Formal Language Selected Homework Chapter 3.2

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

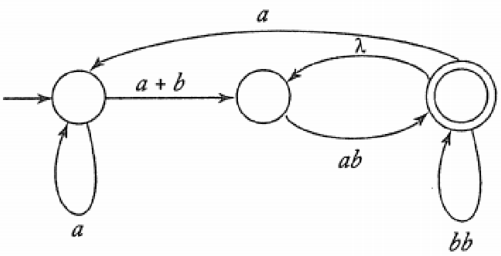
4. Find dfa's that accept the following languages.

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

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

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

8. Consider the following generalized transition graph.

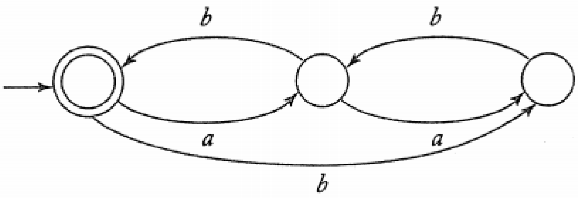


(a) Find an equivalent generalized transition graph with only two states.

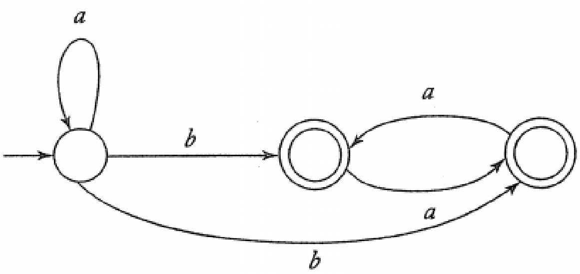
(b) What is the language accepted by this graph?

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

(b)



(c)



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

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 $insert(L)$.