Course: Theory of Computation

HW6： Turing Machine

Lina Mi

@01377283

3.1 This exercise concerns TM M2, whose description and state diagram appear in Example 3.7. In each of the parts, give the sequence of configurations that M2 enters when started on the indicated input string.

a. 0.

The sequence of configurations:

$q_1 0 \rightarrow \sqcup q_2 \sqcup \rightarrow \sqcup \sqcup q_{accept}$

b. 00.

The sequence of configurations:

$q_1 00 \rightarrow \sqcup q_2 0 \rightarrow \sqcup x q_3 \sqcup \rightarrow \sqcup q_5 x \sqcup \rightarrow q_5 \sqcup x \sqcup \rightarrow \sqcup q_2 x \sqcup \rightarrow \sqcup x q_2 \sqcup \rightarrow \sqcup x \sqcup q_{accept}$

c. 000.

The sequence of configurations:

$q_1 000 \rightarrow \sqcup q_2 00 \rightarrow \sqcup x q_3 0 \sqcup \rightarrow \sqcup x 0 q_4 \sqcup \rightarrow \sqcup x 0 \sqcup q_{reject}$

d. 000000.

The sequence of configurations:

$q_1 000000 \rightarrow \sqcup q_2 00000 \rightarrow \sqcup x q_3 0000 \sqcup \rightarrow \sqcup x 0 q_4 000 \sqcup \rightarrow \sqcup x 0 x q_3 00 \sqcup \rightarrow \sqcup x 0 x 0 q_4 0 \sqcup$

$\rightarrow \sqcup x 0 x 0 x q_3 \sqcup \rightarrow \sqcup x 0 x 0 q_5 x \sqcup \rightarrow \sqcup x 0 x q_5 0 x \sqcup \rightarrow \sqcup x 0 q_5 x 0 x \sqcup \rightarrow \sqcup x q_5 0 x 0 x \sqcup \rightarrow \sqcup q_5 x 0 x 0 x \sqcup \rightarrow q_5 \sqcup x 0 x 0 x \sqcup$

$\rightarrow \sqcup q_2 x 0 x 0 x \sqcup \rightarrow \sqcup x q_2 0 x 0 x \sqcup \rightarrow \sqcup x x q_3 x 0 x \sqcup \rightarrow \sqcup x x x q_3 0 x \sqcup \rightarrow \sqcup x x x 0 q_4 x \sqcup \rightarrow \sqcup x x x 0 x q_4 \sqcup \rightarrow \sqcup x x x 0 x \sqcup q_{reject}$

3.2 This exercise concerns TM M1, whose description and state diagram appear in Example 3.9. In each of the parts, give the sequence of configurations that M1 enters when started on the indicated input string.

a. 11.

The sequence of configurations:

$q_1 11 \rightarrow x q_3 1 \rightarrow x 1 q_3 \sqcup \rightarrow q_{reject}$

b. 1#1.

The sequence of configurations:

$q_1 1\#1 \rightarrow x q_3 \#1 \rightarrow x\# q_5 1 \rightarrow x q_6 \# x \rightarrow q_7 x\#x \rightarrow x q_1 \# x \rightarrow x\# q_8 x \sqcup \rightarrow x\#x q_8 \sqcup \rightarrow x\#x \sqcup q_{accept}$

c. 1##1.

The sequence of configurations:

$q_1 1\#\#1 \rightarrow xq_3\#\#1 \rightarrow x\#q_5\#1 \rightarrow q_{reject}$

d. 10#11.

The sequence of configurations:

$q_1 10\#11 \rightarrow xq_3 0\#11 \rightarrow x0q_3\#11 \rightarrow x0\#q_5 11 \rightarrow x0q_6\#x1 \rightarrow xq_7 0\#x1 \rightarrow q_7 x0\#x1 \rightarrow xq_1 0\#x1 \rightarrow xxq_2\#x1$
$xx\#q_4 x1 \rightarrow xx\#xq_4 1 \rightarrow q_{reject}$

e. 10#10

The sequence of configurations:

$q_1 10\#10 \rightarrow xq_3 0\#10 \rightarrow x0q_3\#10 \rightarrow x0\#q_5 10 \rightarrow x0q_6\#x0 \rightarrow xq_7 0\#x0 \rightarrow q_7 x0\#x0 \rightarrow xq_1 0\#x0 \rightarrow xxq_2\#x0 \rightarrow$
$xx\#q_4 x0 \rightarrow xx\#xq_4 0 \rightarrow xx\#q_6 xx \rightarrow xxq_6\#xx \rightarrow xq_7 x\#xx \rightarrow xxq_1\#xx \rightarrow xx\#q_8 xx \rightarrow xx\#xq_8 x \rightarrow xx\#xxq_8 \sqcup \rightarrow$
$xx\#xx\sqcup q_{accept}$

3.6 In Theorem 3.21, we showed that a language is Turing-recognizable iff some enumerator enumerates it. Why didn't we use the following simpler algorithm for the forward direction of the proof? as before, $s_1, s_2, \ldots$ is a list of all strings in Σ^* . E = "Ignore the input.

1. Repeat the following for i = 1, 2, 3, ...

2. Run M on s_i.

3. If it accepts, print out s_i ."

Solution: based on the previous algorithm, the TM M run on one string s_i each time, differing from the algorithm provided by the text book, in which TM M run on all strings of the list at each step, that means the previous algorithm test the string $s_1, s_2, \ldots.$ one by one, however the set Σ^* can be infinite, and if an strings s_k belongs to the list of all strings in Σ^* and recognizable by TM M, it is located in the infinite position of list, the TM M will not be able to test it, and hence s_k will have no chance to be printed out by enumerator E, namely string s_k is not enumerated by enumerated by E. so the forward direction of Theorem 3.21 can not be proved by the algorithm provided in this exercise.

3.8 Give implementation-level descriptions of Turing machines that decide the following languages over the alphabet {0,1}.

a. {w| w contains an equal number of 0s and 1s}

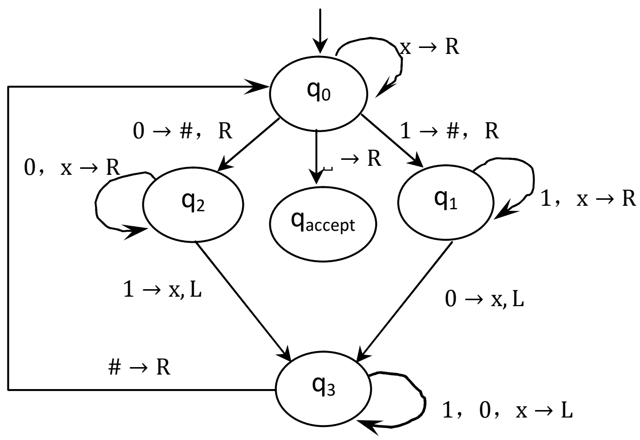
Suppose $M = \{Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject}\}$ is the Turing Machine that decide the language S={w| the number of 0s =the number of 1s},

Where $Q = \{q_0, q_1, q_2, q_3, q_{accept}, q_{reject}\}$

$\Sigma = \{0,1\}; \Gamma = \{0, 1, \#, x, \sqcup\}$

Start state:q_0; accept state: q_{accept} ; reject state: q_{reject}

δ is described as following figure:



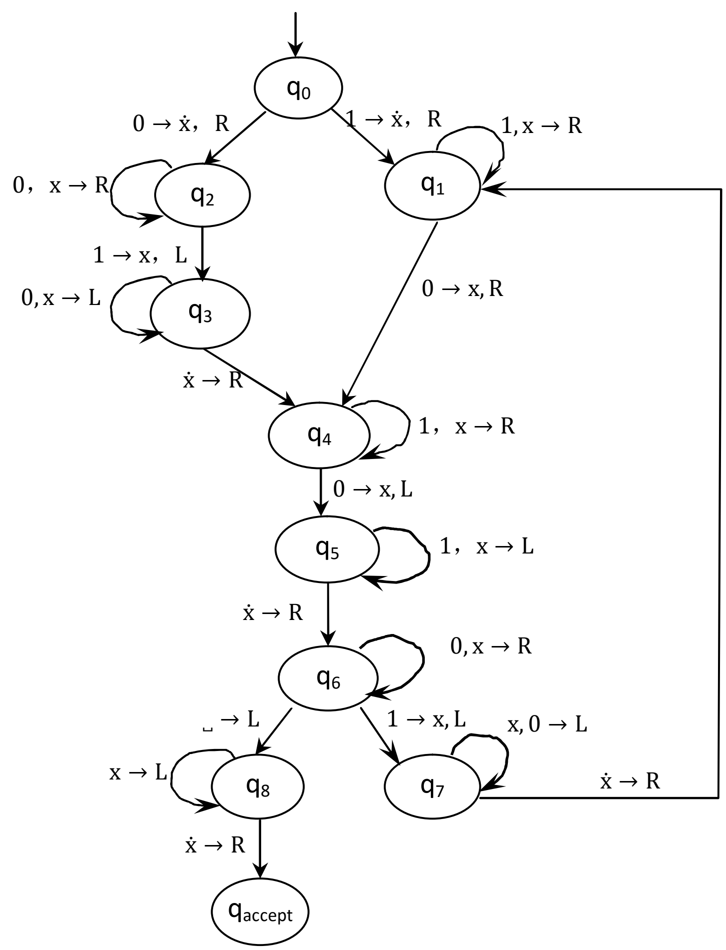
b. {*w*| *w* contains twice as many 0s as 1s}

Suppose $M = \{Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject}\}$ is the Turing Machine that decide the language S={*w*| *w* contains twice as many 0s as 1s },

Where $Q = \{q_0, q_1, q_2, q_3, q_{04}, q_5, q_6, q_7, q_8, q_{accept}, q_{reject}\}$

$\Sigma = \{0,1\}$; $\Gamma = \{0, 1, \dot{x}, x, ␣\}$

Start state:q_0; accept state: q_{accept} ; reject state: q_{reject}



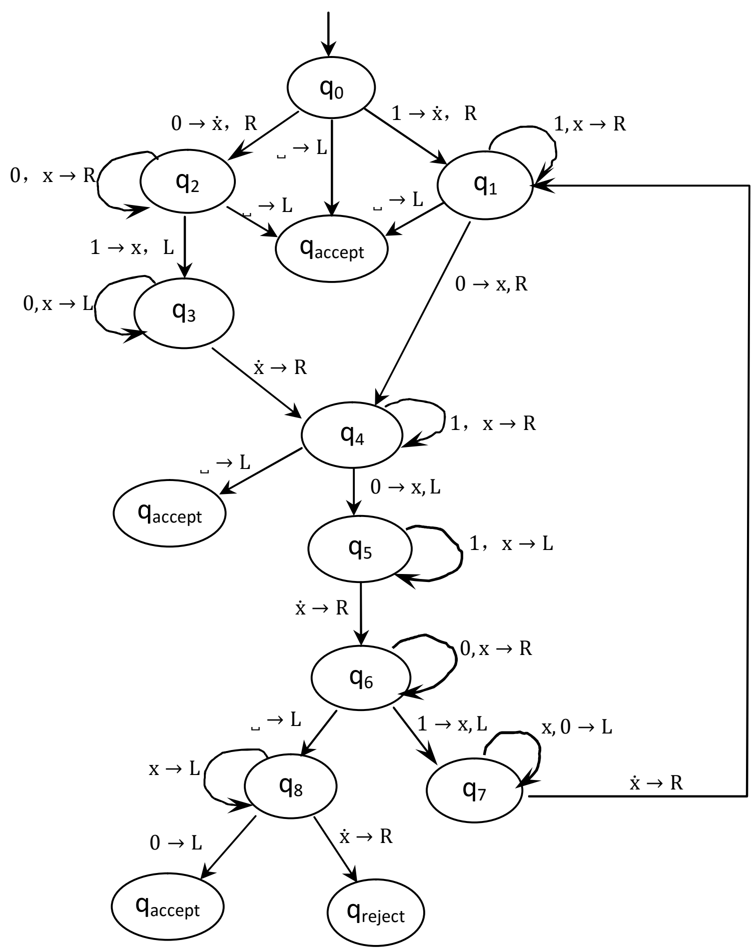
c. {w| w does not contain twice as many 0s as 1s}

Suppose $M = \{Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject}\}$ is the Turing Machine that decide the language S={w| w does not contains twice as many 0s as 1s },

Where $Q = \{q_0, q_1, q_2, q_3, q_{04}, q_5, q_6, q_7, q_8, q_{accept}, q_{reject}\}$

$\Sigma = \{0,1\}$; $\Gamma = \{0, 1, \dot{x}, x, ␣\}$

Start state:q_0; accept state: q_{accept} ; reject state: q_{reject}



3.15 Show that the collection of decidable languages is closed under the operation of

a. union.

Let language A and B are both decidable languages, and there are turing machine $M_1=\{Q_1, \Sigma_1, \Gamma_1, \delta_1, q_{01}, q_{accept1}, q_{reject1}\}$ decides A, and turing machine $M2=\{Q_2, \Sigma_2, \delta_2, q_{02}, q_{accept2}, q_{reject2}\}$ decides B, for language C=AUB, if we can construct a turing machine to decide language C, then C is decidable.

Let $M=\{Q, \delta, \Sigma, q_0, q_{accept}, q_{reject}\}$ and C can be decided by M, then

$Q = Q_1 \cup Q_2$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$\Gamma = \Gamma_1 \cup \Gamma_2$

$$\delta_{(q_i,a)} = \begin{cases} \delta_{1(q_i,a)}, q_i \in Q_1 \\ \delta_{2(q_i,a)}, q_i \in Q_2 \end{cases}$$

$q_0 = q_{01} \cup q_{02}$
$q_{accept} = q_{accept1} \cup q_{accept2}$
$q_{reject} = q_{rejct1} \cup q_{reject2}$

From the definition of M, it can be seen that for any input c, where $c \in \{A \cup B\}$, because if it is accepted by M_1, rejected by M_2, it will accept by M; if it is accepted by M_2, rejected by M_1, it can also be accepted by M; if it is rejected by both of M_1 and M_2, it will be rejected by M. it means for any input c, where $c \in \{A \cup B\}$, it can decided by TM M, namely, language $C = A \cup B$ is decidable. Hence it can conclude that the collection of decidable languages is closed under union operation.

b. concatenation.

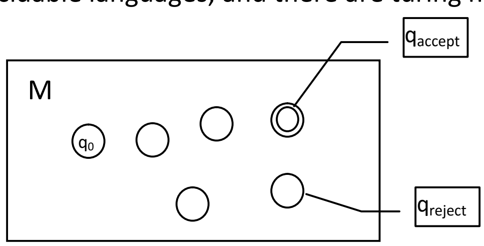
Let language A and B are both decidable languages, and there are turing machine $M_1=\{Q_1, \Sigma_1, \Gamma_1, \delta_1, q_{01}, q_{accept1}, q_{reject1}\}$ decides A, and turing machine $M2=\{Q_2, \Sigma_2, \delta_2, q_{02}, q_{accept2}, q_{reject2}\}$ decides B, for language $C = A \circ B$, we construct TM M= $\{Q, \Sigma, \delta, q_0, q_{accept}, q_{reject}\}$, where $Q = Q_1 \cup Q_2$; $\Sigma = \Sigma_1 \cup \Sigma_2$; $\Gamma = \Gamma_1 \cup \Gamma_2$; $q_0 = q_{01}, q_{accept} = q_{accept2}$; $q_{reject} = q_{reject2}$;

$$\delta(q_i, a) = \begin{cases} \delta_1(q_i, a) & q_i \in Q_1 \text{ and } q_i \notin q_{accept1} \\ q_{02}, \text{head do not move} & q_i \in q_{accep} \quad \text{and } a = \varepsilon \\ \delta_2(q_i, a) & q_i \in Q_2 \end{cases}$$

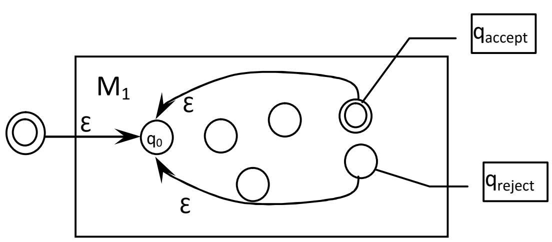
For any input c, $c \in \{C | C = A \circ B\}$, it can be decided by the constructed TM M, that means language C can be decided by M, namely C is decidable. So it can be concluded that the collection of decidable languages is closed under operation concatenation.

c. star

Let language A is decidable languages, and there are turing machine $M=\{Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject}\}$ decides A



For language A^*, we construct M_1 based on previous TM M,



TM M1 can decide Ɛ, which is A^0. For any input A^*, $(* \neq 0)$, it can be broken into * number of pieces, namely $\underbrace{AAAAAAA}_{*\text{ number of A}}$, TM M_1 will can decide every piece of it, end either with state of accept or with state of reject, from previous analysis, the language A^* can be decided by M1, namely A* is decidable. So it can be concluded that the collection of decidable languages is closed under star operation.

d. complementation.

Let language A is decidable language, and there are turing machine M={Q, Σ, Γ , δ, q_0, q_{accept}, q_{reject}} decides A, then for any input $b \in \bar{A}$, it will end the TM M with q_{reject}, namely it can be decide by TM M, so the language $\bar{A}$ can be decide by TM M, namely it is decidable. Now we can reach the conclusion that the collection of decidable language is closed under complementation operation.

e. intersection.

Let language A and B are both decidable languages, and there are turing machine M_1={Q_1, Σ_1, Γ_1 , δ_1, q_{01}, $q_{accept1}$, $q_{reject1}$} decides A, and turing machine M2={Q_2, Σ_2, δ_2, q_{02}, $q_{accept2}$, $q_{reject2}$} decides B, then for any input $c \in \{A \cap B\}$, it can be decided by both of M_1 and M_2, so the collection of input c is decidable language, namely language $C = A \cap B$ is decidable. So it can be concluded that the collection of decidable languages is closed under intersection operation.