CSE332: Theory of Computation
Homework 2

Student Name: Nguyen Minh Duc
Student ID: 20202026

Problem 1 (20pts) Finite automata and pushdown automata always halt and provide the answer whether or not the input string is a member of a specific language. However, this is not always true for the Turing machine. To learn more about the difference between the Turing machine and simpler computational models (i.e., finite automata and pushdown automata), you are highly encouraged to refer to our textbooks (Hopcroft[a] and Sipser[b]).

Choose the right words to complete the description of the Turing machine.

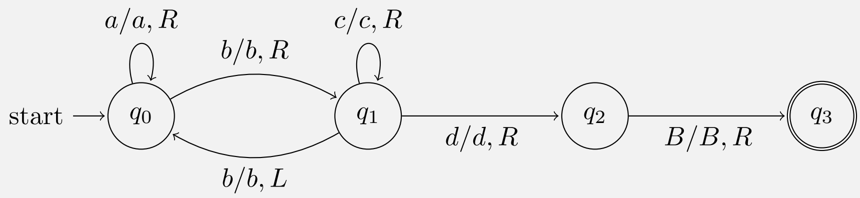
1. (10pts) A language L is called a (***recursively enumerable language***[c] / ***recursive language***[d]) if there exists a Turing machine M that accepts(halts in $q \in F$) every $w \in L$, and rejects(halts in $q \notin F$) or does not halt on $w \notin L$. In other words, M (***recognizes*** / ***decides***) L.
2. (10pts) A language L is called a (***recursively enumerable language*** / ***recursive language***) if there exists a Turing machine M that accepts(halts in $q \in F$) every $w \in L$, and rejects(halts in $q \notin F$) every $w \notin L$. In other words, M (***recognizes*** / ***decides***) L.

[a] *Introduction to Automata Theory, Languages, and Computation*, John Hopcroft, 2006, 3rd Edition, p334.
[b] *Introduction to the Theory of Computation*, Machael Sipser, 2012, 3rd Edition, p170
[c] Also called ***Turing-recognizable***.
[d] Also called ***Turing-decidable***.

Solution.

1. A language L is called a ***recursively enumerable language*** if there exists a Turing machine M that accepts(halts in $q \in F$) every $w \in L$, and rejects(halts in $q \notin F$) or does not halt on $w \notin L$. In other words, M ***recognizes*** L.
2. A language L is called a ***recursive language*** if there exists a Turing machine M that accepts(halts in $q \in F$) every $w \in L$, and rejects(halts in $q \notin F$) every $w \notin L$. In other words, M ***decides*** L.

Problem 2 (20pts) Below is the transition diagram of a Turing machine M that recognizes L but does not decide L. (A label $X/Y, D$ from state q to p corresponds to a transition $\delta(q, X) = (p, Y, D)$.)



1. (5pts) Describe the language L.
2. (5pts) Give a concrete example of an input string $w \notin L$ that M rejects (that is, halt in state $q \notin F$). Show the transitions of IDs (instantaneous descriptions) of M on that input string.
3. (5pts) Give a concrete example of an input string $w \notin L$ that M does not reject (that is, does not halt on). Show why M does not halt on that string by showing the transitions of IDs (instantaneous descriptions).
4. (5pts) Modify M so that it can ***decide*** L. Describe how it should be modified.

Solution.

1. Describe the language L.

$$L = \{a^n bc^m d \mid n, m \geq 0\}$$

2. Give a concrete example of an input string $w \notin L$ that M rejects (that is, halt in state $q \notin F$). Show the transitions of IDs (instantaneous descriptions) of M on that input string.

Let $w = aabcc \implies w \notin L$.
The transitions of IDs are

$$q_0 aabcc \vdash aq_0 abcc \vdash aaq_0 bcc \vdash aabq_1 cc \vdash aabcq_1 c \vdash aabccq_1$$

3. Give a concrete example of an input string $w \notin L$ that M does not reject (that is, does not halt on). Show why M does not halt on that string by showing the transitions of IDs (instantaneous descriptions).

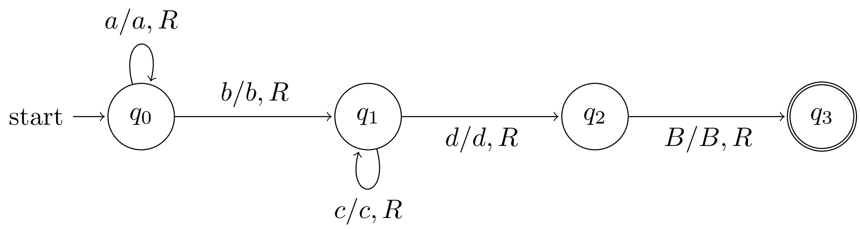
Let $w = aabbc \implies w \notin L$.
The transitions of IDs are

$$q_0 aabbc \vdash aq_0 abbc \vdash aaq_0 bbc \vdash aabq_1 bc \vdash aaq_0 bbc \vdash aabq_1 bc \vdash \ldots$$

Note that M now loops back and forth between q_0 and q_1 since there are two consecutive b's in w.

4. Modify M so that it can ***decide*** L. Describe how it should be modified.

The issue that prevents M from halting is the loop from q_0 to q_1. The infinite loop occurs when two consecutive b are in the input string w. The backward edge $\delta(q_1, b) = (q_0, b, L)$ is redundant as M already halts at a non-terminal state when there are two consecutive b's (of which M does not recognize). Therefore, we only need to remove the edge $\delta(q_1, b) = (q_0, b, L)$ to make M decide L.



Problem 3 (30pts) Complete the Turing machine M which transforms $w \in \{0,1\}^*$ into ww^R. You must define the transition function δ of $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0,1\}, \{0,1,x,y,B\}, \delta, q_0, B, \{q_4\})$ rather than drawing a transition diagram.

Solution.

The desired Turing machine is

$$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0,1\}, \{0,1,x,y,B\}, \delta, q_0, B, \{q_4\}),$$

where δ is the transition function

$$\begin{aligned}
\delta\left(q_0, 0\right) &= \left(q_0, x, R\right)\\
\delta\left(q_0, 1\right) &= \left(q_0, y, R\right)\\
\delta\left(q_0, B\right) &= \left(q_1, B, L\right)\\
\delta\left(q_1, 0\right) &= \left(q_1, 0, L\right)\\
\delta\left(q_1, 1\right) &= \left(q_1, 1, L\right)\\
\delta\left(q_1, x\right) &= \left(q_2, 0, R\right)\\
\delta\left(q_1, y\right) &= \left(q_3, 1, R\right)\\
\delta\left(q_2, 0\right) &= \left(q_2, 0, R\right)\\
\delta\left(q_2, 1\right) &= \left(q_2, 1, R\right)\\
\delta\left(q_2, B\right) &= \left(q_1, 0, L\right)\\
\delta\left(q_3, 0\right) &= \left(q_3, 0, R\right)\\
\delta\left(q_3, 1\right) &= \left(q_3, 1, R\right)\\
\delta\left(q_3, B\right) &= \left(q_1, 1, L\right)\\
\delta\left(q_1, B\right) &= \left(q_4, B, R\right)
\end{aligned}$$

Problem 4 (30pts) To describe storage in the state, a Turing machine M of Example 8.6 of the textbook[a], is expressed using an extended notation for a finite control—the set of states Q is expressed in the form of $\{q_0, q_1\} \times \{0, 1, B\}$ in $M = (Q, \{0,1\}, \{0,1,B\}, \delta, [q_0, B], B, \{[q_1, B]\})$. However, because it is merely another 'view' of an equivalent standard Turing machine, it can be converted into the standard Turing machine notation. Convert the definition of M to the one without the extended notation. In other words, complete the following blank components of $M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \{0,1,B\}, \_\_\_, \_\_\_, B, \_\_\_)$ using the standard Turing machine notation.

[a] *Introduction to Automata Theory, Languages, and Computation*, John Hopcroft, 2006, 3rd Edition, p338.

Solution.

The desired Turing machine is

$$M = \left(\{q_0, q_1, q_2, q_3\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \{q_3\}\right),$$

where δ is the transition defined as follows

$$\begin{aligned}
\delta\left(q_0, 0\right) &= \left(q_1, 0, R\right)\\
\delta\left(q_0, 1\right) &= \left(q_2, 1, R\right)\\
\delta\left(q_1, 1\right) &= \left(q_1, 1, R\right)\\
\delta\left(q_2, 0\right) &= \left(q_2, 0, R\right)\\
\delta\left(q_1, B\right) &= \left(q_3, B, R\right)\\
\delta\left(q_2, B\right) &= \left(q_3, B, R\right)
\end{aligned}$$