Lecture 9
Turing Machines

COSE215: Theory of Computation

Seunghoon Woo

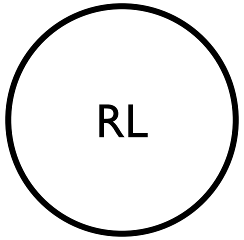
Fall 2023

Contents

- **Turing machines**

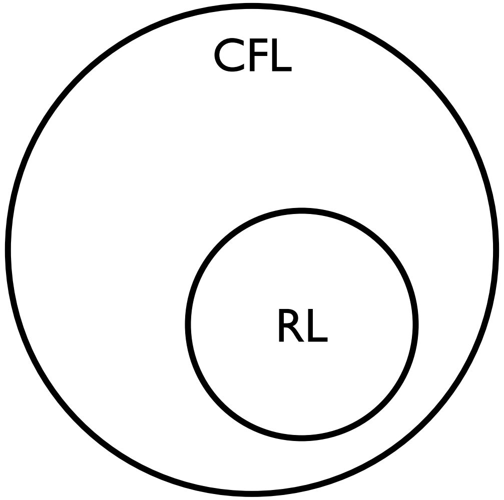
Limitation of Pushdown automata

- **Finite automata**
  - Without any additional storage
  - FA can accept regular languages



Limitation of Pushdown automata

- **Finite automata**
  - Without any additional storage
  - FA can accept regular languages
- **Pushdown automata**
  - FA + stack
  - (n)PDA can accept context-free languages

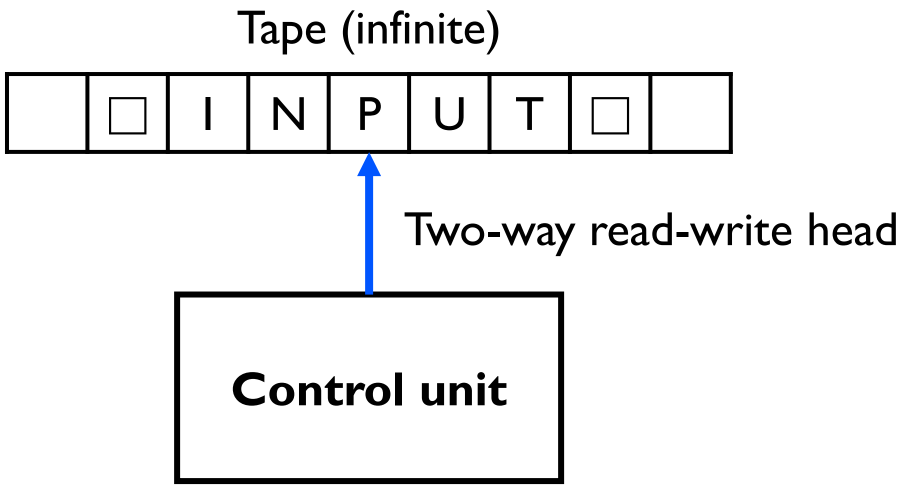


Limitation of Pushdown automata

- **Finite automata**
  - Without any additional storage
  - FA can accept regular languages
- **Pushdown automata**
  - FA + stack
  - (n)PDA can accept context-free languages
- **We can expect to discover even more powerful language families if we give the automaton more flexible storage!**

Turing machine

- **Turing machine (TM)**
  - TM has an extra component called a tape
    - A tape consists of cells
    - A tape, by definition, has infinite length with a two-way read-write head

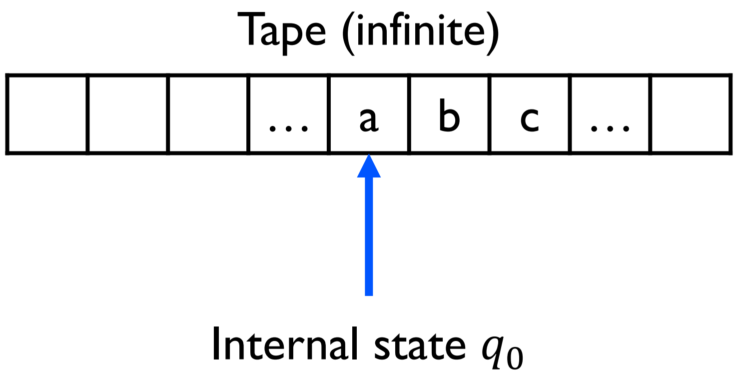


Turing machine

- **Turing machine: Formal definition**
  - A Turing machine (TM) is a 7-tuple: $\boldsymbol{M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)}$
    - ❖ Q is a finite set of **internal states**
    - ❖ Σ is a finite set of **symbols**
      - $\Sigma \subseteq \Gamma - \{\square\}$
    - ❖ Γ is a finite set of symbols called **tape alphabets**
    - ❖ δ is a set of **transition functions**
      - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
    - ❖ $q_0 \in Q$ is **the initial state**
    - ❖ $\square \in \Gamma$ is a special symbol called the **blank**
    - ❖ $F \subseteq Q$ is a set of **final states**

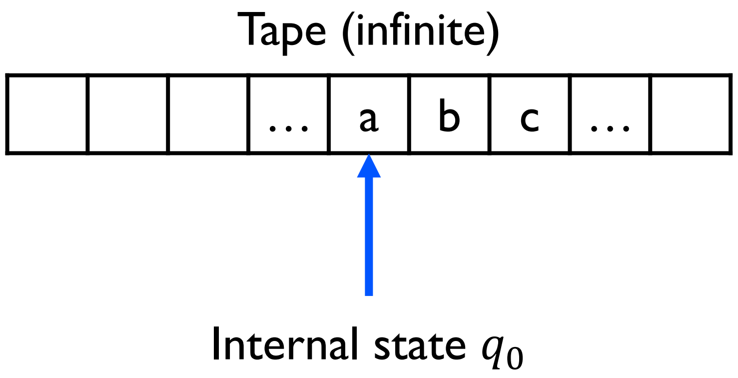
Turing machine

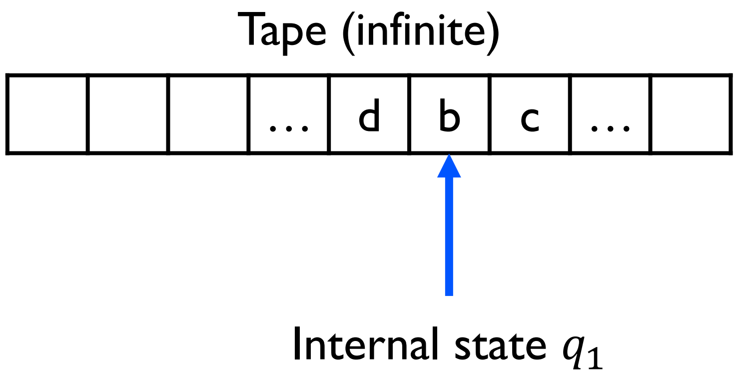
- **Turing machine: Formal definition**
  - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
  - E.g., $\delta(q_0, a) = (q_1, d, R)$



Turing machine

- **Turing machine: Formal definition**
  - $\delta: Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$
  - E.g., $\delta(q_0, a) = (q_1, d, R)$



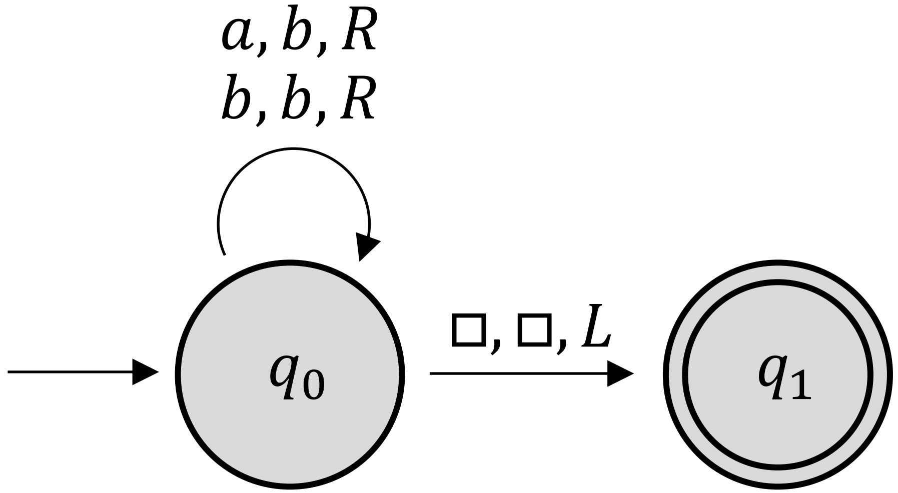


Turing machine

- **Example**
  - $M = (\{q_0, q_1\}, \{a, b\}, \{a, b, \square\}, \delta, q_0, \square, \{q_1\})$
    - $\delta(q_0, a) = (q_0, b, R)$
    - $\delta(q_0, b) = (q_0, b, R)$
    - $\delta(q_0, \square) = (q_1, \square, L)$

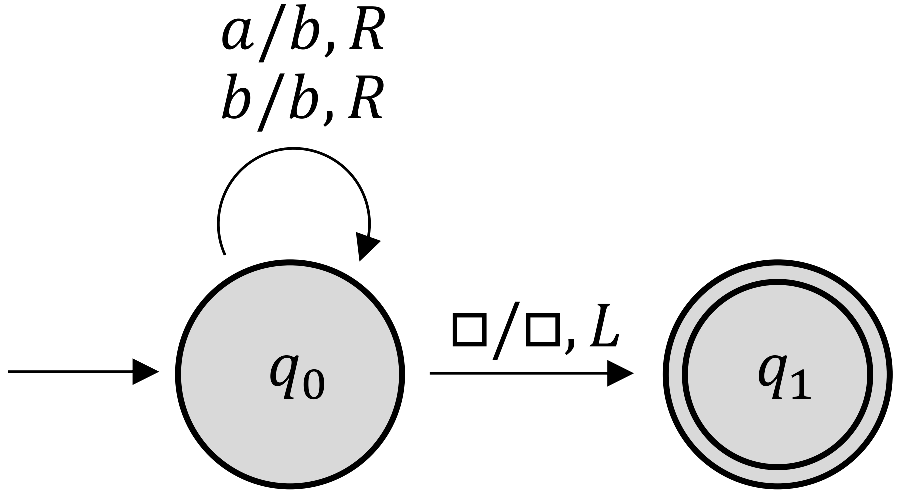
Turing machine

- **Example**
  - $M = (\{q_0, q_1\}, \{a, b\}, \{a, b, \square\}, \delta, q_0, \square, \{q_1\})$
    - ❖ $\delta(q_0, a) = (q_0, b, R)$
    - ❖ $\delta(q_0, b) = (q_0, b, R)$
    - ❖ $\delta(q_0, \square) = (q_1, \square, L)$



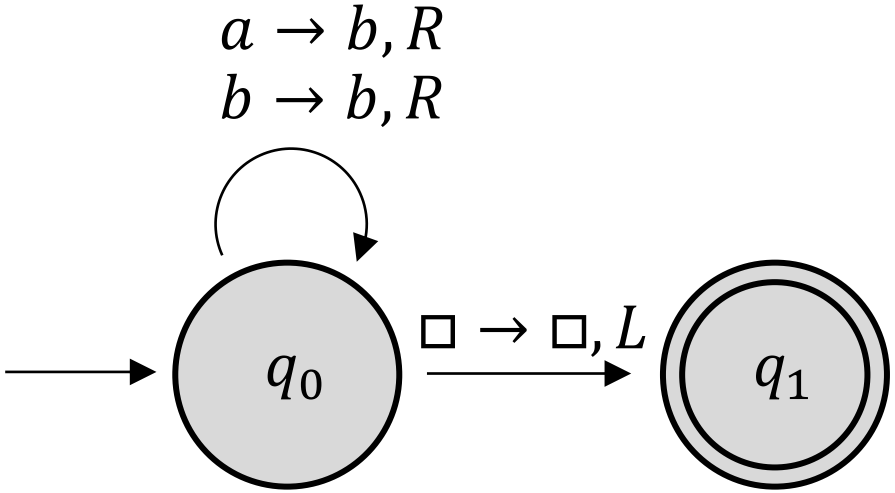
Turing machine

- **Example**
  - $M = (\{q_0, q_1\}, \{a, b\}, \{a, b, \square\}, \delta, q_0, \square, \{q_1\})$
    - $\delta(q_0, a) = (q_0, b, R)$
    - $\delta(q_0, b) = (q_0, b, R)$
    - $\delta(q_0, \square) = (q_1, \square, L)$



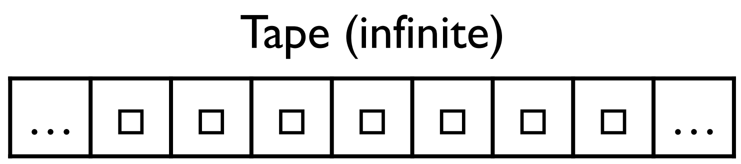
Turing machine

- **Example**
  - $M = (\{q_0, q_1\}, \{a, b\}, \{a, b, \square\}, \delta, q_0, \square, \{q_1\})$
    - $\delta(q_0, a) = (q_0, b, R)$
    - $\delta(q_0, b) = (q_0, b, R)$
    - $\delta(q_0, \square) = (q_1, \square, L)$



Turing machine

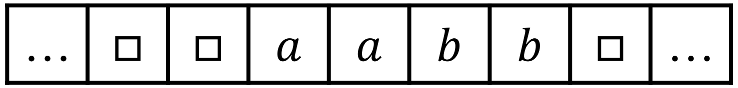
- **Turing machine: process**
  - Initially, all tape symbols are blank



Turing machine

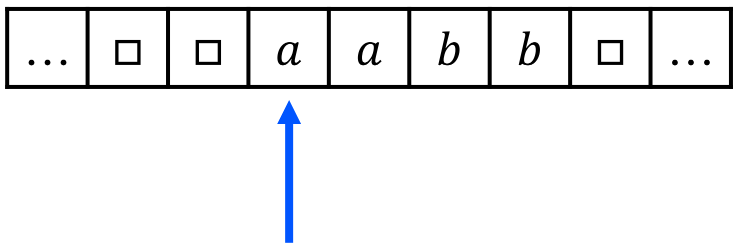
- **Turing machine: process**
  - Initially, all tape symbols are blank
  - The machine is started with the input string written somewhere on the tape

Tape (infinite)



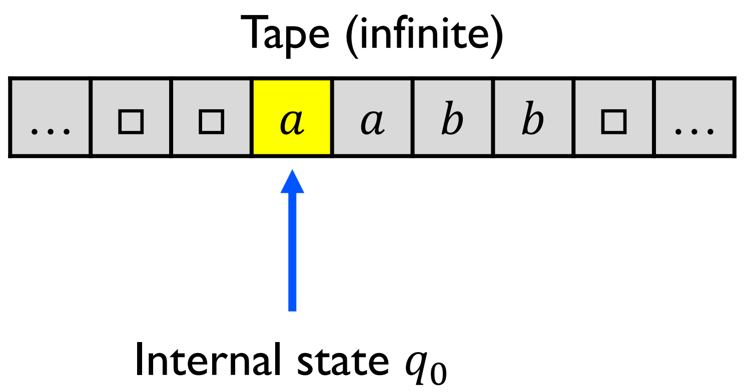
Turing machine

- **Turing machine: process**
  - Initially, all tape symbols are blank
  - The machine is started with the input string written somewhere on the tape
  - The tape head initially points to the first symbol of the input string



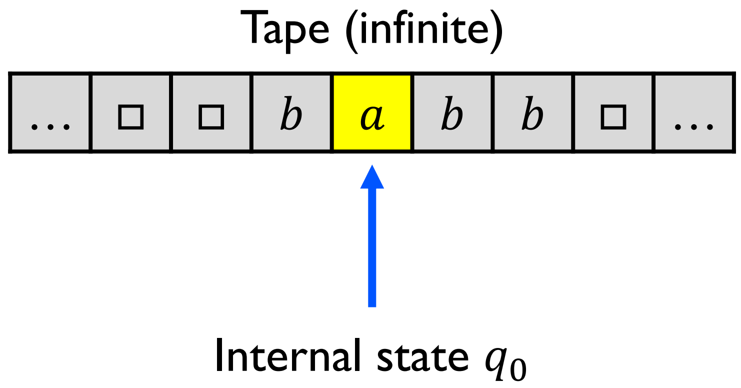
Turing machine

- **Turing machine: process**
  - Initially, all tape symbols are blank
  - The machine is started with the input string written somewhere on the tape
  - The tape head initially points to the first symbol of the input string
  - At each step, the TM only looks at the symbol immediately under the head



Turing machine

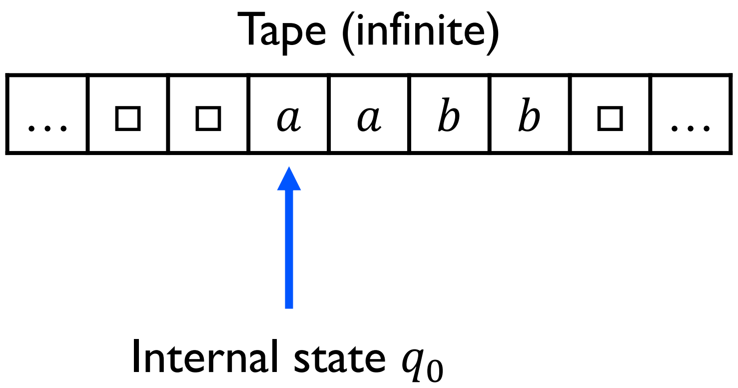
- **Turing machine: process**
  - Initially, all tape symbols are blank
  - The machine is started with the input string written somewhere on the tape
  - The tape head initially points to the first symbol of the input string
  - At each step, the TM only looks at the symbol immediately under the head
  - Read or write the symbol, and move the tape head (L or R)

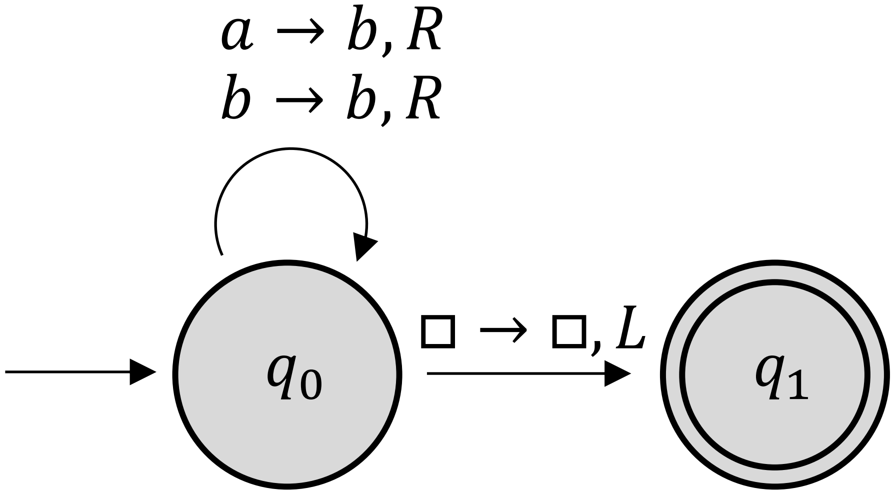


Turing machine

- **Example**
  - E.g., Input string is "$aabb$"

$$aabb$$

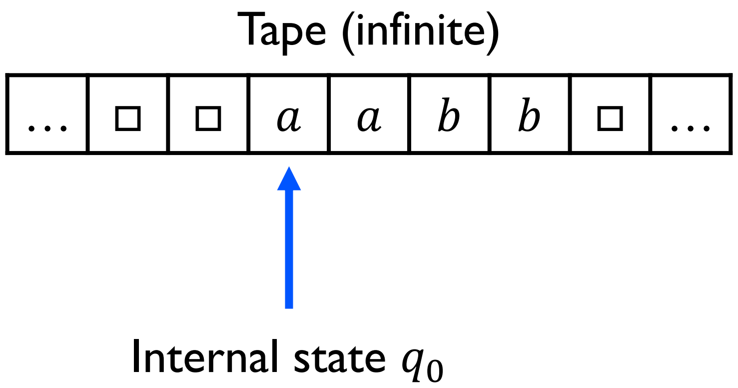


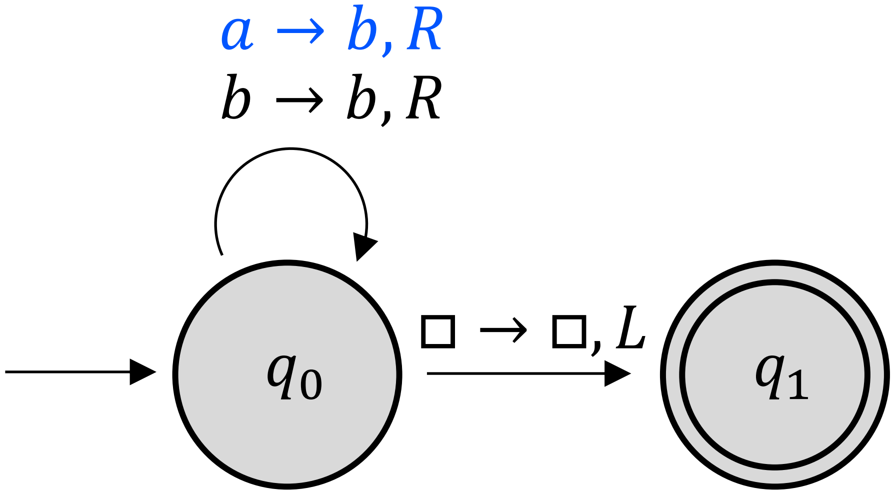


Turing machine

- **Example**
  - E.g., Input string is "$aabb$"

$$aabb$$

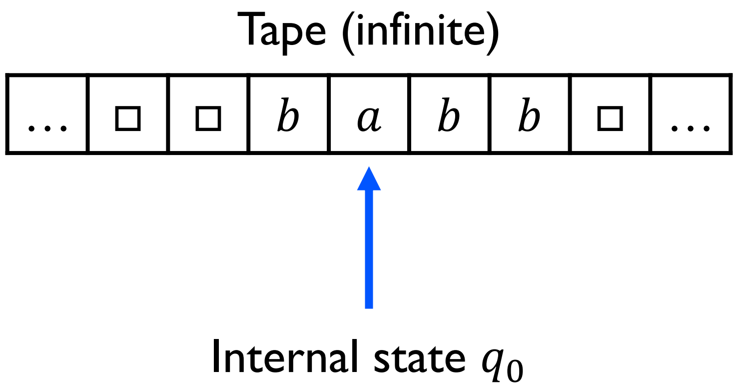


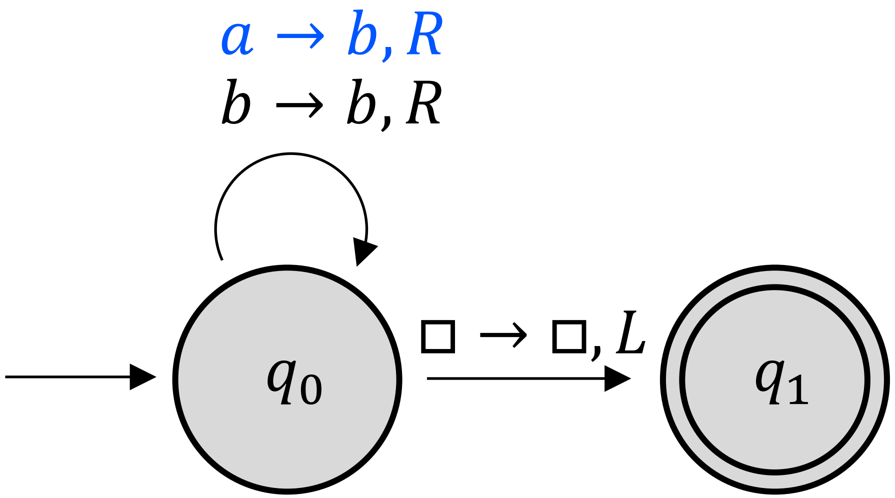


Turing machine

- **Example**
  - E.g., Input string is "$aabb$"

~~a~~abb

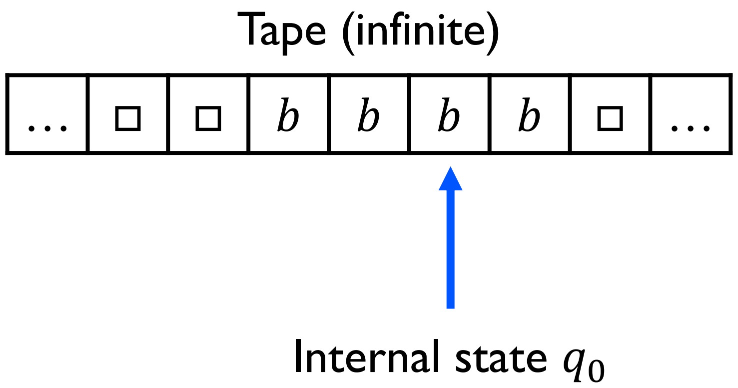


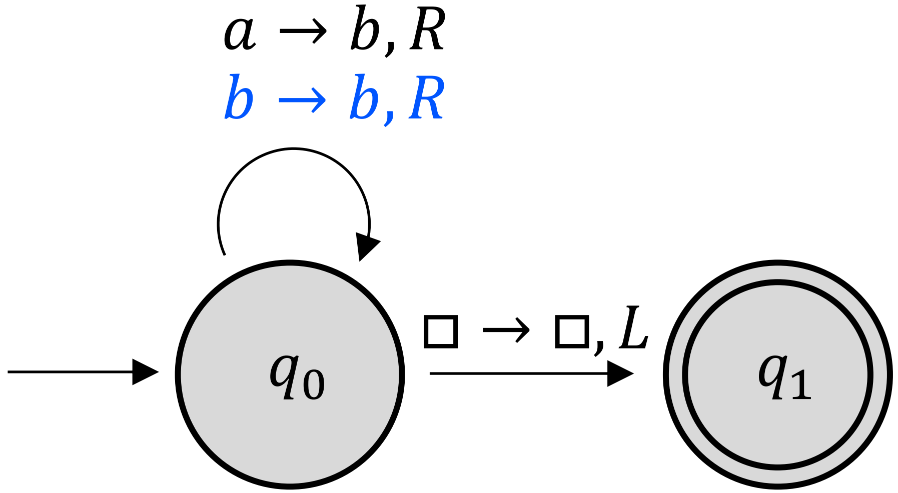


Turing machine

- **Example**
  - E.g., Input string is "$aabb$"

$\sout{aa}bb$

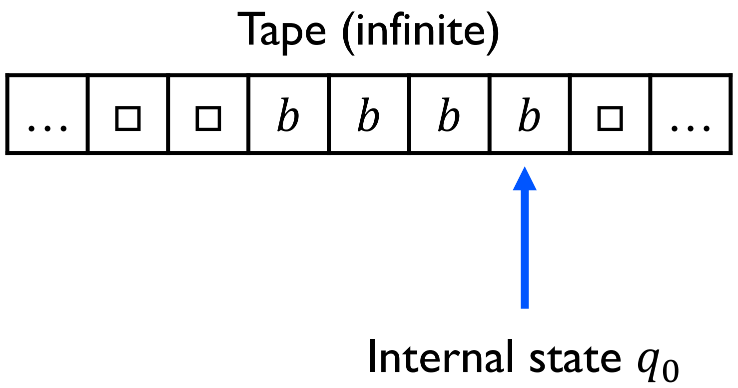


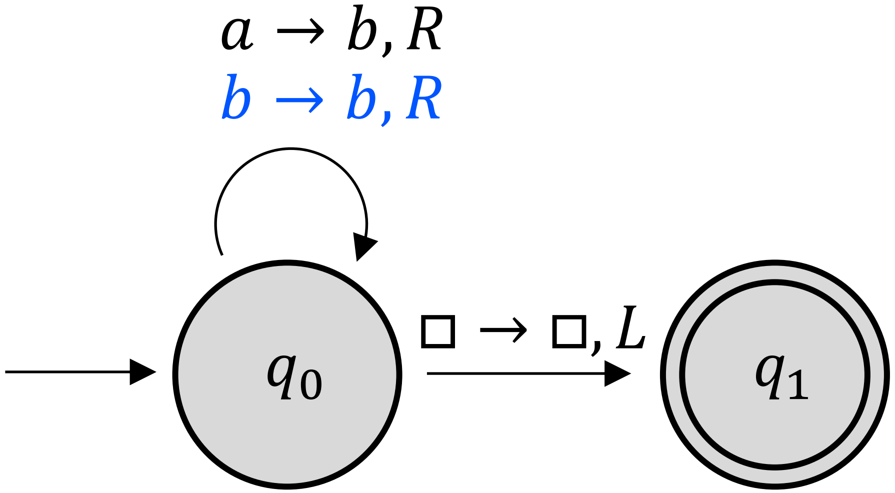


Turing machine

- **Example**
  - E.g., Input string is "$aabb$"

~~aab~~b

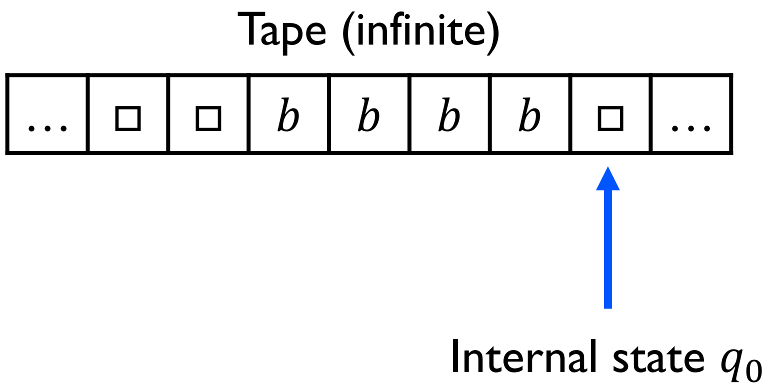


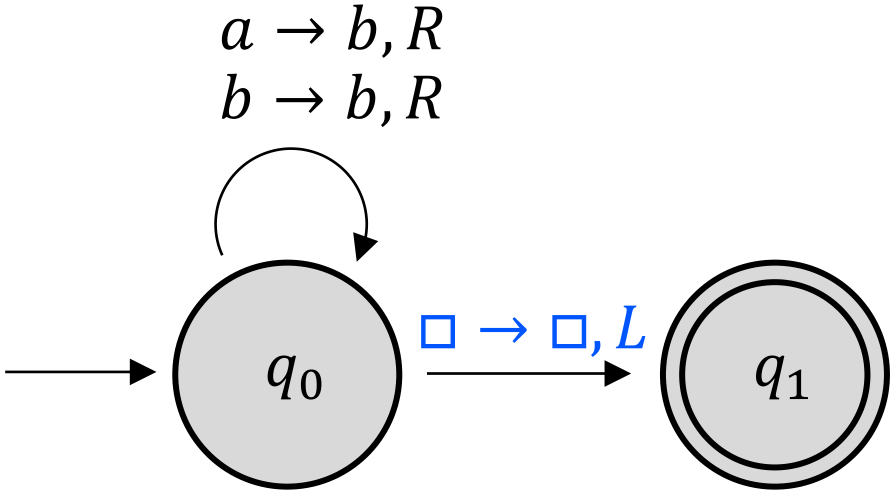


Turing machine

- **Example**
  - E.g., Input string is "$aabb$"

~~$aabb$~~

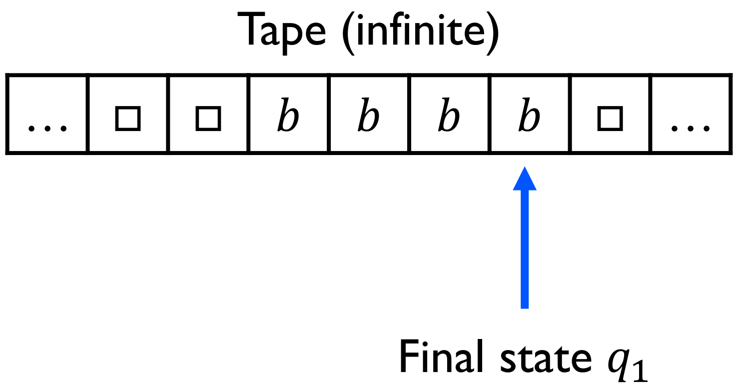


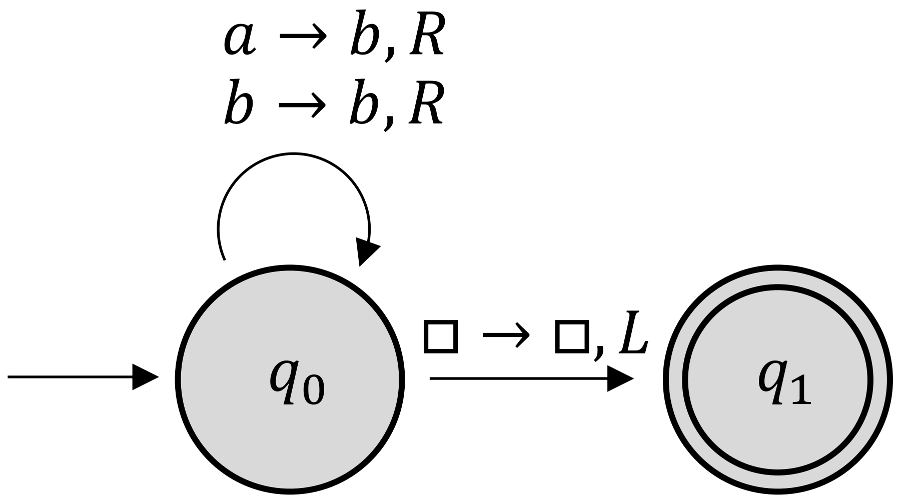


Turing machine

- **Example**
  - E.g., Input string is "$aabb$"

~~$aabb$~~





Turing machine

- **Standard Turing machine**
  - The Turing machine has a tape that is unbounded in both directions, allowing any number of left and right moves
  - The Turing machine is deterministic in the sense that δ defines at most one move for each configuration
  - There is no special input file / output device
    - We use tape!

Turing machine

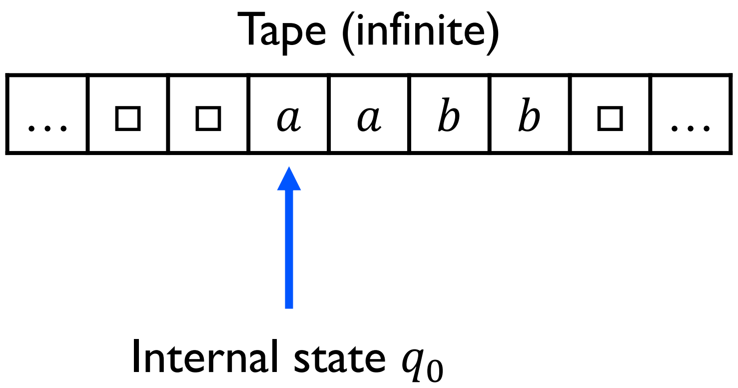
- **Instantaneous Description**
  - Any configuration is determined by
    1. The current state
    2. The contents of the tape
    3. The position of the head

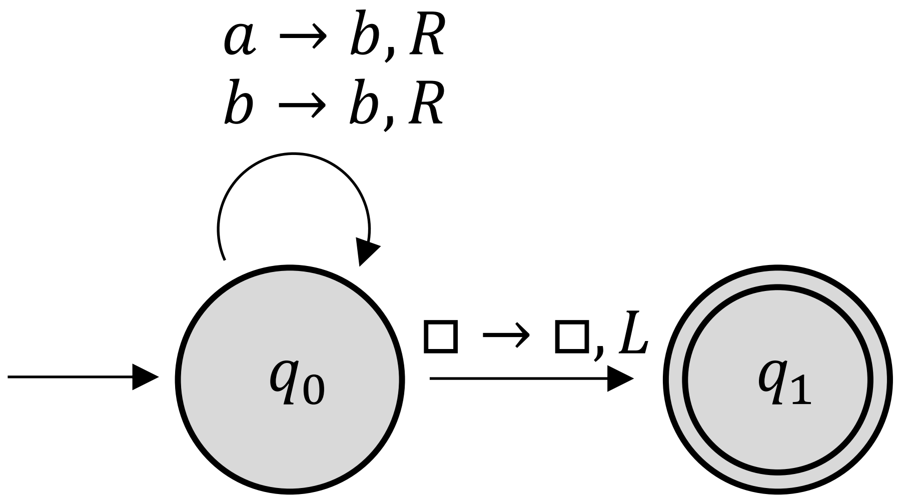
Turing machine

- **Instantaneous Description**
  - Any configuration is determined by
    1. The current state
    2. The contents of the tape
    3. The position of the head
  - Represented as $a_1 a_2 \ldots a_{k-1} q a_k a_{k+1} \ldots a_n$
    - q: current state
    - $a_1 a_2 \ldots a_n$: tape contents
    - The position of head is over the cell containing the symbol immediately following q
      - In this case, a_k

Turing machine

- **Example: ID**
  - E.g., Input string is aabb





Turing machine

- **Example: ID**
  - E.g., Input string is aabb
    - ❖ $q_0aabb \vdash bq_0abb \vdash bbq_0bb \vdash bbbq_0b \vdash bbbbq_0\square \vdash bbbq_1b$
    - ❖ $q_0aabb \vdash^* bbbq_1b$

Turing machine

- **Instantaneous Description**
  - A move

$$a_1 a_2 \dots a_{k-1} q_1 a_k a_{k+1} \dots a_n \vdash a_1 a_2 \dots a_{k-1} b q_2 a_{k+1} \dots a_n$$

is possible if and only if $\delta(q_1, a_k) = (q_2, b, R)$ exist

Turing machine

- **Instantaneous Description**
  - A move

$$a_1 a_2 \dots a_{k-1} q_1 a_k a_{k+1} \dots a_n \vdash a_1 a_2 \dots a_{k-1} b q_2 a_{k+1} \dots a_n$$

    is possible if and only if $\delta(q_1, a_k) = (q_2, b, R)$ exist

  - A TM is said to halt starting from some initial configuration $x_1 q_i x_2$ if

$$x_1 q_i x_2 \vdash^* y_1 q_j a y_2$$

    for any q_j and a, for which $\delta(q_j, a)$ is undefined

Turing machine

- **Instantaneous Description**
  - A move

$$a_1a_2 \ldots a_{k-1}q_1a_ka_{k+1} \ldots a_n \vdash a_1a_2 \ldots a_{k-1}bq_2a_{k+1} \ldots a_n$$

    is possible if and only if $\delta(q_1, a_k) = (q_2, b, R)$ exist
  - A TM is said to halt starting from some initial configuration $x_1q_ix_2$ if

$$x_1q_ix_2 \vdash^* y_1q_jay_2$$

    for any q_j and a, for which $\delta(q_j, a)$ is undefined
  - The sequence of configurations leading to a halt state is called a computation

Turing machine

- **Turing machines as language accepters**
  - Start with initial state with the head positioned on the leftmost symbol of w
  - After a sequence of moves, if the Turing machine enters a final state and halts, then w is considered to be accepted

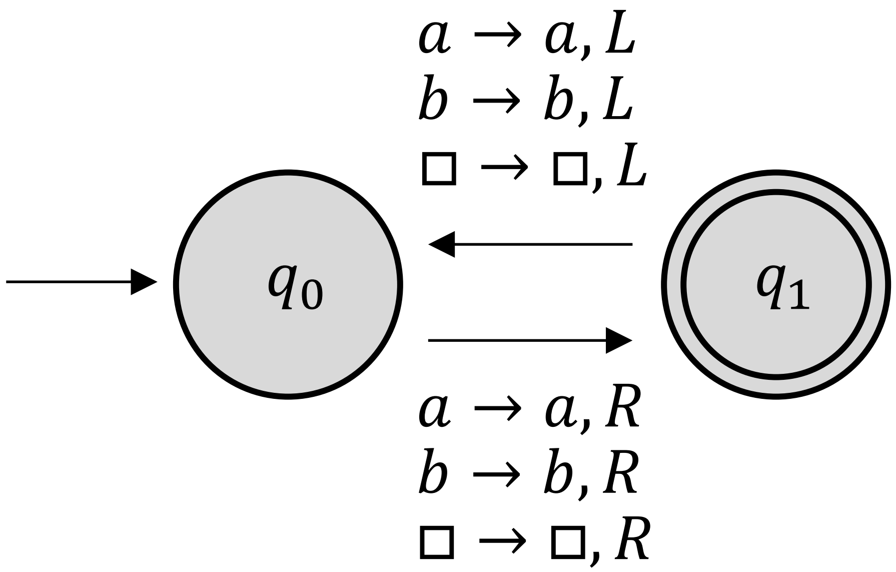
Turing machine

- **Turing machines as language accepters**
  - Start with initial state with the head positioned on the leftmost symbol of w
  - After a sequence of moves, if the Turing machine enters a final state and halts, then w is considered to be accepted
  - Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ be a Turing machine
    - ❖ The language accepted by M is the set

$$L(M) = \{w \in \Sigma^* : q_0 w \vdash^* x_1 q_f x_2,\ q_f \in F, x_1, x_2 \in \Gamma^*\}$$

Turing machine

- **Turing machines as language accepters**
  - Non-accepting input string
    1. The machine halts in a nonfinal state
    2. The machine enters an infinite loop and never halt



Turing machine

- **Turing machines as language accepters**
  - A language L is **recursively enumerable** if there exists a Turing machine M such that $L = L(M)$

Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$

Turing machine

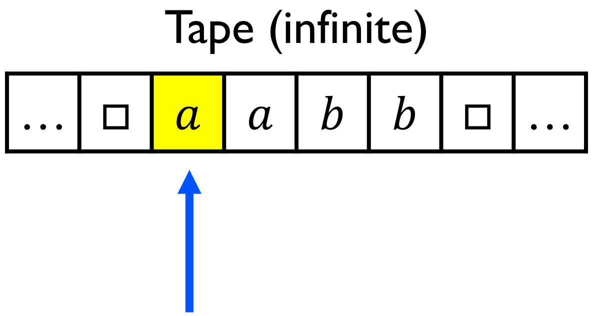
- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - Basic idea
    - ❖ Find leftmost a and replace it with a tape symbol (let A)
    - ❖ Move head to the right to find the leftmost b
    - ❖ Find leftmost b and replace it with a tape symbol (let B)
    - ❖ Move head to the left to find the leftmost a
    - ❖ If after some time no a's or b's remain, then the input string should be in L

Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - Basic idea
    - **While** there are a's **do**
    - **Find** and **Replace** a with **A**
    - **Find** and **Replace** b with **B**
    - **end**
    - If no a's or b's remain => accept

Turing machine

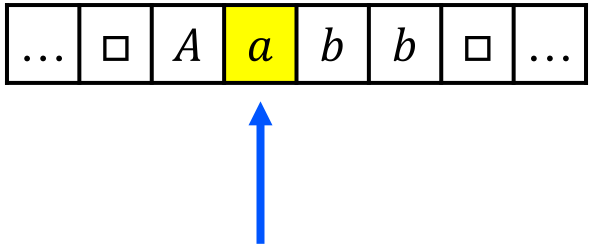
- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - E.g., Input: "$aabb$"



Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - E.g., Input: "$aabb$"

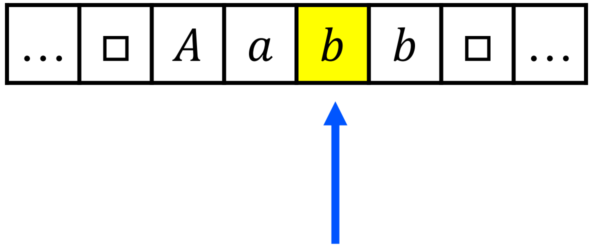
Tape (infinite)



Turing machine

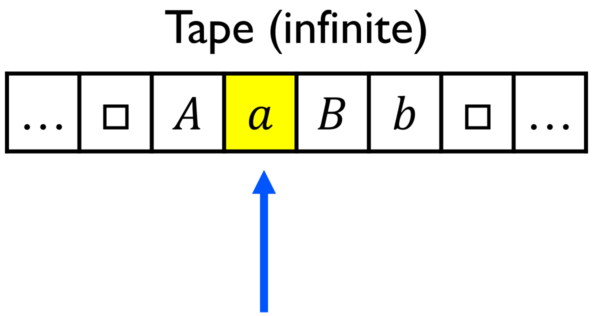
- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - E.g., Input: "$aabb$"

Tape (infinite)



Turing machine

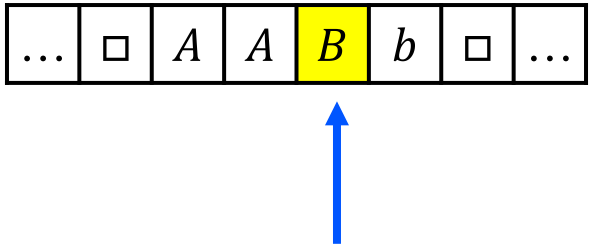
- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - E.g., Input: "$aabb$"



Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - E.g., Input: "$aabb$"

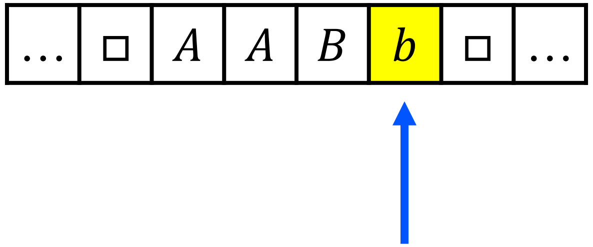
Tape (infinite)



Turing machine

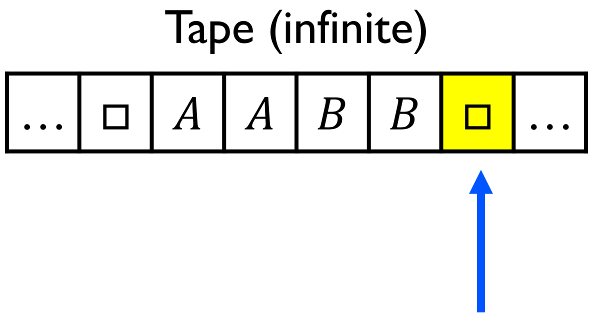
- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - E.g., Input: "$aabb$"

Tape (infinite)



Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - E.g., Input: "$aabb$"



Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - E.g., Input: "$aabb$"

Tape (infinite)

$\ldots$	$\square$	A	A	B	B	$\square$	$\ldots$

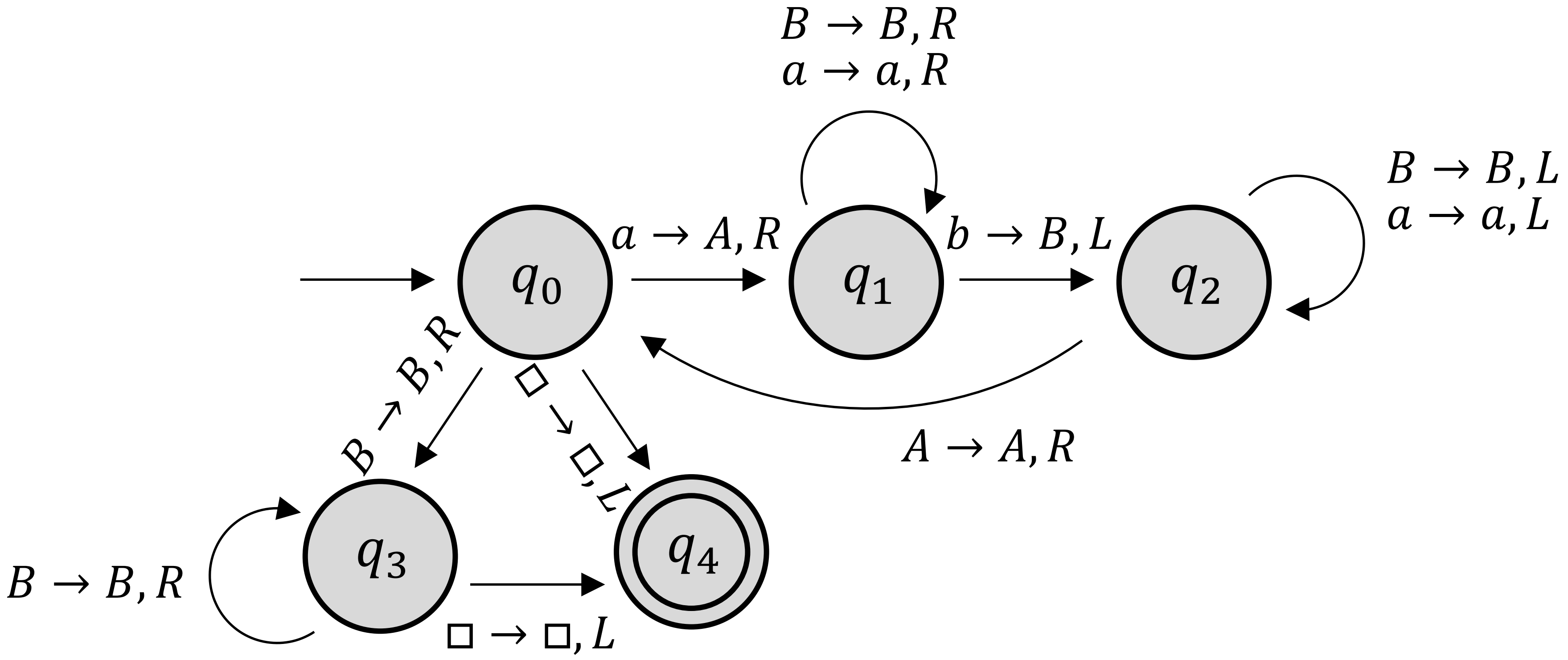
No input symbols => accept

Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \{a, b, A, B, \square\}, \delta, q_0, \square, \{q_4\})$

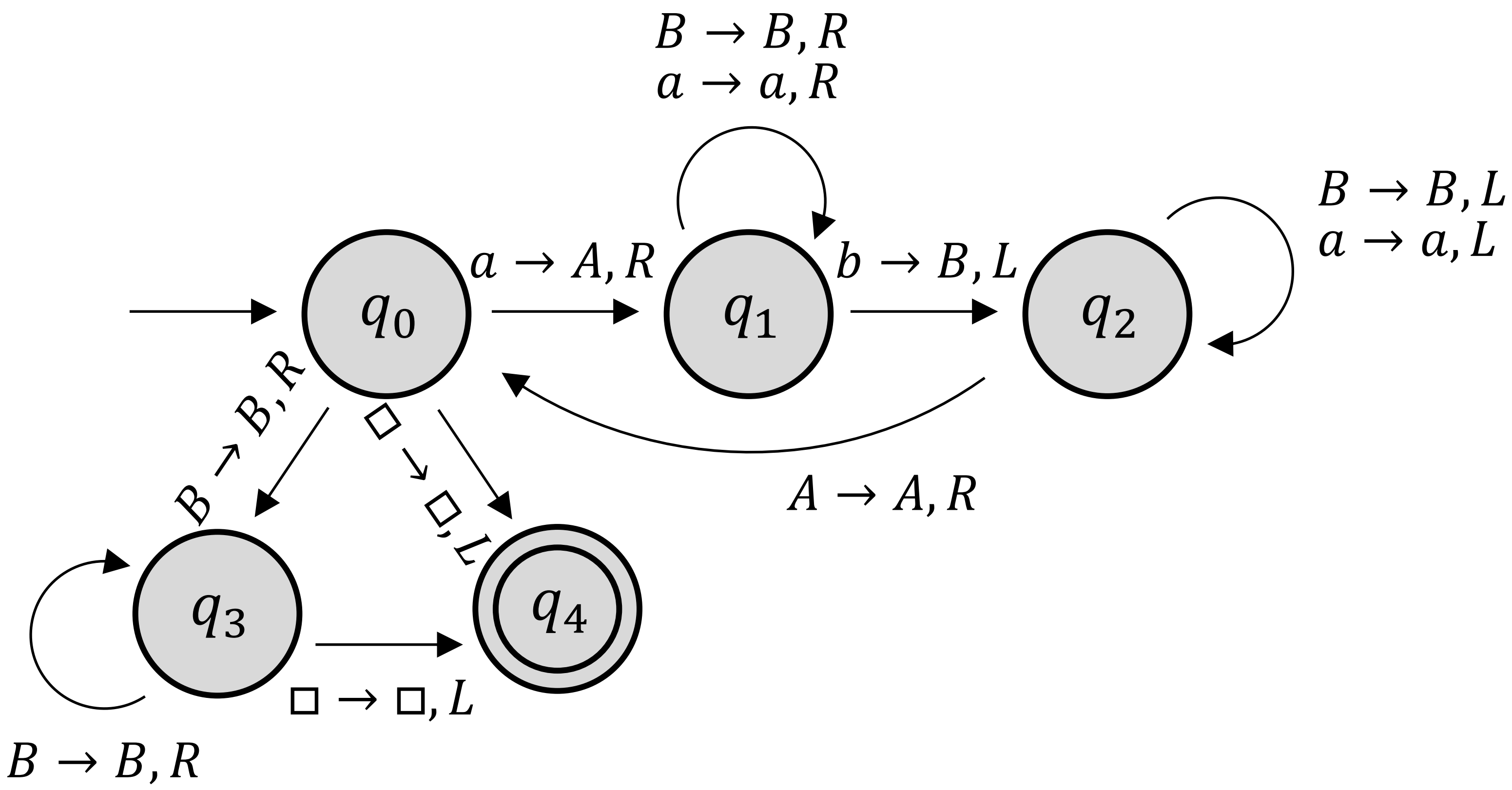
Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$
  - $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \{a, b, A, B, \square\}, \delta, q_0, \square, \{q_4\})$



Turing machine

- **Example:** Design a TM for $L = \{a^n b^n : n \geq 0\}$



ID for the input string $aabb$

$q_0 aabb$
$\vdash Aq_1abb$
$\vdash Aaq_1bb$
$\vdash Aq_2aBb$
$\vdash q_2AaBb$
$\vdash Aq_0aBb$
$\vdash AAq_1Bb$
$\vdash AABq_1b$
$\vdash AAq_2BB$
$\vdash Aq_2ABB$
$\vdash AAq_0BB$
$\vdash AABq_3B$
$\vdash AABBq_3\square$
$\vdash AABq_4B$

Turing machine

- **Practice:** Design a TM for $L = (r)$ where $r = 01^{*}0$

Turing machine

- **Practice:** Design a TM for $L = (r)$ where $r = 01^*0$

Turing machine

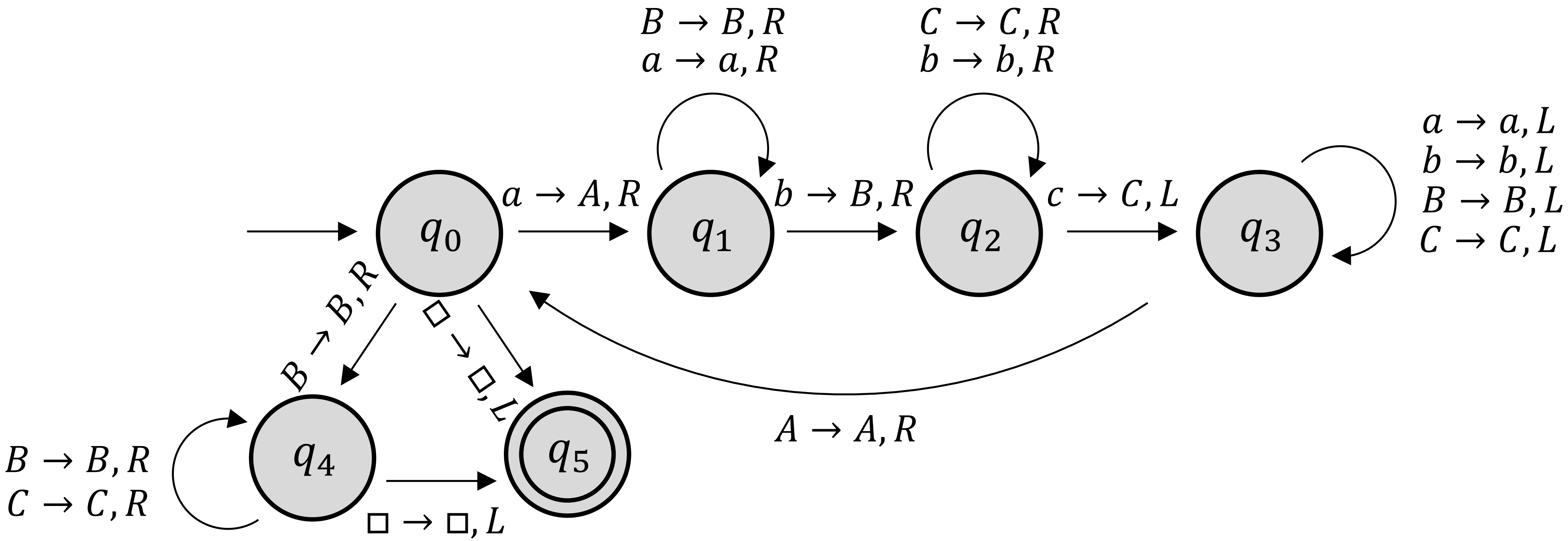
- **Practice:** Design a TM for $L = \{a^n b^n c^n : n \geq 0\}$

Turing machine

- **Practice:** Design a TM for $L = \{a^n b^n c^n : n \geq 0\}$
  - Hint: basic idea
    - ❖ **While** there are a's **do**
    - ❖ **Find** and **Replace** a with **A**
    - ❖ **Find** and **Replace** b with **B**
    - ❖ **Find** and **Replace** c with **C**
    - ❖ **end**
    - ❖ If no symbol a, b, c exist => accept

Turing machine

- **Example:** Design a TM for $L = \{a^n b^n c^n : n \geq 0\}$
  - $M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{a, b, c\}, \{a, b, c, A, B, C, \square\}, \delta, q_0, \square, \{q_5\})$



Next Lecture

- **Turing machines as transducers (변환기)**
- **Turing machines for complicated tasks**