Computational Complexity and Computability

Lecture 2 - Turing Machines

Koushik Pal

University of Toronto

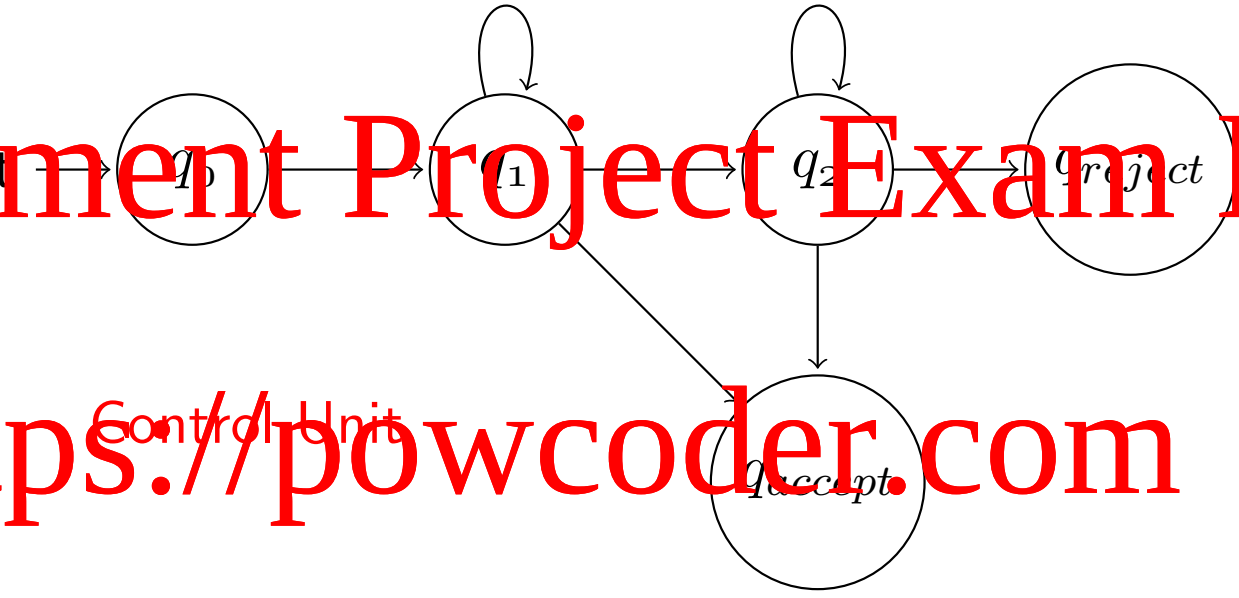
January 13, 2021

Motivation

Goal

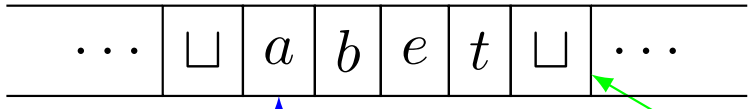
- Define computation as abstractly and generally as possible
- Obtain a rigorous definition of algorithm
- Obtain a rigorous definition of efficiency

Turing Machine



Control Unit

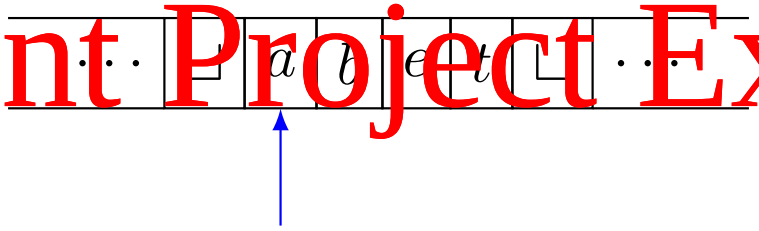
Tape



Read Write Head

Blank Symbol

Tape

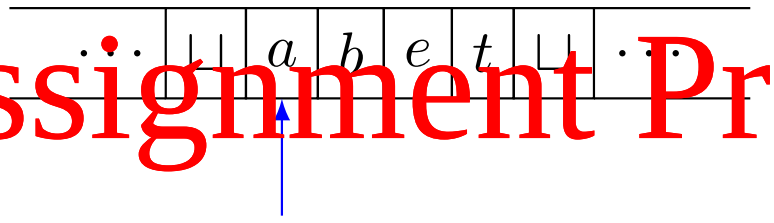


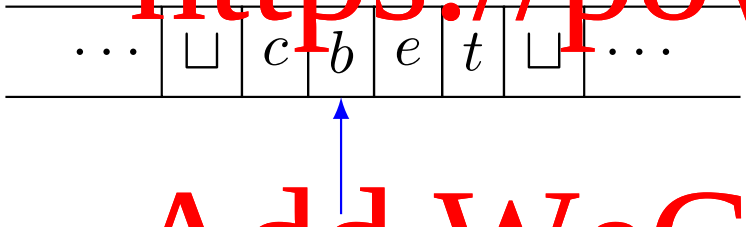
The read write head at each time step does the following:

1. Reads a tape symbol
2. Writes a tape symbol
3. Moves Left or Right

Examples

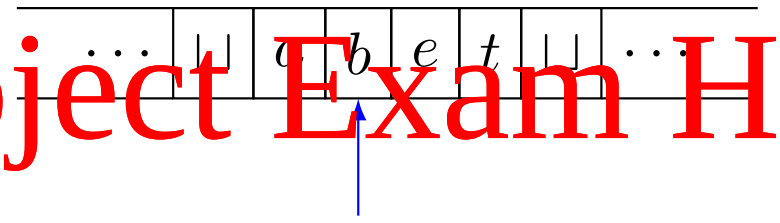
Example1

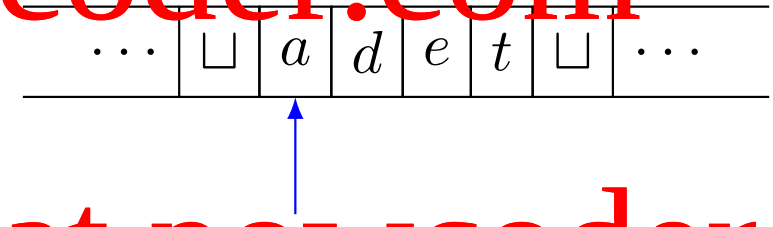




1. Reads a
2. Writes c
3. Moves $Right$

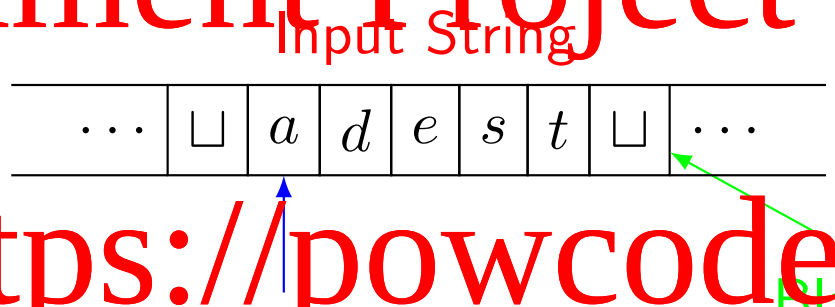
Example2





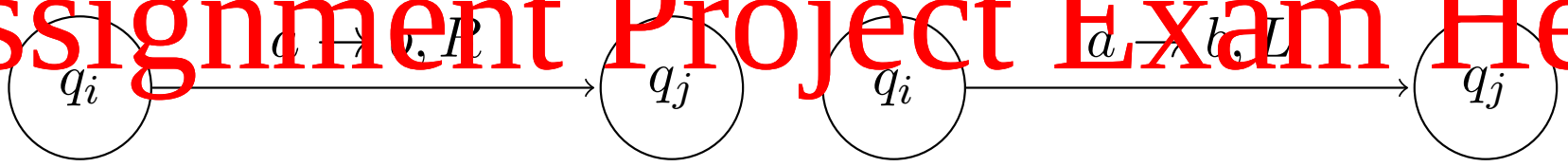
1. Reads b
2. Writes d
3. Moves $Left$

Input String



- Head starts at the leftmost position of the input string
- Rest of the cells on the tape contain the blank symbol

States & Transitions



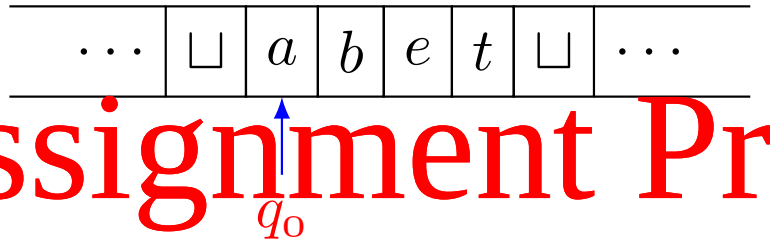
1. Reads a
2. Writes b
3. Moves $Right$
4. Transitions from q_i to q_j

1. Reads a
2. Writes b
3. Moves $Left$
4. Transitions from q_i to q_j

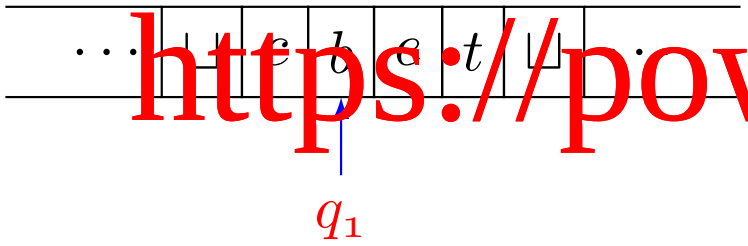
Transitions & Configurations

Example1

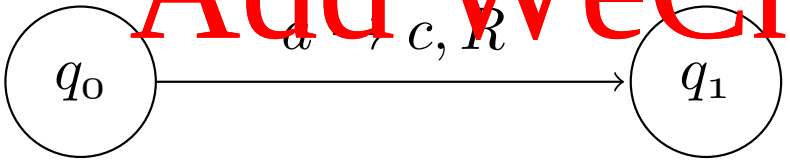
Time 0



Time 1



Transition



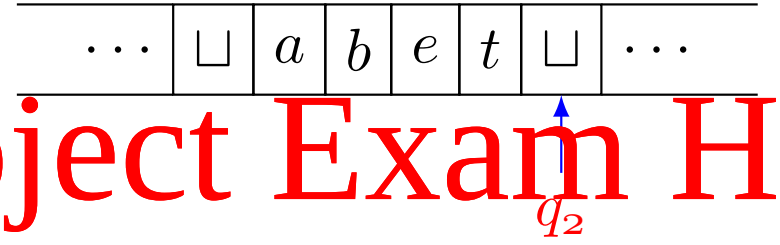
Configurations

Time 0 q_0abet

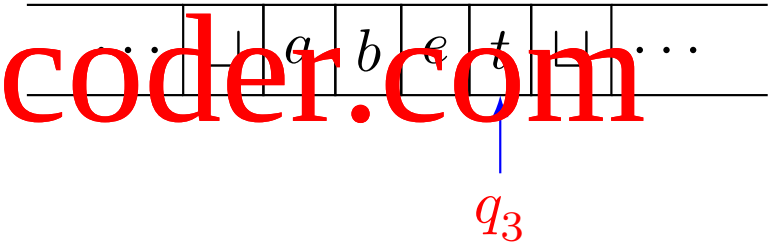
Time 1 cq_1bet

Example2

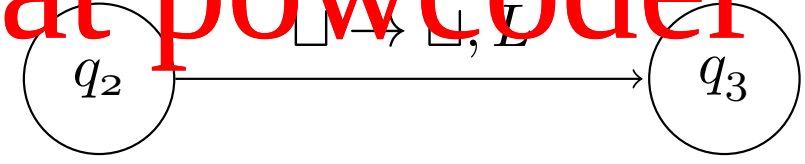
Time 5



Time 6



Transition



Configurations

Time 5 $abetq_2$

Time 6 $abeq_3t$

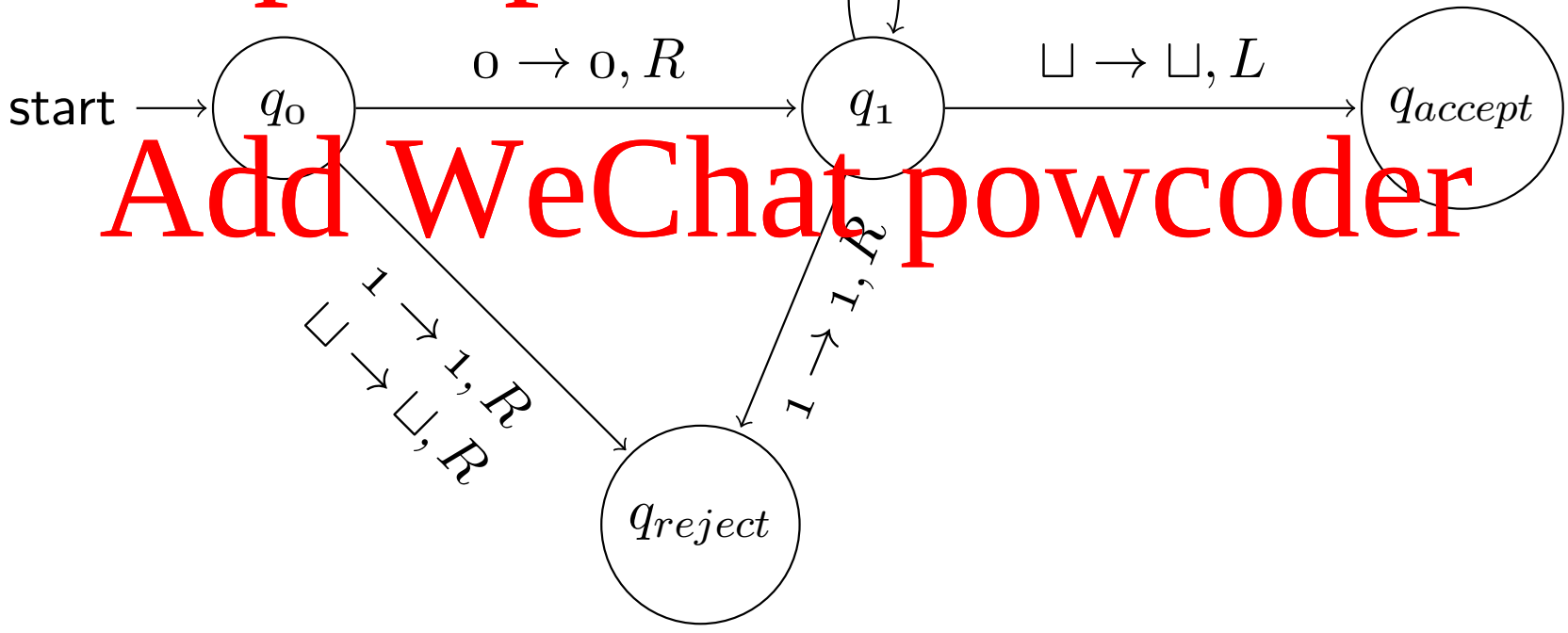
Halting and Acceptance

- There are two special states — q_{accept} and q_{reject}. These are final states, i.e., there are no transitions going out of them. A Turing machine halts if and only if it reaches a final state.
- If the machine halts in q_{accept}, the machine accepts the input.
- If the machine halts in q_{reject} or the machine enters an infinite loop (aka, never reaches a final state), the machine rejects the input.
- The set $C(M) = \{w \mid M \text{ accepts } w\}$ is called the language recognized by the Turing Machine M.

Turing Machine Example

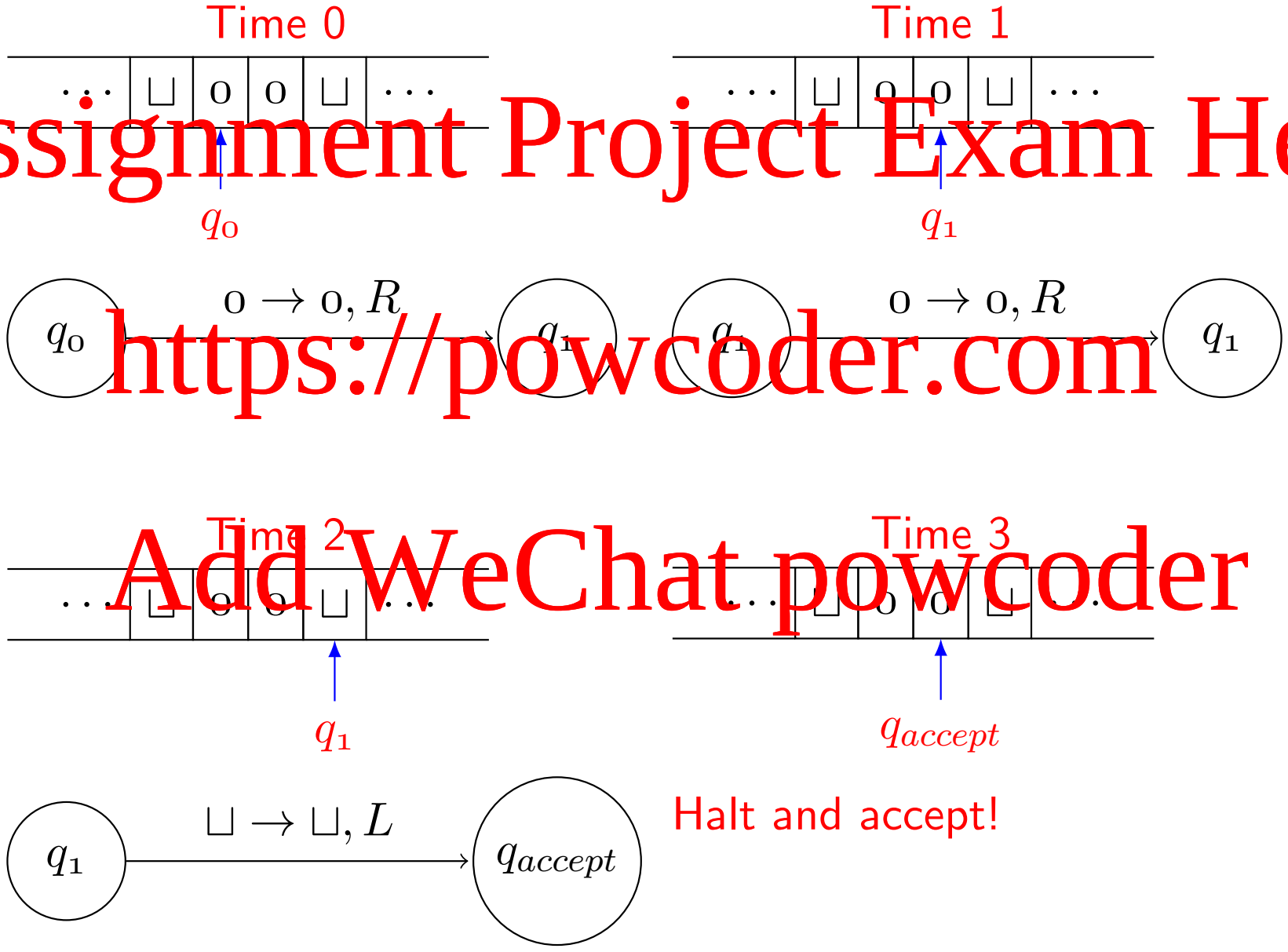
Goal : Describe a TM on the alphabet $\{0, 1\}$ that recognizes the language 0^+.

$$\Sigma = \{0, 1\}$$
$$\Gamma = \{0, 1, \sqcup\}$$



Accept Example

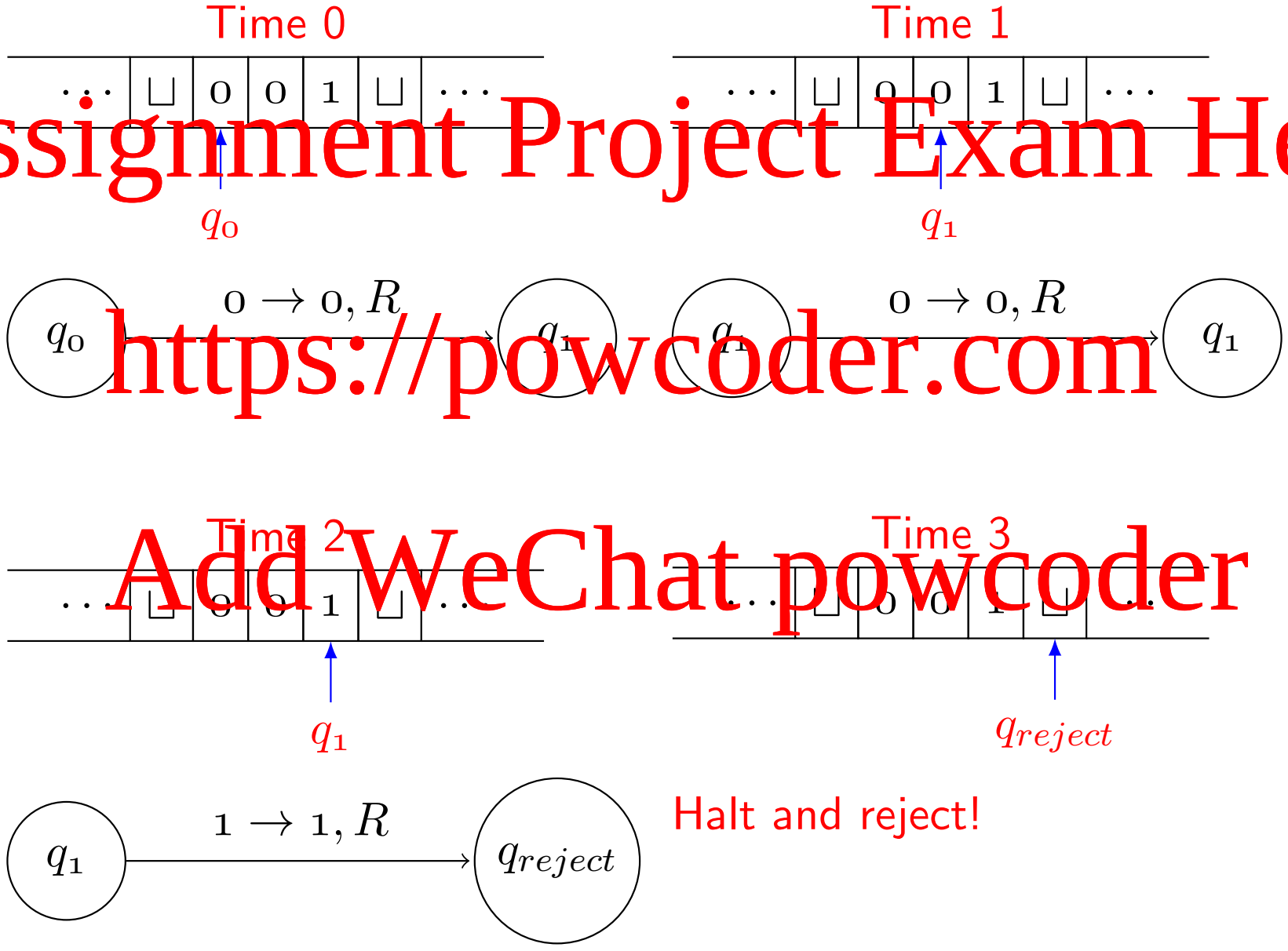
Input : 00



Halt and accept!

Reject Example

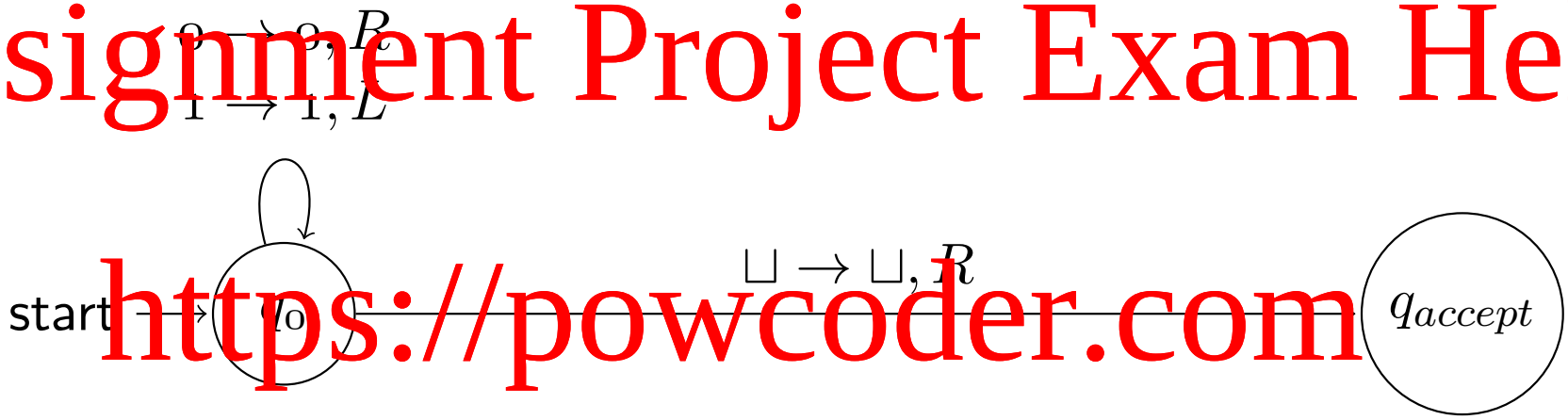
Input : 001



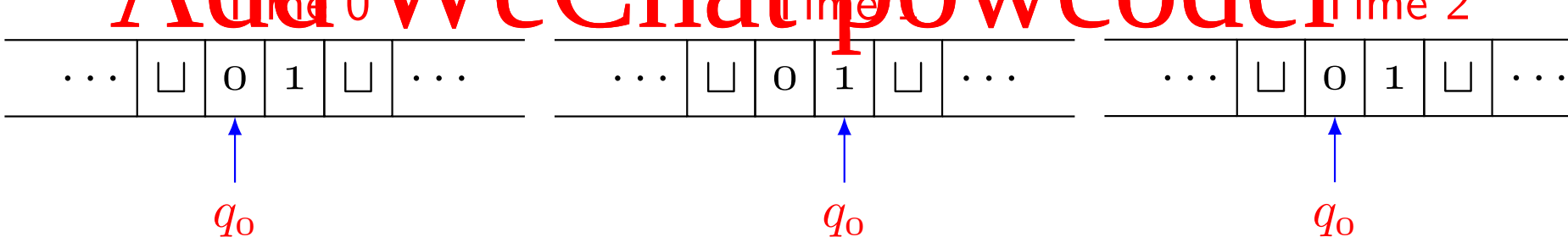
Halt and reject!

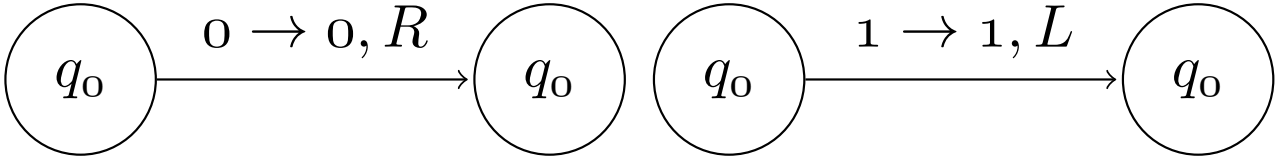
Infinite Loop Example

$$\Sigma = \{0, 1\}, \qquad \Gamma = \{0, 1, \sqcup\}$$



Input : 01





Infinite Loop!

Formal Definition

Definition (Turing Machine)

A Turing machine is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where Q, Σ, Γ are all finite sets and

- Q is the set of states,
- Σ is the input alphabet not containing the blank symbol $\sqcup$,
- Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is the transition function,
- $q_0 \in Q$ is the start state,
- $q_{accept} \in Q$ is the accept state, and
- $q_{reject} \in Q$ is the reject state, where $q_{reject} \neq q_{accept}$.

TM vs FSA

The following are the primary differences between a Turing Machine and a Finite State Automaton:

- TM can read as well as write symbols.
- TM has an infinite tape.
- Head can move left or right, thereby allowing a TM to have no limitation on access to input.
- Special accept and reject states that stop computation immediately.

One more example - PAL

Goal : Describe a TM on the alphabet $\{0, 1\}$ for the languauge

$PAL = \{\text{set of even engtn palindromes}\} = \{yy^R \mid y \in \{0, 1\}^*\}$.

One more example - PAL