore b's and return head to first a not crossed off

	↓												
•					•	•							
A	a	a	b	b	c	c	c	c	c	c	$\sqcup$	$\sqcup$	$\cdots$

- Stage 3: Cross off one a and cross off matching b's and c's

●	●		●	●	●	●	●	●					
A	a	a	b	b	c	c	c	c	c	c	$\sqcup$	$\sqcup$	$\ldots$

- Stage 4: Restore b's and return head to first a not crossed off

●	●				●	●	●	●					
A	a	a	b	b	c	c	c	c	c	c	$\sqcup$	$\sqcup$	$\ldots$

- Stage 3: Cross off one a and cross off matching b's and c's

●	●	●	●	●	●	●	●	●	●	●			
A	a	a	b	b	c	c	c	c	c	c	$\sqcup$	$\sqcup$	$\ldots$

- Stage 4: If all a's crossed off, check if all c's crossed off.
- *accept*

HOW TO TELL WHEN A TM IS AT THE LEFT END OF THE TAPE?

- One Approach: Mark it with a special symbol.
- Alternative method:
  - remember current symbol
  - overwrite it with special symbol
  - move left
  - if special symbol still there, head is at start of tape
  - otherwise, restore previous symbol and move left.

Design a TM to accept the language $L = \{a^n b^n, n \geq 1\}$. Show an ID for the string 'aaabbb' with tape symbols.

When the leftmost 'a' is traversed, that 'a' is replaced by X and the head moves to one right.

$$\delta(q0, a) \rightarrow (q1, X, R)$$

Then, the machine needs to search for the leftmost 'b'. Before that 'b', there exist (n – 1) numbers of 'a'. Those 'a' are traversed by

$$\delta(q1, a) \rightarrow (q1, a, R)$$

When the leftmost 'b' is traversed, the state is q1. That 'b' is replaced by Y, the state is changed to q2, and the head is moved to one left. The transitional functional is

$$\delta(q1, b) \rightarrow (q2, Y, L)$$

Then, it needs to search for the second 'a' starting from the left. The first 'a' is replaced by X, which means the second 'a' exists after X. So, it needs to search for the rightmost 'X'. After traversing the leftmost 'b', the head moves to the left to find the rightmost X. Before that, it has to traverse 'a'. The transitional function is

$$\delta(q2, a) \rightarrow (q2, a, L)$$

After traversing all the ‘a’ we get the rightmost ‘X’. Traversing the X the machine changes its state from q2 to q0 and the head moves to one right. The transitional function is

$$\delta(q2, X) \rightarrow (q0, X, R)$$

Similarly, after traversing ‘b’, the machine has to traverse some Y to get the rightmost ‘X’.

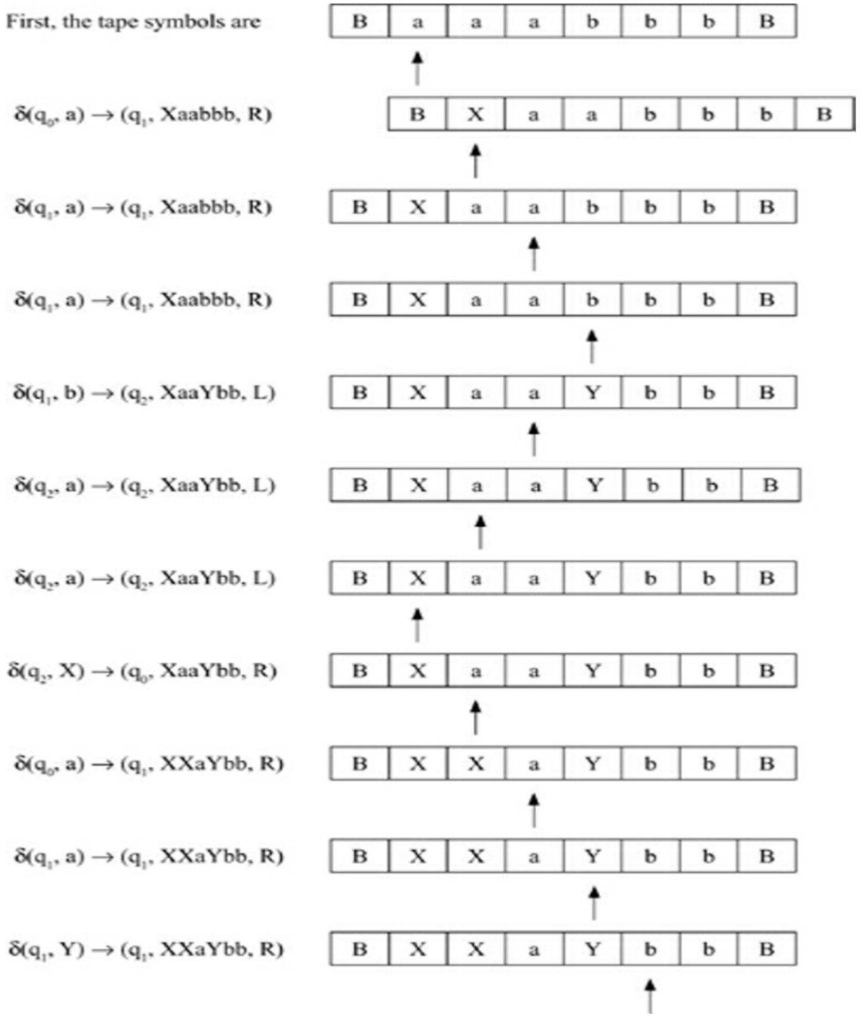
$$\delta(q2, Y) \rightarrow (q2, Y, L)$$

When all the ‘a’s are traversed, the state is q0, because before that state was q2 and the input was X. The head moves to one right and gets a Y. Getting a Y means that all the ‘a’s are traversed and the same number of ‘b’s are traversed. Traversing right, if at last the machine gets no ‘b’ but a blank ‘B’, then the machine halts. The transitional functions are

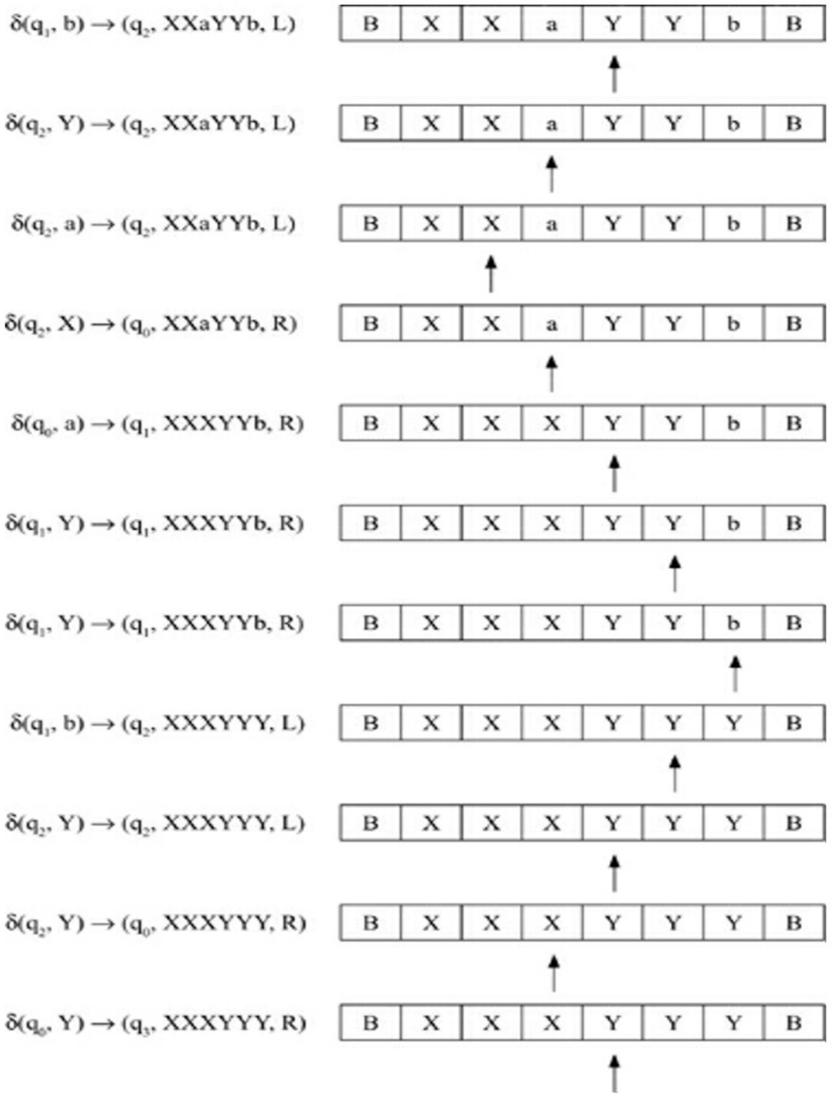
$$\delta(q0, Y) \rightarrow (q3, Y, R)$$

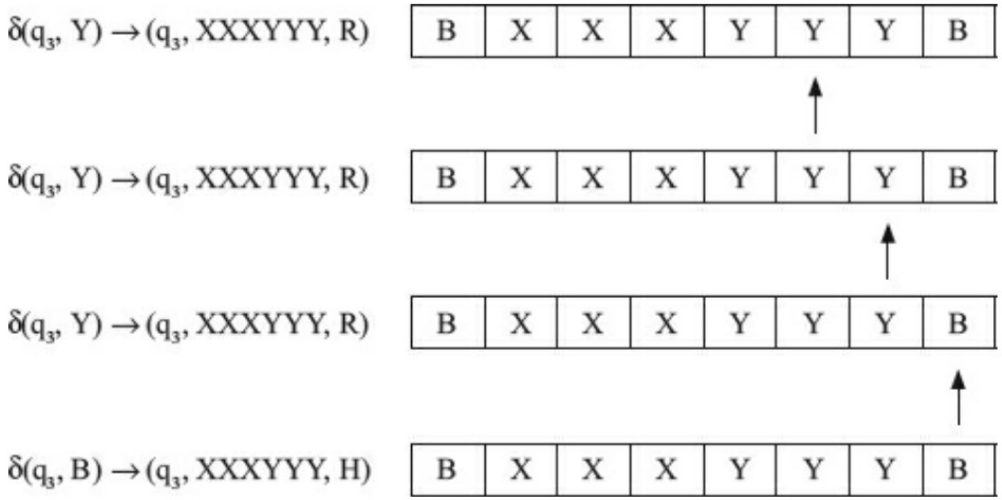
$$\delta(q3, Y) \rightarrow (q3, Y, R)$$

$$\delta(q3, B) \rightarrow (q4, B, H)$$



ID for the String ‘aaabbb’





ID for the String ‘aaabbb’

TRY YOURSELF!

- Design a TM that recognizes the language L = $\{0^n1^n \mid n>0\}$