Formal Language Selected Homework Chapter 3.2

3. Give an nfa that accepts the language $L\left((a+b)^* b\,(a+bb)^*\right)$.

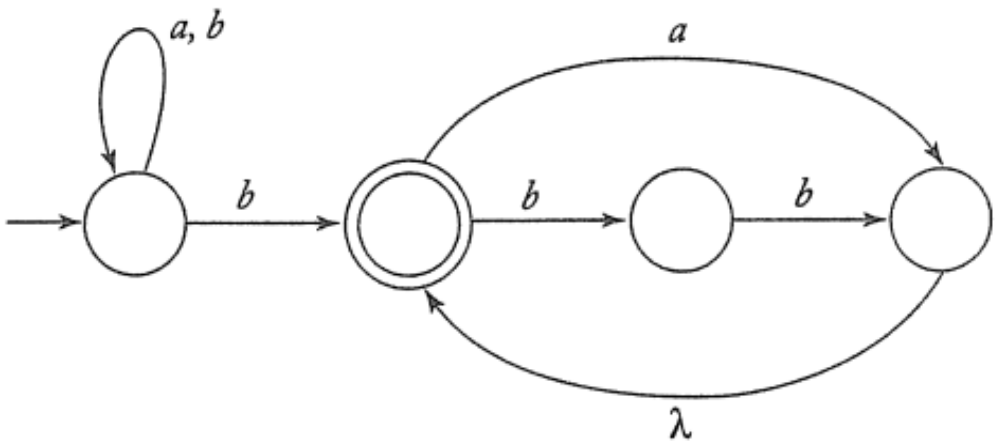
4. Find dfa's that accept the following languages.

(a) $L\left(aa^* + aba^*b^*\right)$.

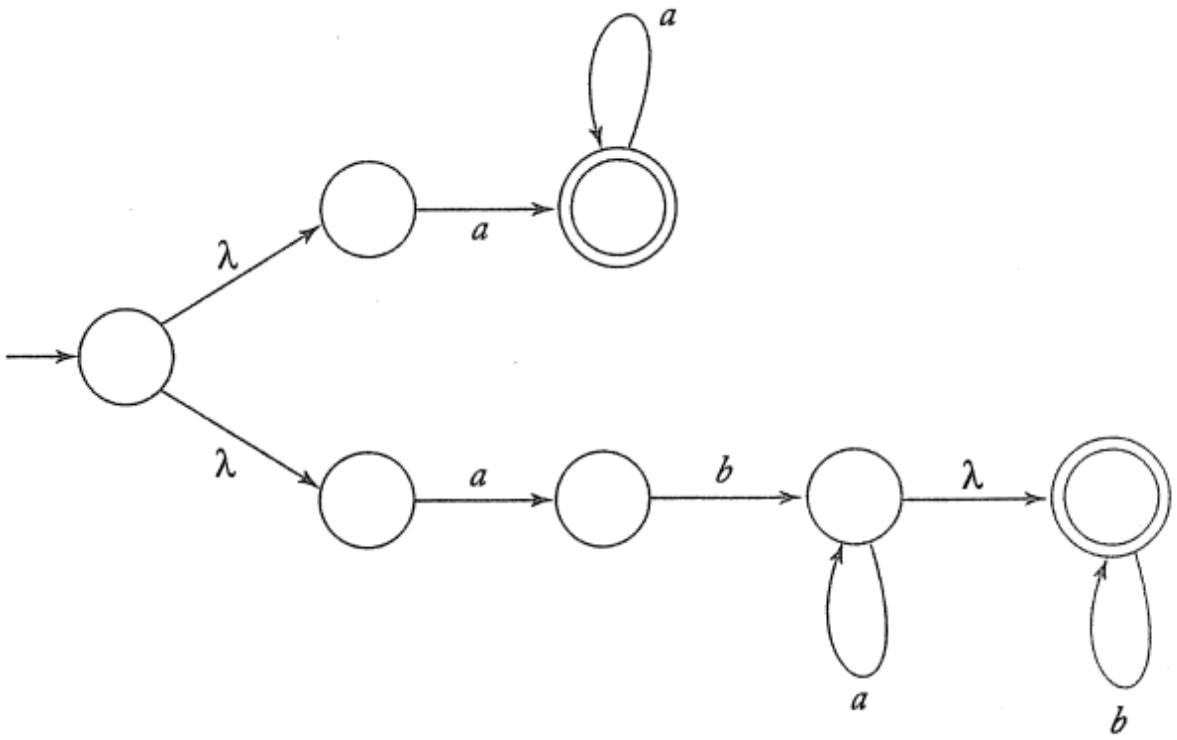
(b) $L\left(ab\,(a+ab)^*\,(a+aa)\right)$.

(c) $L\left((abab)^* + (aaa^* + b)^*\right)$.

3. This can be solved from first principles without going through the regular expression to nfa construction. The latter will, of course, work but gives a more complicated answer. Solution:

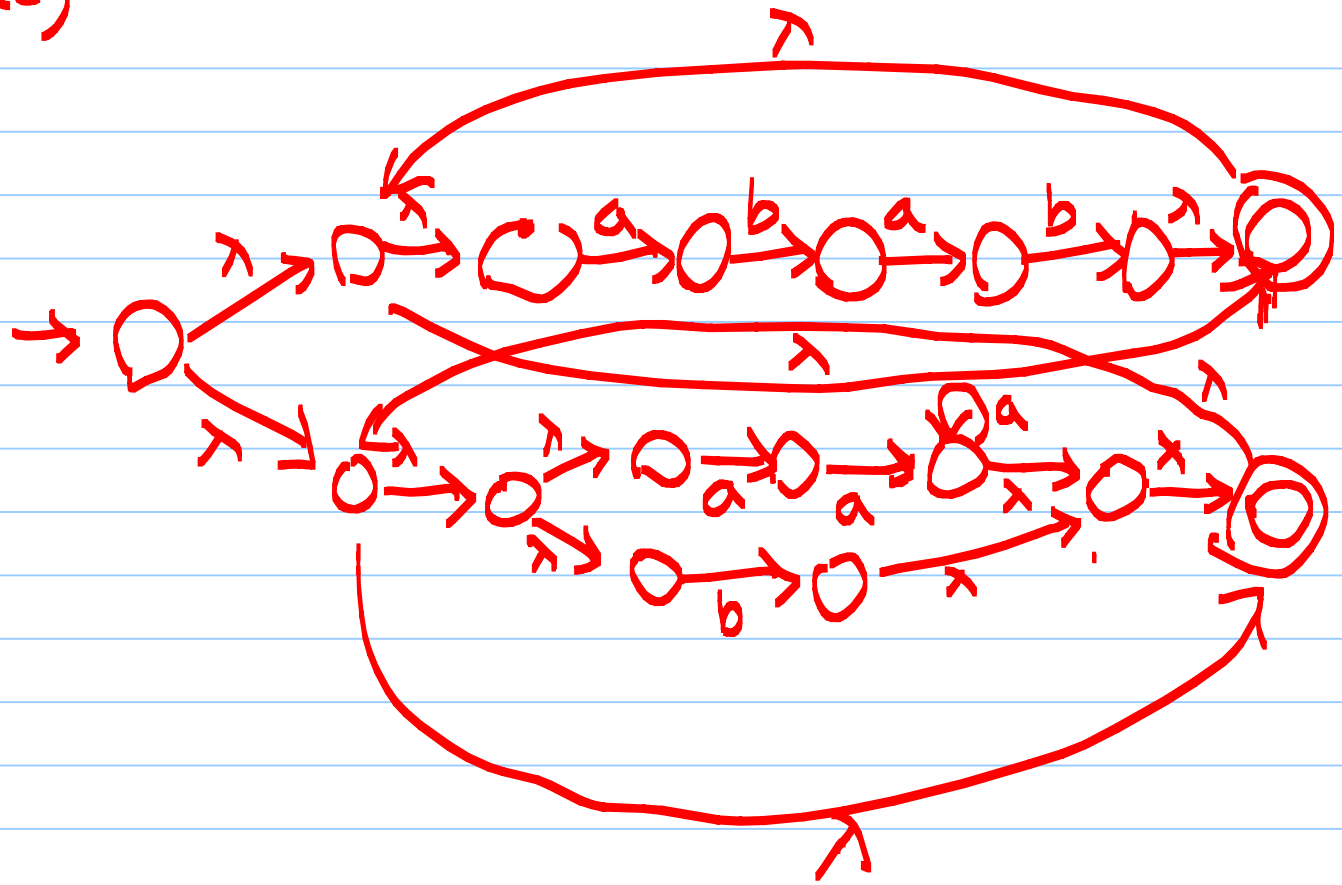


4. (a) Start with



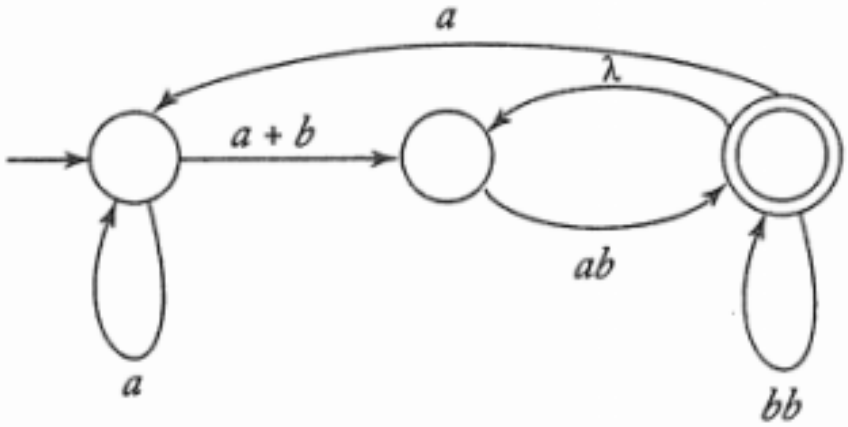
Then use the nfa to dfa algorithm in a routine manner. (Skip!)

(c)



Skip dfa!

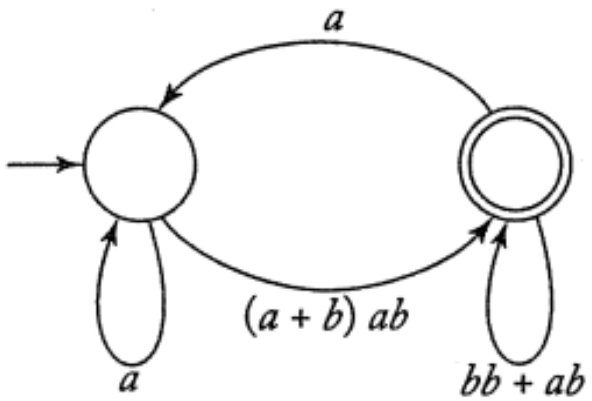
8. Consider the following generalized transition graph.



(a) Find an equivalent generalized transition graph with only two states.

(b) What is the language accepted by this graph?

8. Removing the middle vertex gives

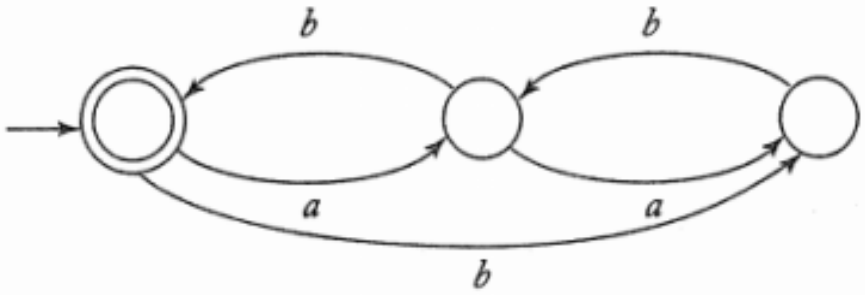


By Equation (3.1), the language accepted then is $L(r)$, where

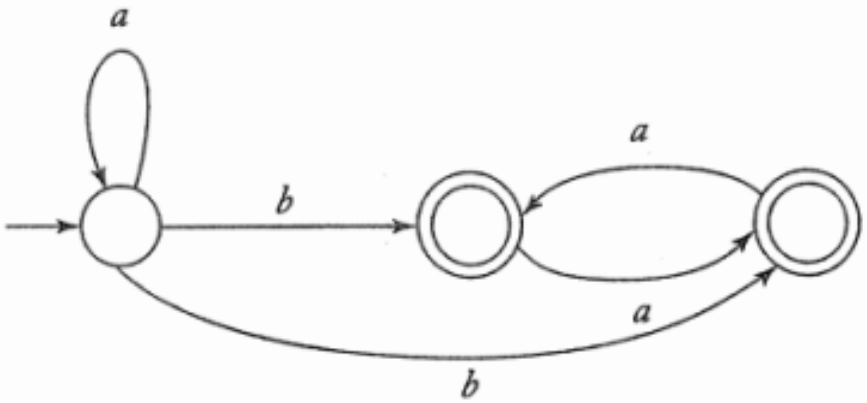
$$r = a^*(a+b)ab(ab+bb+aa^*(a+b)ab)^*.$$

10. Find regular expressions for the languages accepted by the following automata.

(b)

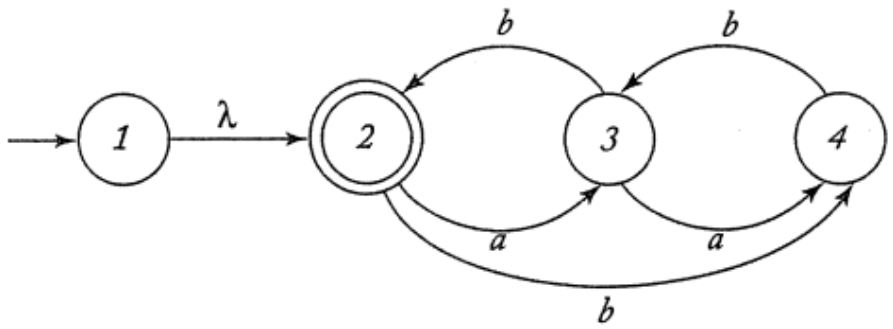


(c)

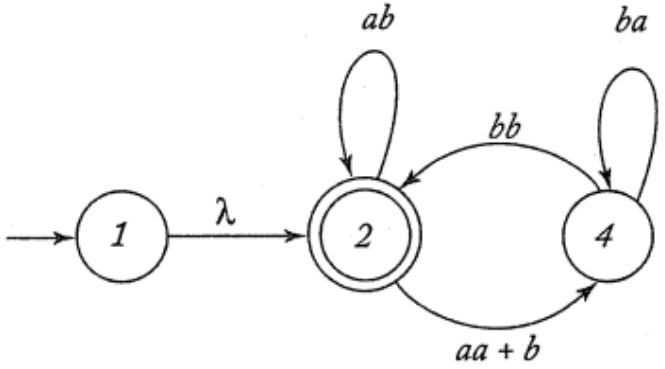


Sol.

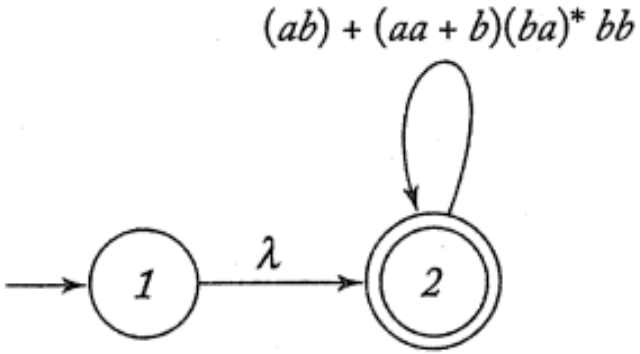
10. (b) First, we have to modify the nfa so that it satisfies the conditions imposed by the construction in Theorem 3.2, one of which is $q_0 \notin F$. This is easily done.



Then remove state 3.



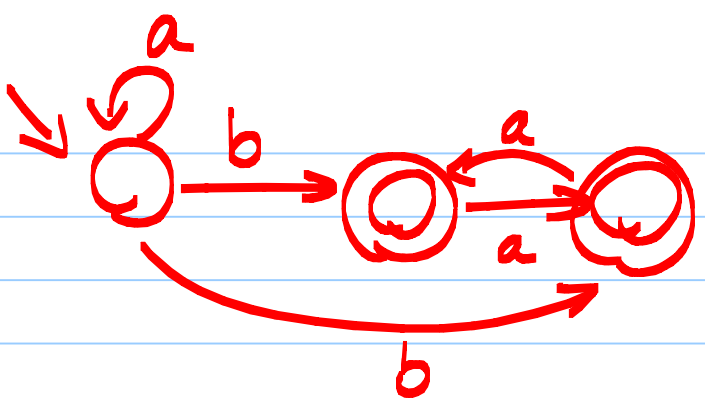
Next, remove state 4.



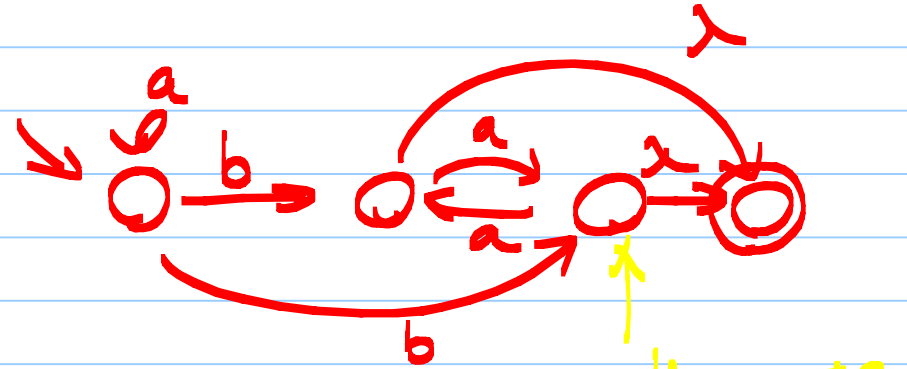
The regular expression then is $r = \left(ab + (aa + b)\,(ba)^*\,bb^*\right)^*$.

Sol. 10. (c)

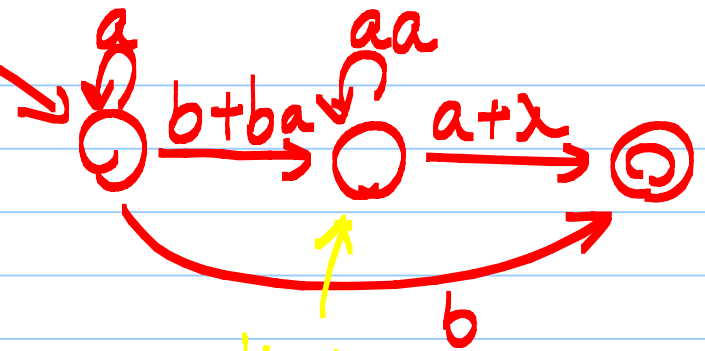
1.



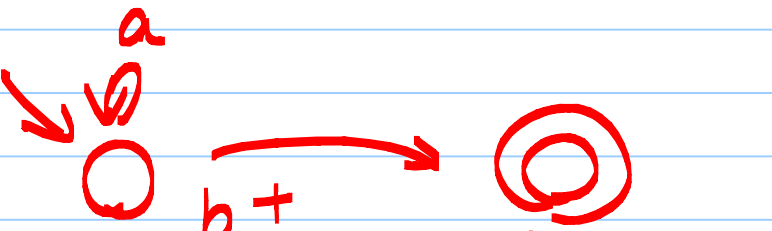
2.



3.



4.



RE

$= a^*(b+(b+ba)(aa)^*(a+\lambda))$

$= a^*(b+b(aa)^*a+ba(aa)^*a$
$+b(aa)^* + ba(aa)^*)$

$= a^*ba^*$

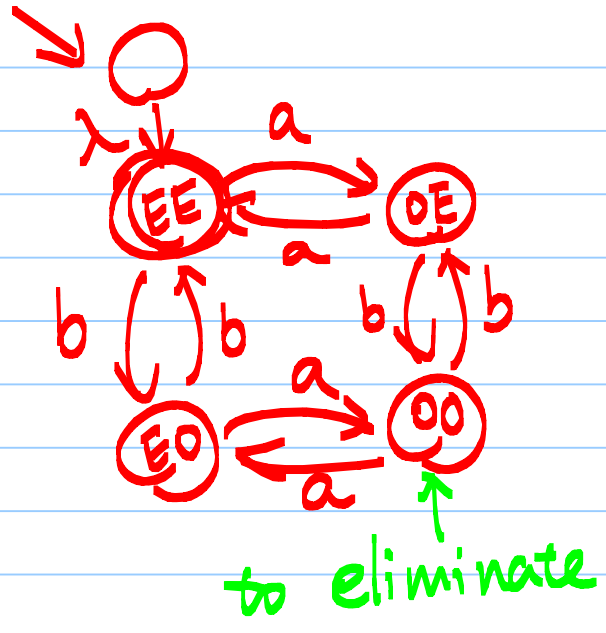
13. Find a regular expression for the following languages on $\{a, b\}$.

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

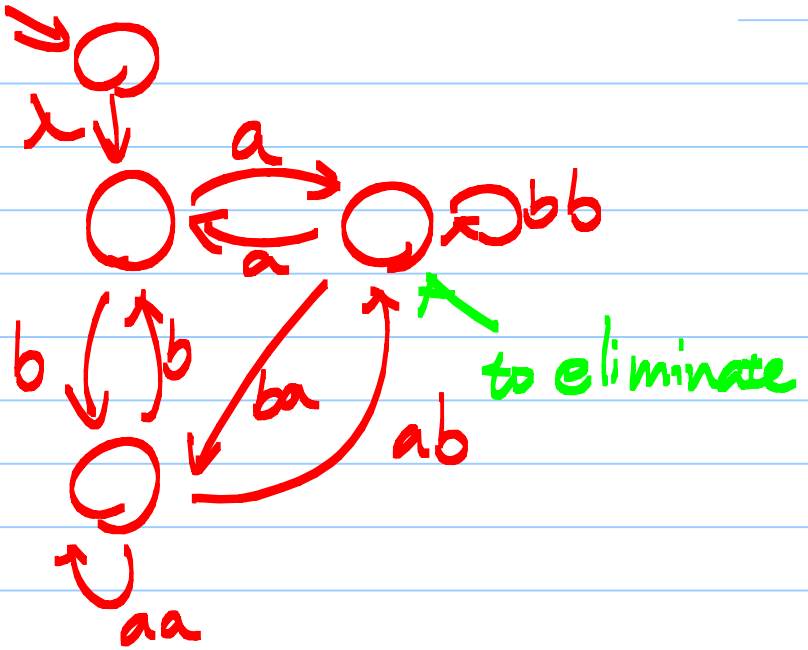
(b) $L = \{w : (n_a(w) - n_b(w)) \bmod 3 = 1\}$.

Sol. 13 (a)

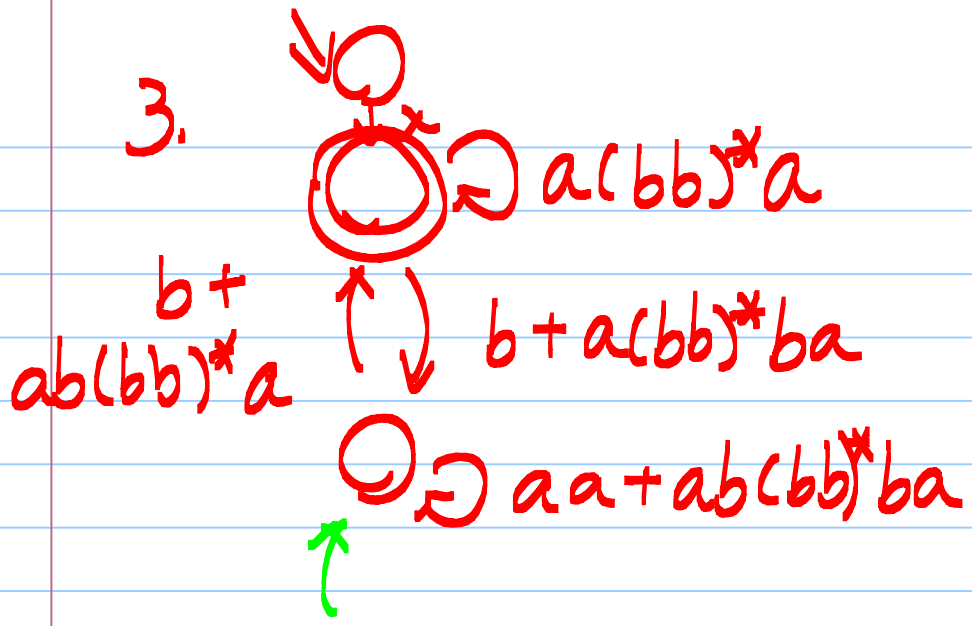
1.



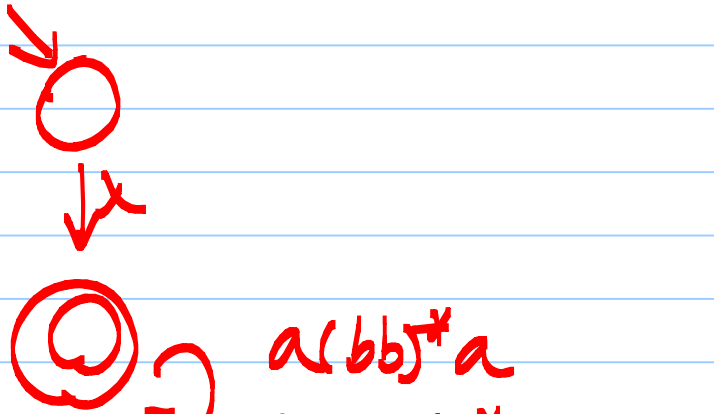
2.



3.



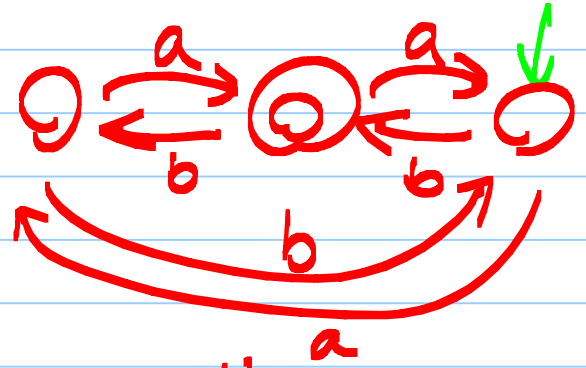
4



$$RE = (a(bb)^*a + (b+a(bb)^*ba)(aa+ab(bb)^*ba)^*(b+ab(bb)^*a)^*$$

Sol. 13 (b)

1.

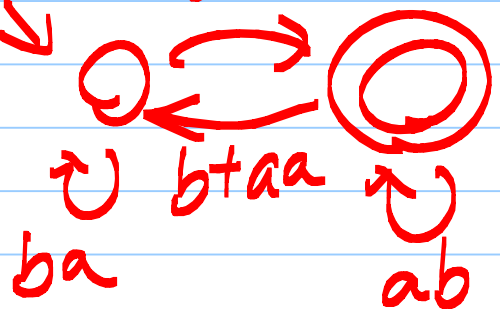


to eliminate

RE =

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

2



16. In some applications, such as programs that check spelling, we may not need an exact match of the pattern, only an approximate one. Once the notion of an approximate match has been made precise, automata theory can be applied to construct approximate pattern matchers. As an illustration of this, consider patterns derived from the original ones by insertion of one symbol.

Let L be a regular language on Σ and define

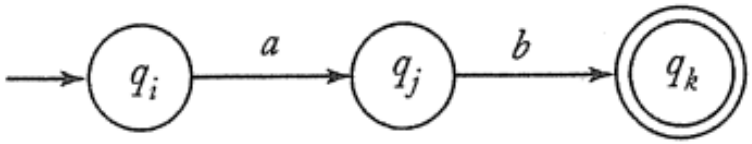
$$insert\,(L) = \{uav : a \in \Sigma, uv \in L\}\,.$$

In effect, $insert\,(L)$ contains all the words created from L by inserting a spurious symbol anywhere in a word.

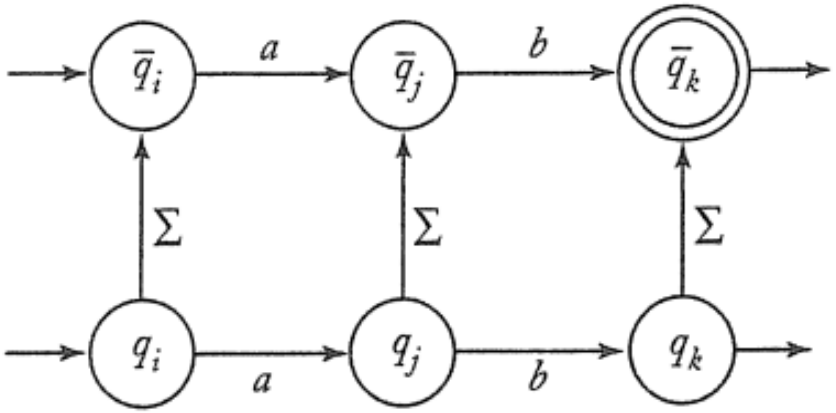
★ (a) Given an nfa for L, show how one can construct an nfa for $insert\,(L)$.

Sol.

16. (a) This is a hard problem until you see the trick. Start with a dfa with states $q_0, q_1, \ldots$, and introduce a "parallel" automaton with states $\overline{q}_0, \overline{q}_1, \ldots$. Then arrange matters so that the spurious symbol nondeterministically transfers from any state of the original automaton to the corresponding state in the parallel part. For example, if part of the original dfa looks like



then the dfa with its parallel will be an nfa whose corresponding part is



It is not hard to make the argument that the original dfa accepts L if and only if the constructed nfa accepts $insert\,(L)$.