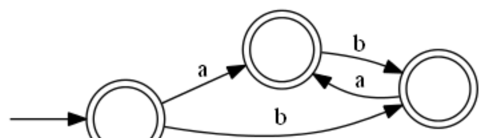
Informatics 2A 2015–16.
Tutorial Sheet 1 - SOLUTIONS

John Longley

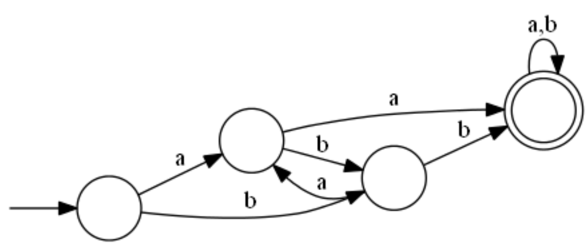
1. The NFAs given here are the 'simplest possible' –however, many other choices of regular expresssions would be equally reasonable.

   (a) NFA:

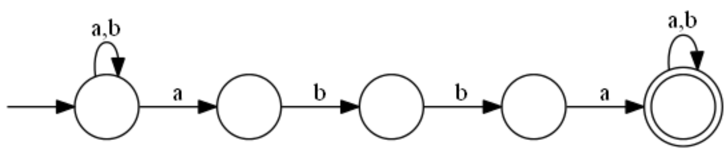
   Regular Expresssion: $(a+\epsilon)(ba)^*(b+\epsilon)$

   (b) NFA:

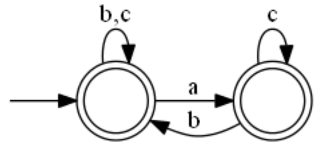
   Regular Expresssion: $(a+b)^*(aa+bb)(a+b)^*$

   (c) NFA:

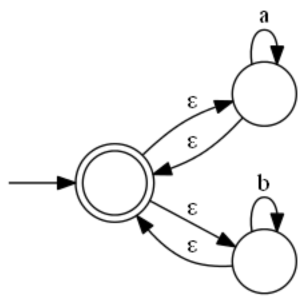
   Regular Expresssion: $(a+b)^*abba(a+b)^*$

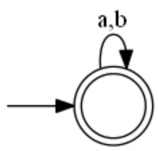
   (d) NFA:

   Regular Expresssion: $\mathcal{Z}(\epsilon+a(\mathcal{Z}b\mathcal{Z}a)^*)\mathcal{Z}$, where $\mathcal{Z}=(b+c)^*$

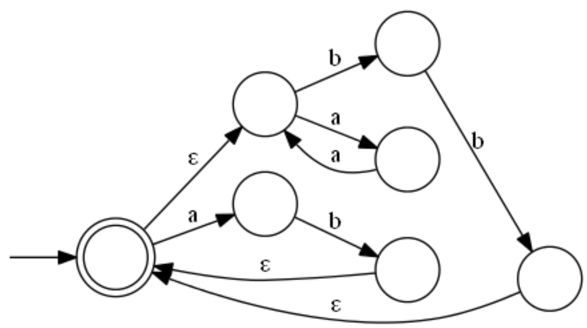
(e)



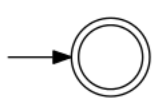
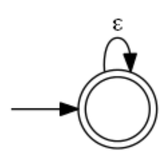
Actually 

would accept the same language, but I wouldn't regard it as following the structure of the regular expression.

(f)

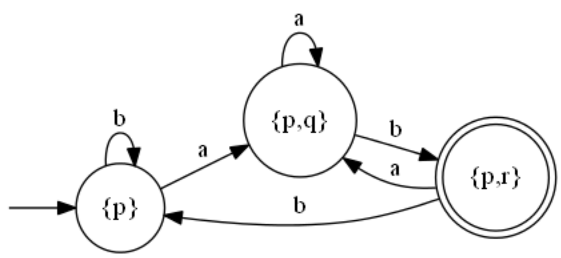


(g)

 or 

[Note that ϵ is the only string accepted.]

2. Three subsets of $\{p, q, r\}$ suffice:



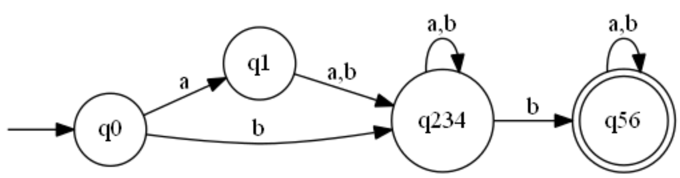
The best way to produce this is to start with the DFA start state $\{p\}$, and then explore the result of applying a and b transitions to states so far constructed, until no new DFA states (i.e. subsets of $\{p, q, r\}$) arise.

3. Suppose $N = (Q, \Delta, S, F)$ is an NFA for L. An NFA, N^r, for the reverse language L^r is produced by: making every start state in N into a final state in N^r; making every final state in N into a start state in N^r; and reversing every transition in N.

   More formally, define $\Delta^r = \{(q, a, q') \mid (q', a, q) \in \Delta\}$. Then the desired NFA for the reversal is $N^r = (Q, \Delta^r, F, S)$.

   Note that even if N is DFA its reversal N^r need not be.

4. The minimized DFA is:



   (**Correction:** please remove the b transition from $q234$ to itself. This self-loop should have an a label only.)

   Please obtain this using the algorithm presented in Lecture 4. However, it can also be obtained more directly by observing the following.

   - States $q5$ and $q6$ may be collapsed, since **any** string takes us from either of these to an accepting state.
   - States $q2$,$q3$ and $q4$ may all be collapsed, since the strings that takes us from these to an accepting state are those matching $a^*b(a+b)^*$.
   - Any other pair of states are differentiated by their behaviour on at least one of the strings: a, ϵ, b, ab.

5. (a) Different: 01 is in $\mathcal{L}((0+1)^*)$ but not $\mathcal{L}(0^* + 1^*)$.

   (b) The same: by the third identity with $a = 1$, $b = 20$,

$$\begin{array}{rcl} (120)^*1 & = & 1(201)^* \\ \text{where} \quad 0(120)^*12 & = & 01(201)^*2 \end{array}$$

(c) Different: 0 is in $\mathcal{L}((0^*1^*)^*)$ but not $\mathcal{L}((0^*1)^*)$.

(d) The same: $(01+0)^*0$
$= (0(1+\epsilon))^*0$ by second indentity and '$a\epsilon = a$'.
$= 0((1+\epsilon)0)^*$ by third indentity.
$= 0(10+0)^*$ by first indentity and '$\epsilon a = a$'.

6. (a) The required language is X_p, where

$$X_p = aX_p + bX_q \quad (1)$$
$$X_q = (a+b)X_q + \epsilon \quad (2)$$

Solving these:

$$\begin{array}{lll} X_q & = (a+b)^* & \text{from (2) by Arden's rule} \\ X_p & = aX_p + b(a+b)^* & \text{substituting in (1)} \\ X_p & = a^*b(a+b)^* & \text{by Arden's rule.} \end{array}$$

(b) The required language is X_p, where

$$X_p = bX_p + aX_q + \epsilon \quad (3)$$
$$X_q = bX_p + aX_r \quad (4)$$
$$X_r = (a+b)X_q \quad (5)$$

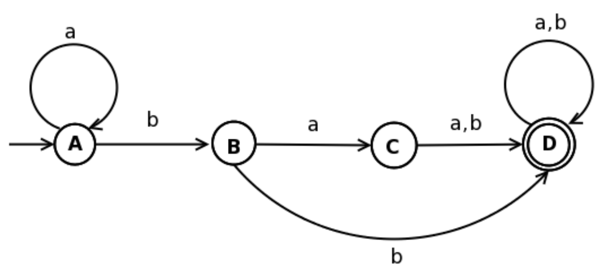
Solving these:

$$\begin{array}{lll} X_q & = bX_p + a(a+b)X_q & \text{substituting (3) in (2)} \\ X_q & = (a(a+b))^*bX_p & \text{by Arden's rule} \\ X_p & = bX_p + a(a(a+b))^*bX_p + \epsilon & \text{substituting in (1)} \\ & = (b + a(a(a+b))^*b)X_p + \epsilon & \text{by distributivity law} \\ & = (b + a(a(a+b))^*b)^* & \text{by Arden's rule.} \end{array}$$

7. The procedure followed in this question is known as Brzozowski's minimization algorithm. It works because, somewhat miraculously, steps (a) and (b) together, always produce the minimized DFA for the reversal of the language accepted by the original DFA.

The answer to the question at the end, after two passes at (a) and (b), is that the resulting DFA is isomorphic to the one obtained as the solution to Question 4.

At the intermediate stage, i.e., after the first pass of steps (a) and (b), you should have obtained the DFA below.



Here $A = \{q5, q6\}$, $B = \{q2, q3, q4, q5, q6\}$, $C = \{q1, q2, q3, q4, q5, q6\}$ and $D = \{q0, q1, q2, q3, q4, q5, q6\}$.