

# CS 410- PROJECT3 DESIGN REPORT

## FALL2022

### 1. Introduction

Turing Machine was invented by Alan Turing in 1936 and it is used to accept Recursive Enumerable Languages (generated by Type-0 Grammar).

A Turing machine consists of a tape of infinite length on which read and writes operation can be performed. The tape consists of infinite cells on which each cell either contains an input symbol or a special symbol called a blank. It also consists of a head pointer that points to the cell currently being read and it can move in both directions. [1]

A TM is expressed as a 7-tuple  $(Q, T, B, \Sigma, \delta, q_0, F)$  where:

$Q$  is a finite set of states

$T$  is the tape alphabet (symbols that can be written on Tape)

$B$  is a blank symbol (every cell is filled with  $B$  except input alphabet initially)

$\Sigma$  is the input alphabet (symbols that are part of the input alphabet)

$\delta$  is a transition function which maps  $Q \times T \rightarrow Q \times T \times \{L, R\}$ . Depending on its present state and present tape alphabet (pointed by head pointer), it will move to a new state, change the tape symbol (may or may not), and move the head pointer to either left or right.

$q_0$  is the initial state

$F$  is the set of final states. If any state of  $F$  is reached, the input string is accepted.

There are 4 possible operations on the tape:

- Read the symbol below the tape head.
- Update or write a symbol below the tape head.
- Move the tape head one step to the left.
- Move the tape head one step to the right.

At each step, the following are applied:

1. Read
2. Update
3. Move

In the project, a simulation of the Turing machine is aimed. Tape alphabet, states, start state, accept state, reject state, transition lists, and strings to be detected information is accepted through the input file. As a result, the program print the console and also creates a file which is named output.txt file will show the route that is taken by following the transition sequence of the string, and whether the string is accepted or not. In Sections 2 and 3, the tools that are used to design and create this project are introduced.

## 2. Tools Used

In the project, the tools are used below.

- Java 8: Used as the software language.
- IntelliJ IDEA Ultimate 2019: Used as the Integrated Development Environment.
- Java Util Package: Used to create ArrayList objects to manipulate simulating Turing Machine.
- Java Io Package: Used to read and write files.

## 3. Software Design

The algorithm contains functions to make the algorithm simplified. The list of the functions is below:

```
getTransitions(ArrayList<String>) : Object[]  
main(String[]) : void  
printResult(ArrayList<Object[]>) : void  
readFile(String) : void  
readString(String) : Object[]  
readStrings() : ArrayList<Object[]>  
updateBlanks() : void
```

Figure 1: List of functions of the simulation of the Turing Machine

**readFile:** It is used to read the input file and get information about the Turing machine.

**readStrings:** It is used to read the strings to check whether the string to be detected is accepted or not.

**readString:** It is used to read one line at a time to check whether the string to be detected is accepted or not.

**getTransitions:** It is used to create returns according to the route given parameter.

**printResult:** It is used to print result and create output file.

**updateBlanks:** It is used to update blank according to the tape has blank symbol.

```
0 X b                // Tape alphabet
q1 q2 q3 q4 q5 qA qR // States
Q1                   // Start State
qA                   // Accept State
qR                   // Reject State
q1 0 b R q2          // Transitions
q1 b b R qR
q1 X X R qR
q2 0 X R q3
q2 X X R q2
q2 b b R qA
q3 X X R q3
q3 0 0 R q4
q3 b b L q5
q4 X X R q4
q4 0 X R q3
q4 b b R qR
q5 0 0 L q5
q5 X X L q5
```

q5 b b R q2

000

// String(s) to be detected

*Figure 2: Example input file of the Turing Machine simulation*

q1 q2 q3 q4 qR (route taken)

Rejected

*Figure 3: Example output file of the Turing Machine simulation*