Contents

1 Introduction **1**

2 Exercise 1: Context Free Grammar & Language **1**

2.1 Sub question (a) 1

2.1.1 Language 1 1

2.1.2 Language 2 2

2.1.3 Language 1 UNION Language 2 2

2.2 Sub question (b) 2

3 Exercise 2: Turing Machine Configuration **3**

4 Exercise 3: Design a Turing Machine **4**

4.1 Implementation-level algorithm description 4

4.2 State diagram 5

5 Conclusion **5**

1 Introduction

The main objective of this assignment is to learn about Context Free Languages, Turing Machine configurations, and design of Turing Machines.

2 Exercise 1: Context Free Grammar & Language

2.1 Sub question (a)

For this question, we describe the language as the union of two sub-languages:

$$\begin{aligned} L &= \left\{x\#y : x, y \in \{0,1\}^*, |x| \neq |y| \cup x = y^R\right\} \\ &= \{x\#y : x, y \in \{0,1\}^*, |x| \neq |y|\} \bigcup \left\{x\#y : x, y \in \{0,1\}^*, x = y^R\right\} \end{aligned}$$

We define $L = L_1 \bigcup L_2$ as listed below, and we describe $L_1 = \{x\#y : x, y \in \{0,1\}^*, |x| \neq |y|\}$ and $L_2 = \left\{x\#y : x, y \in \{0,1\}^*, x = y^R\right\}$ separately.

2.1.1 Language 1

For L_1, we use the Context Free Grammar $G_1 = (V_1, \Sigma, R_1, S_1)$ to describe it, where:

$$\begin{aligned} V_1 &= \{S_1, L, R, X, Y\} \\ \Sigma &= \{0, 1\} \\ R_1 &= \{S_1 \to XL \mid RX \\ &\quad L \to XL \mid Y \\ &\quad R \to RX \mid Y \\ &\quad Y \to XYX \mid \# \\ &\quad X \to 0 \mid 1\} \end{aligned}$$

In G_1, the variables have the following meanings:

- S_1 Strings in the form $x\#y$ where x has longer (exclusive) length **OR** y has longer (exclusive) length
- L Strings in the form $x\#y$ where x has longer (inclusive) length
- R Strings in the form $x\#y$ where y has longer (inclusive) length
- X 0 **OR** 1
- Y Strings in the form $x\#y$ where x and y have the same length

L_1 is unambiguous because no string has two leftmost derivations.

2.1.2 Language 2

For L_2, we use the Context Free Grammar $G_2 = (V_2, \Sigma, R_2, S_2)$ to describe it, where:

$$\begin{aligned} V_2 &= \{S_2\} \\ \Sigma &= \{0, 1\} \\ R_2 &= \{S_2 \to 0S_20 \mid 1S_21 \mid \#\} \end{aligned}$$

In G_2. the variable S_2 stands for strings in the form $x\#y$ where x and y are reverse.

L_2 is unambiguous because no string has two leftmost derivations.

2.1.3 Language 1 UNION Language 2

For $L = L_1 \bigcup L_2$, we use the Context Free Grammar $G = (V, \Sigma, R, S)$ to describe it, where:

$$\begin{aligned} V &= \{S_1, S_2\} \\ \Sigma &= \{0, 1\} \\ R &= \{S \to S_1 \mid S_2\} \end{aligned}$$

In G, the variables have the following meanings:

- S_1 Strings in the form $x\#y$ where x has longer (exclusive) length **OR** y has longer (exclusive) length
- S_2 Strings in the form $x\#y$ where x and y are reverse

L is **unambiguous**, because it is the union of two unambiguous languages.

2.2 Sub question (b)

For this question, we use the Context Free Grammar $G = (V, \Sigma, R, S)$ to describe the language $L = \{a^m b^n c^p : m + n = p\}$, where:

$$\begin{aligned} V =& \{S, X\} \\ \Sigma =& \{a, b, c\} \\ R =& \{S \to aSc \mid bXc \mid \varepsilon \\ & X \to bXc \mid \varepsilon\} \end{aligned}$$

In G, the variables have the following meanings:

- S Strings where a and c are of same count **OR** transition to X to account for the count of b
- X Strings where b and c are of same count

L is **unambiguous** because no string has two leftmost derivations.

3 Exercise 2: Turing Machine Configuration

For this question, we are asked to trace the sequence of configurations for the string $110\#11$ in the Turing Machine presented in Figure 1.

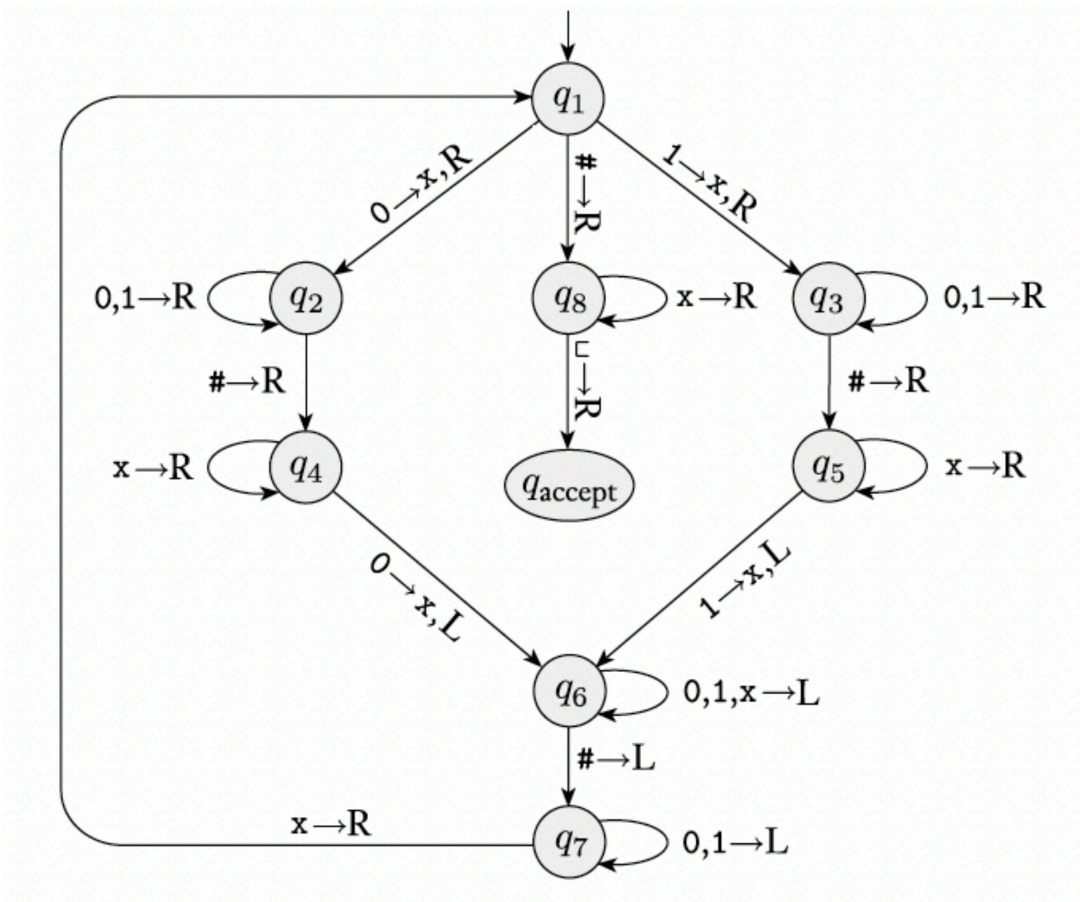

Figure 1: Turing Machine diagram

This string is **rejected**. The sequence of configurations is as follows:

$$
\begin{aligned}
&q_1 110\#11 \\
&\times\ q_3 10\#11 \\
&\times\ 1q_3 0\#11 \\
&\times\ 10q_3 \#11 \\
&\times\ 10\#q_5 11 \\
&\times\ 10q_6 \# \times 1 \\
&\times\ 1q_7 0\# \times 1 \\
&\times\ q_7 10\# \times 1 \\
&q_7 \times 10\# \times 1 \\
&\times\ q_1 10\# \times 1 \\
&\times\ \times q_3 0\# \times 1 \\
&\times\ \times 0q_3 \# \times 1
\end{aligned}
$$

$$
\begin{aligned}
&\times \times 0\#q_5 \times 1\\
&\times \times 0\# \times q_5 1\\
&\times \times 0\#q_6 \times \times\\
&\times \times 0 q_6\# \times \times\\
&\times \times q_7 0\# \times \times\\
&\times q_7 \times 0\# \times \times\\
&\times \times q_1 0\# \times \times\\
&\times \times \times q_2\# \times \times\\
&\times \times \times \#q_4 \times \times\\
&\times \times \times \# \times q_4 \times\\
&\times \times \times \# \times \times q_4 \_\\
&\times \times \times \# \times \times \_ q_{\text{reject}}
\end{aligned}
$$

4 Exercise 3: Design a Turing Machine

For this question, we design a Turing Machine that decides $A = \{w\#w^R : w \in \{0,1\}^*\}$.

4.1 Implementation-level algorithm description

We propose to check the reverse in a digit-by-digit manner. That is to say, we remember the first digit of w, cross it, then scan right till the first digit of w^R and check if it's the reverse digit; same for the second digit and so on until the end... Specifically, we describe the algorithm as follows:

1. If input is 0, cross it, read right, and switch to (2); if input is 1, cross it, read right, and switch to (5); if input is #, read right, switch to (8)
2. (Start of sub-branch 1 to check inverse of 0) Scan right until # while input is in 0*1*, Reject if not
3. Scan right until the first input other than cross
4. (End of sub-branch 1) If input is 1, cross it, read left, and switch to (10); Reject if not
5. (Start of sub-branch 2 to check inverse of 1) Scan right until # while input is in 0*1*, Reject if not
6. Scan right until the first input other than cross
7. (End of sub-branch 2) If input is 0, cross it, read left, and switch to (10); Reject if not
8. (Start of sub-branch 3 when all digits are checked, or in the case where input is a pure # without digit strings) While input is cross, keep scanning right
9. (End of sub-branch 3) If input is _, Accept; Reject if not
10. Scan left until # while checking input is 0, 1, or cross; Reject if not

11. Keep scanning left until cross while checking input is 0^*1^*; Reject if not
12. If input is cross, read right and repeat from (1); Reject if not

4.2 State diagram

We present the Turing Machine state diagram in Figure 2.

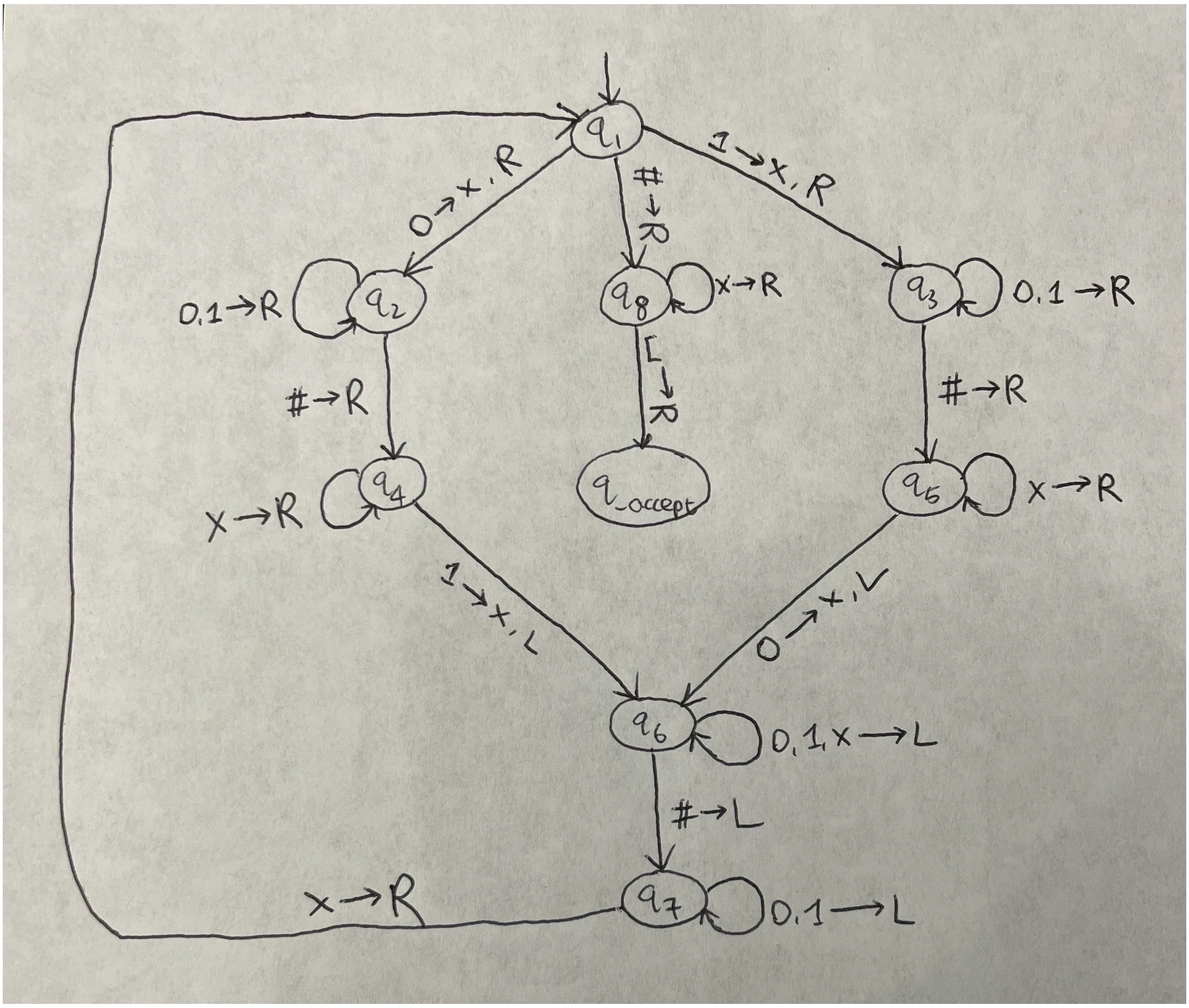

Figure 2: State diagram of Turing Machine recognizing $A = \{w\#w^R : w \in \{0,1\}^*\}$

5 Conclusion

In this assignment, we learned about Context Free Languages, Turing Machine configurations, and design of Turing Machines.