# Lecture 13: Turing Machines

CSC 320: Foundations of Computer Science

Quinton Yong

quintonyong@uvic.ca

University of Victoria

# Turing Machine: Formal Definition

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

- $\boldsymbol{Q}$ : set of states

- $\boldsymbol{\Sigma}$ : input alphabet (**not containing** blank symbol $\sqcup$)
- $\boldsymbol{\Gamma}$ : tape alphabet ($\sqcup \in \boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma} \in \boldsymbol{\Gamma}$ as well as other symbols)

- $\boldsymbol{\delta}: \boldsymbol{Q} \times \boldsymbol{\Gamma} \to \boldsymbol{Q} \times \boldsymbol{\Gamma} \times \{\mathbf{L}, \mathbf{R}\}$ : transition function (**must move** tape head)

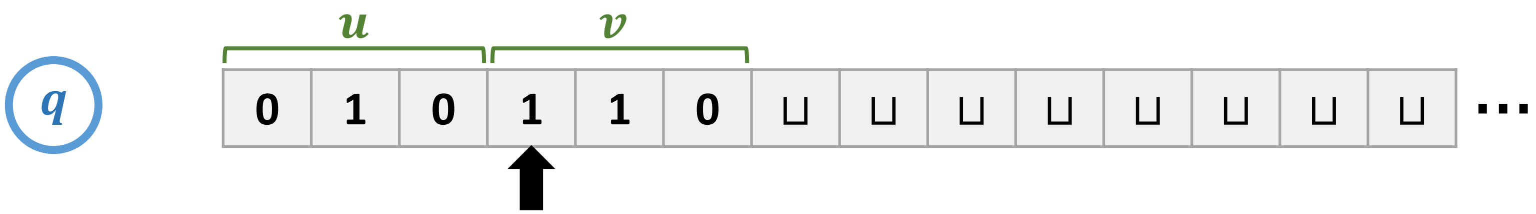
$$q_i \xrightarrow{\mathbf{0} \to \mathbf{x}, \mathbf{R}} q_j$$

- If tape head reading $\mathbf{0}$, write $\mathbf{x}$ and move head **right**
- If we don't modify tape cell, can use shorthand $\mathbf{0} \to \boldsymbol{R}$
- This transition function is **deterministic** (one transition for every tape alphabet symbol)

- $\boldsymbol{q_0} \in \boldsymbol{Q}$ : single start state
- $\boldsymbol{q_{accept}} \in \boldsymbol{Q}$ : single accept state
- $\boldsymbol{q_{reject}} \in \boldsymbol{Q}$ : single reject state (with $\boldsymbol{q_{reject}} \neq \boldsymbol{q_{accept}}$)

# Configuration of Turing Machines

- A TM **configuration** is a string **description of the current state** of the TM
- Given a TM $M$, a configuration of $M$ consists of a description of:
  - Its current **state**
  - Its current **tape content**
  - Its current **tape head location**

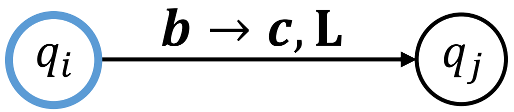


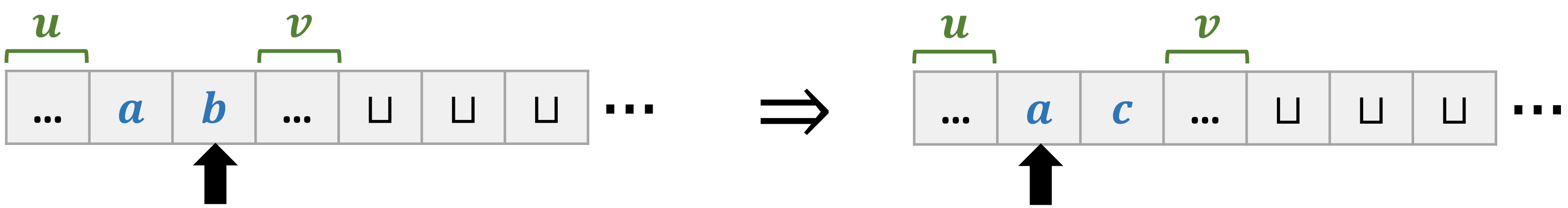

- We write $\boldsymbol{u\ q\ v}$ for the configuration where:
  - Current state is $\boldsymbol{q}$
  - Current tape content is $\boldsymbol{uv}$ where strings $\boldsymbol{u, v} \in \boldsymbol{\Gamma}^*$
  - Current tape head location is first symbol of $\boldsymbol{v}$
  - Tape contains only blanks following last symbol of $\boldsymbol{v}$

# Configuration of Turing Machines

- For a TM $\boldsymbol{M}$, let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \boldsymbol{\Gamma}$, let $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{\Gamma}^*$, and $\boldsymbol{q_i}, \boldsymbol{q_j} \in \boldsymbol{Q}$
- Let $\boldsymbol{ua}\ \boldsymbol{q_i}\ \boldsymbol{bv}$ , $\boldsymbol{u}\ \boldsymbol{q_j}\ \boldsymbol{acv}$ , and $\boldsymbol{uac}\ \boldsymbol{q_j}\ \boldsymbol{v}$ be configurations

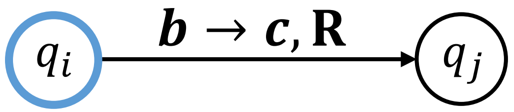
- $\boldsymbol{ua}\ \boldsymbol{q_i}\ \boldsymbol{bv}$ **yields** $\boldsymbol{u}\ \boldsymbol{q_j}\ \boldsymbol{acv}$ if $\boldsymbol{\delta}(\boldsymbol{q_i}, \boldsymbol{b}) = (\boldsymbol{q_j}, \boldsymbol{c}, \mathbf{L})$

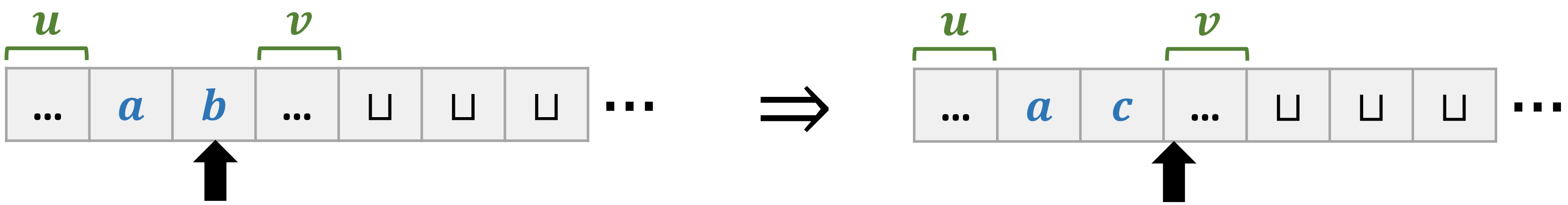



Read $\boldsymbol{b}$, write $\boldsymbol{c}$, move **left**

- $\boldsymbol{ua}\ \boldsymbol{q_i}\ \boldsymbol{bv}$ **yields** $\boldsymbol{uac}\ \boldsymbol{q_j}\ \boldsymbol{v}$ if $\boldsymbol{\delta}(\boldsymbol{q_i}, \boldsymbol{b}) = (\boldsymbol{q_j}, \boldsymbol{c}, \mathbf{R})$

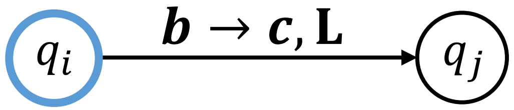


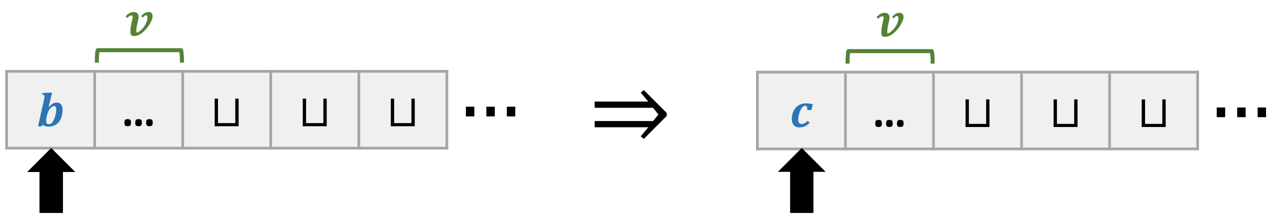

Read $\boldsymbol{b}$, write $\boldsymbol{c}$, move **right**

# Special Configurations of Turing Machines

Tape head is at **left end**:

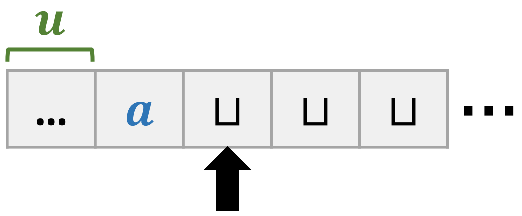



- $\boldsymbol{\delta(q_i, b) = (q_j, c, L)}$: configuration $\boldsymbol{q_i\ bv}$ yields $\boldsymbol{q_j\ cv}$ (if transition moves left, prevent tape head from going off left-end of tape)




Tape head is at **right end**:

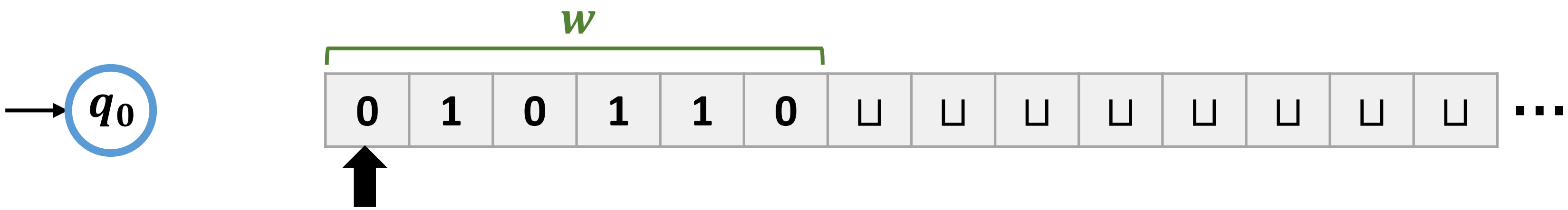
- Configuration $\boldsymbol{ua\ q_i}$ is equivalent to $\boldsymbol{ua\ q_i}$ ⊔ since we assume that blanks follow the last part of tape in configuration



# Special Configurations of Turing Machines

**Start** configuration $q_0$ $w$:

- Input string is $w$, $M$ is in start state $q_0$, tape head at leftmost position




**Halting** configurations:

- **Accepting** configuration: any configuration where state is $q_{accept}$


- **Rejecting** configuration: any configuration where state is $q_{reject}$

# Computation of a Turing Machine

- Turing machine $\boldsymbol{M}$ **accepts input** $\boldsymbol{w}$ if a sequence of configurations

$$\boldsymbol{C_1, C_2, \ldots, C_k}$$

exists where:

1. $\boldsymbol{C_1}$ is the **start** configuration
2. Each $\boldsymbol{C_i}$ **yields** $\boldsymbol{C_{i+1}}$
3. $\boldsymbol{C_k}$ is an **accepting** configuration

- The **language** $\boldsymbol{L(M)}$ of $\boldsymbol{M}$, or the **language recognized by** $\boldsymbol{M}$, is the collection of all strings that $\boldsymbol{M}$ accepts

# Outcomes of TM Computation

Possible outcomes of a TM on an input string $\boldsymbol{w}$ are:

- **Accept** (halt and accept)
- **Reject** (halt and reject)
- **Loop infinitely** (considered **non-accept**, but **not a halt reject**)

A Turing machine that **halts on every input** (never loops) is called a **decider**

# Turing-recognizable and Turing-decidable

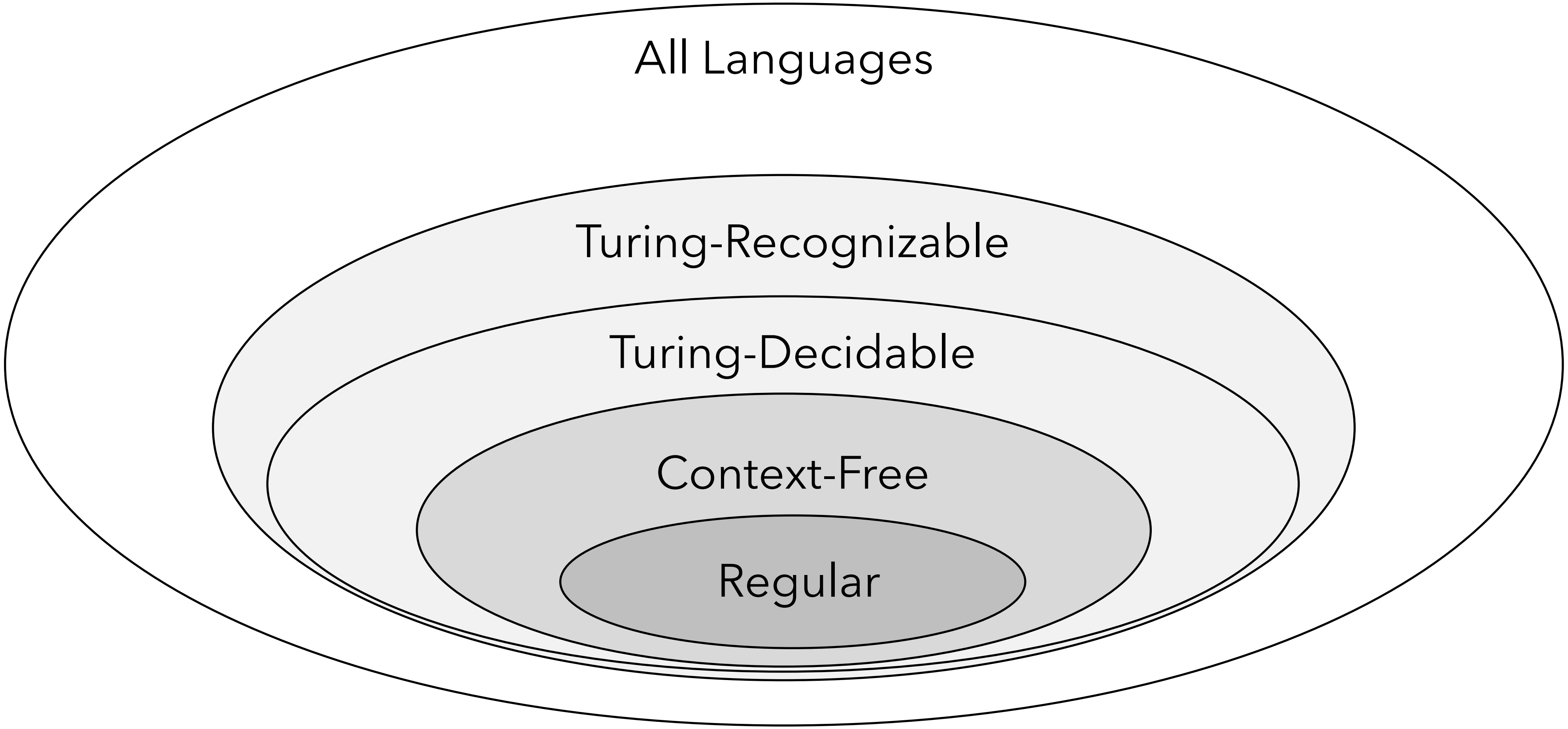
If a language $\boldsymbol{L}$ is **recognized** by some TM, we call $\boldsymbol{L}$ **Turing-recognizable**

- **Halts and accepts** on input strings **in $\boldsymbol{L}$**
- **Halts and rejects or loops infinitely** on inputs strings **not in $\boldsymbol{L}$**

We say a language $\boldsymbol{L}$ is **Turing-decidable** or **decidable** if there exists a **decider** that recognizes $\boldsymbol{L}$ (we also say that $\boldsymbol{M}$ decides $\boldsymbol{L}$)

- **Halts and accepts** on strings **in $\boldsymbol{L}$**
- **Halts and rejects** on strings **not in $\boldsymbol{L}$**
- Never loops infinitely

# Turing-recognizable and Turing-decidable




- **Note**: Every Turing-decidable language **is also Turing-recognizable**

# Decidable Language Example

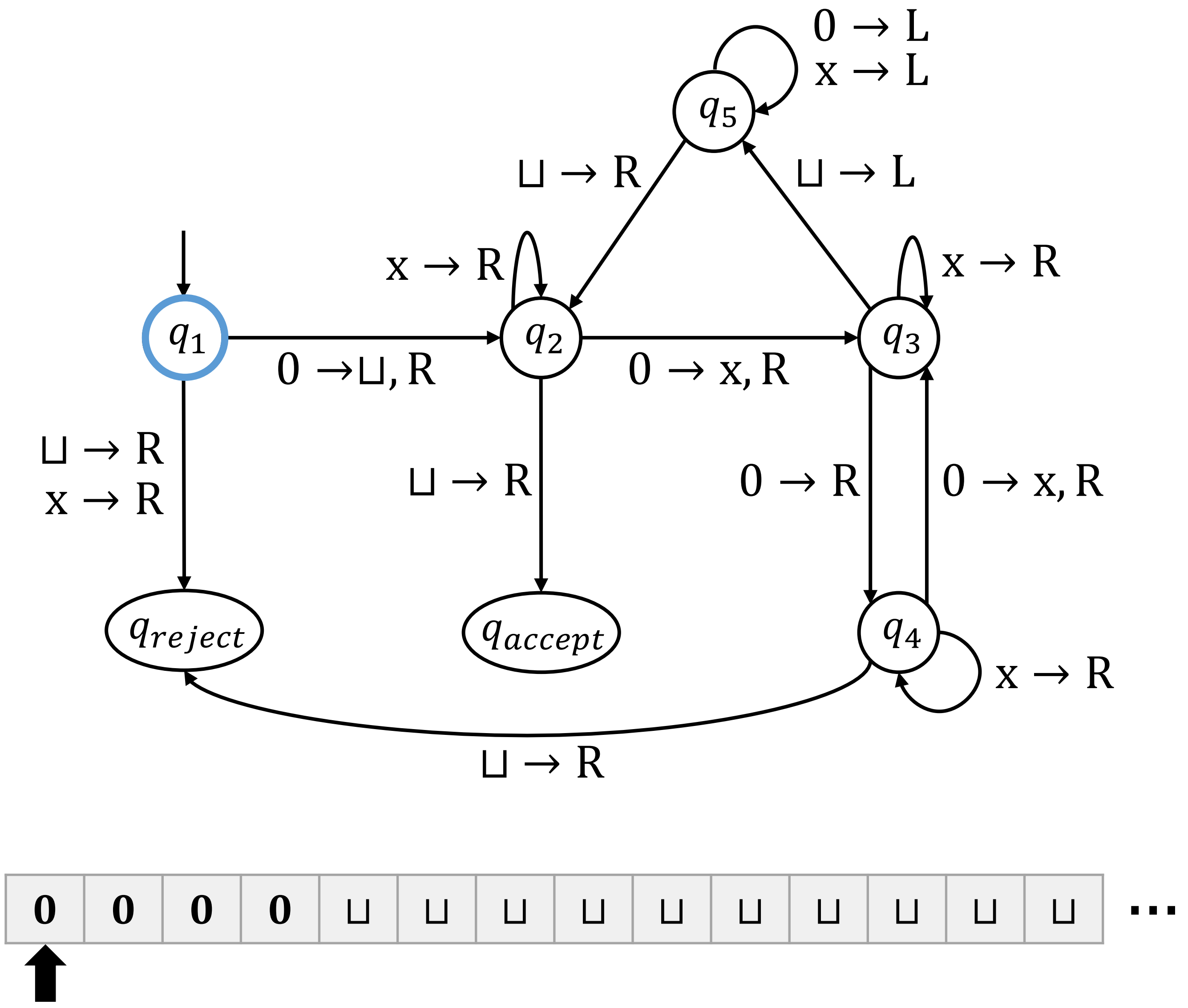
- Let $\boldsymbol{L} = \left\{ \mathbf{0}^{2^{\boldsymbol{n}}} \mid \boldsymbol{n} \geq \mathbf{0} \right\}$ over $\boldsymbol{\Sigma} = \{\mathbf{0}\}$. We describe a TM $\boldsymbol{M}$ that decides $\boldsymbol{L}$.

**Implementation level description**:

- On input string $\boldsymbol{w} \in \{\mathbf{0}\}^*$
  1. Sweep left to right, crossing off every other $\mathbf{0}$
  2. If the tape contains exactly one $\mathbf{0}$, **accept**
  3. If the tape contains more than one $\mathbf{0}$, and the number of $\mathbf{0}$'s is odd, **reject**
  4. Otherwise, return to the left end of the tape
  5. Go to step 1

# Decidable Language Example

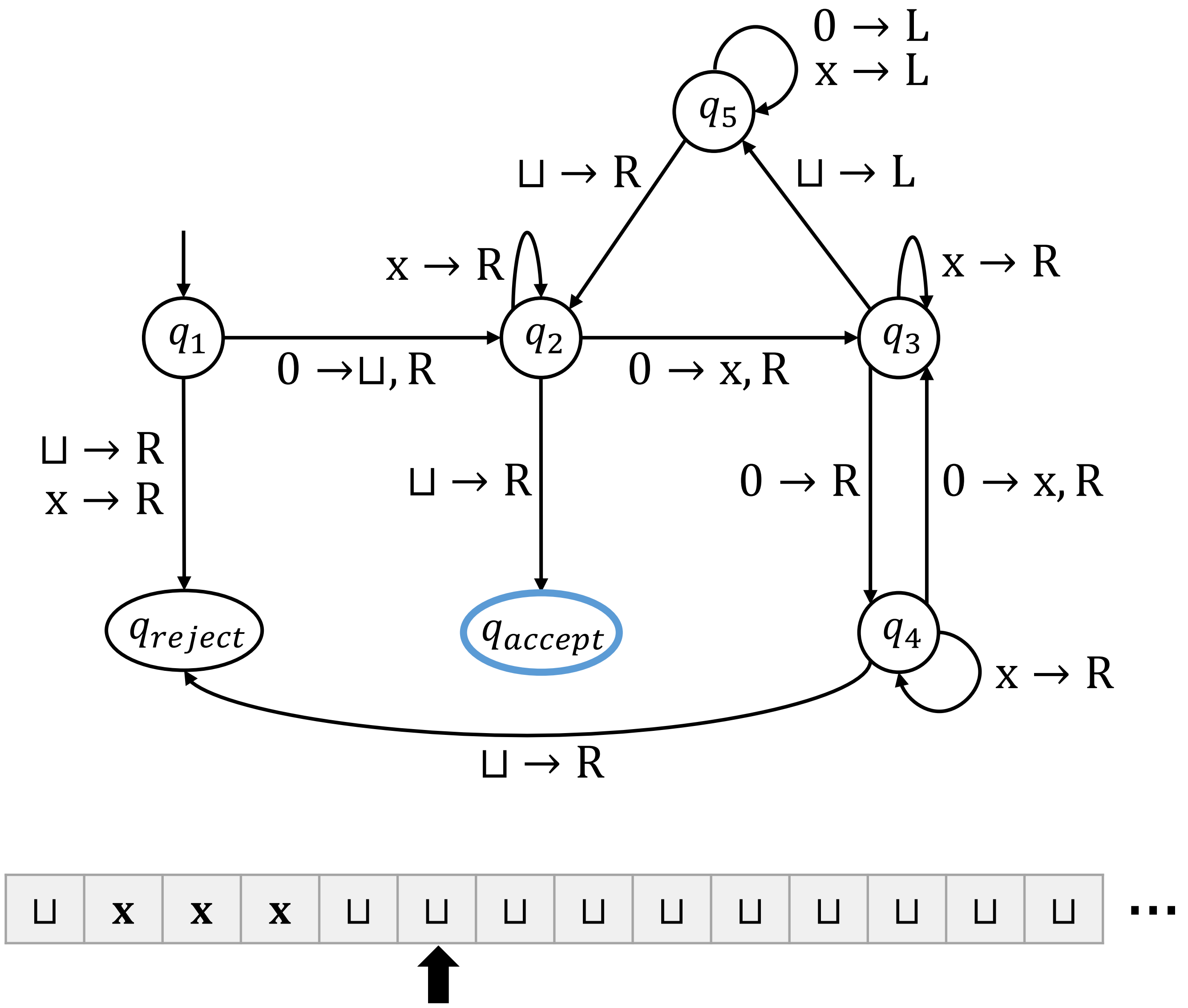
**Input: 0000**



# Decidable Language Example

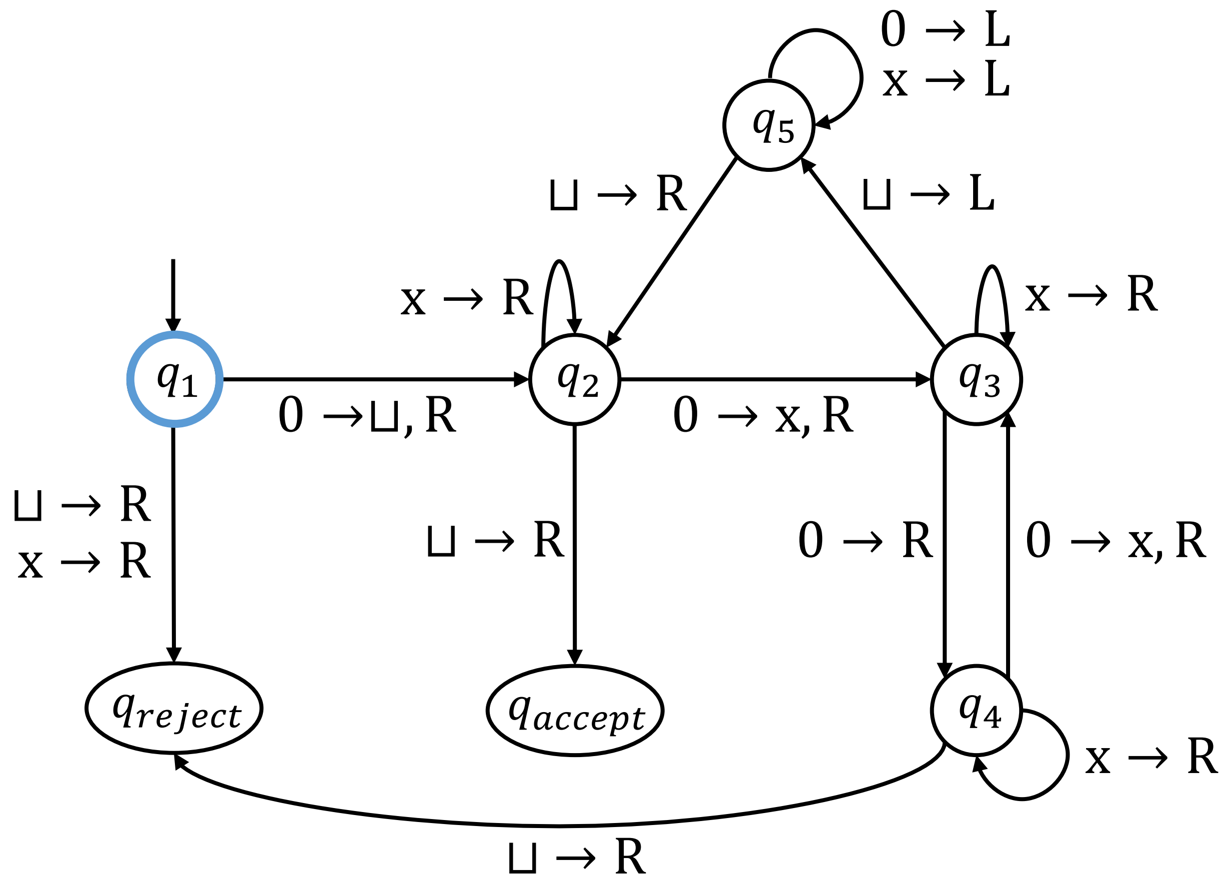
**Input: 0000**

**Accept**



# Decidable Language Example

**Input: 000**

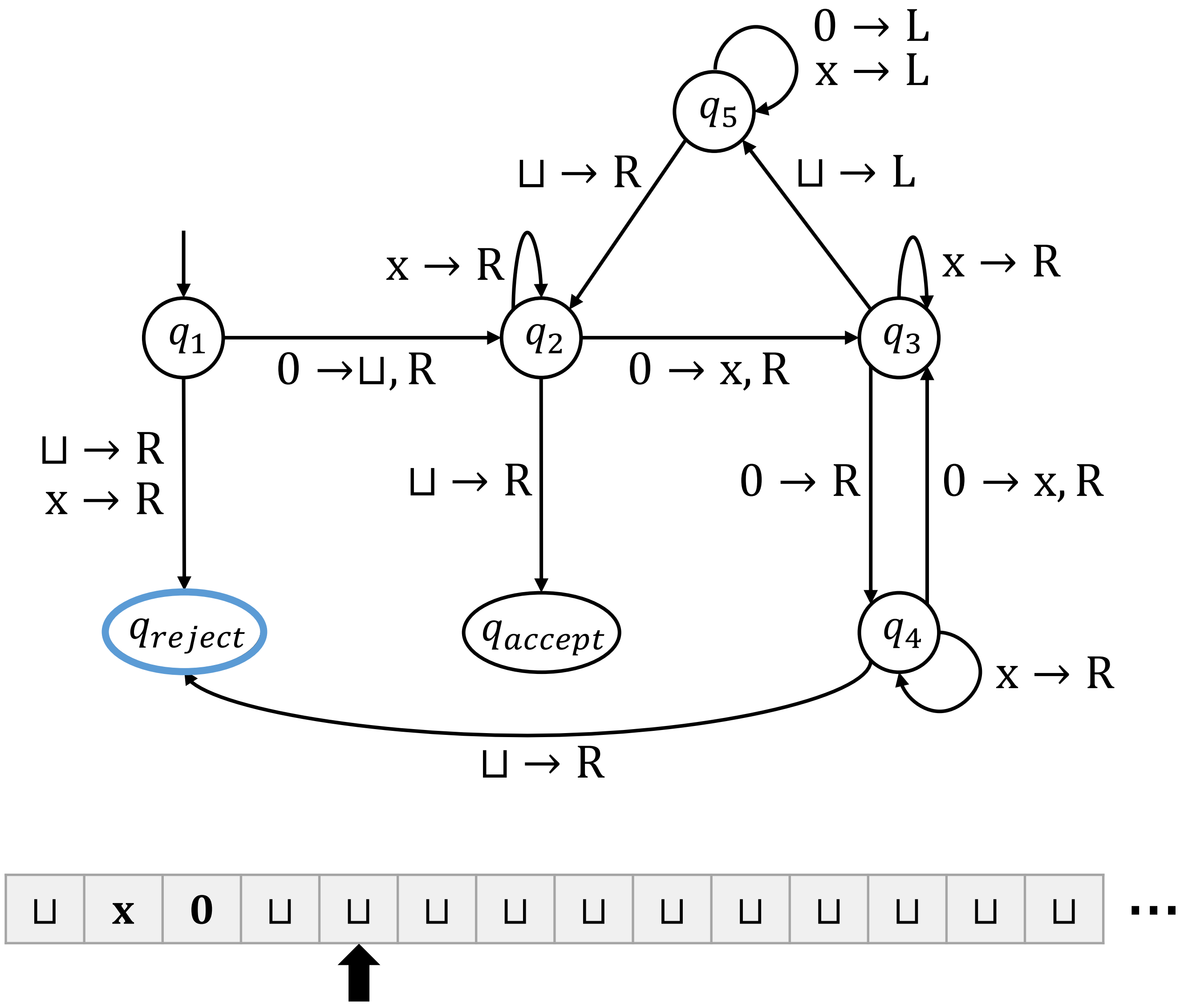



| 0 | 0 | 0 | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

# Decidable Language Example

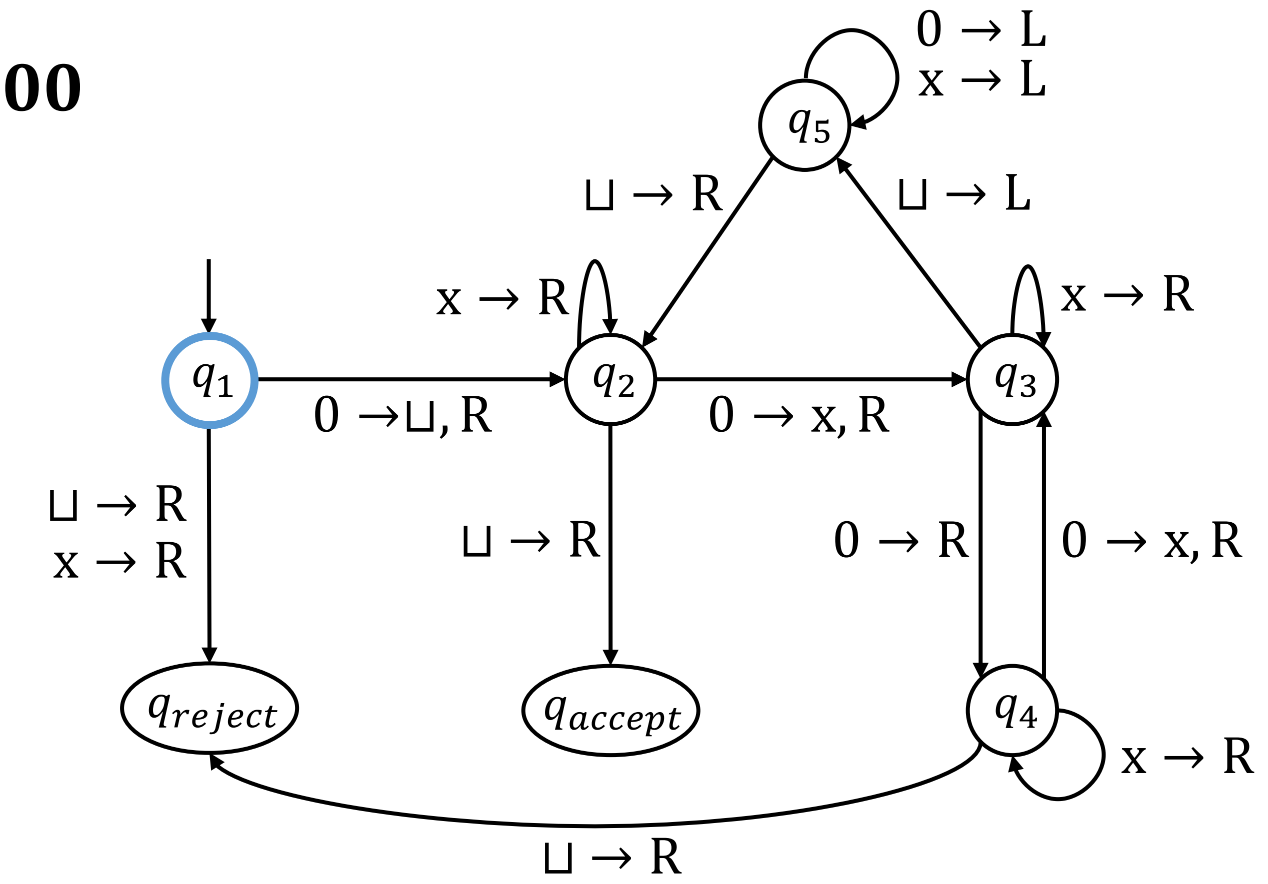
**Input: 000**

**Reject**



# Decidable Language Example

**Input: 000000**

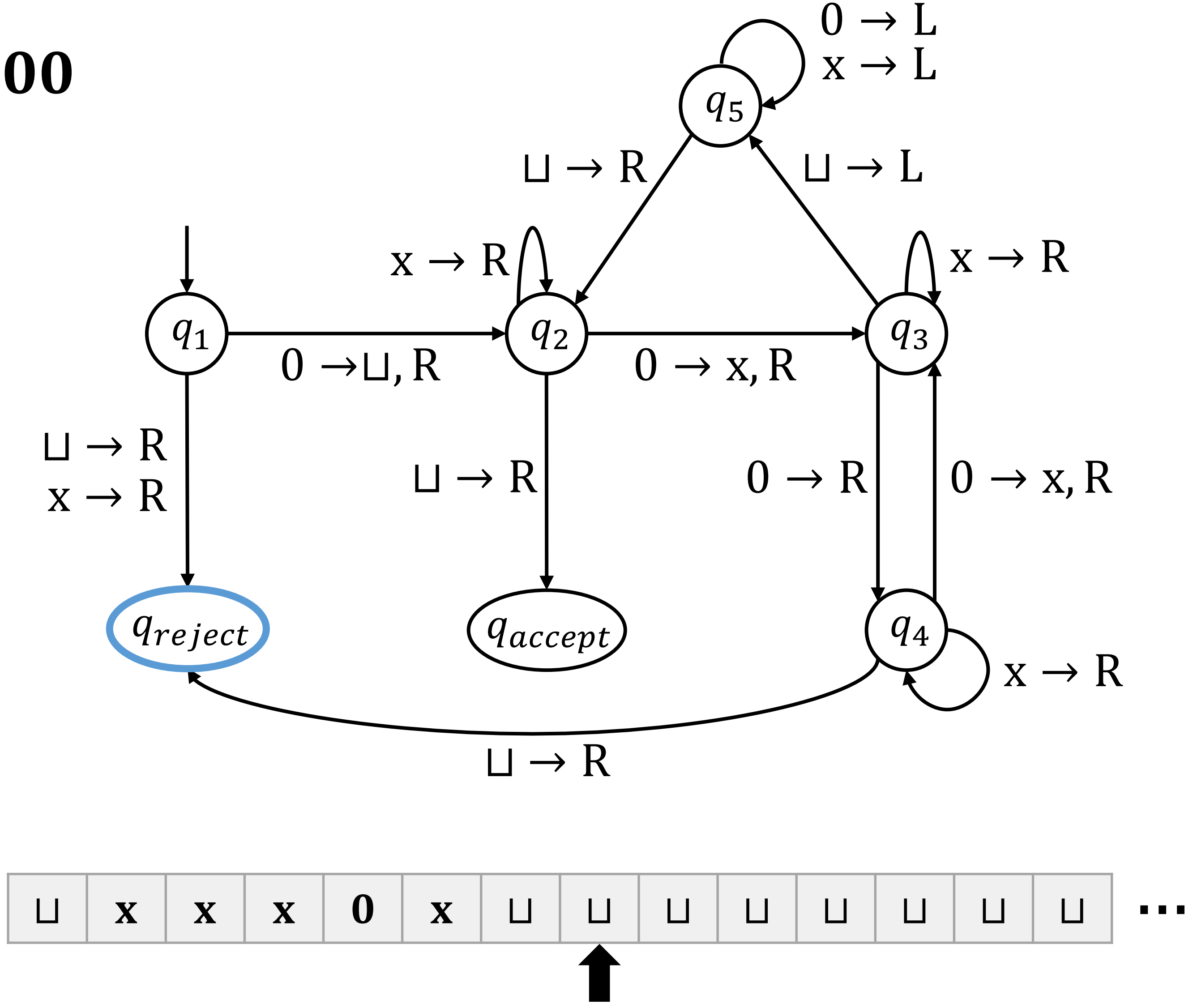



| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | **0** | **0** | **0** | **0** | **0** | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ··· |

# Decidable Language Example

**Input: 000000**

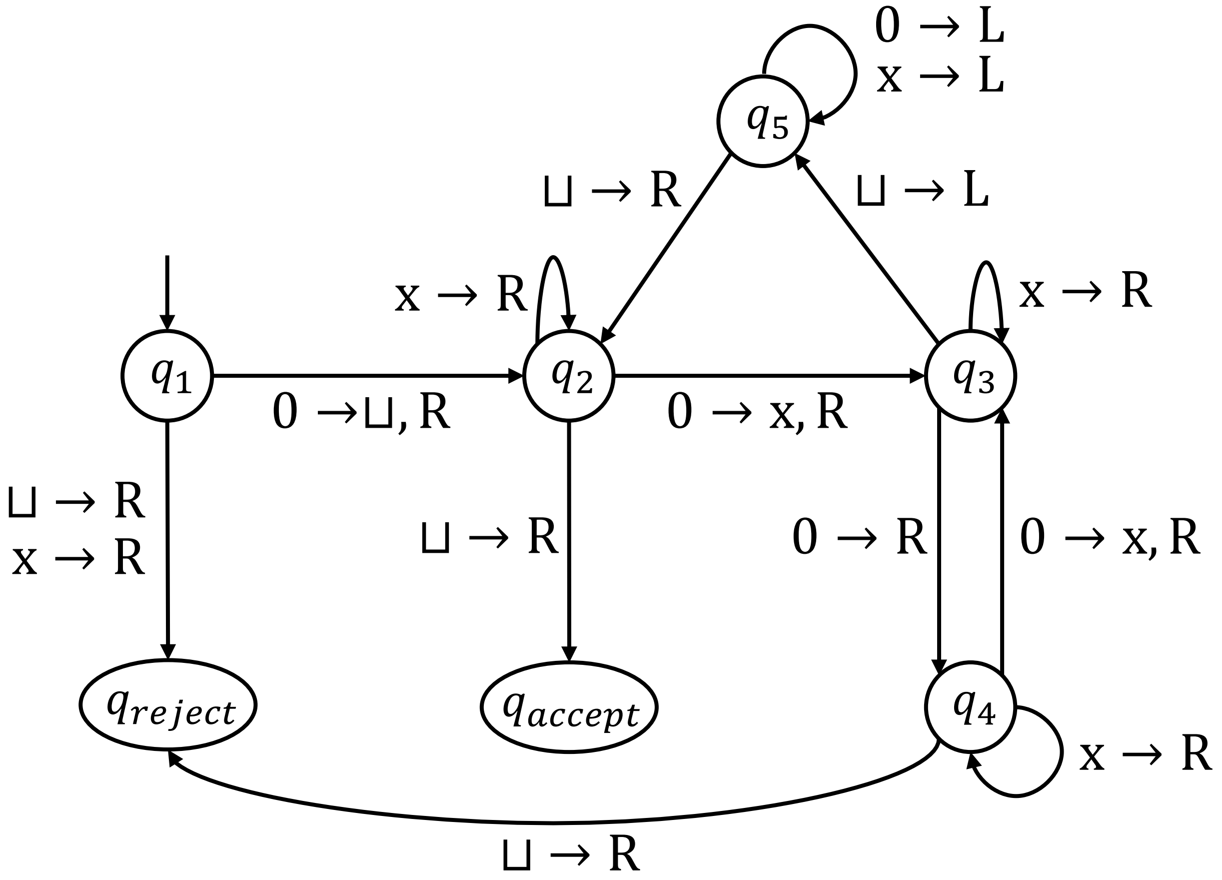
**Reject**



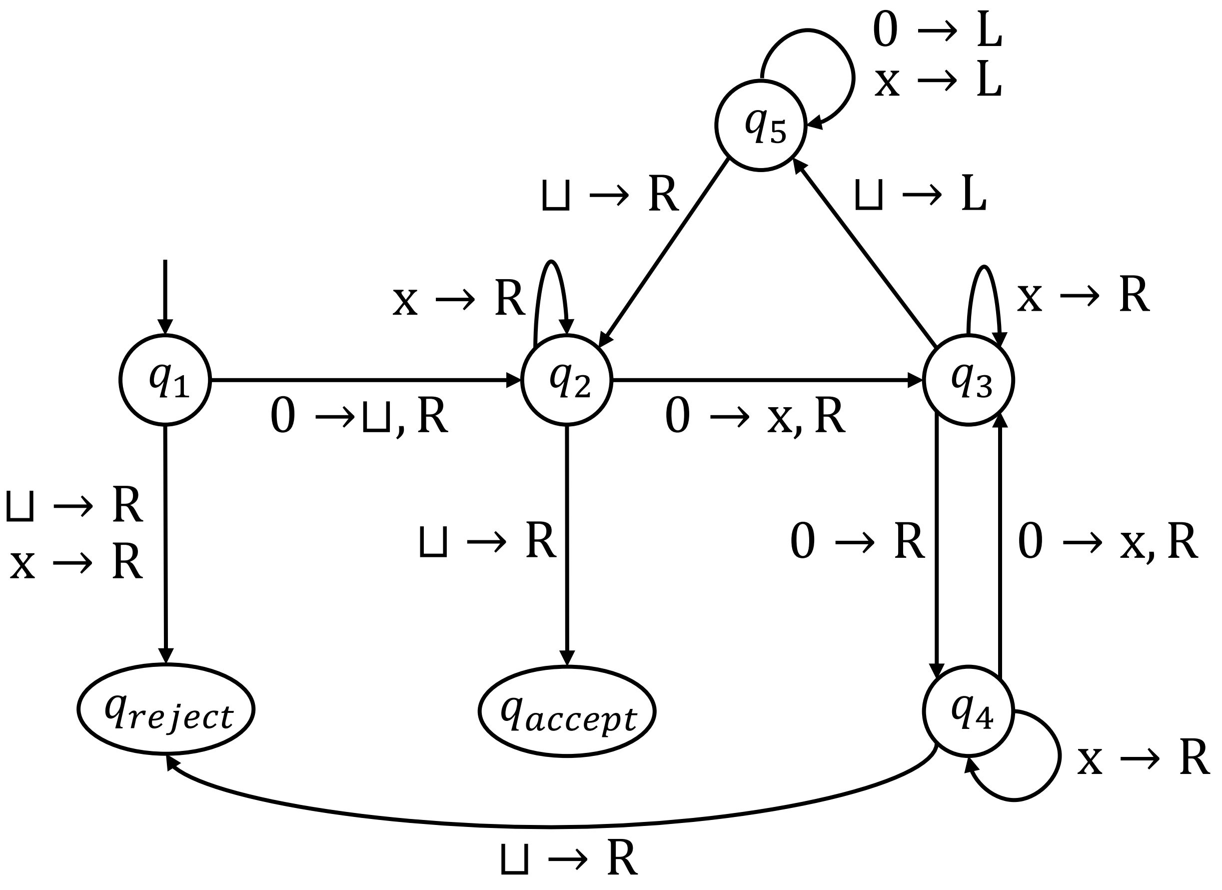
# Formal Description of TM

$M = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$ where:

- $Q = \{q_1, q_2, q_3, q_4, q_5, q_{accept}, q_{reject}\}$
- $q_1$ is the start state
- $q_{accept}$ is the accept state
- $q_{reject}$ is the reject state
- $\Sigma = \{0\}$
- $\Gamma = \{0, x, \sqcup\}$
- $\delta$ is described by state diagram



# Decidable Language Example




- This TM is a **decider** for $\boldsymbol{L = \{ 0^{2^n} \mid n \geq 0 \}}$ since it will not infinitely loop
- Therefore, $\boldsymbol{L}$ is a **Turing-decidable** language

# Turing Machine Example 2

- Design a TM that **increments a binary number** given as input by $\mathbf{1}$

- Note: Adding $\mathbf{1}$ to a binary number will not change the length unless the entire string is $\mathbf{1}$'s

**Implementation level description:**

- If input consists of only $\mathbf{1}$'s, shift the string one to the right

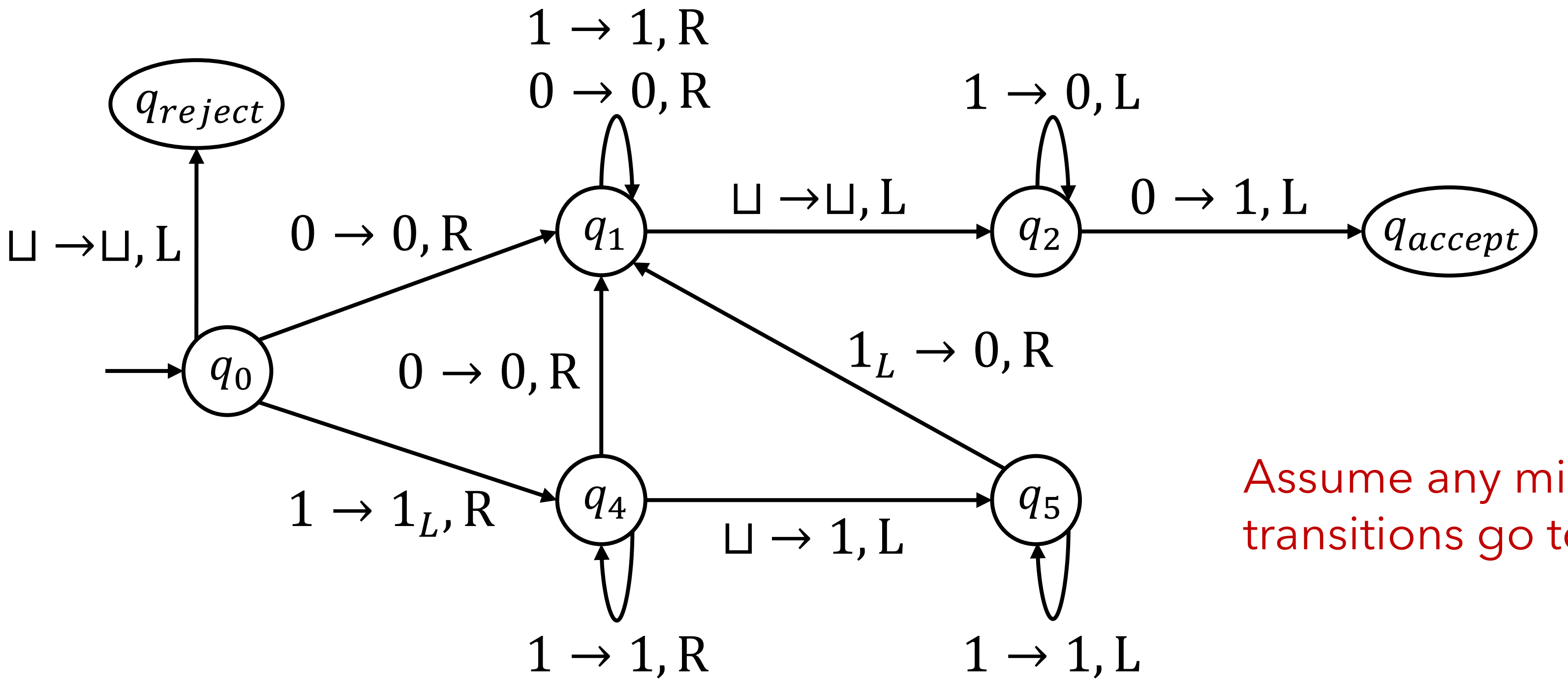
- If right-most symbol is $\mathbf{0}$, change it to one. **Done**.

- If right-most symbol is $\mathbf{1}$, change suffix of $\mathbf{1}$'s to $\mathbf{0}$'s, and previous right-most $\mathbf{0}$ to $\mathbf{1}$

# Turing Machine Example 2

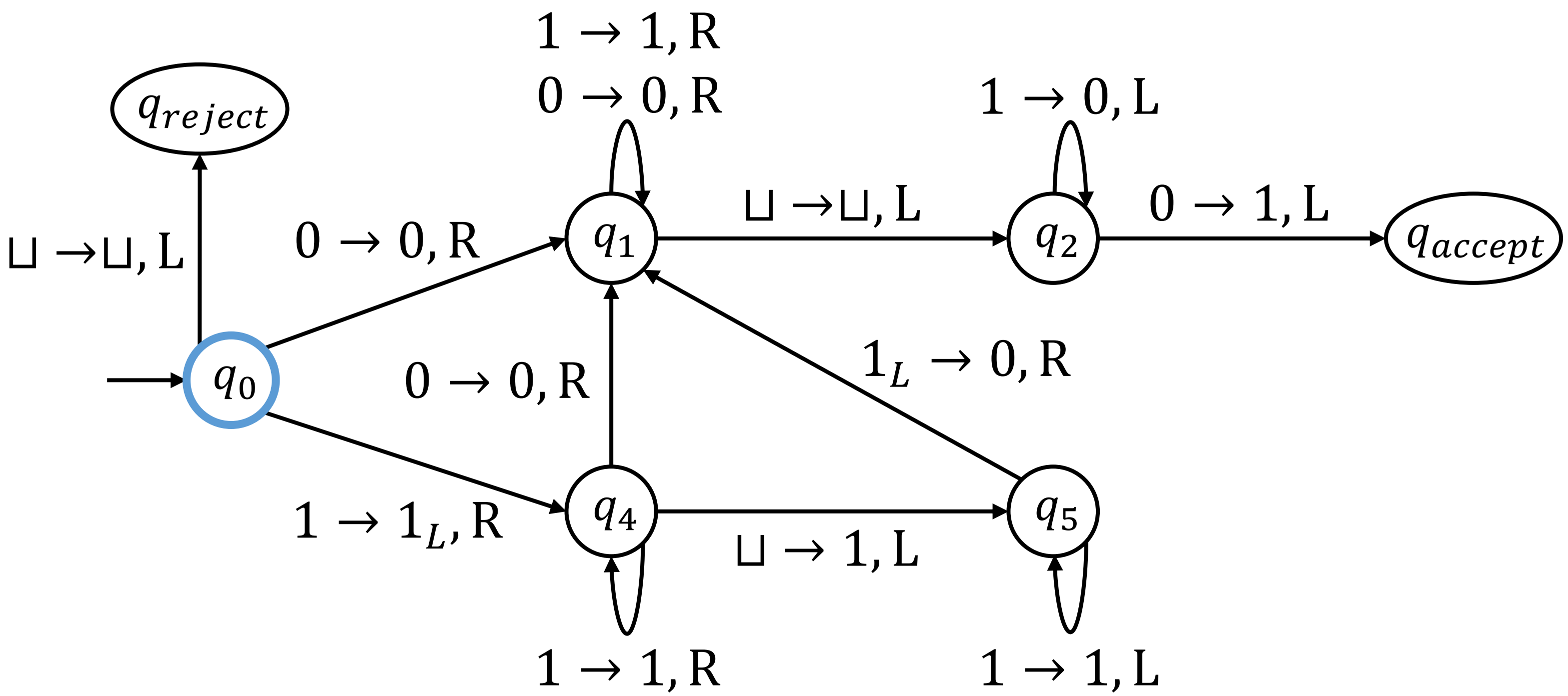
**Implementation level description:**

- If input consists of only $\mathbf{1}$'s, shift the string one to the right
- If right-most symbol is $\mathbf{0}$, change it to one. **Done**.
- If right-most symbol is $\mathbf{1}$, change suffix of $\mathbf{1}$'s to $\mathbf{0}$'s, and previous right-most $\mathbf{0}$ to $\mathbf{1}$




Assume any missing outgoing transitions go to $\boldsymbol{q_{reject}}$

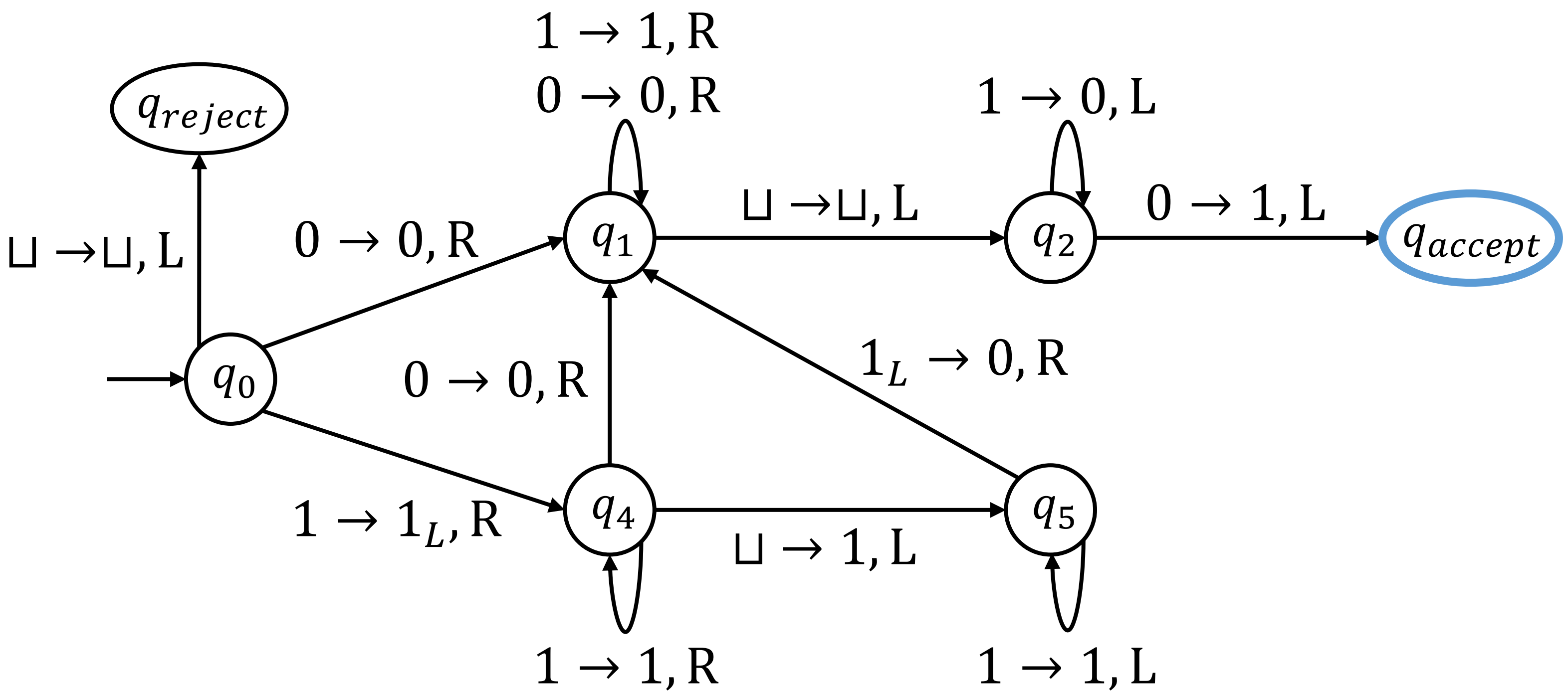
# Turing Machine Example 2




**Input: 10**

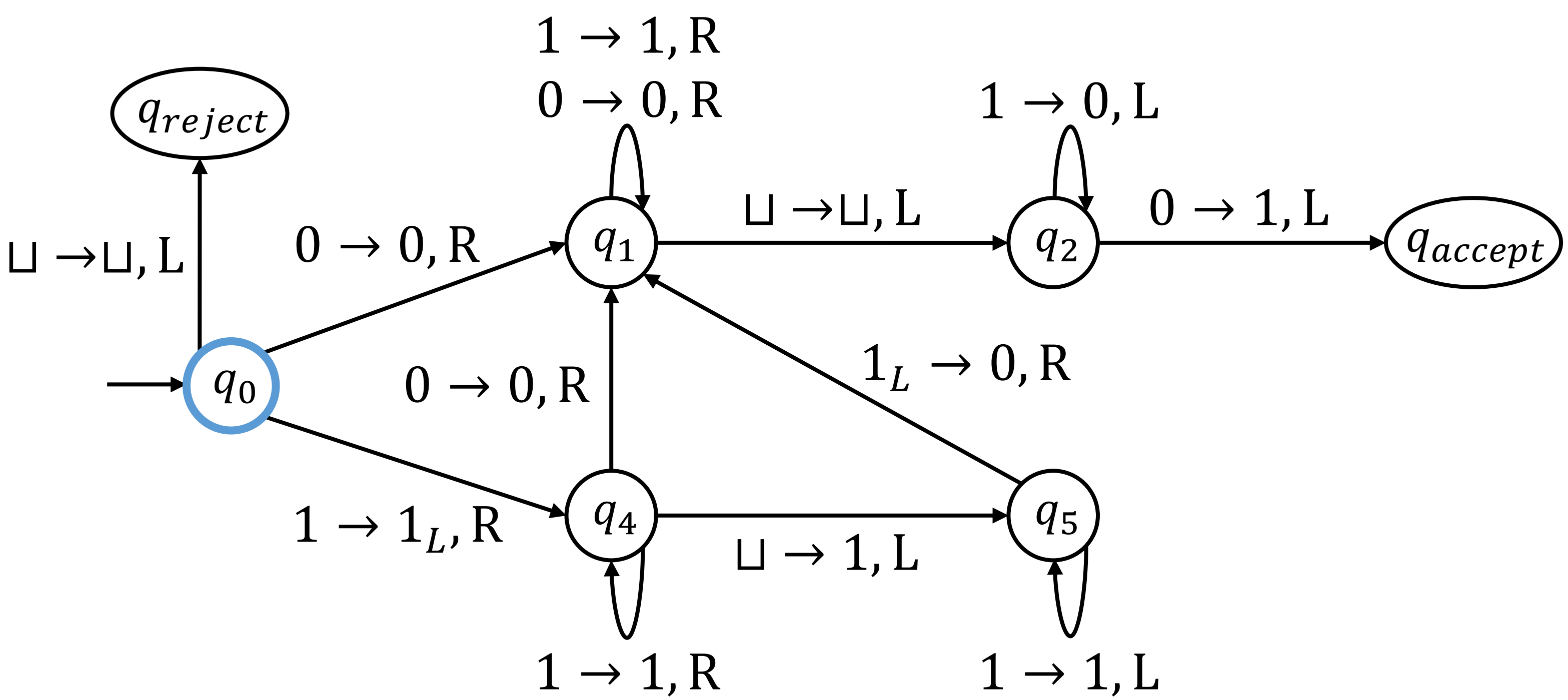
| 1 | 0 | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ⬆ | | | | | | | | | | | | | | |

# Turing Machine Example 2




**Input: 10**

# Turing Machine Example 2

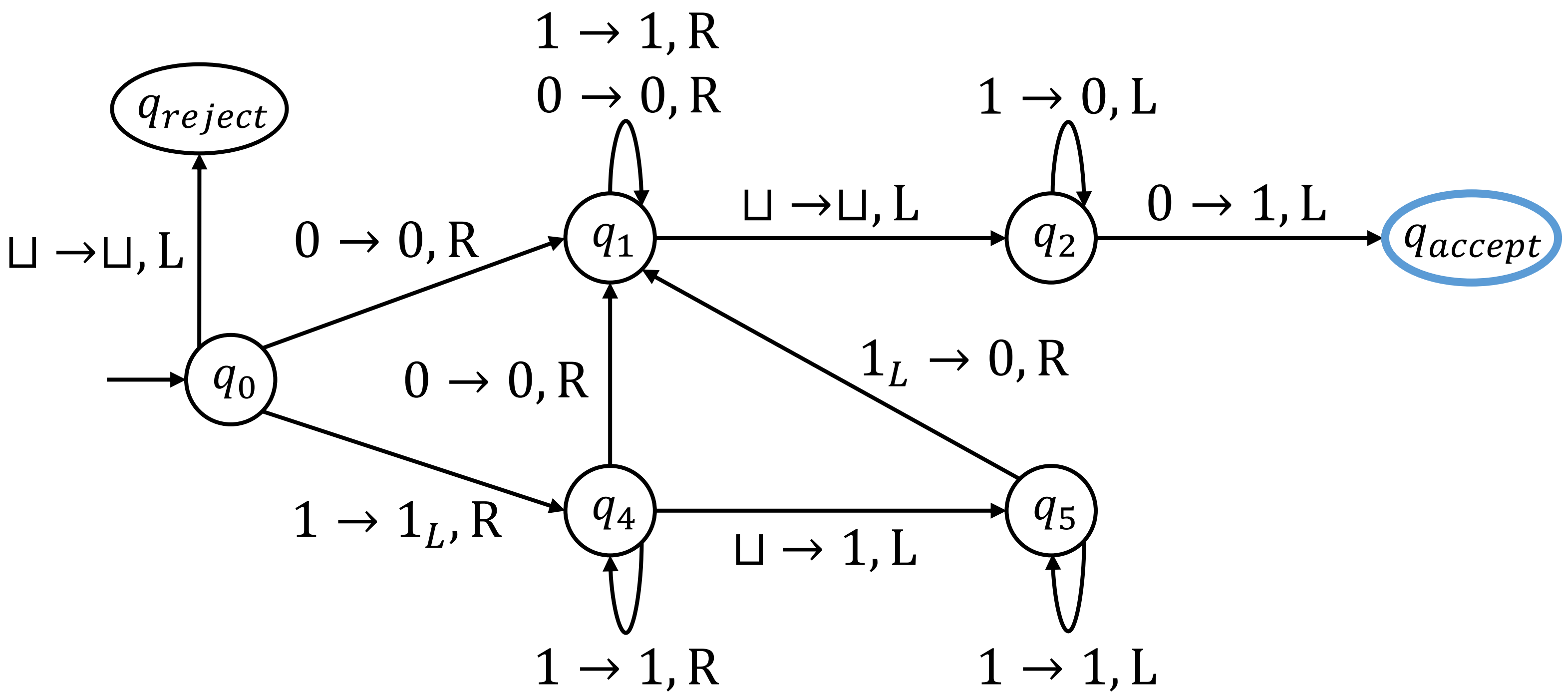



**Input: 111**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | **1** | **1** | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ |

⋯

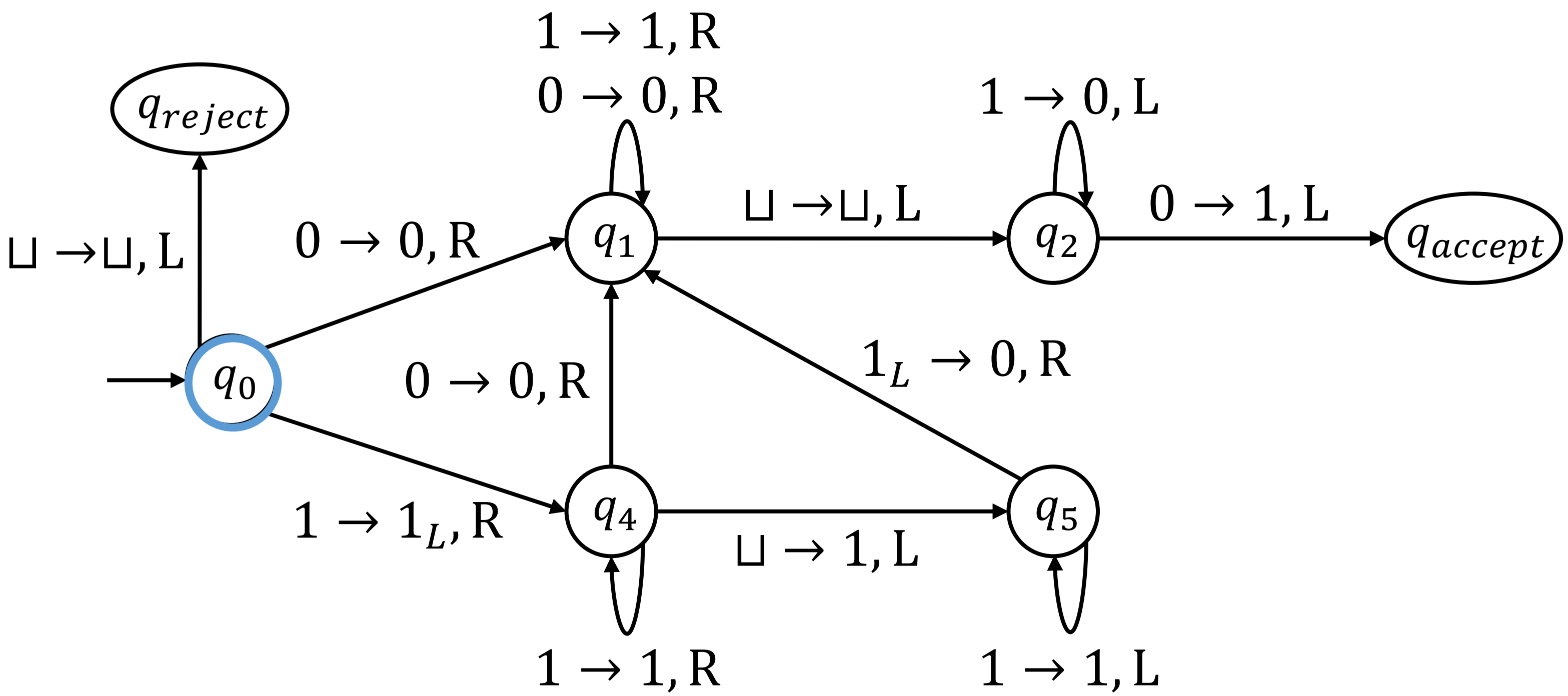
# Turing Machine Example 2




**Input: 111**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⋯ |

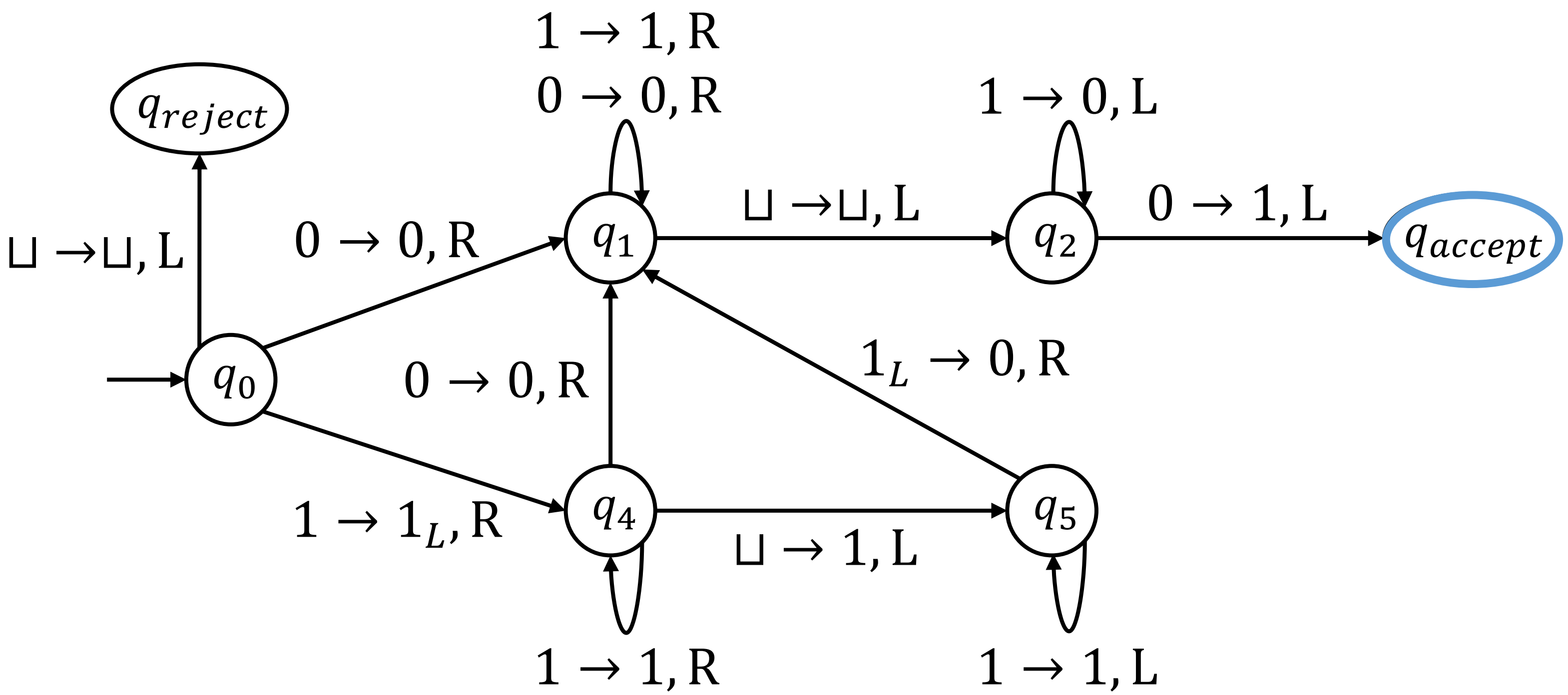
# Turing Machine Example 2




**Input: 1100111**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | **1** | **0** | **0** | **1** | **1** | **1** | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⋯ |

# Turing Machine Example 2




**Input: 1100111**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $1_L$ | 1 | 0 | 1 | 0 | 0 | 0 | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⊔ | ⋯ |

# Next Lecture

The Turing machine we have learned so far has the **following characteristics**:

- **Deterministic**
- **One tape**, left-ended, unlimited to the right
- Read / write head

In the next lecture, we will learn about **TM variants**:

- Tape head can also **stop / pause**, not just move left and right
- **Multiple tapes**
- **Nondeterministi**c Turing machines
- Enumerators