Concordia University

Dept. of Computer Science and Software Engineering

COMP 335 – Introduction to Theoretical Computer Science

Fall 2017

Solutions for $\mathcal{A}$ssignment 2

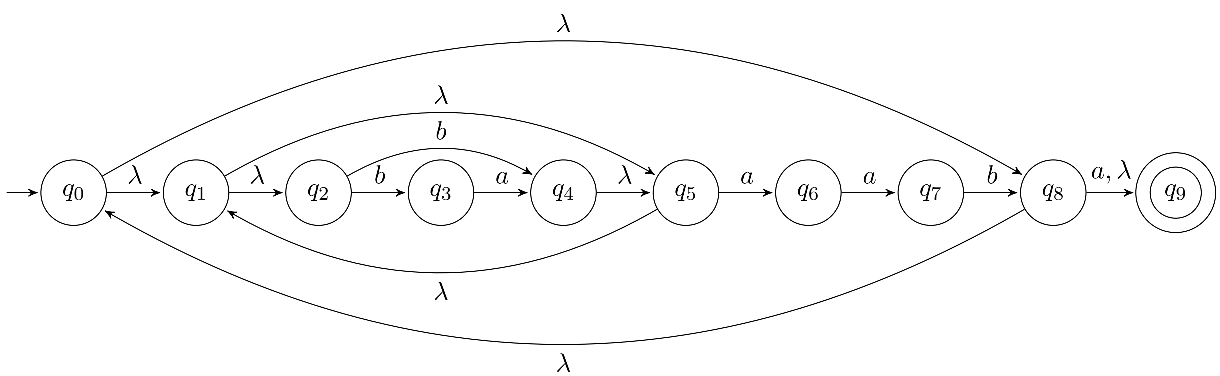
1. (10 points) Give regular expressions for the following languages on the alphabet $\{a, b\}$.

   (a) the set of all strings with an even number of a's
   Soln. $(b + ab^*a)^*$

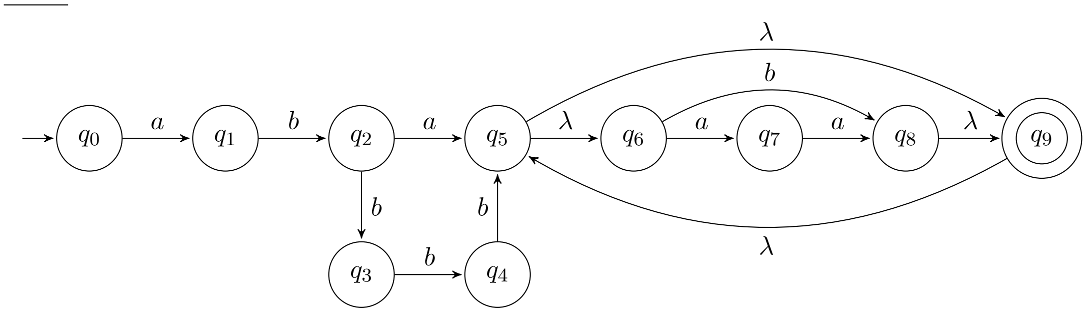
   (b) the set of strings in which all runs are of length < 3. (A run in a string is a non-extendable substring of length at least 2 which contains repetitions of the same symbol. For example (a) the string $abab$ has no runs, (b) the string $aabbbababbaaa$ has four runs, in order: a run of length 2 of a's, a run of length 3 of of b's, a run of length 2 of b's and a run of length 3 of a's.)
   Soln. $(b + bb + \lambda)(ab + aabb + aab + abb)^*(a + aa + \lambda)$

2. (10 points) For each of the following regular expressions r, convert it to to an NFA that accepts the language $L(r)$:

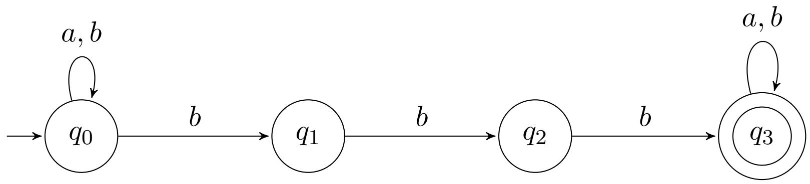
   (a) $((ba + b)^*aab)^*(a + \lambda)$
   Soln.



   (b) $ab(a + bbb)(b + aa)^*$
   Soln.



3. (10 points) Consider the following NFA M:



(a) Convert it to a DFA M' such that $L(M') = L(M)$.

Soln. We use the subset construction to derive the transition function of the DFA.

State	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0\}$	$\{q_0, q_1, q_2, q_3\}$
$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$\{q_0, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_2, q_3\}$

The set of final states for the DFA is $\{\{q_0, q_1, q_2, q_3\}, \{q_0, q_3\}.\{q_0, q_1, q_3\}\}$. Notice that this DFA can be minimized. The states $\{q_0, q_3\}$ and $\{q_0, q_1, q_3\}$ can be removed, and we can set $\delta(\{q_0, q_1, q_2, q_3\}, a) = \{q_0, q_1, q_2, q_3\}$ instead.

(b) Convert it to a right-linear grammar G such that $L(G) = L(M)$.

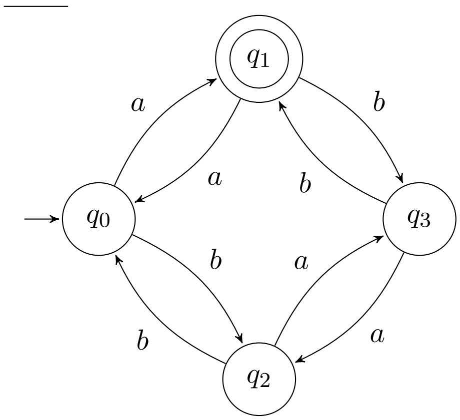
Soln. For convenience, we map the states q_0, q_1, q_2, q_3 to the variables S, A, B, C respectively, with S the start variable of the grammar. The productions of the desired grammar are:

$S \longrightarrow aS \mid bS \mid bA$
$A \longrightarrow bB$
$B \longrightarrow bC$
$C \longrightarrow aC \mid bC \mid \lambda$

4. (10 points) Let $L \subseteq \{a, b\}^*$ be the set of all strings with an odd number of a's and an even number of b's.

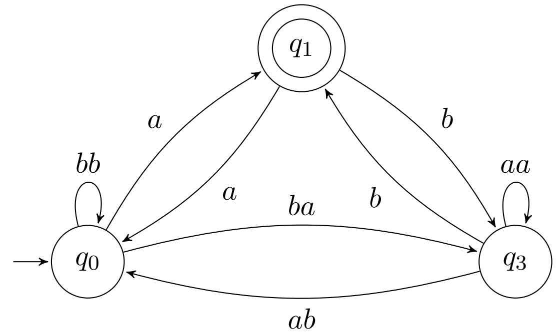
(a) Give a DFA M that accepts L.

Soln.

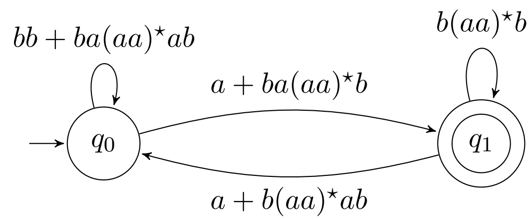


(b) Convert M to a regular expression r such that $L(M) = L(r)$.

Soln. Note that the DFA above has a single final state, which is different from the start state. So we begin by removing state q_2, yielding the extended NFA below.



Next we remove state q_3 to obtain the following:



Finally, this yields the regular expression
$(bb + ba(aa)^\star ab)^\star(a + ba(aa)^\star b)\,(b(aa)^\star b + (a + b(aa)^\star ab)(bb + ba(aa)^\star ab)^\star(a + ba(aa)^\star b))^\star$

5. (10 points) Let L be a regular language defined on the alphabet $\Sigma = \{a, b\}$. Which of the following languages is regular? Prove your answer.

(a) $f_2(L) = \{a_1a_1a_2a_2 \ldots a_na_n \; : \; a_1a_2 \ldots a_n \in L\}$

Soln. If L is regular, then $f_2(L)$ is regular. This can be seen in multiple ways.

i. Define a homomorphism $h(a) = aa$ and $h(b) = bb$. Then $f_2(L) = h(L)$. Since the regular languages are closed under homomorphisms, $f_2(L)$ must be regular, if L is regular.

ii. Since L is regular, there must exist a regular expression r such that $L = L(r)$. We can convert r to a new regular expression r' by replacing every occurrence of a in r by aa and every occurrence of b by bb. Clearly r' is a regular expression, and furthermore, $f_2(L) = L(r')$. Since $f_2(L)$ can be expressed by a regular expression, it must be regular.

iii. Since L is regular, there exists a DFA $M = (Q, \Sigma, q_0, \delta, F)$ that accepts L. From M, we can construct the following DFA $M' = (Q', \Sigma, q_0, F, \delta')$ that accepts $f_2(L)$. First we define $Q' = Q \cup \{[q, a] \mid q \in Q, a \in \Sigma\} \cup \{q_{trap}\}$, that is, we add new states corresponding to every state-symbol pair in M, as well as a new trap state. Further, $F' = F$. Finally, we define δ' as follows. If $\delta(q, a) = p$, then we define $\delta'(q, a) = [q, a]$, and $\delta'([q, a], a) = p$. Additionally, for every state $q \in Q$, $\delta'([q, a], b) = q_{trap}$ and $\delta'([q, b], a) = q_{trap}$. Finally $\delta'(q_{trap}, a) = \delta'(q_{trap}, b) = q_{trap}$.

It is easy to see from the definition of δ' that in even steps, M' is in states in Q or in q_{trap}, and in odd steps, it is in the new states of type $[q, a]$ or $[q, b]$ (for some $q \in Q$) or q_{trap}. This implies that M' only accepts even length strings. Notice also that processing any string with a prefix $a_1a_2 \ldots a_{2i-1}a_{2i}$ with $a_{2i-1} \neq a_{2i}$ puts the machine in q_{trap}. That is, only strings of the form $a_1a_1a_2a_2 \cdots a_na_n$ are accepted by M'.

We now give a proof by induction on n that for any $q, p \in Q$,

$$\delta^*(q, a_1a_2\cdots a_n) = p \text{ if and only if } \delta'^*(q, a_1a_1a_2a_2\cdots a_na_n) = p \tag{1}$$

For the basis, $n = 1$. Observe that by the definition of the new transition function δ', we have $\delta(q, a) = p$ if and only if $\delta'^*(q, aa) = \delta'([q, a], a) = p$, as needed. We now assume that the statement is true for strings of length n. Consider a string of length $n+1$. Suppose $\delta^*(q, a_1a_2\cdots a_n) = p$. Let q' be such that $\delta^*(q, a_1a_2\cdots a_n) = q'$, and $\delta(q', a_{n+1}) = p$. From the inductive hypothesis, we know that $\delta'^*(q, a_1a_1a_2a_2\cdots a_na_n) = q'$. Also, $\delta'^*(q', a_{n+1}a_{n+1}) = \delta([q', a_{n+1}], a_{n+1}) = p$. Therefore $\delta'^*(q, a_1a_1a_2a_2\cdots a_na_{n+1}a_{n+1}) = p$ as needed.
Conversely, suppose $\delta'^*(q, a_1a_1a_2a_2\cdots a_na_na_{n+1}a_{n+1}) = p$. Then there must be a state $q' \in Q$ so that $\delta'^*(q, a_1a_1a_2a_2\cdots a_na_n) = q'$, and $\delta'(q', a_{n+1}) = [q', a_{n+1}]$ and $\delta([q', a_{n+1}], a_{n+1}) = p$. It follows now from the inductive hypothesis that $\delta^*(q, a_1a_2\cdots a_n) = q'$ and from the definition of the transition function δ' that $\delta(q', a_{n+1}) = p$, leading to the conclusion that $\delta^*(q, a_1a_2\cdots a_n) = p$. This completes the proof of (1).
It follows from (1) that $a_1a_2a_n \in L(M)$ if and only if $a_1a_1a_2a_2\cdots a_na_n \in L(M')$. It remains only to recall that only strings of the form $a_1a_1a_2a_2\cdots a_na_n$ are accepted by M'. We conclude that $L(M') = f_2(L)$. Since $f_2(L)$ is accepted by a DFA, it is regular.

(b) $f_3(L) = \{ww \, : \, w \in L\}$
Soln. $f_3(L)$ is not necessarily regular. As a counter-example, consider $L = \Sigma^*$. Clearly L is regular, but $f_3(L) = \{ww \, : \, w \in \Sigma^*\}$ is not regular.

6. (10 points) For each of the following languages, say whether or not it is regular. Prove your answer.

(a) $L_1 = \{a^nb^n \, : \, n \bmod 3 = 0\}$
Soln. We claim that L_1 is not regular, and will show it using the pumping lemma. Suppose, for the purpose of contradiction, that L_1 is regular. Let m be the constant of the pumping lemma, and let n be the smallest multiple of 3 that is $\geq m$. Choose $w = a^nb^n$. Clearly $|w| > m$, and $w \in L_1$. Let $w = xyz$ with $|xy| \leq m$ and $y \geq 1$. Then y can contain only a's. Let $y = a^j$ where $1 \leq j \leq n$. Then $xz = a^{n-j}b^n \notin L_1$, a contradiction to the pumping lemma. Therefore, L_1 is not regular.

(b) $L_3 = \{a^n : n \text{ is not a perfect square}\}$ (an integer n is a perfect square if $n = i^2$ for some integer i).
Soln. L_3 is not regular. If L_3 is regular, then $L = \overline{L_3} = \{a^n \mid n \text{ is a perfect square }\}$ must also be regular, since the regular languages are closed under complementation. However, we will show that L is not regular. Suppose L is regular; then let m be the constant of the pumping lemma. Choose $w = a^{m^2}$. Clearly, $w \in L$ and $|w| > m$. Let $w = xyz$ with $|xy| \leq n$ and $y \geq 1$. Then $y = a^j$ where $1 \leq j \leq n$. We have

$$m^2 < m^2 + j \leq m^2 + m < m^2 + 2m + 1 = (m+1)^2$$

that is, $m^2 + j$ lies strictly between two consecutive perfect squares, and therefore it cannot itself be a perfect square. We conclude that $xy^2z = a^{m^2+j} \notin L$, a contradiction to the pumping lemma. Therefore, L cannot be regular, and consequently, the given language $L_3 = \overline{L}$ is not regular.

7. (10 points) Let L_1 and L_2 be any two languages over $\Sigma^\star$ where $\Sigma = \{a, b\}$. Prove or disprove:

(a) If $L_1 \subseteq L_2$ and L_2 is regular, then $L_2 - L_1$ is regular.
Soln. False. Take $L_1 = \{a^n b^n \mid n \geq 0\}$, and $L_2 = \{a, b\}^*$. Then $L_2 - L_1 = \overline{L_1}$ which is not regular, since L_1 is not regular.

(b) If $L_1 = L_1 L_2$ and $\lambda \notin L_2$, then $L_1 = \phi$
Soln. We consider the cases $L_2 = \phi$ and $L_2 \neq \phi$ separately. If $L_2 = \phi$, then $L_1 = L_1 L_2 = L_1 \phi = \phi$ as needed. Suppose instead that $L_2 \neq \phi$, and $\lambda \notin L_2$. Let m be the length of the shortest strings in L_2. Since $\lambda \notin L_2$, it must be that $m \geq 1$. We claim that $L_1 = \phi$. Suppose for the purpose of contradiction that $L_1 \neq \phi$. Then L_1 contains at least one string; consider a shortest string $w \in L_1$ and let n be the length of w. Since $L_1 = L_1 L_2$, any string in L_1, including w, can be written as the concatenation of a string in L_1 and a string in L_2. It follows that $|w| \geq n + m \geq n + 1$, a contradiction to the assumption that $|w| = n$. We conclude that $L_1 = \phi$.