YONSEI UNIVERSITY
Department of Computer Science

CSI3109 Automata and Formal Languages, SPRING 2017

Homework No.2

DUE: 2017.04.17 13:00:00pm

Student ID: 2014117007

Student Name: Chung, Ji-Wan

1. Given a DFA $A = (Q, \Sigma, \delta, s, F)$, where

$$Q = \{1, 2, 3, 4, 5, 6\}, \Sigma = \{a, b\}, s = 1, F = \{2, 5, 6\}$$

and δ is defined as follows:

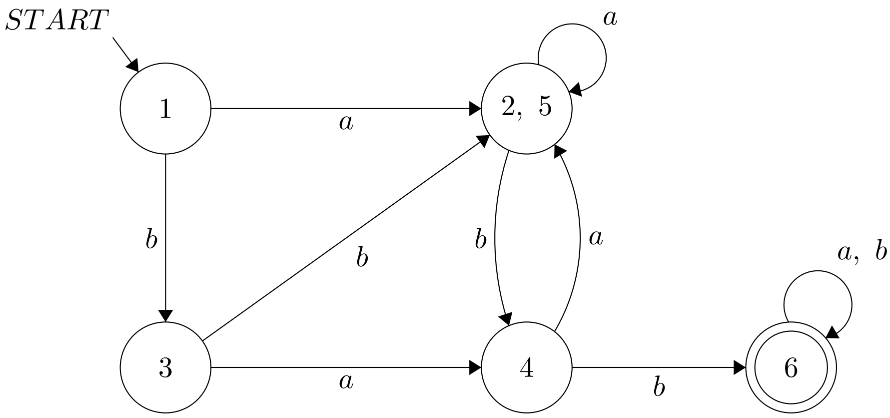
	a	b
1	2	3
2	2	4
3	4	5
4	2	6
5	5	4
6	6	6

Run the TF (Table Filling) algorithm and draw a minimal DFA for $L(A)$.

$$\begin{bmatrix} 2 & X & - & - & - & - \\ 3 & X & X & - & - & - \\ 4 & X & X & X & - & - \\ 5 & X & & X & X & - \\ 6 & X & X & X & X & X \\ / & 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

Equivalence Class $= (1), (3), (4), (6), (2, 5)$

$\rightarrow$ *drawing on the next page*


START
1
2, 5
3
4
6
a
b
a
b
a
b
a
a, b
a
b

2. Prove that the following languages are not regular. You may use the pumping lemma or the closure properties of regular languages under union, intersection and complement.
   a) $L = \{0^n 1^m 0^n \mid m, n \geq 0\}$.

Answer:

i. Assume that L is regular. Then, there is a pumping constant $n \geq 1$ strings with length greater than or equal to n can be 'pumped' according to pumping lemma.

ii. For an arbitrary natural number $l \geq n$, we take a string $w = 0^l 1^l 0^l \in L$.

iii. Following pumping lemma, since $|w| > l \geq n$ w can be divided into $w = xyz$ of which $|xy| \leq n \leq l$ and $|y| > 0$ holds.

iv. Since $|xy| \leq l$, we can safely deduct the substrings x, y are composed of the symbol 0 only. Likewise we define $x = 0^i, y = 0^{j-i}$ for some $0 \leq i < j \leq n \leq l$.

v. Then we consider another string $w' = xy^2z = 0^i 0^{2(j-i)} 0^{l-j} 1^l 0^l = 0^{l+(j-i)} 1^l 0^l$ which can be pumped from w. Since $|y| > 0$, $j - i > 0$. Then $l + (j - i) > l$, rendering $w' \notin L$. For that contradicts our assumption that the pumping lemma holds for L, L is not regular.

b) $L = \{w \mid w \in \{0,1\}^* \text{ is not a palindrome}\}$.
(A palindrome is a string that reads the same forward and backward. e.g.: racecar)

Answer:

i. Assume L is regular. Then, $P = \overline{L} = \{w|w \in \{0,1\}^* \text{is a palindrome}\}$ is also regular since a regular language is closed under complement.

ii. We take contrapositive of i). Then, if we can prove the nonregularity of P we prove the nonregularity of L automatically.

iii. Now, define a Language $L' = L(0^*1^*0^*)$. We can see L' is regular by its defition relying on a regular expression. Then we recognize $L' \cap P$ equals the language of a), which is already proven unregular. If P is regular, the language of a) should be regular since a regular language is closed under intersection, which is not the case. So P is nonregular.

iv. Combining ii)and iii), we can deduct L is also a nonregular language.

3. Show that the regular languages are closed under the following operations:

   a)
   $$\mathbb{DROPOUT}(L) = \{xz \mid xyz \in L, \text{ where } x, z \in \Sigma^*, y \in \Sigma\}.$$

   Namely, $\mathbb{DROPOUT}(L)$ is the language containing all strings that can be obtained by removing one symbol from a string in L.
   For example, if $L = \{012\}$, then $\mathbb{DROPOUT}(L) = \{12, 02, 01\}$.

   Answer:

   i. If L is regular, there is at least one corresponding regular expression E recognizing L.

   ii. Now we construct a regular expression E' recognizing the language of $\mathbb{DROPOUT}(L)$. Beforehand, we define a function f which accepts a RegEx E'' and returns a set formed by processing $\mathbb{DROPOUT}$ on the language accepted by that expression.

   iii. For $E'' = \emptyset$, $f(\emptyset) = \emptyset$ by definition.

   iv. For $E'' = \{\epsilon\}$, $f(\{\epsilon\}) = \emptyset$ by definition.

   v. For $E'' = \{\sigma\}$, $f(\{\sigma\}) = \{\epsilon\}$ by definition.

   vi. For union operation, $f(E_1 + E_2) = f(L(E_1) \cup L(E_2))$. Note that for a string w, $w \in L(E_1) \cup L(E_2)$ implies either $w \in L(E_1)$ or $w \in L(E_2)$ holds. For the former, $f(\{w\}) \subset f(L(E_1))$ and $f(L(E_1) \cup L(E_2))$ respectively. For the latter, $f(\{w\}) \subset f(L(E_2))$ and $f(L(E_1) \cup L(E_2))$. So every w is in $f(L(E_1) \cup L(E_2))$ if and only if it is in $f(L(E_1)) \cup f(L(E_2))$. Hence $f(E_1 + E_2) = f(E_1) + f(E_2)$.

   vii. For catenation, a string w' in $f(E_1 \cdot E_2)$ has a symbol deleted from w in $E_1 \cdot E_2$. Since that deletion happened either in E_1 or E_2, w is either in $f(E_1) \cdot E_2$ or $E_2 \cdot f(E_2)$. Hence $f(E_1 \cdot E_2) = f(E_1) \cdot E_1 + E_1 \cdot f(E_2)$.

   viii. For Kleene Star, we can divide the case of R^* into ϵ and R^+. The epsilon case was dealt with in *iv*). For the latter, consider R^* can be divided into an arbitrary sequence of R^*. With R^+, it can be followed that $R^+ = R^* \cdot R \cdot R^*$ for R^+ contains at least one R. Now, $f(R^+) = f(R^* \cdot R \cdot R^*)$ for any arbitrary choice of the single R from the sequence. Since $mathbbDROPOUT$ has to remove a symbol from an ocurrence of R, let that be our R. Then, $f(R^+) = f(R^* \cdot R \cdot R^*) = R^* \cdot f(R) \cdot R^*$.

   ix. Now that for any RegEx E we can construct $E' = f(E)$, we can conclude a regular language is closed under $\mathbb{DROPOUT}$.

b)
$$\mathbb{INIT}(L) = \{w \mid w \text{ for some } x, wx \in L\}.$$

For example, if $L = \{01, 110\}$, then $\mathbb{INIT}(L) = \{0, 01, 1, 11, 110\}$.
(*HINT:* Start with a DFA A for L and describe how to construct an FA for $\mathbb{INIT}(L)$ using A. We assume that A has no sink states.)

Answer:

i. Let a DFA accepting L be $A = (Q, \Sigma, \delta, s, F)$. Then, we construct a DFA A' accepting $\mathbb{INIT}(L)$.

ii. Let A' be $A' = (Q, \Sigma, \delta, s, F')$. The single difference between A and A' is the final states F and F'. We define $F' = Q/\{s\}$. Namely, we let every state of A' except for the start state be final state.

iii. Pick an arbitrary string $w = \sigma^1\sigma^2 \cdots \sigma^n \in L$. Note $|w| = n$. Now we prove $\sigma^1 \in L(A'), \sigma^1\sigma^2 \in L(A'), \cdots w \in L(A')$.

iv. For w to be accepted in A, there has to be a sequence $s \mapsto_A^{\sigma_1} q_1 \mapsto_A^{\sigma_2} \cdots f$. Since A' has the same set of transition functions as A, it follows $s \mapsto_{A'}^{\sigma_1} q_1 \mapsto_{A'}^{\sigma_2} \cdots f$.

v. Consider $w_1 = \sigma_1$. With A, processing w_1 takes one transition which takes the current counter to the state q_1. The situation is almost alike with A' except that in A', $q_1 \in F$. So A' accepts w_1.

vi. Assume A' accepts $w_n = \sigma_1\sigma_2 \cdots \sigma_n$. Then we consider whether A' accepts $w_{n+1} = \sigma_1\sigma_2 \cdots \sigma_n\sigma_{n+1}$.

vii. Since $q_n \mapsto_A^{\sigma_{n+1}} q_{n+1}$, $q_n \mapsto_{A'}^{\sigma_{n+1}} q_{n+1}$. However, q_{n+1} is a final state of A'. So A' accepts w_{n+1} as well.

viii. With mathematical induction from *vi*) and *vii*), we can deduct A' accepts the language of $mathbbINIT(L)$.

ix. Since there is a DFA accepting $\mathbb{INIT}(L)$ for every regular language L, we conclude regular languages are closed under the operation of $\mathbb{INIT}$.

4. Given two NFAs $A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and $A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$, suggest an NFA construction for $L(A_1) \cap L(A_2)$ and justify the construction (in other words, prove the correctness of your construction.)

Answer: We define an NFA $A = (Q, \Sigma, \delta, s, F)$ where $Q = Q_1 \times Q_2, s = (s_1, s_2), F = F_1 \times F_2$ and the transition function δ is defined by $\delta((q_{11}, q_{21}), \sigma) = Q'$ where $Q' = \{(q_{12}, q_{22}) | \delta_1(q_{11}, \sigma) = q_{12} \wedge \delta_2(q_{21}, \sigma) = q_{22}\}$. We suppose A accepts the language of $L(A_1) \cap L(A_2)$.
Then we prove that it indeed does.
For a string w of which $|w| = n$ holds, $w \in L(A_1) \cap L(A_2)$ means $w \in L(A_1)$ and $w \in L(A_2)$, which them implies both A_1 and A_2 accepts w. That says there are two sequences $s_1 \mapsto^{\sigma_1}_{A_1} q_{11} \mapsto^{\sigma_2}_{A_1} q_{12} \cdots \mapsto^{\sigma_n}_{A_1} f_1, f_1 \in F_1$ and $s_2 \mapsto^{\sigma_1}_{A_2} q_{21} \mapsto^{\sigma_2}_{A_2} q_{22} \cdots \mapsto^{\sigma_n}_{A_2} f_2, f_2 \in F_2$ respectively.
Now we do the mathematical induction on the length of a string $w \in L(A_1) \cap L(A_2)$.
If $|w| = 0$, $w = \epsilon$. We call this case vacuously true since if $s_1 \in F_1$ and $s_2 \in F_2$, $(s_1, s_2) \in F_1 \times F_2$.
For $|w| = 1$, we can denote $w = \sigma$ for a $\sigma \in \Sigma$. Since $s_1 \mapsto^{sigma}_{A_1} Q_1$ and $s_2 \mapsto^{\sigma}_{A_2} Q_2$ results in $(s_1, s_2) \mapsto^{\sigma}_{A} Q, Q = Q_1 \times Q_2$ by defintion, $w \in L(A)$.
We assume A accepts all w of which length is $|w| \leq n$. Then we prove A accepts w' of which length is $|w'| = n + 1$. Since w' is accepted by both A_1 and A_2, there are two respective sequences from s_1 and s_2 to q_{1n+1} and q_{2n+1}. That implies $s_1 \mapsto^{*}_{A_1} q_{1n} \mapsto^{\sigma_{n+1}}_{A_1} q_{1n+1}$ and $s_2 \mapsto^{*}_{A_2} q_{2n} \mapsto^{\sigma_{n+1}}_{A_2} q_{2n+1}$. Here, from our assumption $(s_1, s_2) \mapsto^{*}_{A} (q_{1n}, q_{2n})$. By definition of A we can deduct $(q_{1n}, q_{2n}) \mapsto^{\sigma_{n+1}}_{A} (q_{1n+1}, q_{2n+1})$. Hence, there is a sequence from (s_1, s_2) to (q_{1n+1}, q_{2n+1}).
From the induction, we can claim that A accepts any length of $w \in L(A_1) \cap L(A_2)$.

5. Consider the following two languages:
   - $L_1 = \{w \mid w$ has the same number of a's and b's$\}$.
   - $L_2 = \{w \mid w$ has the same number of the substrings ab and $ba.\}$.

   a) Is L_1 regular? Justify your answer—If L_1 is regular, show an regular expression or an FA. If not, prove it.

   Answer:

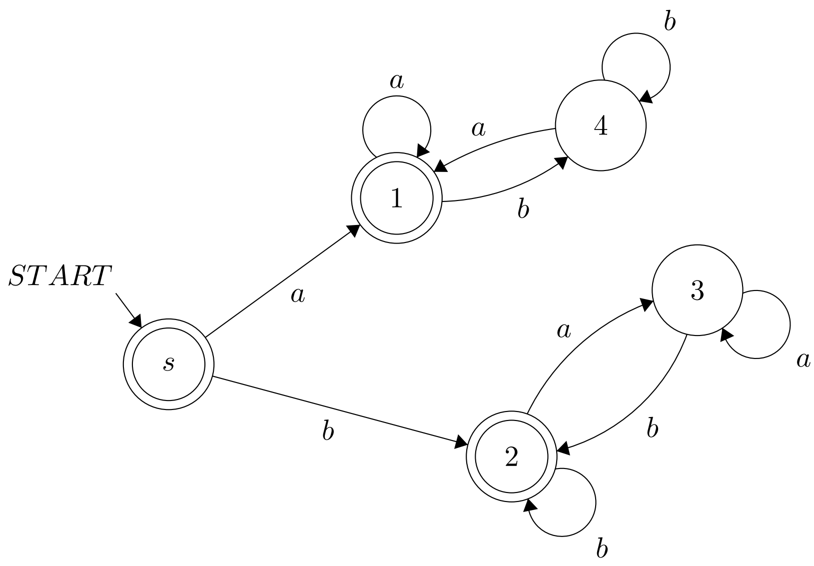
   If L_1 is regular, $L_1 \cap L_a (L_a = L(a^*b^*))$ is regular since a regular language is closed under intersection. However $L_1 \cap L_a = \{a^i b^i | i \geq 0\}$, which is a nonregular language. Hence our assumption is wrong, so L_1 is not regular.

b) Is L_2 regular? Justify your answer.

Answer:

We construct a DFA $A = (Q, \Sigma, \delta, s, F)$ in which $Q = \{q_0, q_1, q_2, q_3, q_4\}, \Sigma = \{a, b\}, s = q_0, F = \{q_0, q_1, q_2\}$.

The transition functions are defined as following: $\delta(q_0, a) = q_1, \delta(q_1, a) = q_1, \delta(q_1, b) = q_4, \delta(q_4, b) = q_4, \delta(q_4, a) = q_1, \delta(q_0, b) = q_2, \delta(q_2, b) = q_2, \delta(q_2, a) = q_3, \delta(q_3, a) = q_3, \delta(q_3, b) = q_2$.



First we should notice that the difference between $|w|_{ab}$ and $|w|_{ba}$ is at most 1. Because $s \in F$, $\epsilon \in L(A)$. Additionally, $s \mapsto_A^{a^*} q_1$ and $s \mapsto_A^{b^*} q_2$ makes $L(A)$ accept a^* and b^* since $q_1, q_2 \in F$.

Now we assume for a string w for which $w \in L_2 \wedge 0 < |w| \leq n$, A accepts w. Then, we prove the same holds for w' that $w' \in L_2 \wedge |w'| = n + 1$.

For $w' = v\sigma$, A can distinguish the string v, $(|v| \geq 1)$.

- If A accepts v, A would be at state either q_1 or q_2.
  - If A is at state q_1 after reading v, the last symbol of v is a.
    - If σ is a, w' has the same ocurrence of ab, ba with v, so $w' \in L_2$. Since A terminates in q_1 after reading w', $w' \in L(A)$ too.
    - If σ is b, w' has one more occurrence of ab than v, so ab occurs one more time than ba in w' and thus $w' \notin L_2$. Since A terminates in q_4 after reading w', $w' \notin L(A)$ too.
  - If A is at state q_2 after reading v, the last symbol of v is b.
    - If σ is b, w' has the same ocurrence of ab, ba with v, so $w' \in L_2$. Since A terminates in q_2 after reading w', $w' \in L(A)$ too.

    * If σ is a, w' has one more occurrence of ba than v, so ba occurs one more time than ab in w' and thus $w' \notin L_2$. Since A terminates in q_3 after reading w', $w' \notin L(A)$ too.

- If A does not accept v, A would be at state either q_4 or q_3.
  - If A is at state q_4 after reading v, the last symbol of v is b.
    * If σ is a, w' has one more occurrence of ba than v, so ba occurs the same times as ab in w' and thus $w' \in L_2$. Since A terminates in q_1 after reading w', $w' \in L(A)$ too.
    * If σ is b, w' has the same ocurrence of ab, ba with v, so $w' \notin L_2$. Since A terminates in q_4 after reading w', $w' \notin L(A)$ too.
  - If A is at state q_3 after reading v, the last symbol of v is a.
    * If σ is b, w' has one more occurrence of ab than v, so ab occurs the same times as ba in w' and thus $w' \in L_2$. Since A terminates in q_2 after reading w', $w' \in L(A)$ too.
    * If σ is a, w' has the same ocurrence of ab, ba with v, so $w' \notin L_2$. Since A terminates in q_3 after reading w', $w' \notin L(A)$ too.

There we proved $L(A) = L_2$ for any length of string $w \in L_2$.