Theory of Algorithms. Spring 2000. Homework 4 Solutions.

Section 3.1

(1) $b, ab, bb, ba, aab, abb, bab, bbb, aba, bba, baa$. (All strings that contain at least one b.)

(5) (a) $aaaaa^*(\lambda + b + bb + bbb)$.

(b) $(\lambda + a + aa + aaa)(\lambda + b(a + b)^*) + aaaaa^*b(bbb + b^*a)(a + b)^*$.

(6) $\emptyset^0 = \{\lambda\}$ so $\emptyset^* = \{\lambda\}$ so $(\emptyset^*)^* = \{\lambda\}^* = \{\lambda\}$. $a\emptyset = \{\, aw \mid w \in \emptyset \,\} = \emptyset$.

(8) $(a + ba)^*b(a + b)^*$.

(12) $aa(a + b)^*aa + ab(a + b)^*ab + ba(a + b)^*ba + bb(a + b)^*bb$.

(14) (a) $(b + c)^*a(b + c)^*$.

(b) $(b + c)^*(a + \lambda)(b + c)^*(a + \lambda)(b + c)^*(a + \lambda)(b + c)^*$.

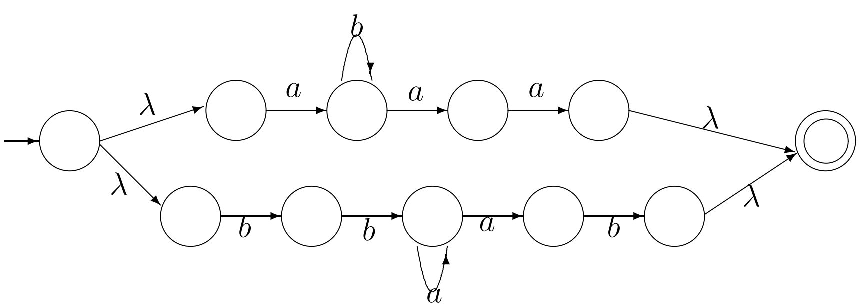
(c) $(a+b+c)^*a(a+b+c)^*b(a+b+c)^*c(a+b+c)^* + (a+b+c)^*a(a+b+c)^*c(a+b+c)^*b(a+b+c)^* + (a+b+c)^*b(a+b+c)^*a(a+b+c)^*c(a+b+c)^* + (a+b+c)^*b(a+b+c)^*c(a+b+c)^*a(a+b+c)^* + (a+b+c)^*c(a+b+c)^*b(a+b+c)^*a(a+b+c)^* + (a+b+c)^*c(a+b+c)^*a(a+b+c)^*b(a+b+c)^*$.

(d) $((\lambda + a + aa)(b + c))^*(\lambda + a + aa)$.

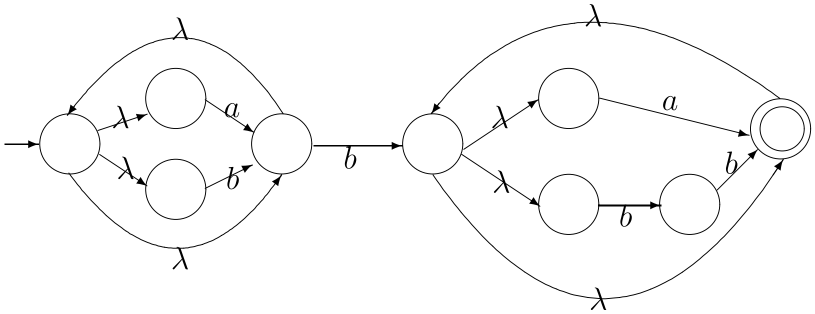
(e) $(aaa + b + c)^*$.

Section 3.2

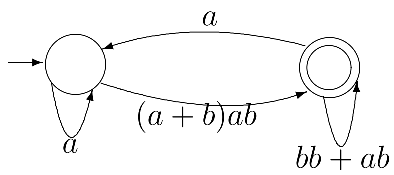
(1) As I mentioned in class, I want you to find nfa's for the small sub-expressions by hand, and then use the techniques of the proof to combine the smaller nfa's into a larger nfa. Thus there will not be a unique solution. One possible solution is the following:



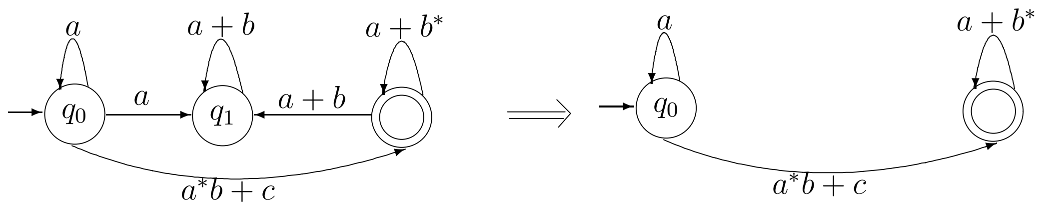
(3)



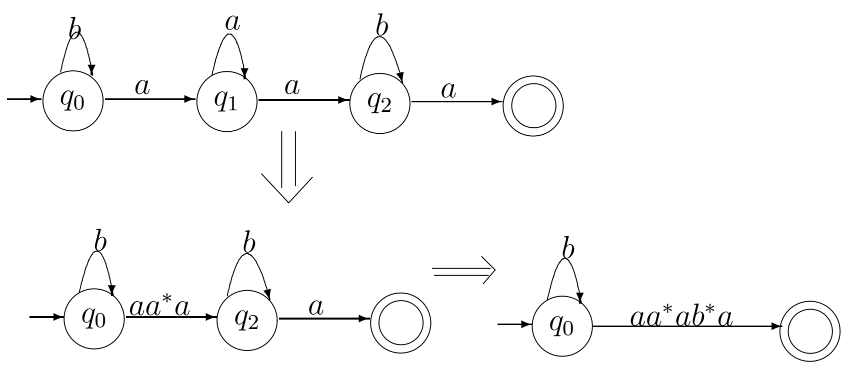
(7) $L(r)$ where $r = a^*(a+b)ab\big((bb+ab)+aa^*(a+b)ab\big)^*$.



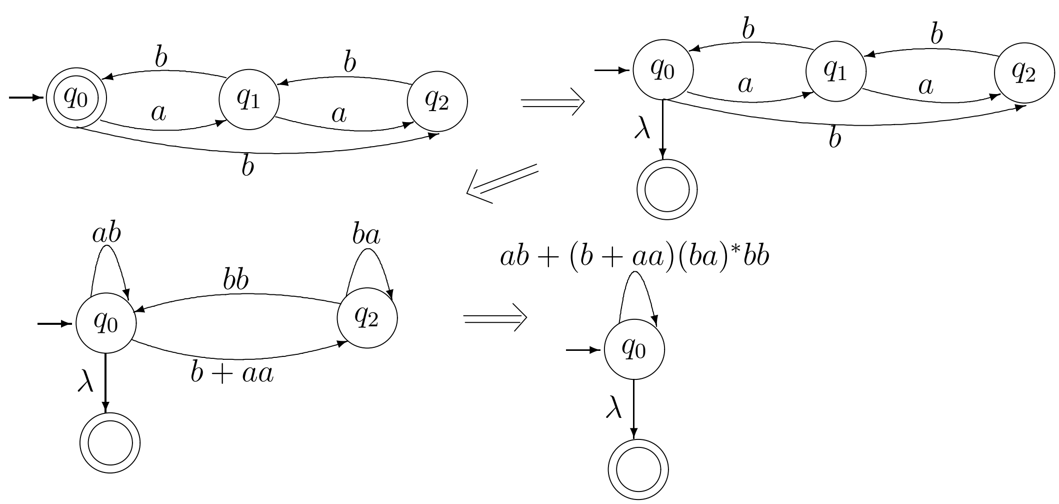
(8) $L(r)$ where $r = a^*(a^*b+c)(a+b^*)^*$.



(9) **(a)** $b^*aa^*ab^*a$.



(b) $\big(ab+(b+aa)(ba)^*bb\big)^*$.



(9c) $a^*\big[b + (b+ba)(aa)^*(\lambda + a)\big]$. Notice that this is equivalent to the simpler expression: a^*ba^*. So our technique for transforming nfa'a into regular expressions does not necessarily produce an *efficient* regular expression.