University of Nevada, Las Vegas Computer Science 456/656 Fall 2019

Answers to Assignment 2: Due Wednesday September 4, 2019

1. Problem 11(d) on page 38 of the fifth edition, problem 14(d) on page 29 of the sixth edition.

   If $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA, we can transform M to a regular grammar G which generates $L(M)$, as follows. there is a construction which uses M to define a regular gammar for L. There is one variable of G for each state of M, and there is one production of G for each transition (arc) of M and one production for each final state of M.

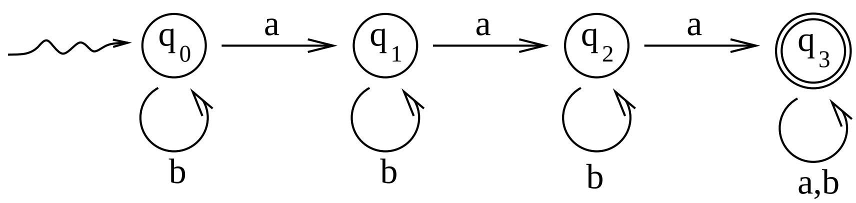
   (a) Let $Q = \{q_0, q_1, \ldots q_k\}$. The variables of G will be $A_0, A_1, \ldots A_k$, and A_0 is the start symbol.

   (b) If M has a transition $q_i \xrightarrow{a} q_j$ for $a \in \Sigma$, then G has a production $A_i \to aA_j$.

   (c) If M has a transition $q_i \xrightarrow{\lambda} q_j$, then G has a production $A_i \to A_j$.

   (d) If $q_i \in F$, then G has a production $A_i \to \lambda$.

   The language consisting of all strings over $\Sigma = \{a, b\}$ with at least 3 a's is accepted by the following DFA (which is, of course, also an NFA):



   The grammar G obtained from M by the transformation given in problem 1 has the productions:

   $A_0 \to bA_0 | aA_1$

   $A_1 \to bA_1 | aA_2$

   $A_2 \to bA_2 | aA_3$

   $A_3 \to aA_3 | bA_3 | \lambda$

2. Problem 14(a) on page 39 of the fifth edition,

   $S \to aSb$

   $S \to Tb$

   $T \to Tb$

   $T \to \lambda$

3. problem 17(h) on page 29 of the sixth edition. We introduce a third variable, S', which plays the same role as S does in the previous problem. The new start symbol S then generates arbitrarily many copies of S', each of which generates a member of L_1.

   $S \to S'S$

   $S \to \lambda$

   $S' \to aS'b$

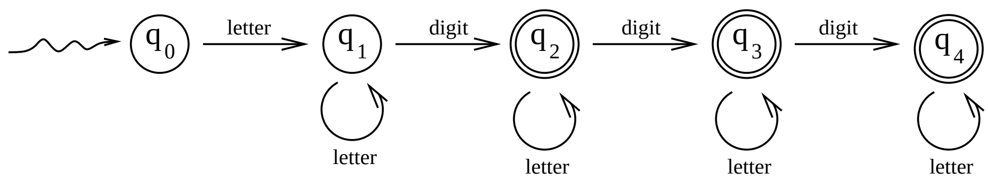
   $S' \to Tb$

   $T \to Tb$

   $T \to \lambda$

4. Problem 4 on page 44 of the fitth edition, problem 6 on page 35 of the sixth edition.

   Instead of showing all 26 letters and ten digits, we simplify the figure by simply writing "digit" instead of $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, similarly for letters.



   We use the transformation given in problem 1, and we introduce two variables, L for letter and D for digit. Our grammar has variables $A_0, A_1, A_2, A_3, A_4, L, D$, and the start symbol is A_0.

   $A_0 \to LA_1$

   $A_1 \to DA_2|LA_1$

   $A_2 \to DA_3|LA_2|\lambda$

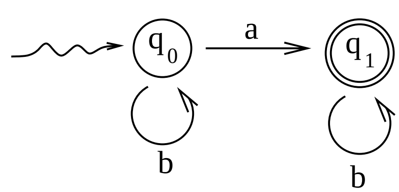
   $A_3 \to DA_4|LA_3|\lambda$

   $A_4 \to LA_4|\lambda$

   $L \to a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z$

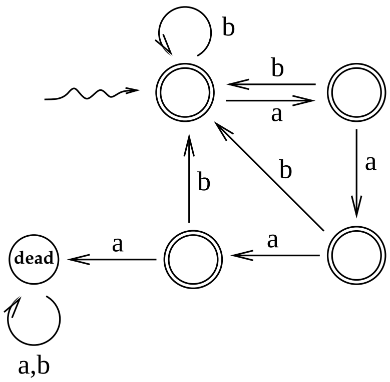
   $D \to 0|1|2|3|4|5|6|7|8|9$

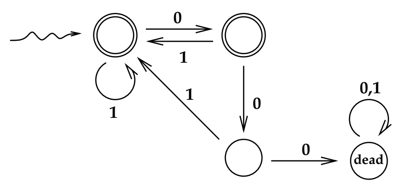
5. Problem 2(a) on page 56 of the fifth edition, problem 4(a) on page 48 of the sixth edition.

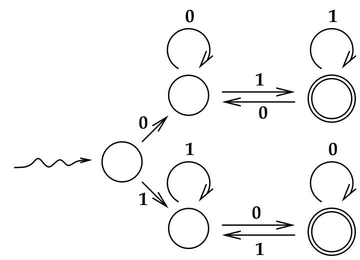


   There is a dead state, not shown in the figure.

6. Problem 8(b), 9(a), 9(c) on page 56 of the fifth edition, 8(b), 11(a), 11(c) on page 49 of the sixth edition.





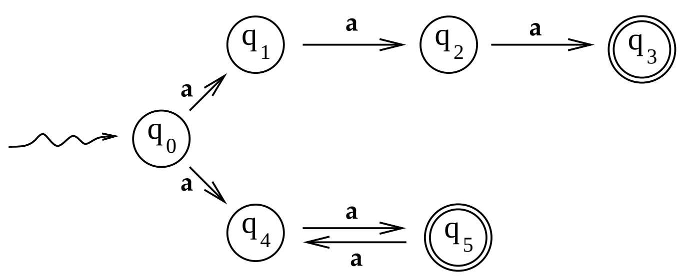


7. Problem 26 on page 58 of the fifth edition., problem 28 on page 51 of the sixth edition.

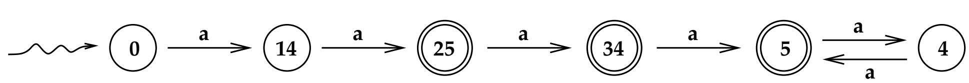
   Figure 2.4 shows a DFA with four states, one of which is a dead state. If we apply the algorithm to find the minimal equivalent DFA, we will still have four states. Thus, the answer is no.

8. Problem 6 on page 63 of the fifth edition, problem 3 on page 57 of the sixth edition.

   This is the NFA shown in Figure 2-8. Let's call it M.

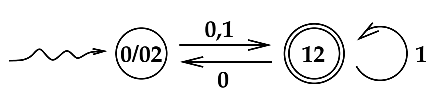


   We now apply the algorithm to convert an NFA into an equivalent DFA. Each state of the DFA is a subset of the set of states of M. Since M has 6 states, there are $2^6 = 64$ such subsets altogether. Fortunately, most of those states are useless, and thus we don't need to include them in our figure. The only ones we show are $\{q_0\}, \{q_1, q_4\}, \{q_2, q_5\}, \{q_3, q_4\}, \{q_5\}, \{q_4\}$. In the figure, we abbreviate these names as 0, 14, 25, 34, 5, and 4.



9. Problem 12 on pages 64 of the fifth edition, problem 13 on page 57 of the sixth edition.

   The safest way to solve this is to compute a DFA equivalent to the NFA. Using the same notation as in the previous problem, we get three states, which we call 0, 02, and 12. Of these, 0 and 02 are equivalent, hence the NFA in the figure is equivalent to a DFA with only two states:



   We can now see that the strings 01001 and 000 are the only ones accepted.