Automata Formal Languages & Logic

Preet Kanwal

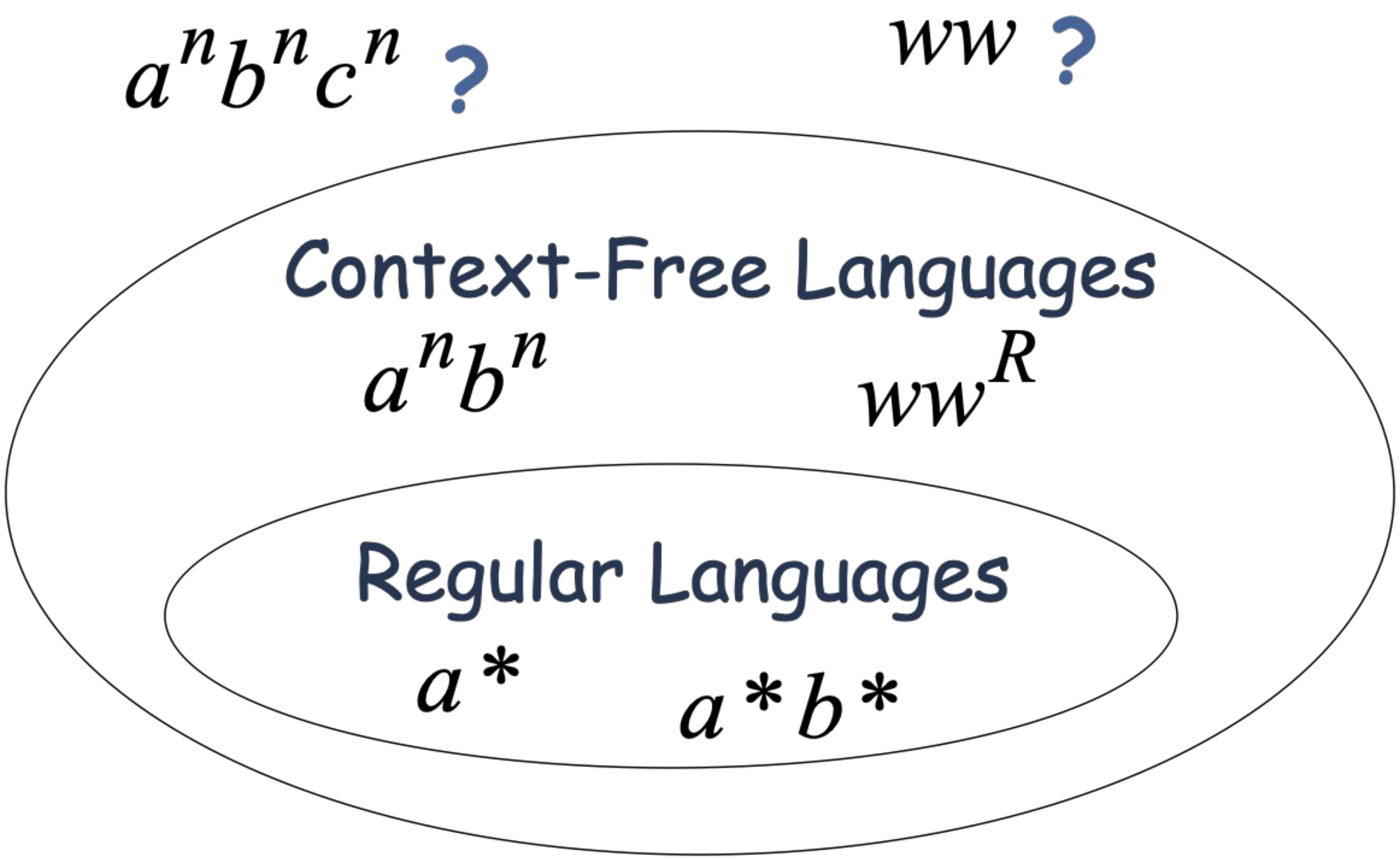
Department of Computer Science & Engineering

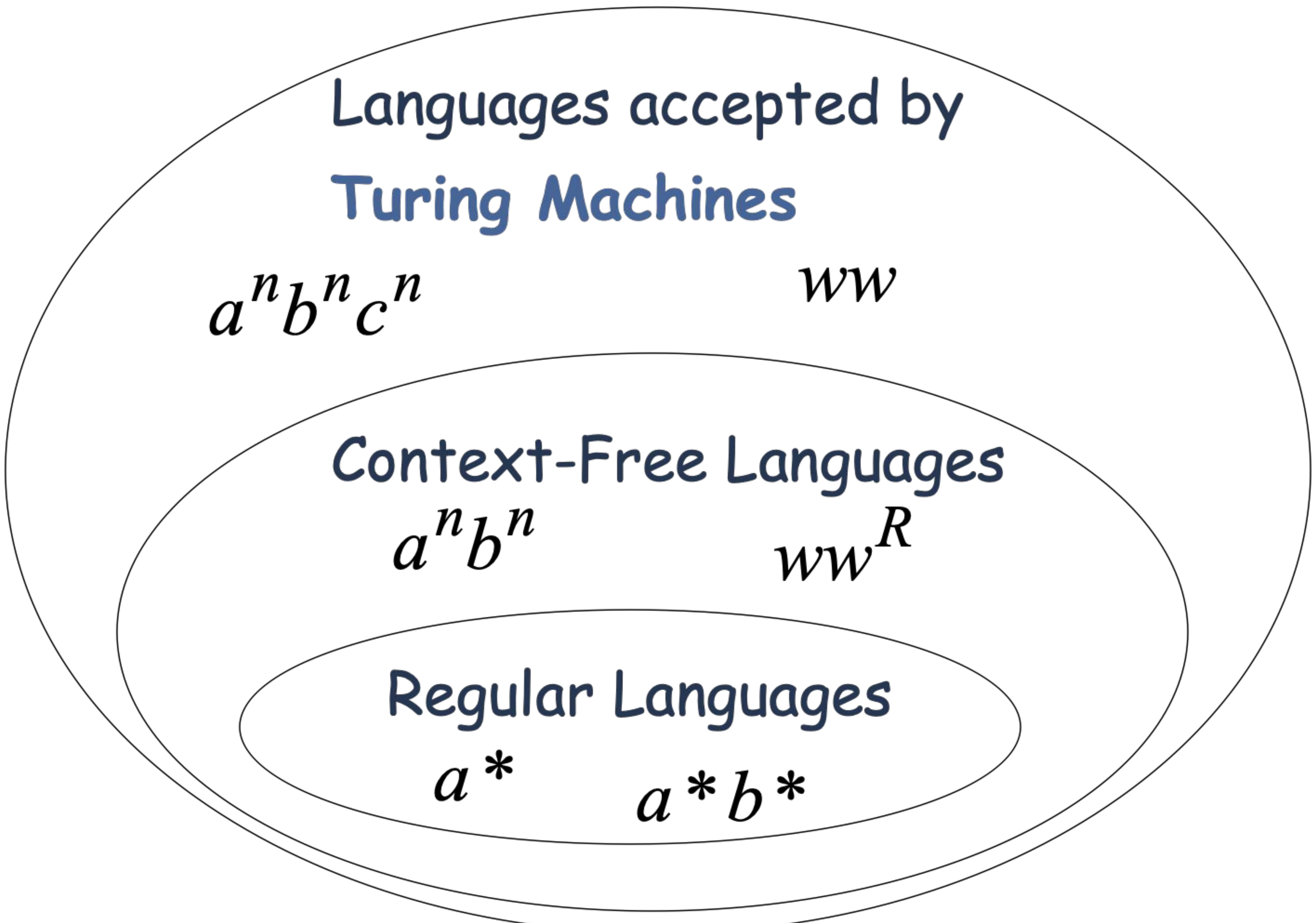
Automata Formal Languages & Logic

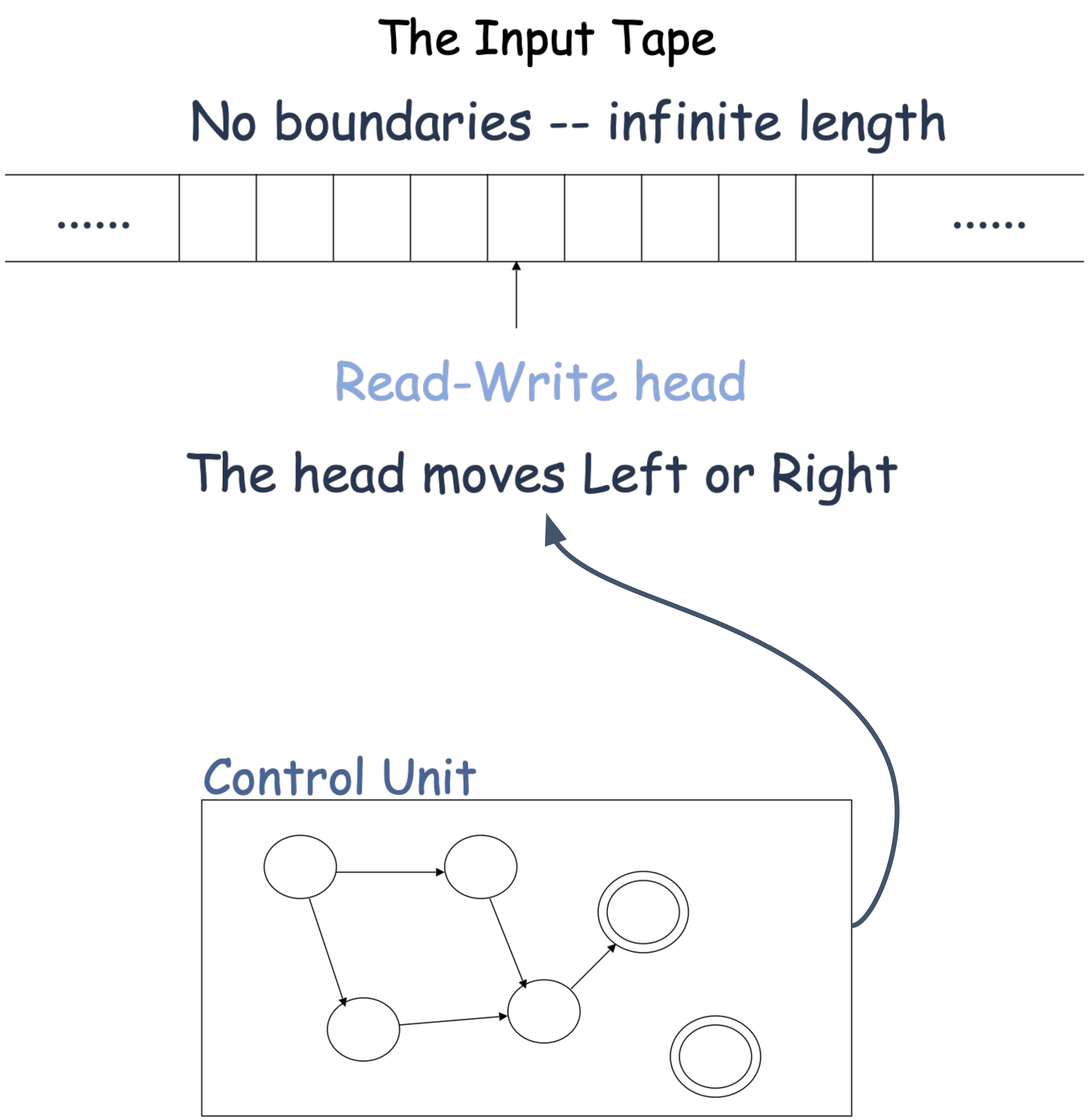
Unit 4 - Turing Machines

Preet Kanwal

Department of Computer Science & Engineering


$a^n b^n c^n$?
ww ?
Context-Free Languages
$a^n b^n$
ww^R
Regular Languages
a^*
$a^* b^*$


Languages accepted by Turing Machines
$a^n b^n c^n$
ww
Context-Free Languages
$a^n b^n$
ww^R
Regular Languages
a^*
$a^* b^*$



A Standard Turing Machine has the following components:

1. Infinite Input tape - unbounded in both directions
2. Control Unit
3. R/W head - changes position in each move

- By convention, input string is preloaded on to the tape.
- Special markers can be used to indicate start and end of the input string.

Turing Machine M, is a 7-tuple :

States

Input alphabet

Tape alphabet

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$$

Transition function

Initial state

blank

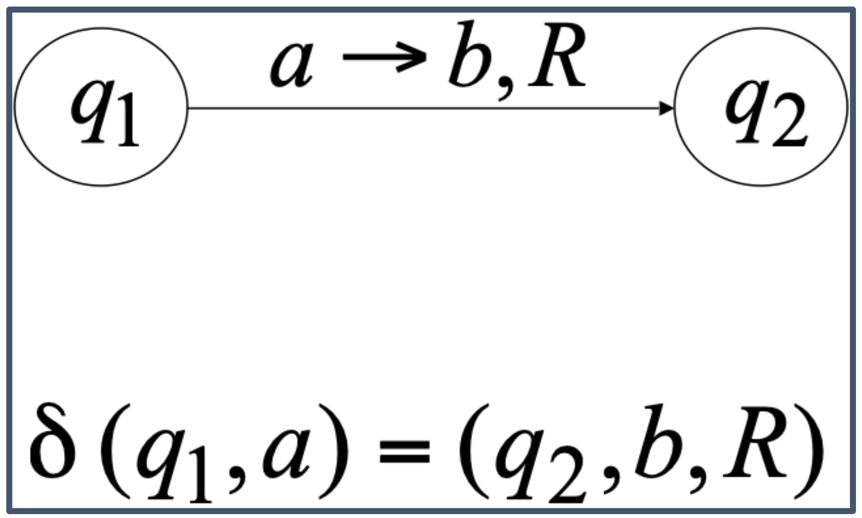
Accept states

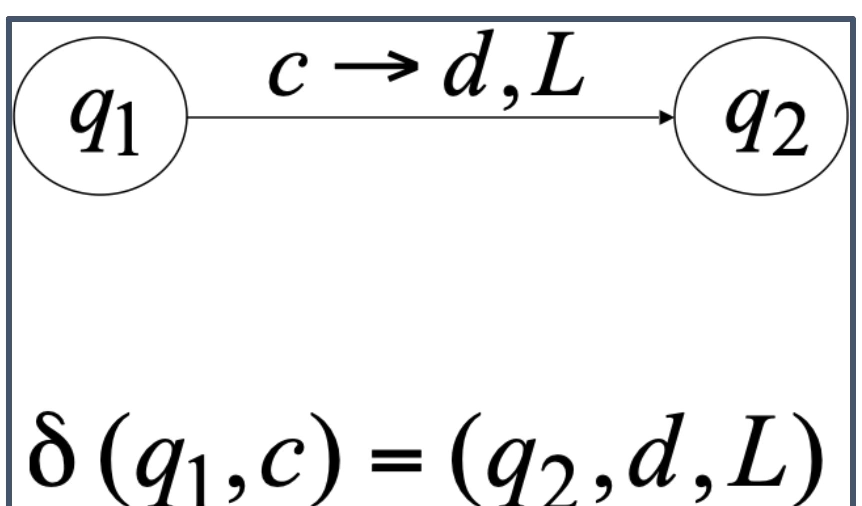
$$\Sigma \subseteq \Gamma$$

$$\Box \in \Gamma$$

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

Unit 4 - Transition Function

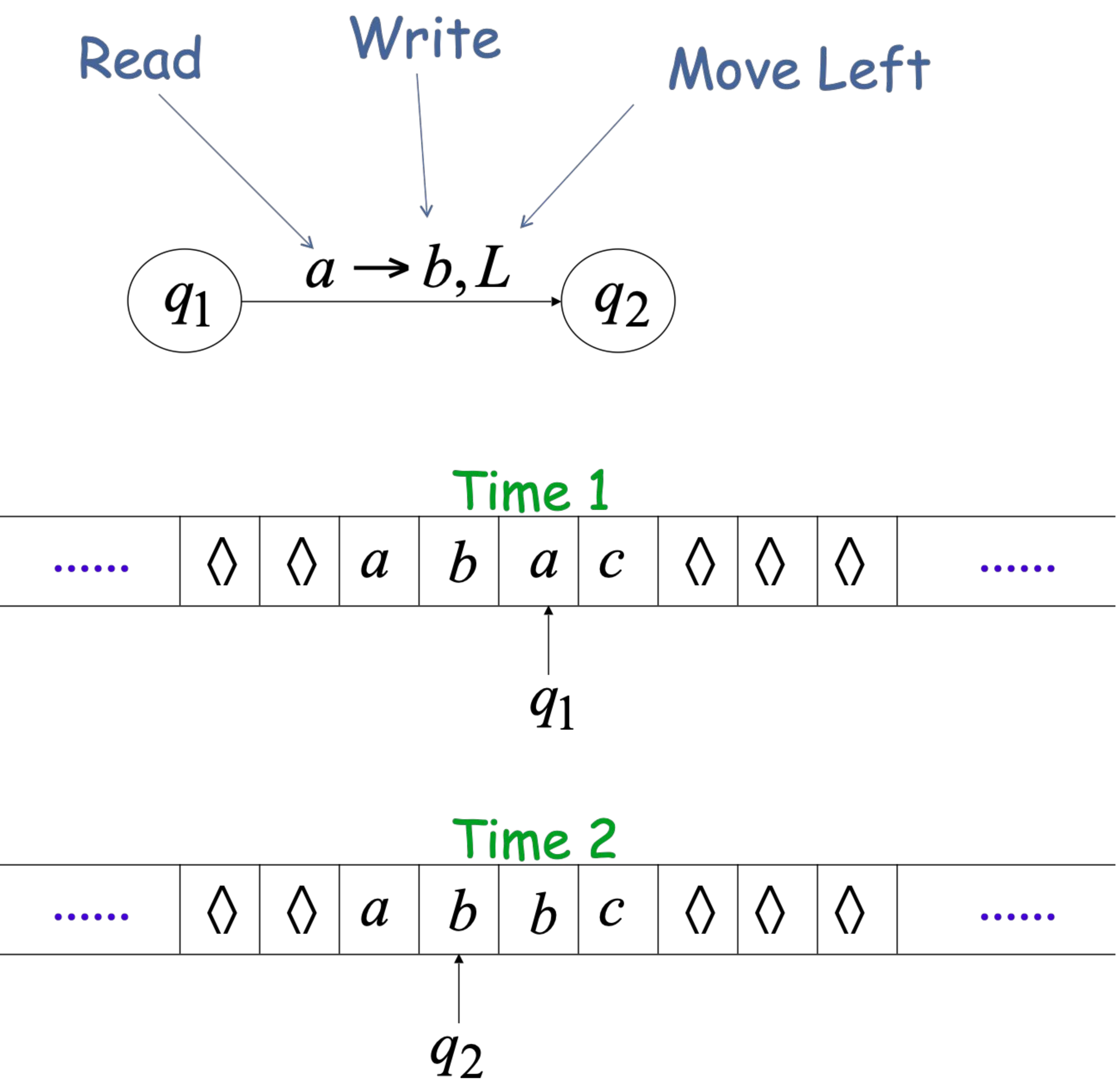




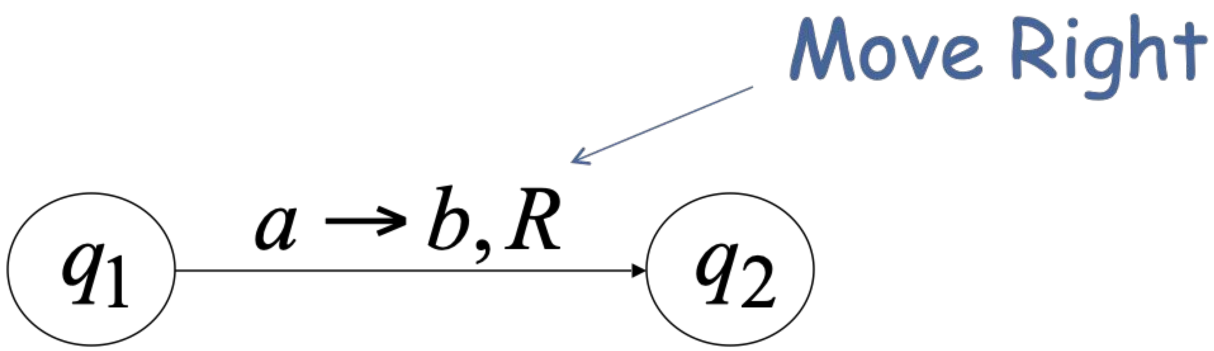
Note:
Textbook specifies the transition as:
a,a,R

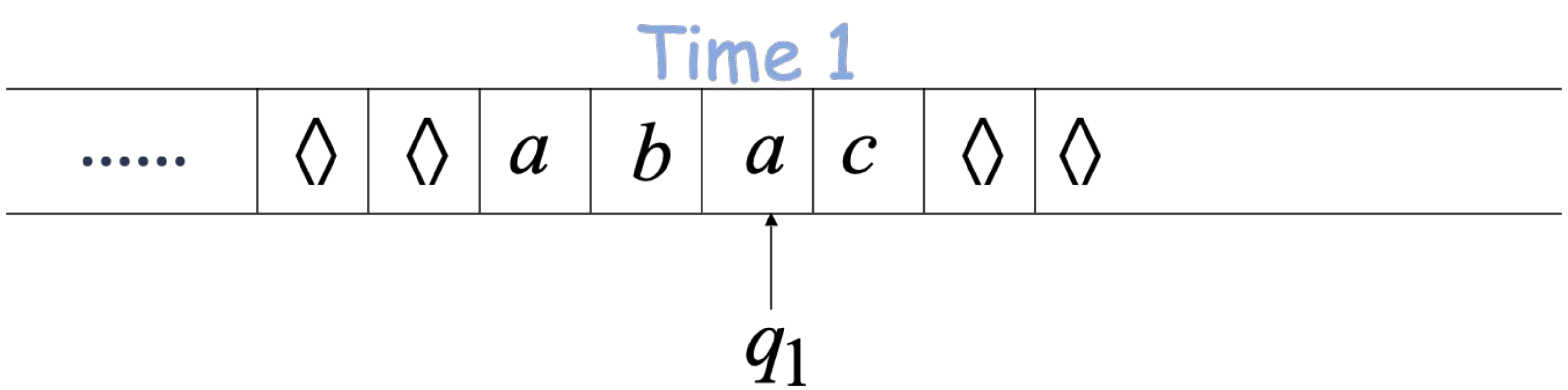
Other accepted notations:
a,a|R
or
a|a|R
or
a|a,R

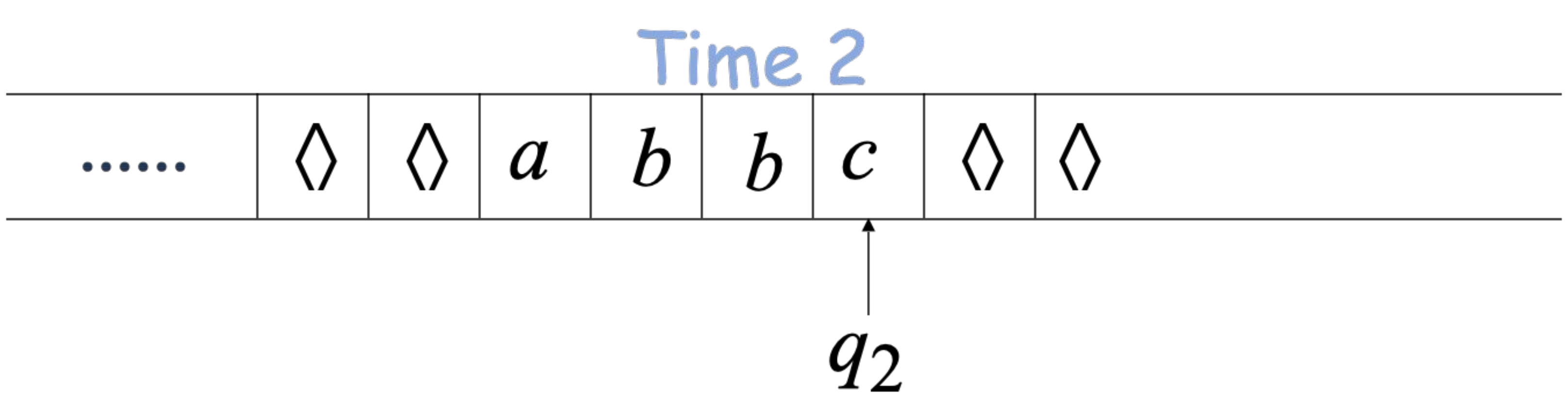
Unit 4 - Transition in a Turing Machine



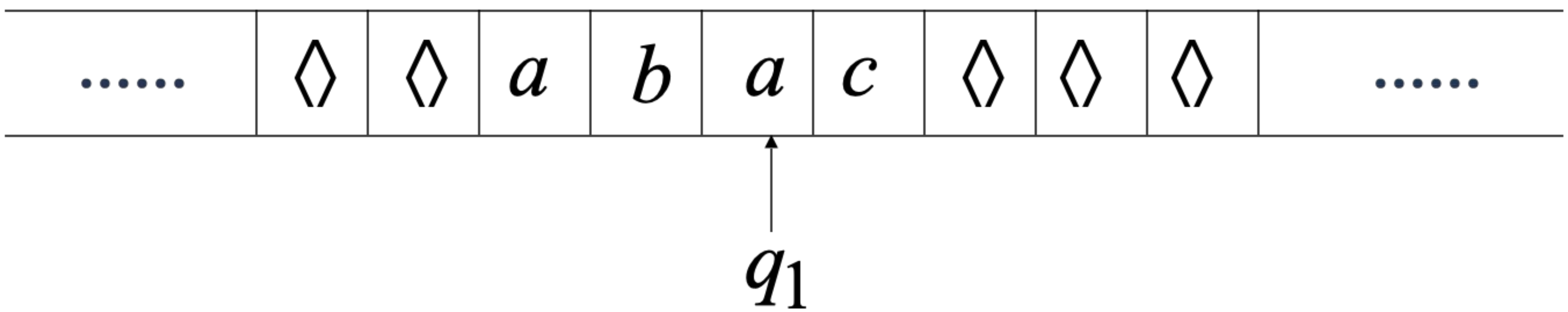
Unit 4 - Transition in a Turing Machine

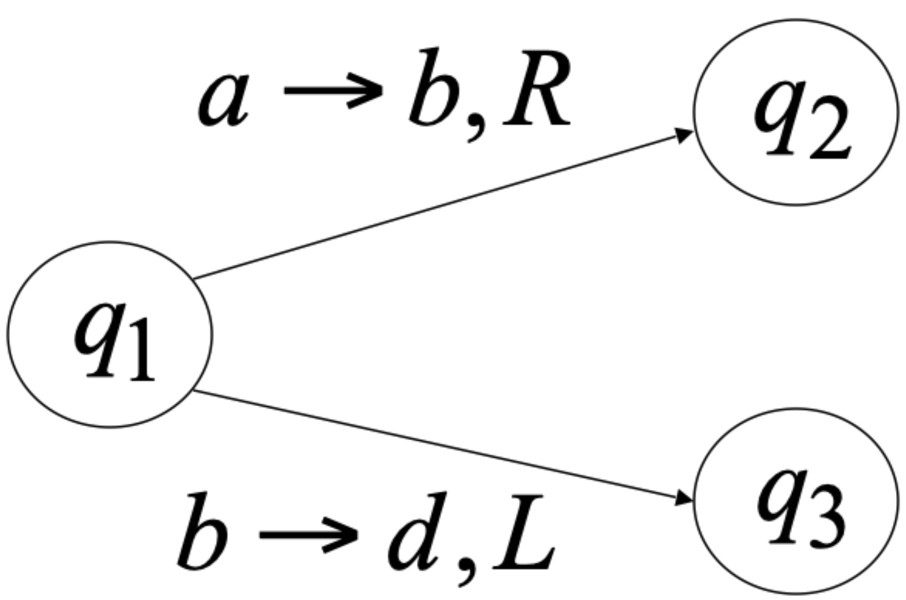






Unit 4 - Partial Transition Function





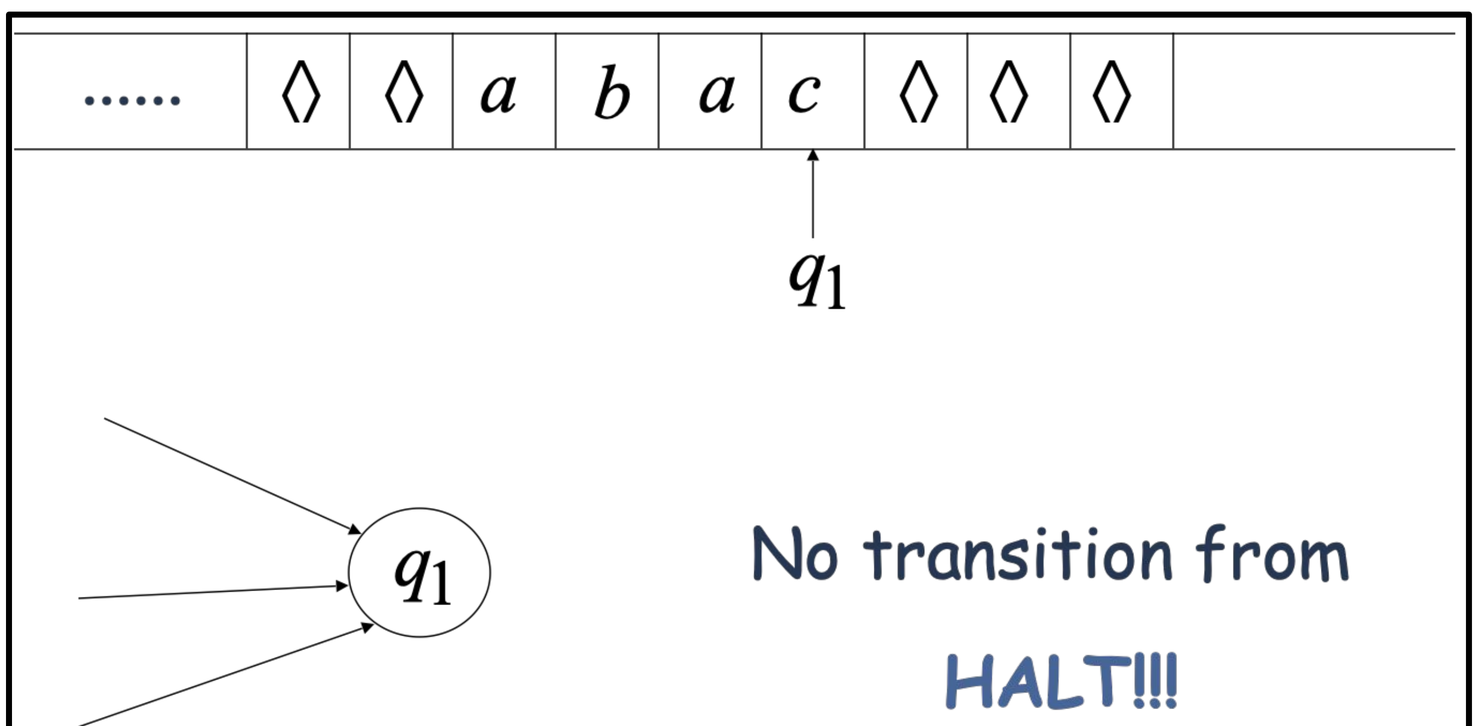
Allowed:

No transition

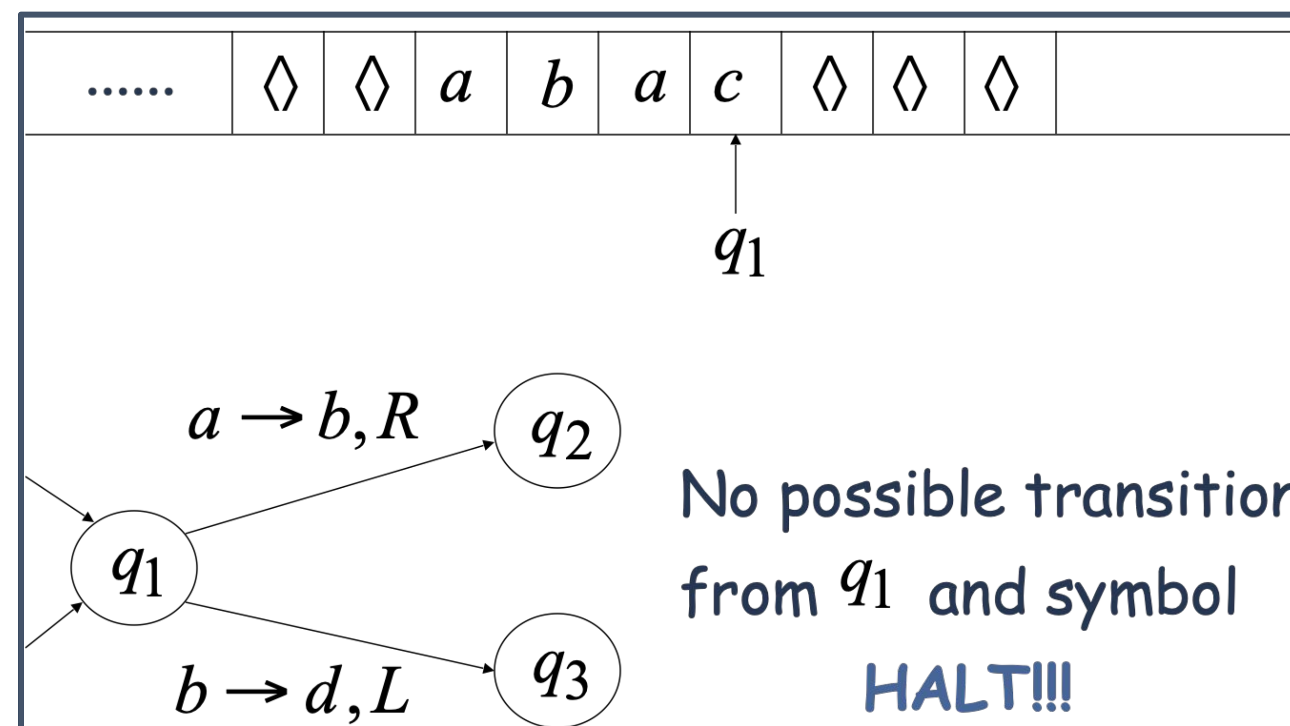
for input symbol c

The machine halts in a state if there is no transition to follow.

For example :



or



Unit 4 - Acceptance in Turing Machine

- **Accepting states have no outgoing transitions.**
- **The machine halts and accepts.**

Allowed

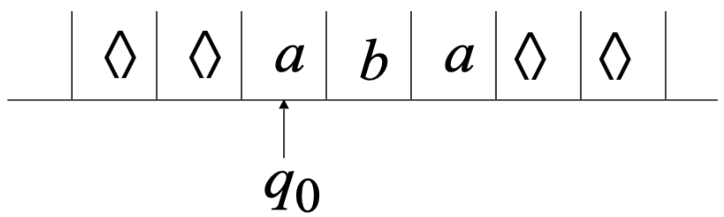
Not Allowed

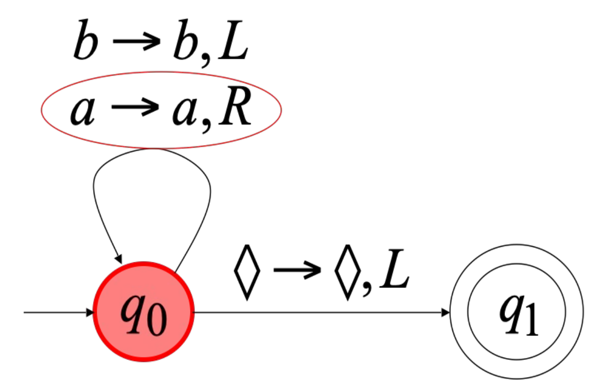
Accept Input string → If machine halts in an accept state

Reject Input string → If machine halts in a non-accept state
or
If machine enters an infinite loop

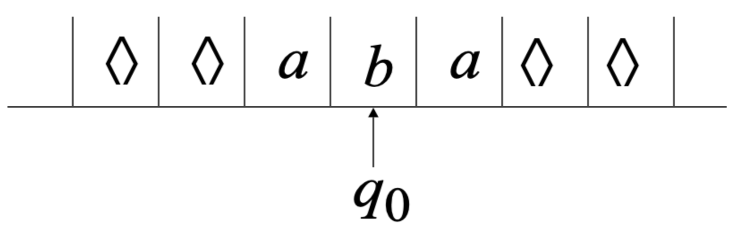
Unit 4 - Infinite Loop Example

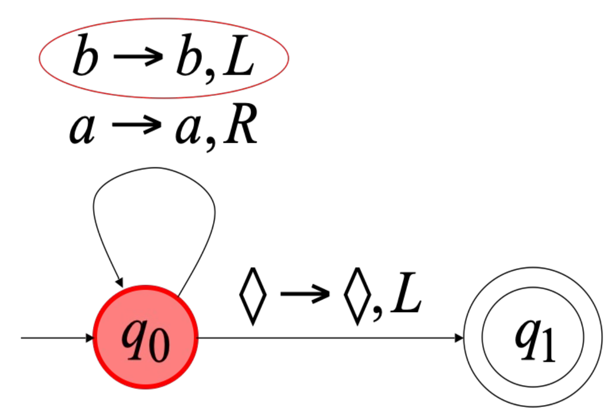
Time 0



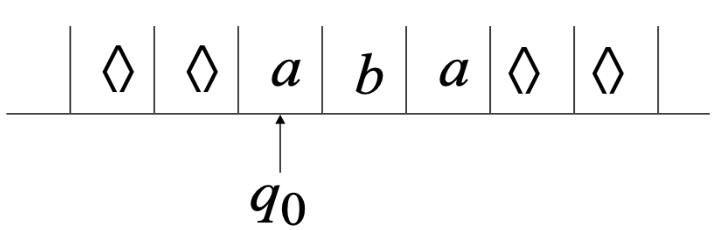


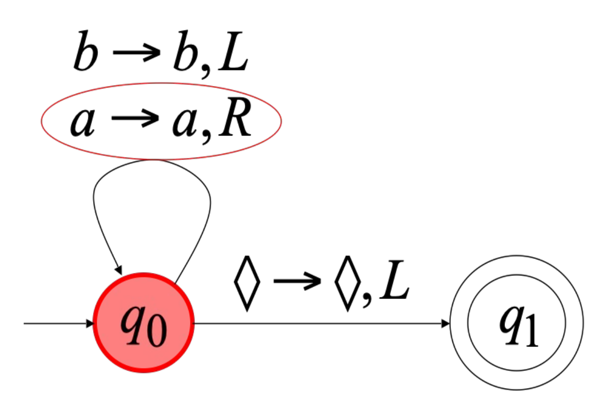
Time 1





Time 2

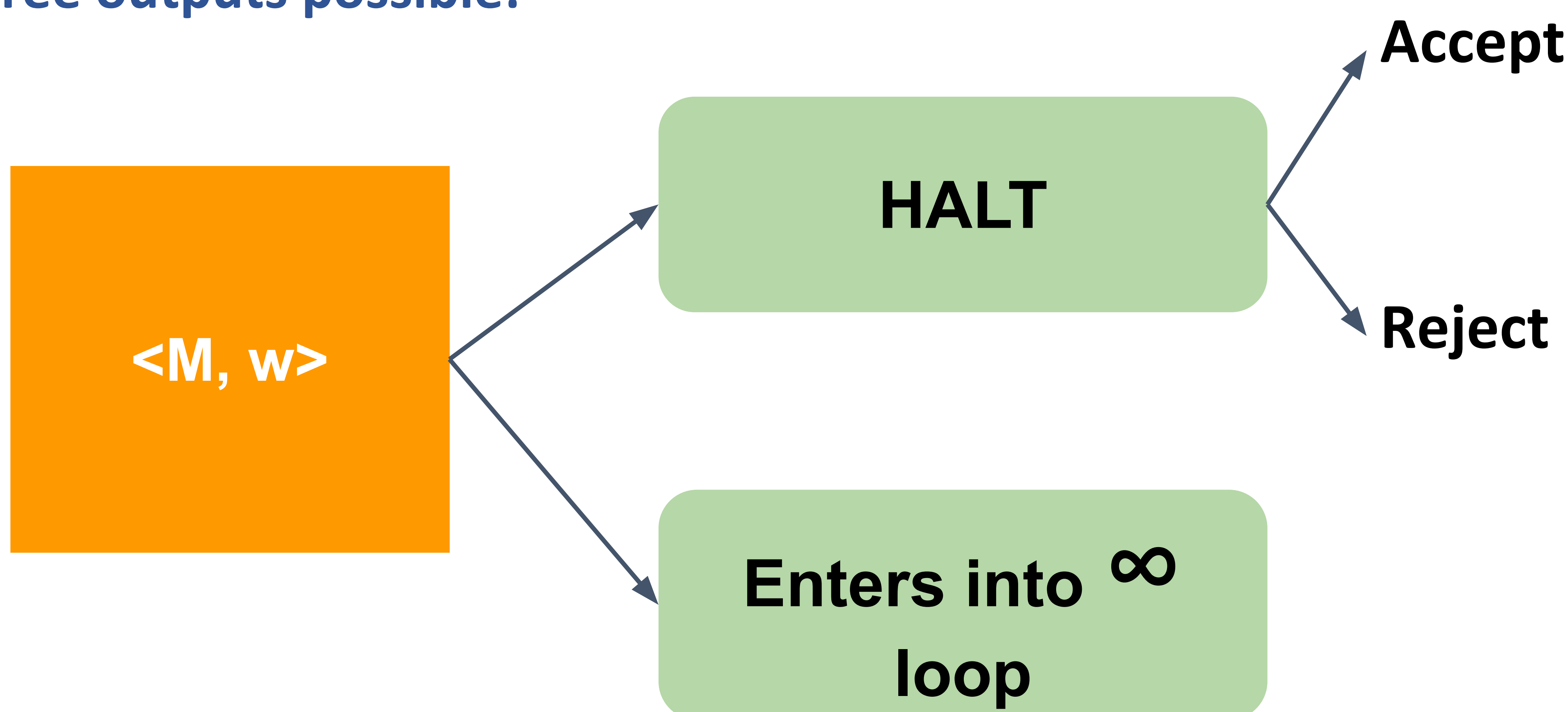




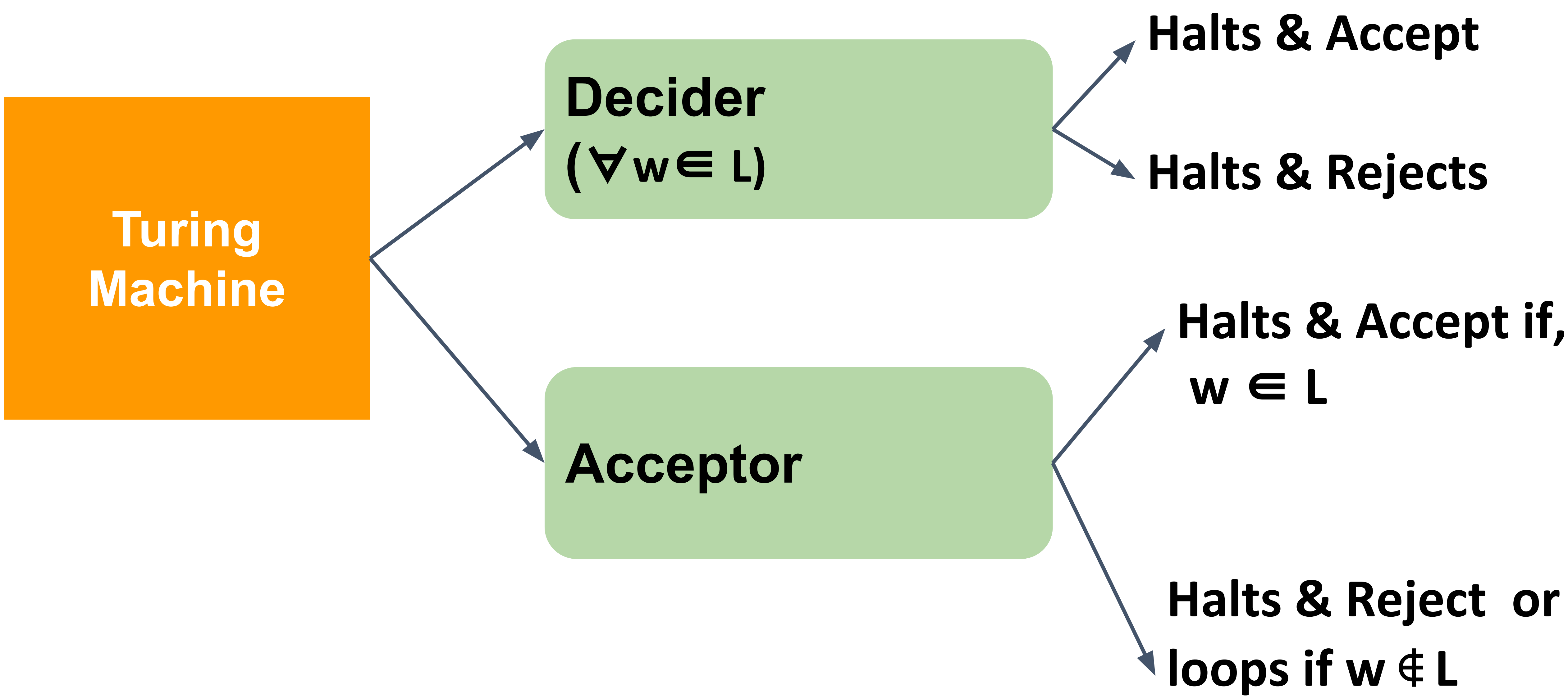
Because of the infinite loop:

- The accepting state cannot be reached
- The machine never halts
- The input string is rejected

Given a Turing Machine M and a string w, when we run w on M, there are three outputs possible:



Given a Turing Machine M and a string w, when we run w on M, there are three outputs possible:



Language recognized by a Turing Machine is called :

Turing Recognizable Language

or

Turing Acceptable Language

or

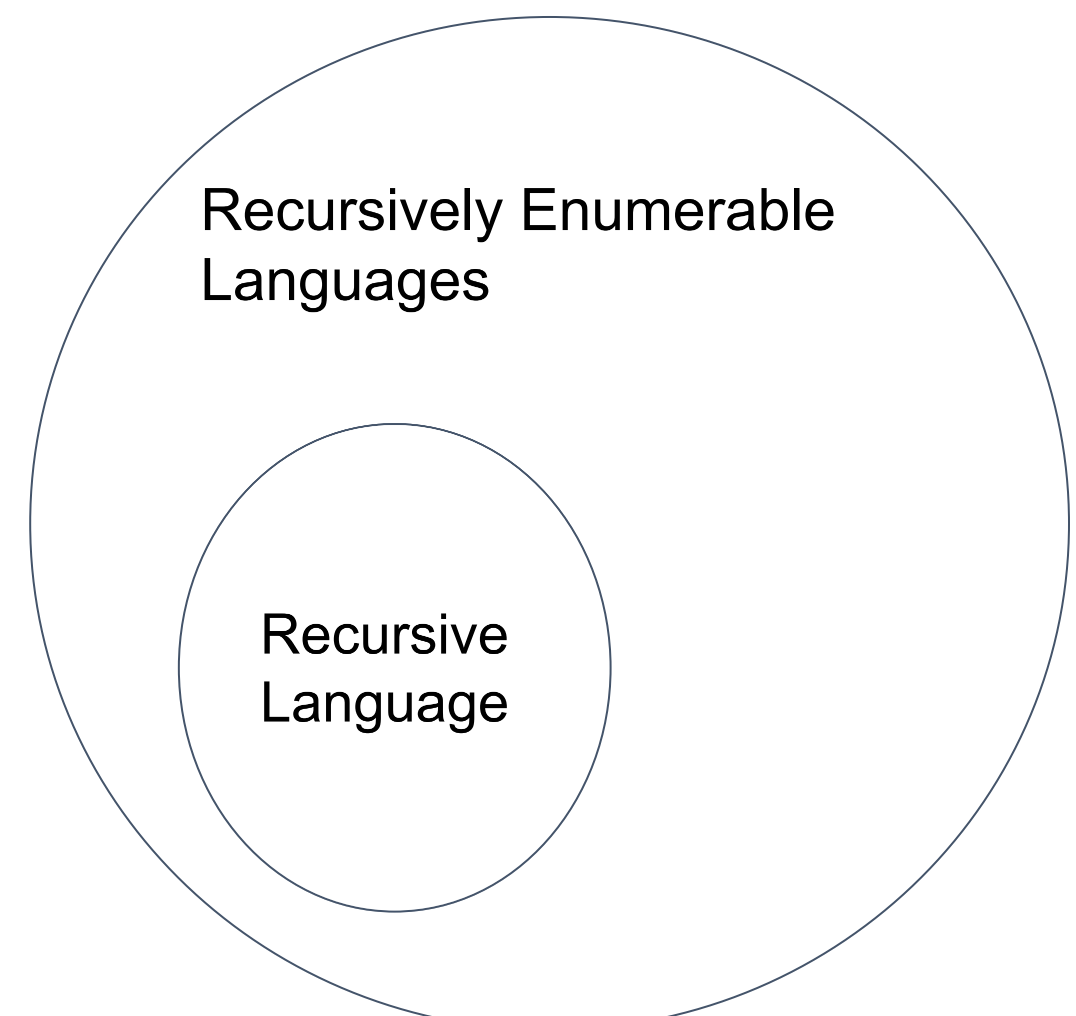
Recursively enumerable Language

Turing Machine Decider:

Turing Machines that halt on all inputs.

Language recognized by a Turing Machine Decider is called Recursive Language.

Recursive languages are closed under Complement.



Computation :

- It is a series of moves a Turing Machine makes given an input until it halts in some configuration.

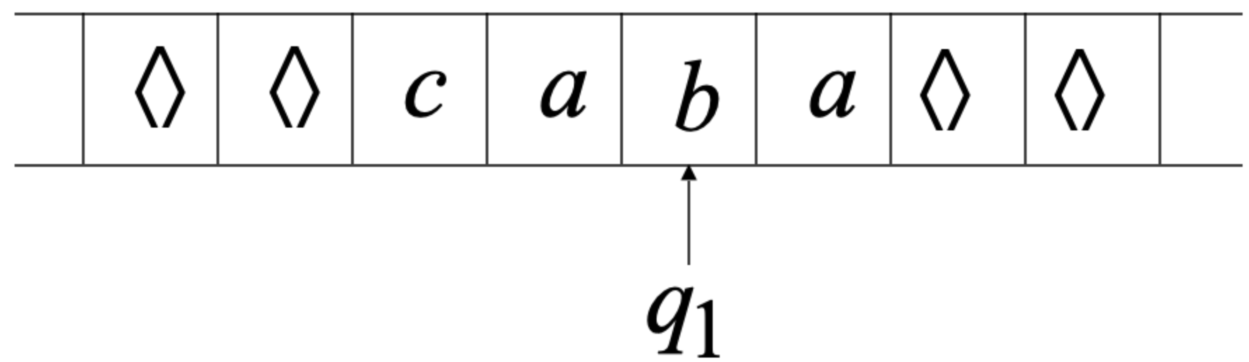
Computable Function :

- A function f is computable if we can construct a Turing Machine Decider for that function.

For example: Following functions are computable:

1. f(x) = 2x
2. f(x,y) = x + y
3. $$f(x,y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x \leq y \end{cases}$$

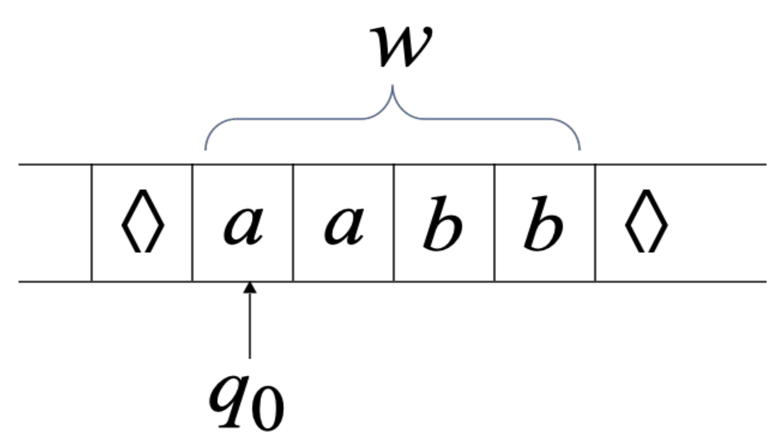
Current Configuration



Instantaneous description: $ca\ q_1\ ba$

Initial configuration: $q_0 \, w$

Input string



Final Configuration : **w q_f**

where q_f is the final state

Unit 4 - Examples

Construct Turing Machine for the following examples:

1. $a^n b^n$, n>=1. Trace the input string : aabb
2. $0^n 1^n 2^n$, n>=1
3. Given input : $a^n b^m c^k$ such that k=n-m. Write the difference(in terms of no. of c's) on the tape. If the difference is negative append a -ve sign. For example if k=3 write ccc on the tape. If k=-2, write -cc on the tape. (Language of subtraction). Input and output must be separated by a ☐.
4. Given input : $0^m 1 0^n$, write 0^{m*n} as output on the tape (Language of Unary multiplication). Input and output must be separated by a ☐.
5. 0^{n^2}, n>=1
6. ww, w $\in \{a,b\}^+$

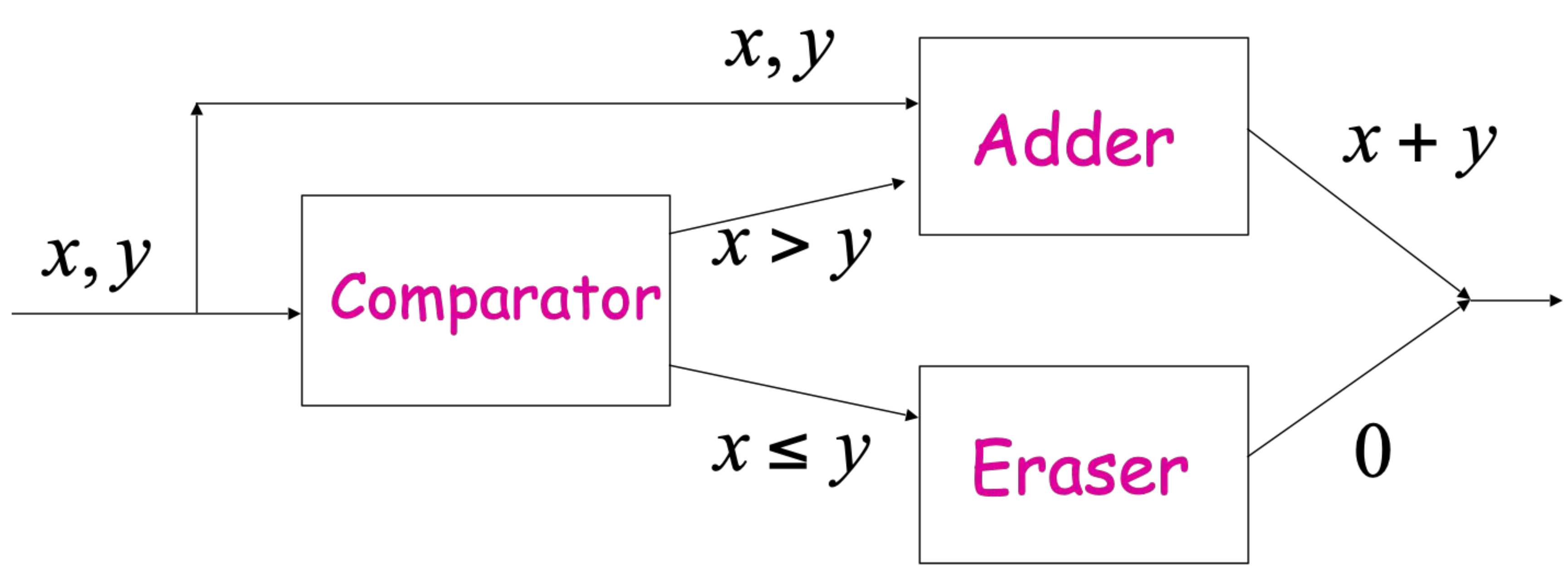
PES UNIVERSITY ONLINE

Construct Turing Machine to recognize the following language:

1. $a^n b^m c^n d^m$, n>=1
2. Given input : $0^m 1 0^n$, write 0^{m+n} as output on the tape (Language of addition), m,n>=1
3. $0^n 1^{n\text{^}2}$, n>=1
4. Given a string w $\in \{a,b\}^+$, write w as output on the tape separated by a blank. Final contents on the tape must look like : w□w (Copy operation)
5. Given a string w$\in \{a,b\}^+$, sort the symbols in the input string. For example if the input is babaa, output should be : aaabb. Final contents on the tape must look like: babaa□aaabb
6. Given a string w$\in \{a,b\}^+$, write w^R as the output on the tape (Reversing a string).Final contents on the tape must look like : w□w^R

Example:

$$f(x,y) = \begin{cases} x+y & \text{if} \quad x > y \\ 0 & \text{if} \quad x \leq y \end{cases}$$



By combining Turing Machines that perform simple tasks, complex algorithms can be implemented.

The Church-Turing Thesis claims that every effective method of computation is either equivalent to or weaker than a Turing machine.

Alan turing came up with the notion that, what can be computed using Turing machine is known as computable.

All variations of the Turing Machine are equivalent in computing capability.

Algorithmically computable means computable by Turing Machine.

Completely different term.

Used in Artificial Intelligence.

It is a test to determine whether a computer or program has same kind of intelligence as that of a human.

Alan turing designed both Turing Machine and Turing Thesis. They are completely different.

THANK YOU

Preet Kanwal

Department of Computer Science & Engineering

preetkanwal@pes.edu

+91 80 6666 3333 Extn 724