# CS 273: Intro to Theory of Computation, Fall 2007
# Problem Set 2 (due Tuesday, September 11th)

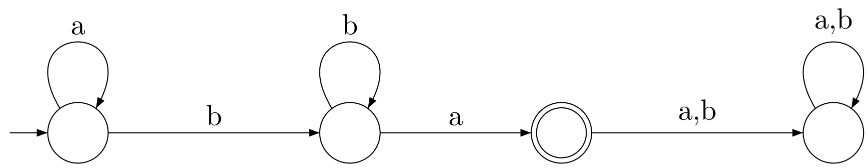
This homework is due 12:30 in class or by noon at Elaine Wilson's office (3229 Siebel).

This homework contains 5 problems, one of which is bonus. Please remember to follow the homework format guidelines, e.g. each problem on a separate page.

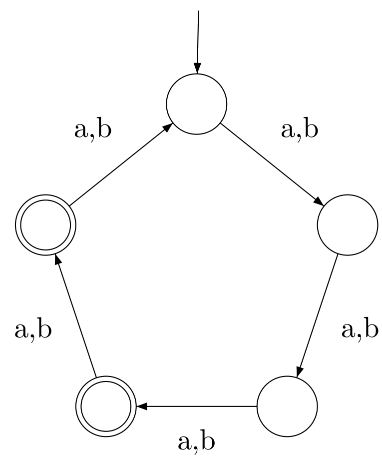
1. Interpreting DFAs (10 points)

   Let $\Sigma = \{a, b\}$. What is the language that each of the following finite state machine accepts.

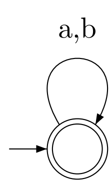
   (a)



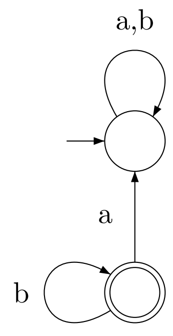

   (b)



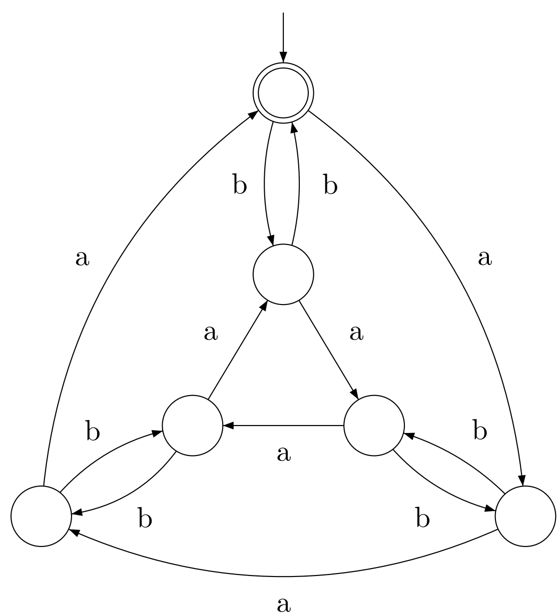

   (c)



(d)




(e)




2. Designing DFAs (16 points)

   Let $\Sigma = \{0, 1\}$. For each of the following languages, give the state diagram for a DFA that recognizes it.

   (a) $L_1 = \{w | w$ is the binary representation of integers divisible by $4\}$. Assume that integers are written with their most significant bit first. Notice that $L_1$ does not contain the empty string.

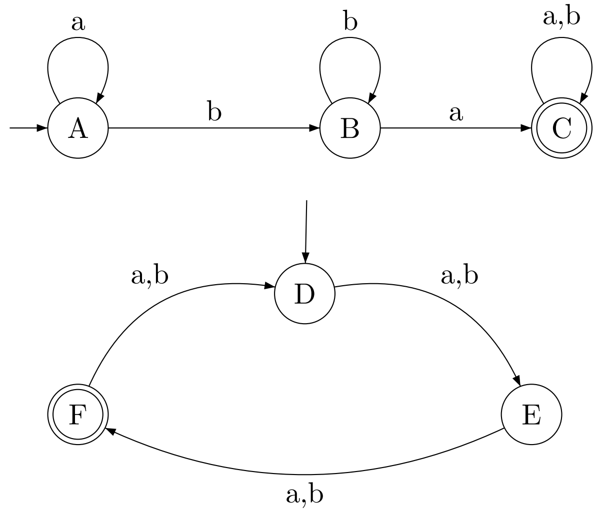
   (b) $L_2$ is the language that consists of all strings such that between every two 0s, there is an even number of 1s.

   (c) $L_3$ is the language that consists of all strings $w$ such that $w$ ends in a 1 and $w$ contains an even number of 0s.

   (d) $L_4 = \{w | w$ does not contain any instance of the substring $001\}$.

3. Product construction (4 points)

   Consider the following two DFAs. Use the product construction (pp. 45–47 in textbook) to construct the state diagram for a DFA recognizing the intersection of the two languages.




4. Algorithms for modifying DFAs (10 points)

   Suppose that $M = (Q, \Sigma, \delta, q_0, F)$ and $N = (R, \Sigma, \gamma, r_0, G)$ are two DFAs sharing a common alphabet.

   (a) Define a new DFA $M' = (Q \cup \{q_S, q_E, q_R\}, \Sigma \cup \{\#\}, \delta', q_S, \{q_E\})$ whose transition function is defined as follows

$$\delta'(q_S, \#) = q_0$$
$$\delta'(q_S, c) = q_R \text{ if } c \neq \#$$
$$\delta'(q, c) = \delta(q, c) \text{ if } q \in Q \text{ and } c \neq \#$$
$$\delta'(q, \#) = q_E \text{ if } q \in F$$
$$\delta'(q, \#) = q_R \text{ if } q \in Q \text{ and } q \notin F$$
$$\delta'(q_E, c) = q_R \text{ for all } c$$
$$\delta'(q_R, c) = q_R \text{ for all } c$$

   Describe the language accepted by $M'$ in terms of the language accepted by $M$.

   (b) Show how to design a DFA $N'$ which accepts the language

$$L' = \{z | z \text{ is a string and there are strings } w \in L(M) \text{ and } x \in L(N) \text{ such that } z = w\#x\}$$

   Define your DFA using notation similar to the definition of $M'$ in part (a).

5. More DFA modification (10 point bonus)

Fix $\Sigma = \{a, b, c\}$. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA. Consider the transformation that takes any such DFA A and constructs the DFA $B = (Q \cup \{q_{new}\}, \Sigma, \delta', q_0, F')$ where $q_{new}$ is a new state (i.e. not in Q), $F' = \{q_{new}\}$, and $\delta'$ is defined as follows:

For every $q \in Q$,

if from q, reading aab takes the automaton A to some state in F,

then $\delta'(q, c) = q_{new}$, $\delta'(q, a) = \delta(q, a)$ and $\delta'(q, b) = \delta(q, b)$

else $\delta'(q, t) = \delta(q, t)$ for each $t \in \Sigma$.

Also, $\delta'(q_{new}, t) = q_{new}$ for each $t \in \Sigma$.

a) Draw a simple DFA with three states, and draw the corresponding transformed DFA.

b) For any DFA A, express the language accepted by its transformed DFA B in terms of the language accepted by A.

c) Prove (b) formally.