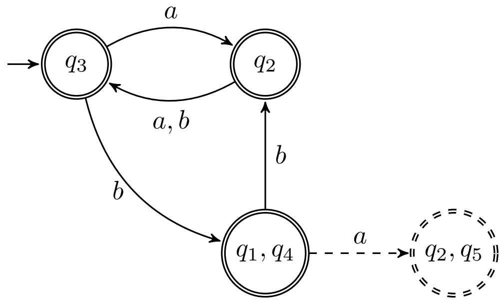
Automata and Formal Languages — Sample Solution 5

Due 13.11.2015

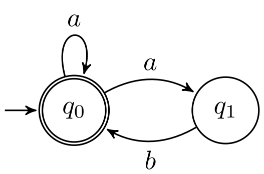
Solution 5.1

Perform *NFAtoDFA* and check on-the-fly whether generated states are final. The subsumption test prevents adding (q_2, q_5) to the worklist.

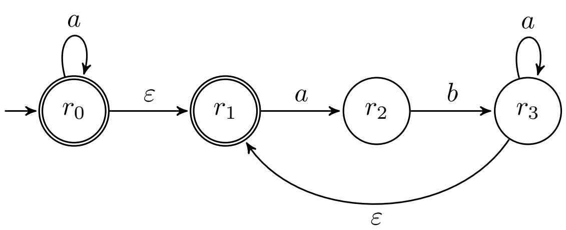


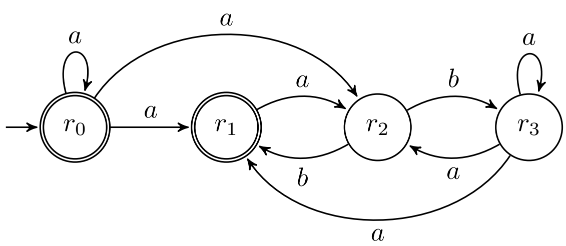
Solution 5.2

(a) NFA A_1 for $(a + ab)^*$

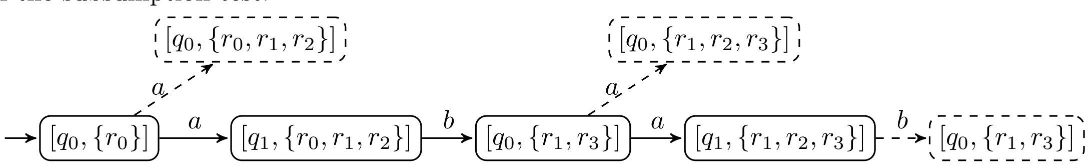


NFA-ε and NFA A_2 for $a^*(aba^*)^*$

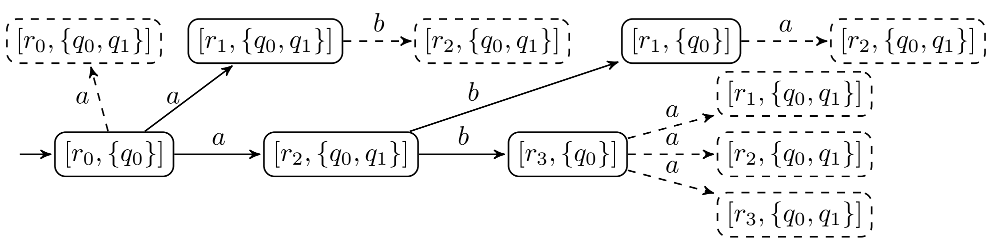




Run *InclNFA(A_1, A_2)*. The following graph illustrates the algorithm execution, where each node represents a state $[q_1, Q_2]$ and edge $u \stackrel{a}{\to} v$ means that v can be constructed from u using the letter a. Dashed nodes are ignored because of the subsumption test.

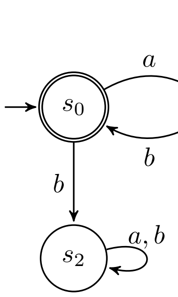


Run *InclNFA(A_2, A_1)*:

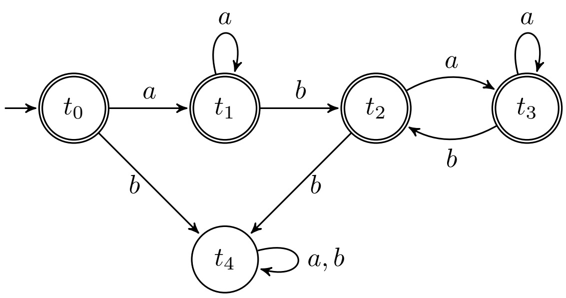


For each state $[q_1, Q_2]$, whenever q_1 is a final state, Q_2 always contains a final state. So, $L(A_1) \subseteq L(A_2)$, $L(A_2) \subseteq L(A_1)$, and therefore $L(A_1) = L(A_2)$.

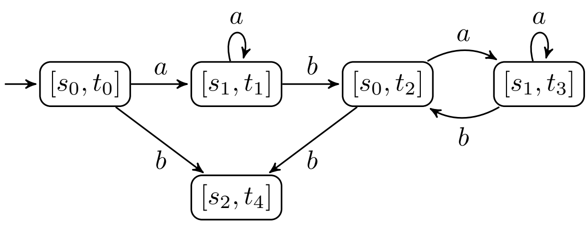
(b) DFA $B_1 = \mathit{NFAtoDFA}(A_1)$:



DFA $B_2 = \mathit{NFAtoDFA}(A_2)$:

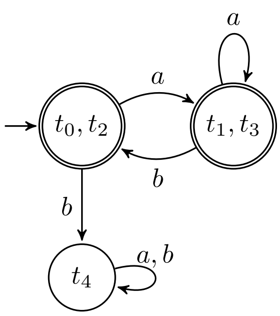


The algorithm execution:



For each state $[q_1, q_2]$, q_1 is final if and only if q_2 is final. Therefore, $L(B_1) = L(B_2)$.

(c) By executing $\mathit{LanPar}(B_2)$, we have $P_l = \{\{t_0, t_2\}, \{t_1, t_3\}, \{t_4\}\}$. Therefore, B_2/P_l can be constructed as follows:



which is obviously isomorphic to B_1.

Solution 5.3

$p = a^n$ and $t = a^{n-1}b$.

Solution 5.4

Use the pattern matching algorithm to search for the pattern $p = w'$ in the text $t = ww$.