Course: Automata Theory

Lecture 9: Models of Computation – Turing Machines (TMs)

Lecturer: Martha Gichuki

Course description

- The course begins with an introduction to logic and formal grammar where learners will do a recap on sets, logic and truth tables, sequences, relations and functions
- A coverage of finite state machines, Push Down automata and Turing Machines (The Church's thesis) will culminate the study of various models of computation.
- Formal language and grammar will then follow to enable learners differentiate regular and context free languages.
- An evaluation of the computability and complexity of practical computational problems which are the foundations of automata theory will then be done and the outcome will be problem description.

Learning outcomes:

Lecture 9: Models of Computation – Turing Machines

At the end of the lecture the learner will be able to:

- Describe Turing Machines
- Describe Turing Machines using State Diagrams
- Formally describe various Turing Machines

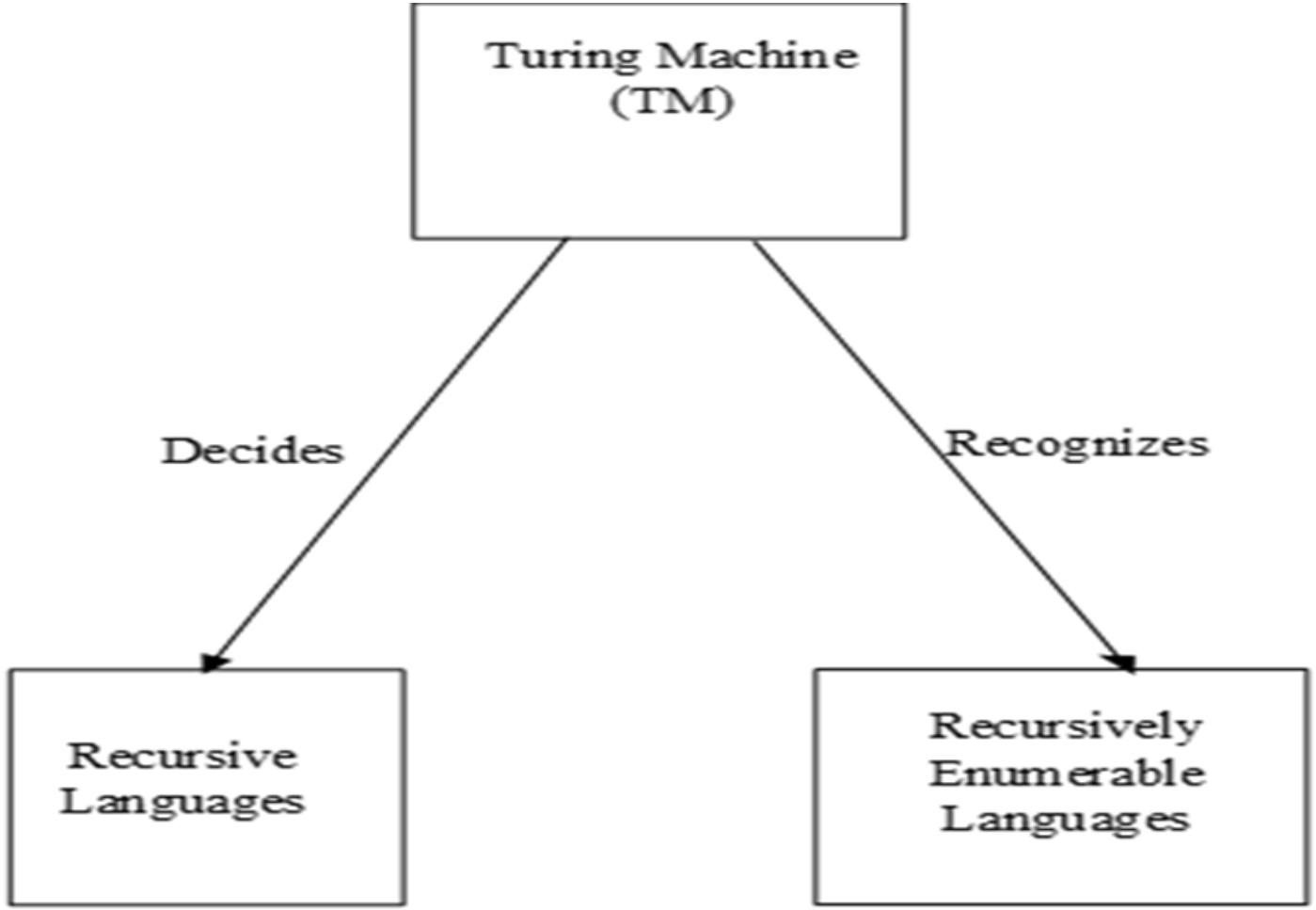
Introduction to Turing Machines (TMs)

- These are the **most powerful computational machines**.
- They possess an **infinite memory** in the form of **a tape**, and **a Read/Write head** which can read and change the tape, while moving in either direction along the tape.

Which language does a Turing Machine (TMs) recognize/decide?

- Turing machines are equivalent to algorithms and are the theoretical basis for modern computers.
- Turing machines decide recursive languages and recognize the recursively enumerable languages.
- Recall that Finite automata recognizes regular languages while Push Down Stack machines recognize Context Free Languages

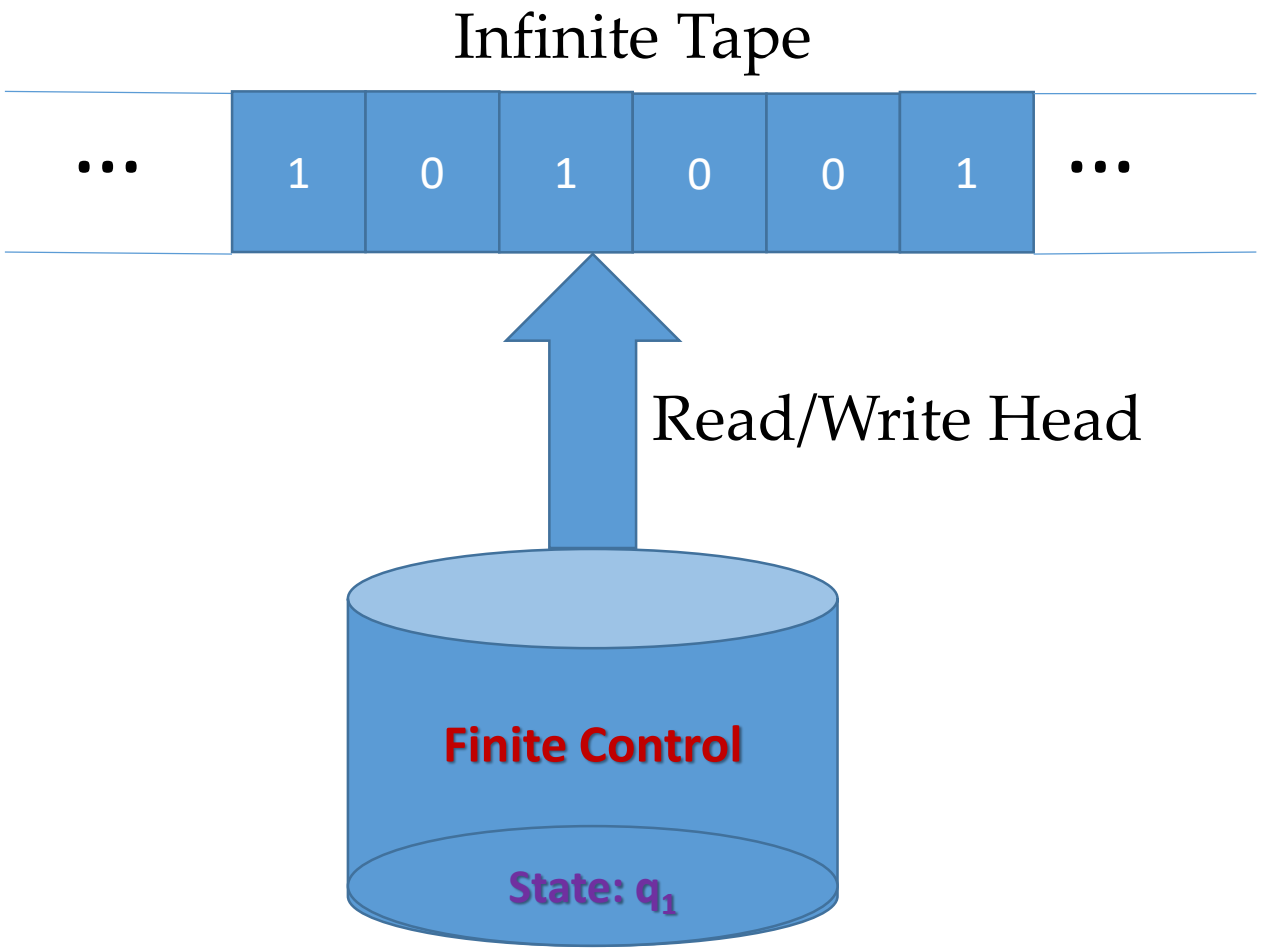
Turing machines decide recurve languages and recognize the recursively enumerable languages.



Turing Machines compared to FA and PDA

- Turing Machines have the essential feature of unrestricted access to memory, distinguishing them from weaker models like Finite Automata and Push Down Automata.
- They possess an **infinite memory in the form of a tape**, and **a Read/Write head which can read and change the tape.**
- The Read/Write head moves in **either direction** along the tape i.e. **Left (L)** or **Right (R)** direction.

Tape Operation



Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 166, 169

Formal Definition of a Turing Machine

A Turing Machine is a **7-tuple** $(Q, \sum, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where

i. Q is a finite set of states

ii. $\sum$ is the input alphabet – blank symbol (∪) not included

iii. Γ is the tape alphabet - blank symbol (∪) included

iv. $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function

Formal Definition of a Turing Machine...

v. $q_0 \in Q$ is the start state

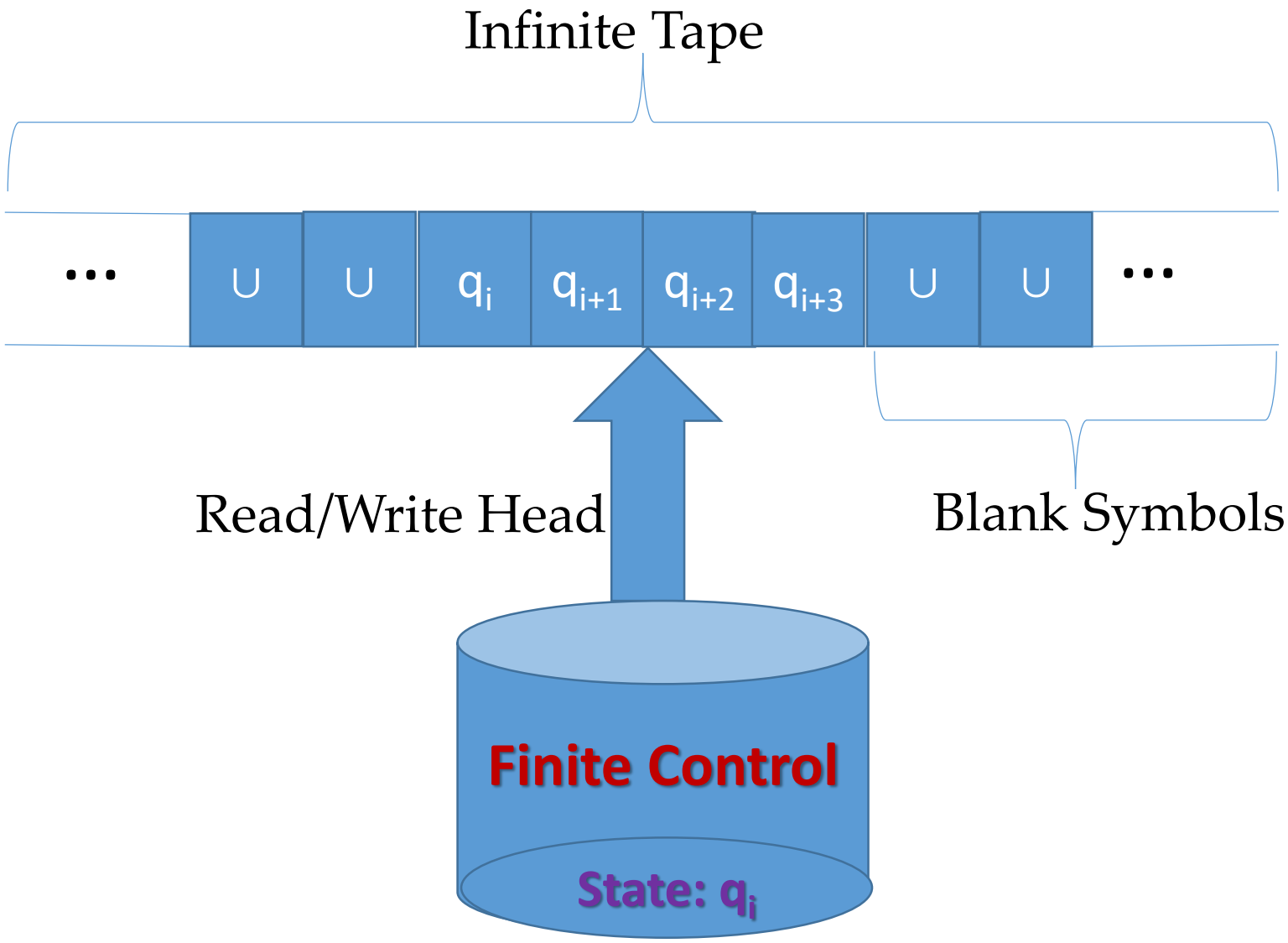
vi. $q_{accept,} \in Q$ is the Accept state, and

vii. $q_{reject} \in Q$ is the Reject state.

The Transition Function

- δ: Q X $\Gamma \rightarrow$ Q X Γ X {L, R} is the transition function whereby while the machine is in State Q, reads a symbol from the tape alphabet, it moves to another state Q, replaces the symbol with another one and the Read/Write head moves either to the Left (L) or to the Right(R)
- Even if the Input alphabet is exhausted, the tape has infinite blank symbols along the tape

Turing Machine Configuration



Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 166, 169

Example 1:

- Given a Turing machine M_6, which recognizes the language consisting of all strings of 0s whose length is a power of 2.
- This machine decides the language:

  $A = \{0^{2^n} \mid n>=0\}$.

- Examples of strings in this language are:
  - ✓ 2^0- 0
  - ✓2^1- 00
  - ✓2^2- 0000
  - ✓2^3 -0000000
  - ✓…2^n

Formal Definition of the Turing Machine M_6

The formal definition of $M_6 = (Q, \sum, \Gamma, \delta, q_1, q_{accept}, q_{reject})$.

Where;

1) Q is a finite set of states (seven states in this case)

- $Q = \{q1, q2, q3, q4, q5, q_{accept}, q_{reject}\}$;

2) $\sum$ is the input alphabet with the blank symbol (∪) not included

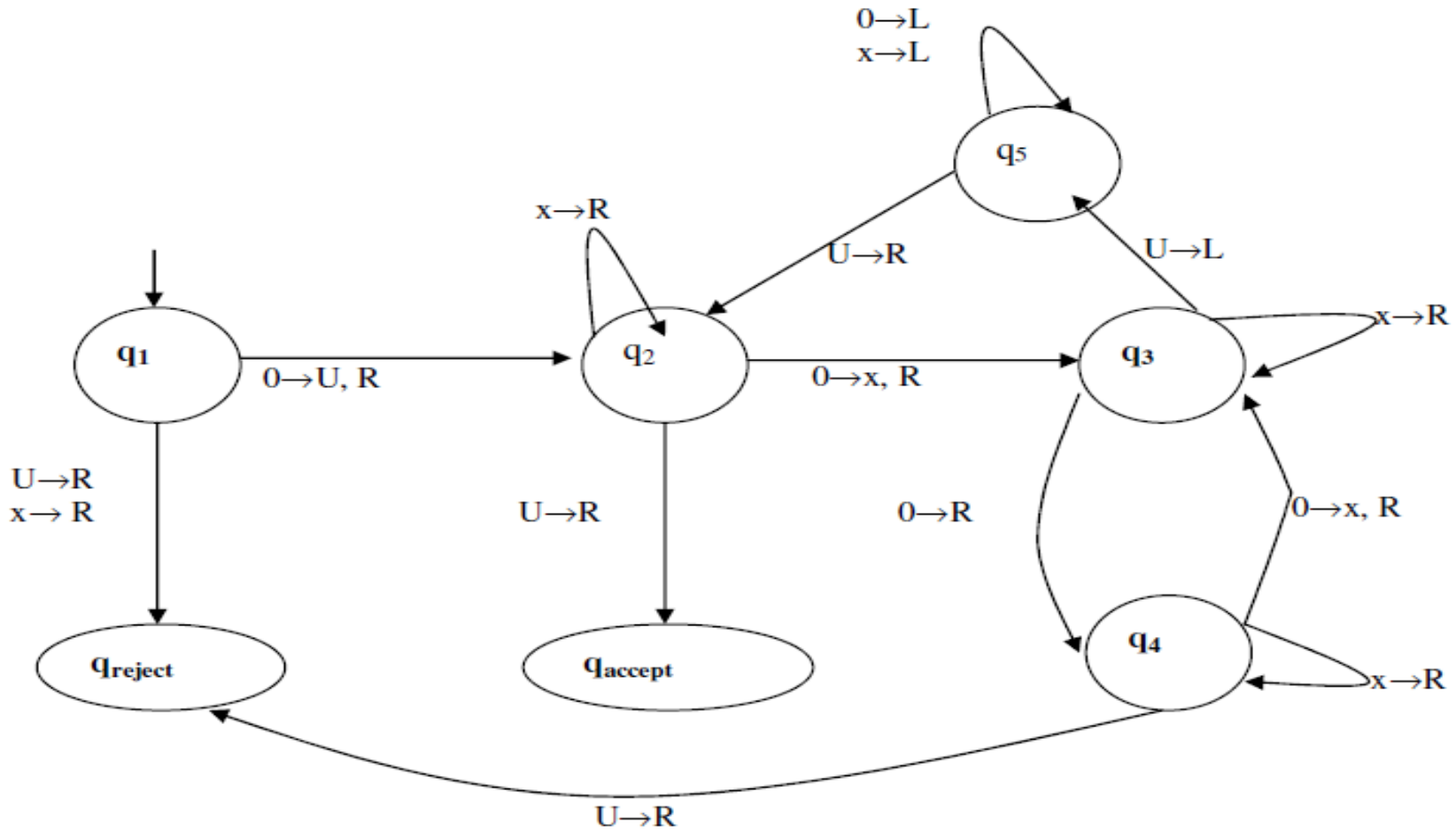
- $\sum = \{0\}$;

Formal Definition of the Turing Machine M_6…

3) Γ is the tape alphabet with the blank symbol (∪) included

- ***Γ = {0, x, U};***
- The read/write head is able to read and write(replace) symbols 0, x, U and replace it with either 0, x, U from the tape alphabet.
- Sometimes replacement does not happen

4) The transition function $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ of the Turing Machine M_6 is described using the state diagram below:-



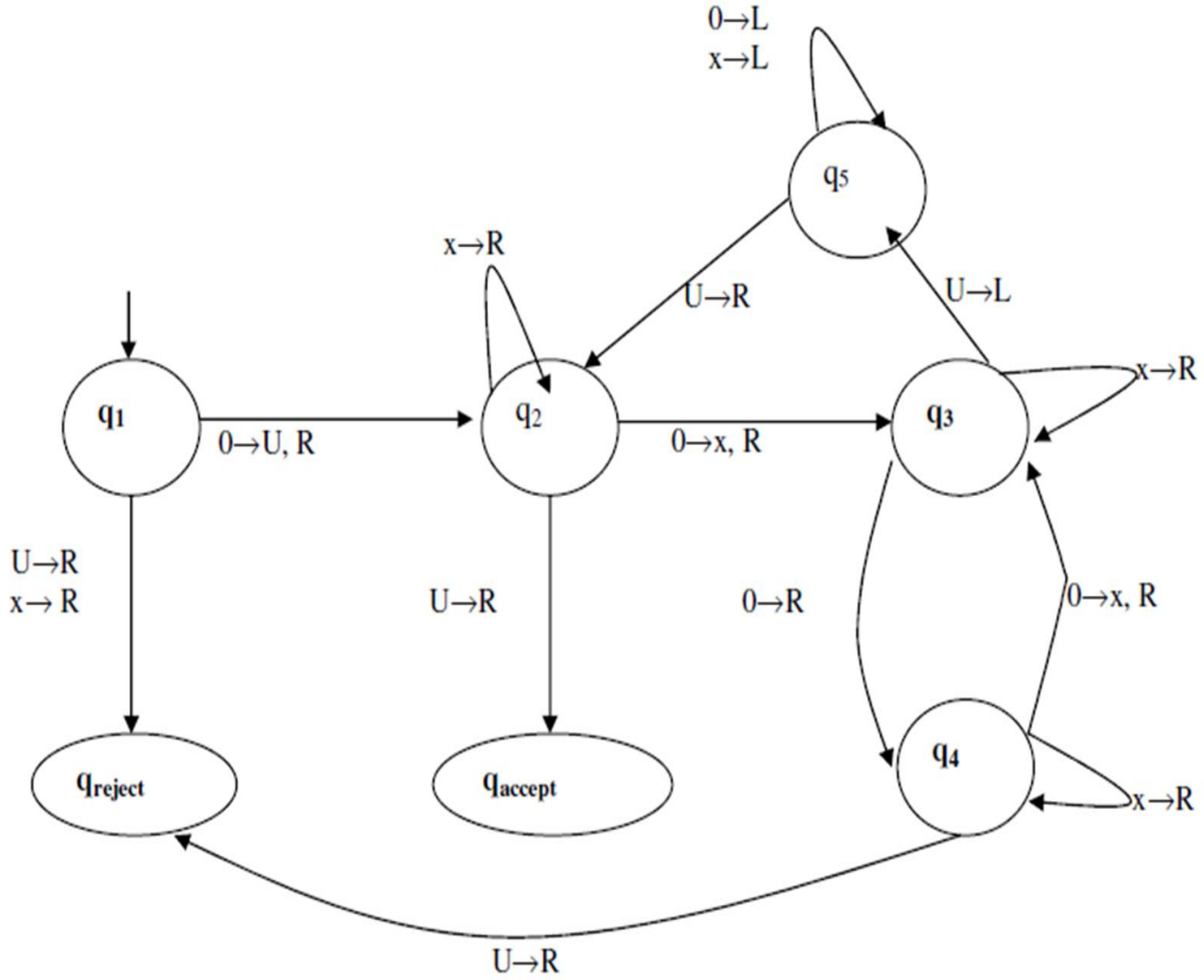
Turing Machine M1

Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 172.

Formal Definition of the Turing Machine M_6…

5) $q_1 \in Q$ is the Start state
6) $q_{accept} \in Q$ is the Accept state, and
7) $q_{reject} \in Q$ is the Reject state

Show that the language 00 is accepted by machine M_6

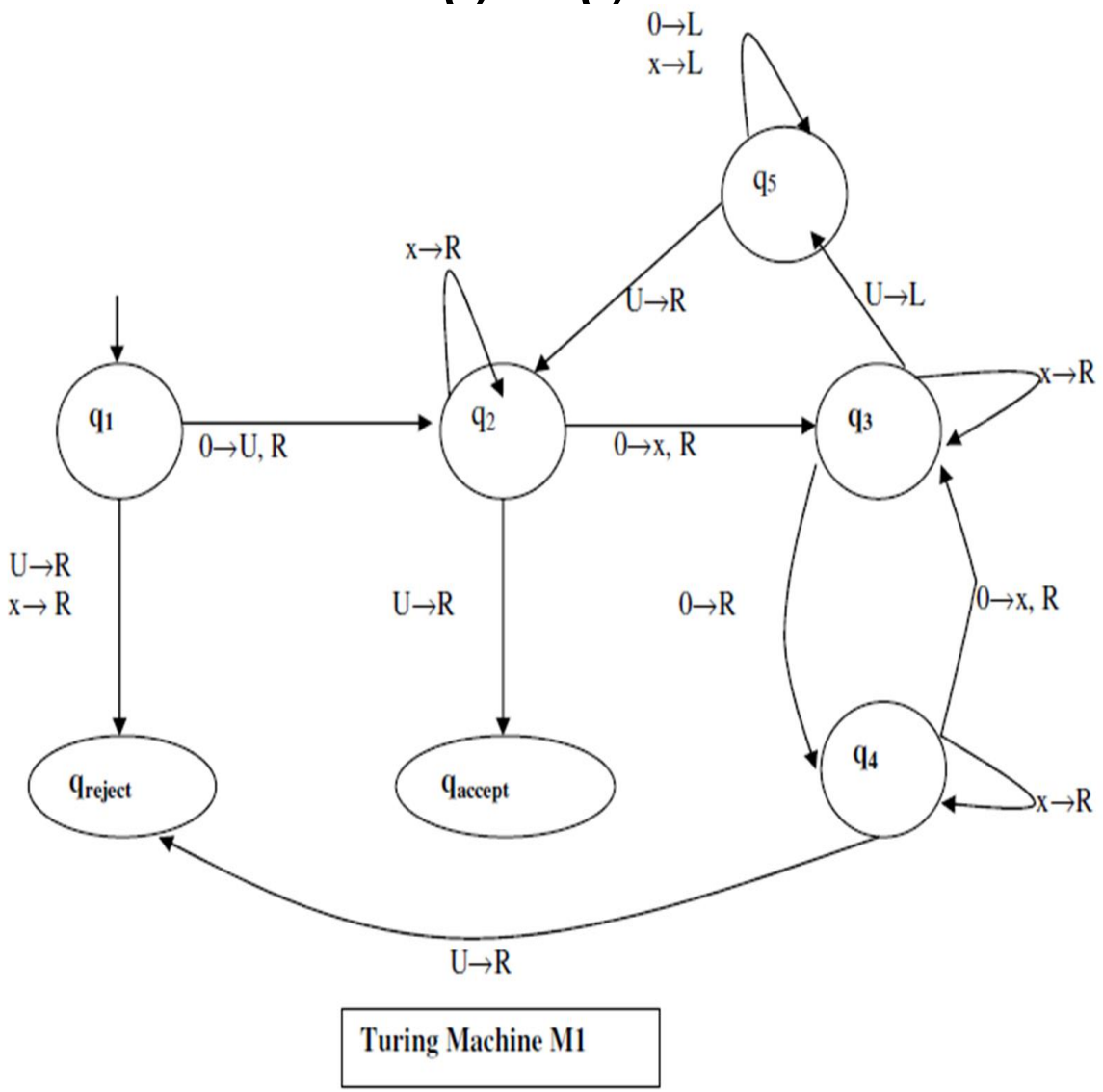


Turing Machine M1

$q_1$00UUU…
U$q_2$0UUU…
U xq_3UUU…
U q_5xUUU…
U q_5UxUUU…
U q_2xUUU…
U x q_2UUU…
U xU q_{accept}

Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 172.

- Show that the language 000 is not accepted by machine M_6



$_1$000UUU…

U$q_2$00UUU…

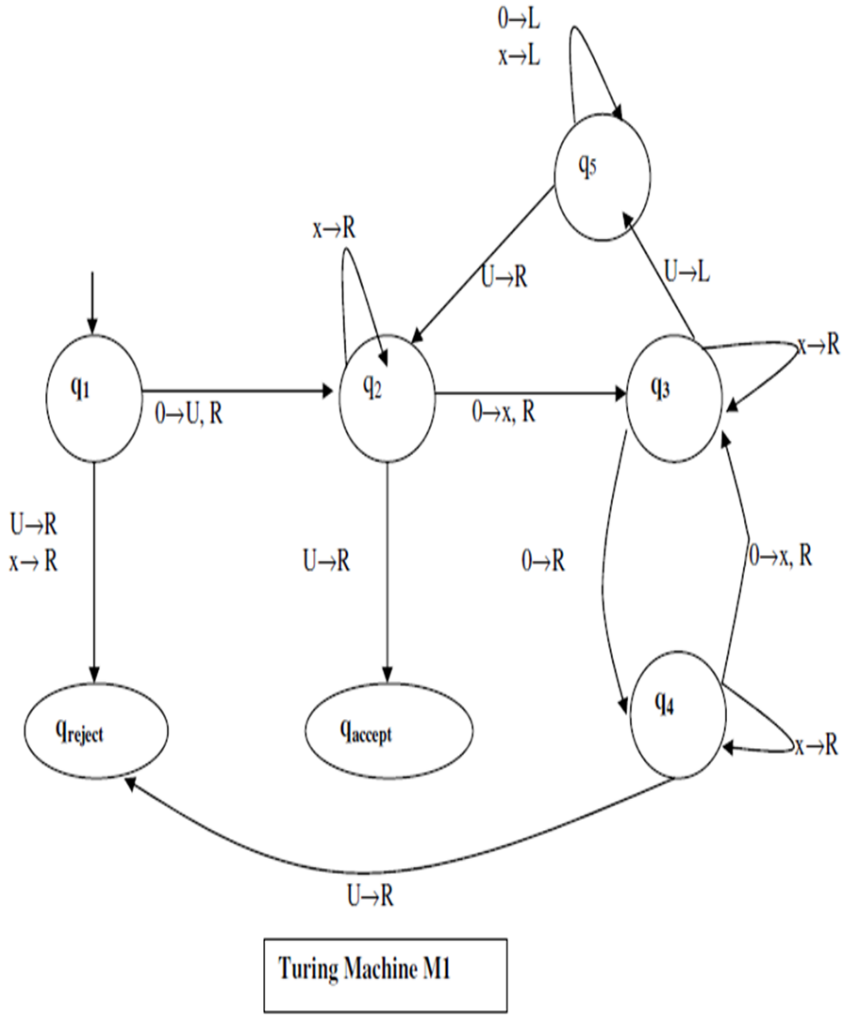
U x$q_3$0UUU…

Ux0q_4UUU…

Ux0U q_{reject} UU…

Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 172.

- Show that the language 0000 is accepted by machine M_6



$q_1$0000UUU…

U$q_2$000UUU…

U x$q_3$00UUU…

U x0$q_4$0UUU…

Ux0x q_3UUU…

Ux0 q_5 x UUU…

Ux $q_5$0x UUU…

U q_5 x0x UUU…

q_5 Ux0x UUU…

U q_2 x0x UUU…

U x q_2 0x UUU…

U xx q_3 x UUU…

U xx x q_3 UUU…

U xx q_5x UUU…

U x q_5 xx UUU…

U q_5 xx x UUU…

q_5 U xx x UUU…

U q2 xx x UUU…

U xq2 x x UUU…

U xx q2 x UUU…

Uxx xq2 UUU…

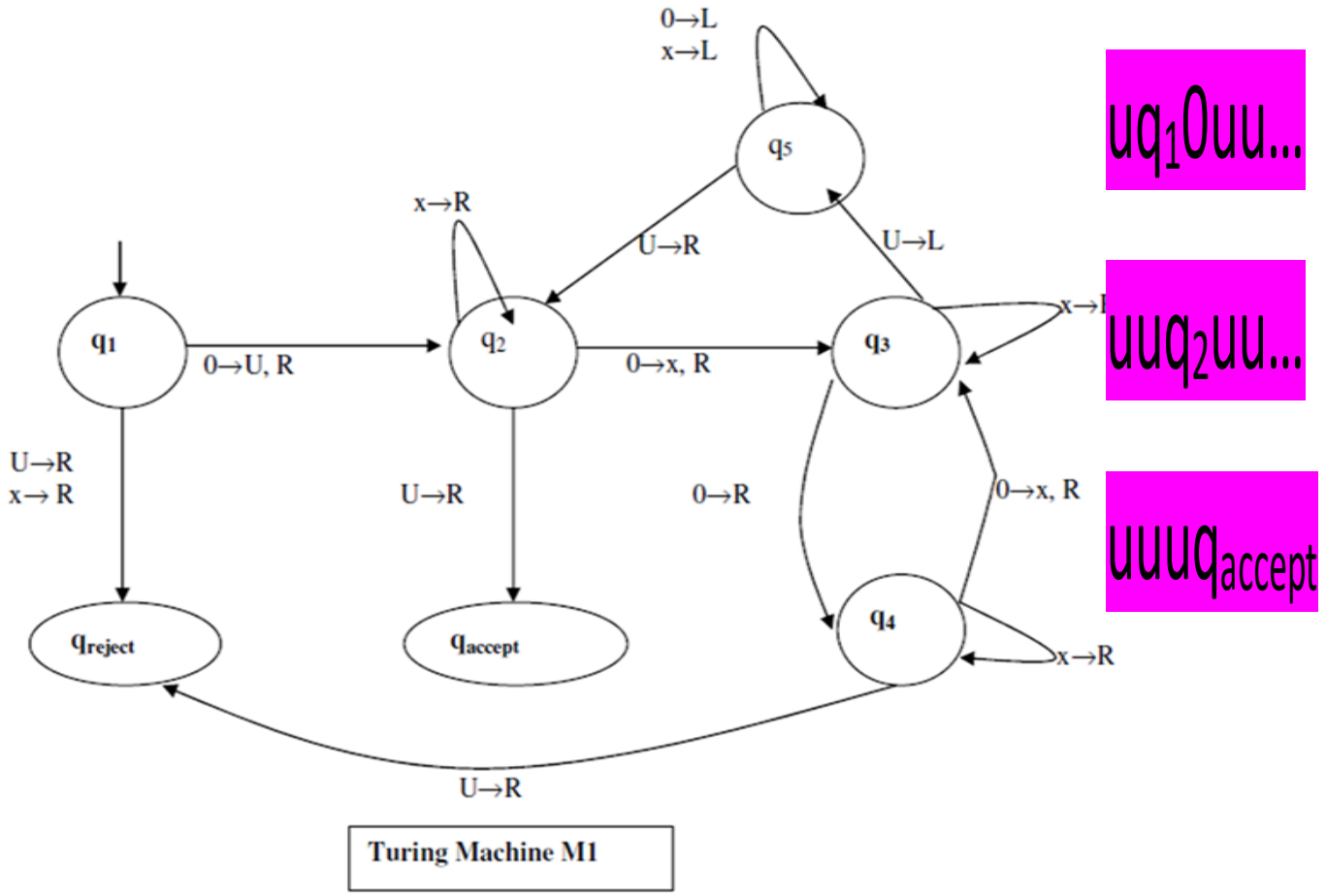
U xx x Uq_{accept}

Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 172.

Question:

- Show whether the following strings are accepted or rejected by the Turing Machine

i). 0



Turing Machine M1

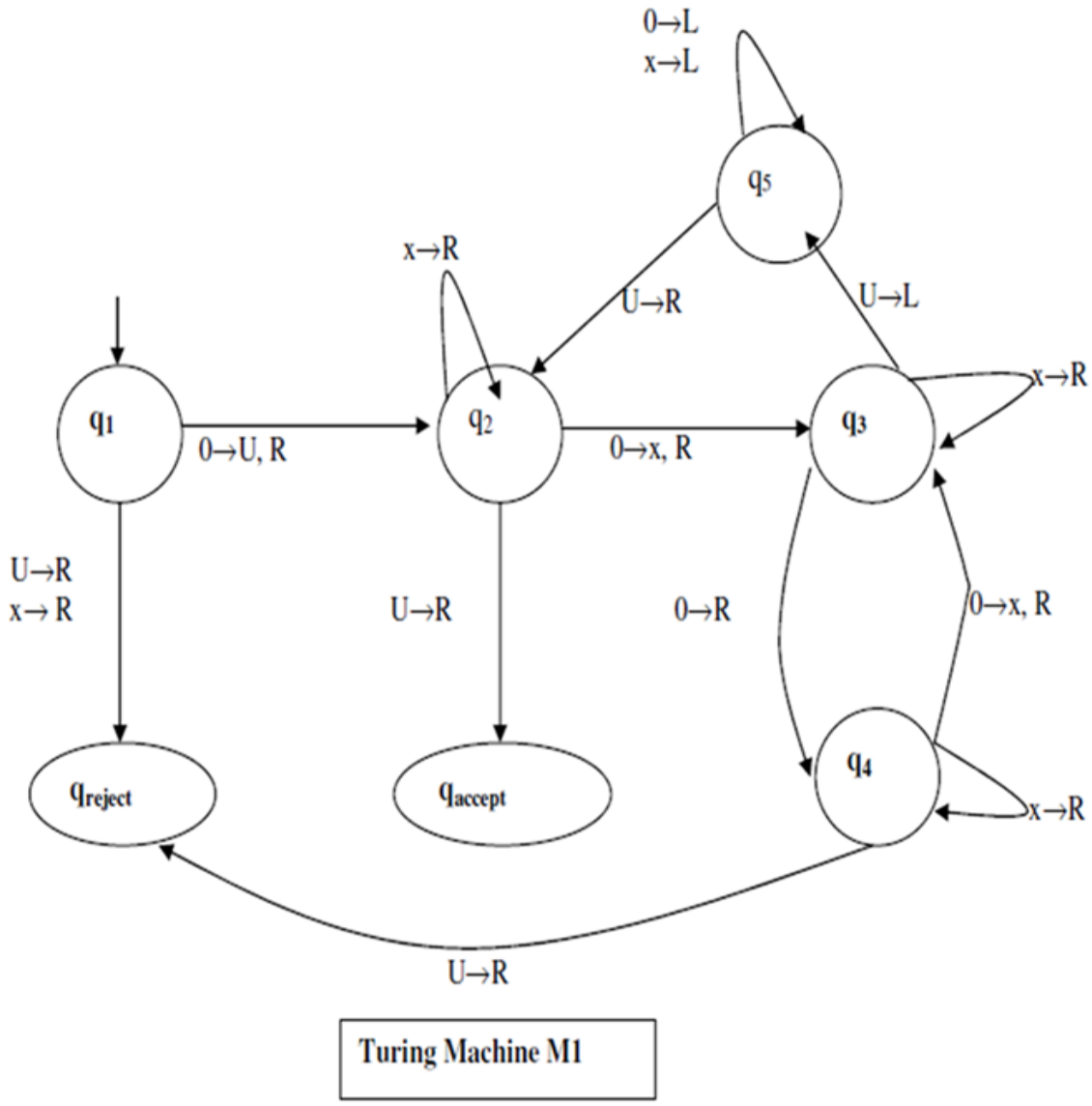
$uq_1 0uu...$

$uuq_2 uu...$

$uuuq_{accept}$

Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 172.

ii). 00000



Turing Machine M1

$uq_100000uu...$

$uuq_20000uu...$

$uuxq_3000uu...$

$uux0q_400uu...$

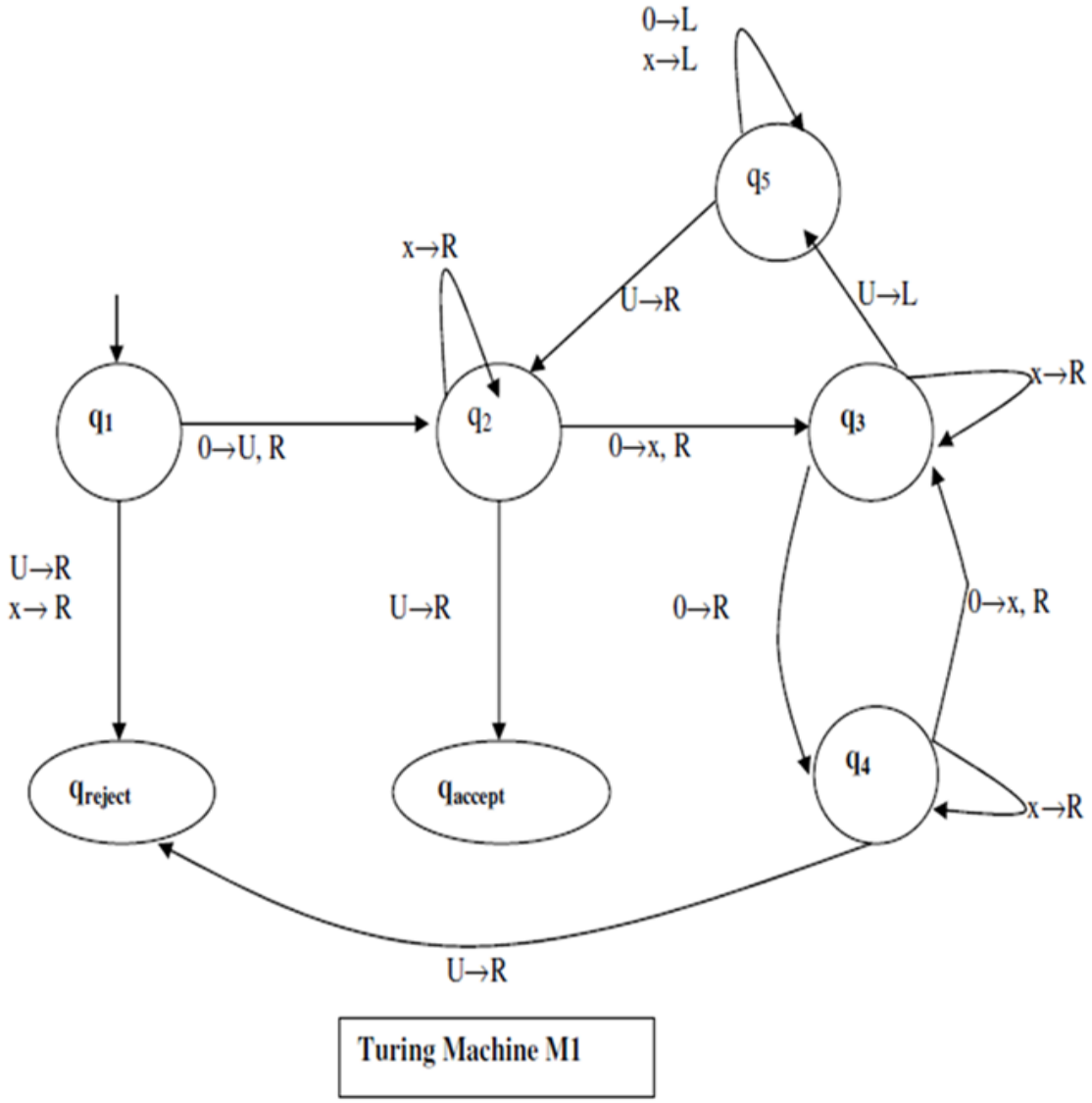
$uuxoxq_30uu...$

$uux0x0q_4uu...$

$uux0x0uq_{reject}$

Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 172.

iii). 000000



Turing Machine M1

$uq_1000000uu...$
$uuq_200000uu...$
$uuxq_30000uu...$
$uux0q_4000uu...$
$uuxoxq_300uu...$
$uux0x0q_40uu...$
$uux0x0xq_3uu...$
$uux0x0q_5xuu...$
$uux0xq_50xuu...$
$uux0q_5x0xuu...$
$uuxq_50x0xuu...$
$uuq_5x0x0xuu...$
$uq_5ux0x0xuu...$
$uuq_2x0x0xuu...$
$uuxq_20x0xuu...$
$uuxxq_3x0xuu...$
$uuxxxq_30xuu...$
$uuxxx0q_4xuu...$
$uuxxx0xq_4uu...$
$uuxxx0xuq_{reject}$

Source: Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012). Page 172.

Variants of Turing Machines: -

1) **Multi-tape Turing Machine** – Has several tapes, each with its own head for reading and writing
2) **Enumerator** – Has a printer attached to it.
3) **Non-Deterministic Turing Machine** – At any point, the machine may proceed according to several possibilities

1. Multi-tape Turing Machine

- As the name suggests, a multi-tape Turing machine has several tapes (k) for reading and writing information, instead of only one.

- The first tape is initialized with the input, and the rest are initially empty.
- The transition function is defined as:

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

- We can transition based on all k current input symbols, write a symbol to each of the k tapes, and move each of the k tapes either left or right.
- However, multi-tape Turing machines are not more capable than regular ones.

2. Enumerator

- This is a Turing machine with a printer attached to it.
- At the beginning of the tape, there is no input
- Instead of having accept or reject strings, the enumerator prints out a set of strings and this becomes the language of the enumerator.
- This string on the tape can be printed any time as decided by the enumerator

3. Non-Deterministic Turing Machine

- At each step in the computation, the machine can take one of several actions
- The transition function is defined as:

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

- Similar to the NFA, the computation of a non-deterministic TM is a **tree**, where each branch is a possible action.
- If any branch of the computation results in an accept state, the machine accepts the input.
- In terms of powr, the Non-deterministic Turing machines are not powerful than deterministic ones.

Equivalence with Other Models

- Researchers have proposed several models of computation with time with all of them exhibiting the characteristic of unrestricted access to unlimited memory.
- Some of these models are similar to the Turing machines, while others are different e.g. the lambda calculus by Alonzo Church.
- This points at the fact that computational power depends on the computer model in use.
- Whatever can be handled by a powerful super-computer can also be handled by a basic Turing machine.

Summary

- At the heart of every machine definition is the transition Function
- Finite Automata can either be DFA or NFA.
- There are other extensions of Finite Automata including the Push Down Automata (PDA) and the Turing Machine (TM) among others.
- Turing machines decide recursive languages and recognize the recursively enumerable languages.

References

- Rowan G. & John T., (2009), *Discrete Mathematics: Proofs, Structures and Applications*, CRC Press, ISBN: 9781439812808.
- W. D. Wallis (2003), *A Beginners Guide to Discrete Mathematics*, Springer Science & Business Media, ISBN: 978-0817642693.
- Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012).
- Introduction to languages and the theory of computation (3rd ed.), Martin, J., New York: McGraw-Hill. ISBN-13: 978-0072322002, (2002)