CS21004 - Tutorial 4

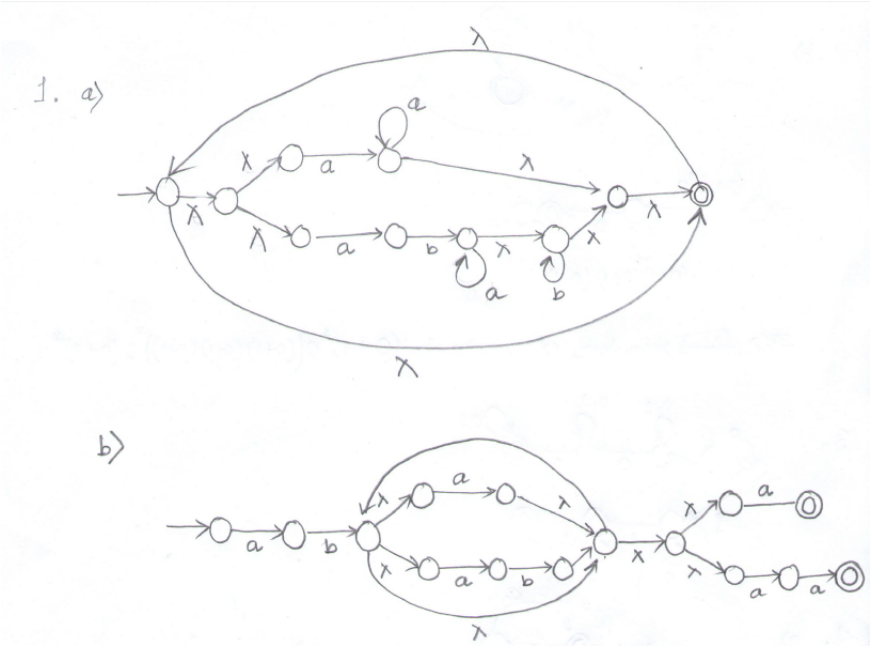
Solutions

Instructions: For the problems with (To submit), please write the answers neatly in loose sheets and submit to the TA before the end of the tutorial.

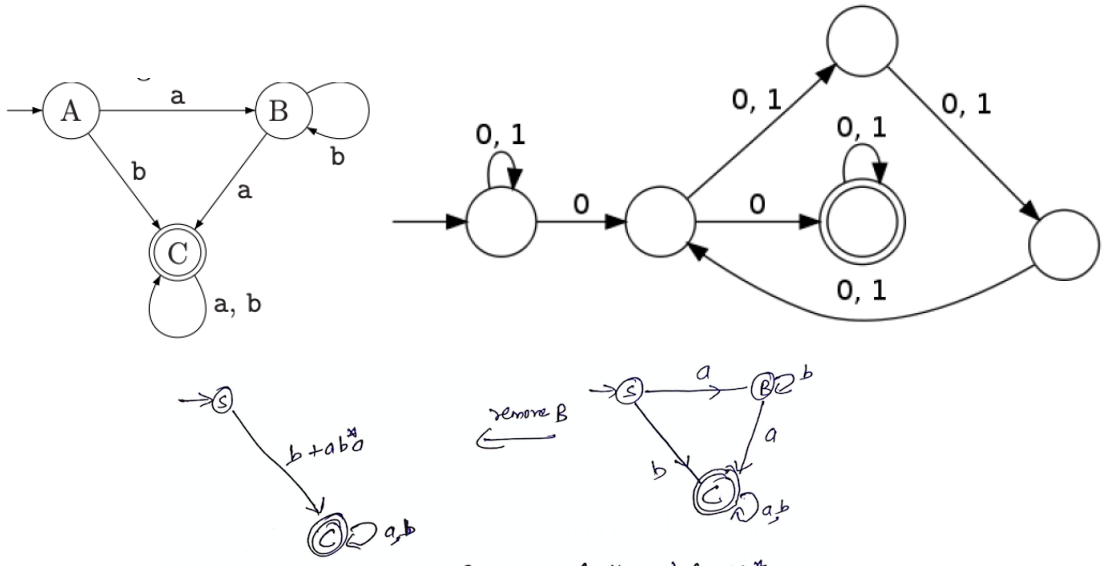
1. Design NFAs for the following regular expressions over $\Sigma = \{a, b\}$:

   a. $(aa^* + aba^*b^*)^*$

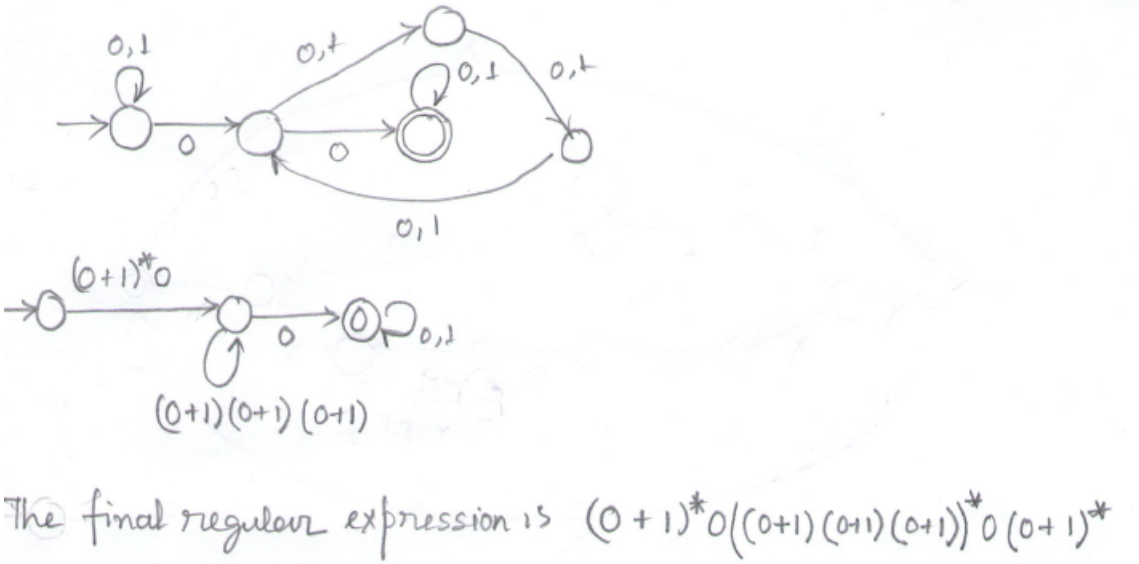
   b. $(ab(a + ab)^*(a + aa))$ (To submit)



2. Consider the following NFAs. Draw regular expressions corresponding to these. (Submit the second)



Solution:



3. Find the regular grammars for the following languages on $\{a, b\}$

   a. $L = \{w : n_a(w) \text{ and } n_b(w) \text{ are both even}\}$ (To submit)

   b. $L = \{a^n b^m : n \geq 2, m \geq 3\}$

Solution

a.

$$q_0 \to aq_1|bq_2|\lambda$$
$$q_1 \to bq_3|aq_0$$
$$q_2 \to aq_3|bq_0$$
$$q_3 \to aq_2|bq_1$$

b.

$$S \to aaA$$

$$A \to aA|B$$

$$B \to bbbC$$

$$C \to bC|\lambda$$

4. Find the regular expressions for the following languages on $\{a, b\}$

   a. $L = \{a^n b^m : n \geq 4, m \leq 3\}$.
   Solution: Generate 4 or more $a's$, follows by the requisite number of $b's$. Hence, $aaaaa^*(\lambda + b + bb + bbb)$

   b. The complement of L.
   Solution: A string not in L if it is of the form $a^n b^m$, with either $n < 4$ or $m > 3$, but this does not completely describe $\bar{L}$. We must also take in the strings in which a b is followed by an a. Hence, $(\lambda + a + aa + aaa)b^* + a^*bbbbb^* + (a + b)^*ba(a + b)^*$

   c. All strings that do not end with aa.
   Solution: $\lambda + a + b + (a + b)^*(ab + ba + bb)$

   d. All strings that contain an even number of b-s.
   Solution: $a^*(ba^*ba^*)^*$

   e. All strings which do not contain the substring ba.
   Solution: a^*b^*

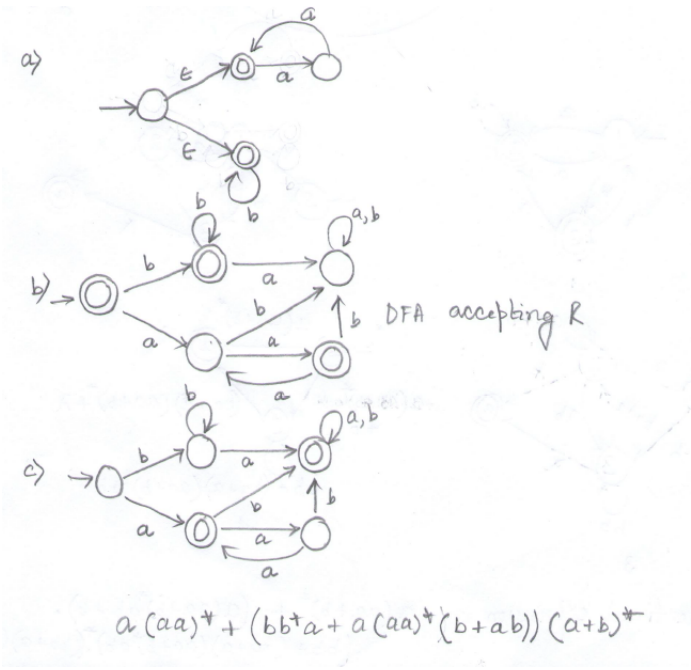
5. Consider the regular expression $R = (aa)^* + b^*$ (Home).

   a. Draw an NFA of the above regular expression with not more than 4 states.

   b. Draw the equivalent DFA.

   c. Find R' which recognizes the complement of language recognized by R.

Solution:



6. Provide an algorithm for converting a left linear grammar to a right linear grammar. (Home)

   Solution: Our algorithm assumes that the left linear grammar doesnt have any rule with the start symbol on the right hand side. If the left linear grammar has a rule with the start symbol S on the right hand side, simply add this rule: $S_0 \Rightarrow S$ to the given grammar and use the algorithm on the modified grammar with start symbol S_0. Let A, B denote non-terminal symbols, p denote zero or more terminal symbols, ϵ denote the empty symbol.

   (a) If the left linear grammar has a rule $S \to p$, then make that a rule in the right linear grammar

   (b) If the left linear grammar has a rule $A \to p$, then add the following rule to the right linear grammar: $S \to pA$

   (c) If the left linear grammar has a rule $B \to Ap$, add the following rule to the right linear grammar: $A \to pB$

   (d) If the left linear grammar has a rule $S \to Ap$, then add the following rule to the right linear grammar: $A \to p$

   Example : $\{S \to Ab,\ S \to Sb,\ A \to Aa,\ A \to a\} \Rightarrow \{S_0 \to S,\ S \to Ab,\ S \to Sb,\ A \to Aa,\ A \to a\} \Rightarrow \{S_0 \to aA,\ A \to bS,\ A \to aA,\ S \to bS,\ S \to \epsilon\}$