CS 311 Homework 1

due 16:00, Thursday, 7^{th} October 2010

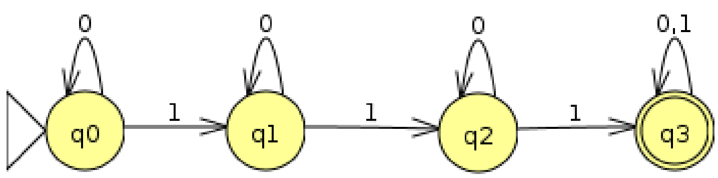
Homework must be submitted on paper, in class.

Question 1. [20 pts.; 5 pts. each]

Draw state diagrams for DFAs recognizing the following languages over the alphabet $\{0, 1\}$. remember to indicate the initial state and the final state(s), and to label all the transitions.

a. $\{ w \mid w$ contains the symbol 1 at least three times.$\}$.

Answer.



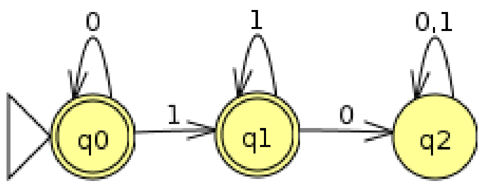
b. $\{ w \mid w$ has all the 0 symbols precede all the 1 symbols.$\}$. *Hint:* What does the question state about how many times each symbol must appear? (*end of hint*).

Answer.

For help with this problem, consider what strings should be rejected. All strings where a 0 is preceded by a 1 should be rejected. Now consider several examples:

- 000111 would not be rejected.
- 111000 would be rejected, because there is at least one 0 that is preceded by a 1.
- 000 would not be rejected. No 0 symbol is preceded by a 1 symbol.
- 111 would not be rejected. No 0 symbol, of which there are none, are preceded by a 1 symbol.
- ε would not be rejected, for the same reason.

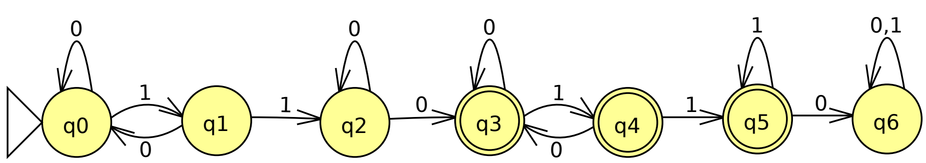
With this in mind, it is easy to see what the DFA should accept.



c. { w | w contains the substring 110 exactly once}.

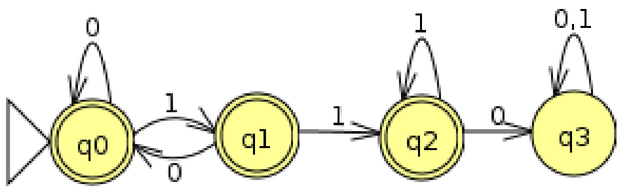
Answer.

In this graphic, pay attention to the transistion from $q2$ to $q3$. Here we see the acceptance of the first occurrence of 110. Now notice the transtion from $q4$ to $q5$. The machine has seen a second occurrence of 11 and is stuck forever accepting 1's or it must fail. I find these graphics helpful in understanding how these machines work. Layout of these graphics becomes important for more sophisticated machines — you can see sections of the machine that perform different tasks and how those tasks interconnect.



d. { w | w does not contain 110}.

Answer.



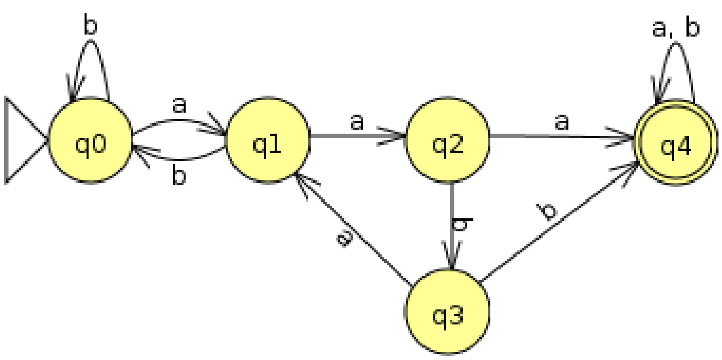
Question 2. [12 pts.; 4 pts. each]

Given $M = \langle \{0, 1, 2, 3, 4\}, \{\mathtt{a}, \mathtt{b}\}, \delta, 0, \{4\} \rangle$ where δ is given by the table below:

	a	b
0	1	0
1	2	0
2	4	3
3	1	4
4	4	4

a. Draw the state diagram for this DFA.

Answer.



b. Informally describe the language that M accepts.

Answer.

Working back from the final state, you can see that there are two ways to arrive. The first way, by way of states q1 or q3 and then through q2 requires the string *aaa*. The second way, through state q2 and then q3 requires the string *aabb*. Once in state q4, the machine remains there on any input. The machine accepts any string that contains either *aaa* or *aabb*.

c. For each of the following three strings, determine whether the string is accepted. List the sequence of states $r_0, r_1, \ldots, r_n$ (from the formal definition) through which the machine moves as it reads the string. (*Hint*, the final state r_n should correspond to your determination of whether the machine accepts or not!).

- *bbabaabb*,

  Answer.

  $r_0 = q_0 = 0$
  $r_1 = \delta(0, b) = 0$
  $r_2 = \delta(0, b) = 0$
  $r_3 = \delta(0, a) = 1$
  $r_4 = \delta(1, b) = 0$
  $r_5 = \delta(0, a) = 1$
  $r_6 = \delta(1, a) = 2$
  $r_7 = \delta(2, b) = 3$
  $r_8 = \delta(3, b) = 4 \in \{4\}$. Accept.

- *abaabaaa*, and

  Answer.

  $r_0 = q_0 = 0$
  $r_1 = \delta(0, a) = 1$
  $r_2 = \delta(1, b) = 0$
  $r_3 = \delta(0, a) = 1$
  $r_4 = \delta(1, a) = 2$
  $r_5 = \delta(2, b) = 3$
  $r_6 = \delta(3, a) = 1$
  $r_7 = \delta(1, a) = 2$
  $r_8 = \delta(2, a) = 4 \in \{4\}$. Accept.

- *aabab*.

  Answer.

  $r_0 = q_0 = 0$
  $r_1 = \delta(0, a) = 1$

$r_2 = \delta(1, a) = 2$
$r_3 = \delta(2, b) = 3$
$r_4 = \delta(3, a) = 1$
$r_5 = \delta(1, b) = 0 \notin \{4\}$. Fail.

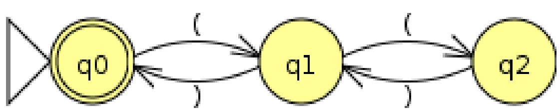
Question 3. [8 pts.]

It is common to use computers to warn us of mistakes we might otherwise miss. (One such mistake to leave unbalanced parentheses (like this!).

It is not possible to determine whether parentheses are balanced using a DFA. Discuss informally why this is so.

Answer.

The difficulty in balancing parentheses lies in the nesting. Accepting one pair of balanced parentheses is easy, as is accepting concatenated pairs of balanced parentheses. But how do you accept nested pairs like (())? Here is the diagram of a DFA that will accept nested pairs to one level.



It will accept (), (()), (()())() and so forth. But what if you need more levels of nesting? Balancing parentheses is a counting task. The machine must count how many open-parentheses have been encountered. For each level of nesting, another state is required. Because you cannot know in advance how many levels of nesting are required, you must provide an infinite number of states. But a DFA by definition has a finite number of states. So we cannot use a DFA to count parentheses. In fact we use another type of machine, a stack based machine called a pushdown automata, to handle these tasks.

Question 4. [10 pts.]

Develop a formal definition for the DFA M, defined as the union of the DFA's described in Question 1a and 1b. Use the 5-tuple notation with a transition table.

Answer.

First we formally define the two machines:

Question 1a

$$
\begin{aligned}
M_a &= \langle Q_a, \Sigma_a, \delta_a, q_{0a}, F_a \rangle \\
\text{where } Q_a &= \{0, 1, 2, 3\} \\
\Sigma_a &= \{0, 1\} \\
q_{0a} &= 0 \\
F_a &= \{3\}
\end{aligned}
$$

and, δ_a is represented by the transition table:

δ_a	0	1
0	0	1
1	1	2
2	2	3
3	3	3

Question 1b

$$
\begin{aligned}
M_b &= \langle Q_b, \Sigma_b, \delta_b, q_{0b}, F_b \rangle \\
\text{where } Q_b &= \{0, 1, 2\} \\
\Sigma_b &= \{0, 1\} \\
q_{0b} &= 0 \\
F_b &= \{0, 1\}
\end{aligned}
$$

and, δ_b is represented by the transition table:

δ_b	0	1
0	0	1
1	2	1
2	2	2

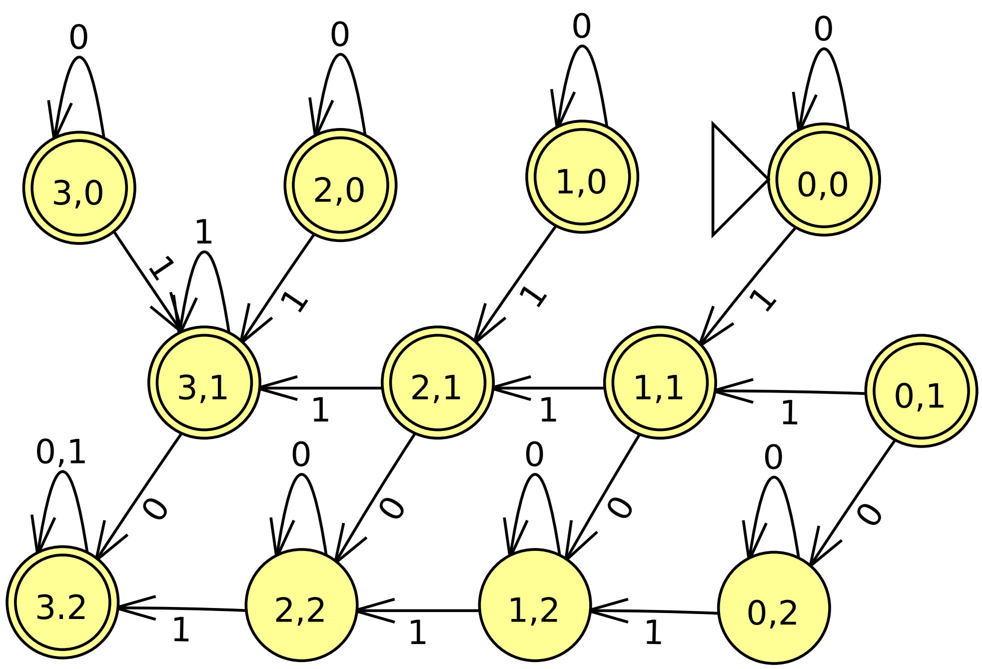
From the formal definition of the union of two languages, we know $L(M_u) = L(M_a) \cup L(M_b)$ if

$$
\begin{aligned}
M_u &= \langle Q, \Sigma, \delta, q_0, F \rangle \\
\text{where} & \\
Q = Q_a \times Q_b &= \{(0,0), (0,1), (0,2), (1,0), (1,1), \\
&\quad (1,2), (2,0), (2,1), (2,2), (3,0), (3,1), (3,2)\} \\
\Sigma &= \{0, 1\} \\
q_0 &= (q_{0a}, q_{0b}) = (0, 0) \\
F &= \{(x, y)) \mid x \in Q_a \vee y \in Q_b\} \\
&= \{(0,0), (1,0), (2,0), (0,1), (1,1), (2,1), (3,0), (3,1), (3,2)\}
\end{aligned}
$$

and, δ is represented by the transition table:

δ	0	1
$(0,0)$	$(0,0)$	$(1,1)$
$(0,1)$	$(0,2)$	$(1,1)$
$(0,2)$	$(0,2)$	$(1,2)$
$(1,0)$	$(1,0)$	$(2,1)$
$(1,1)$	$(1,2)$	$(2,1)$
$(1,2)$	$(1,2)$	$(2,2)$
$(2,0)$	$(2,0)$	$(3,1)$
$(2,1)$	$(2,2)$	$(3,1)$
$(2,2)$	$(2,2)$	$(3,2)$
$(3,0)$	$(3,0)$	$(3,1)$
$(3,1)$	$(3,2)$	$(3,1)$
$(3,2)$	$(3,2)$	$(3,2)$

The table is generated by methodically stepping through the states of each machine using the state pairs. Here is the diagram of this machine:



Note that this diagram is of the complete transition table. No state elimination has been done purely for pedagogical reasons. Examining this diagram, you can see the way this machine works. In the first row are three states that are never reached — they have no transitions leading into them. In the second row, we have one unreachable state, (0,1), and three states that represent acceptance for M_a. These three states also represent counting the number of 1 symbols seen so far. Overall, the second row represents M_a when that machine has seen its first 1 symbol. In the third row we have another unreachable state, (0,2), and three states that again represent the counting of 1 symbols seen. The third row represents the failure state for M_a, a 0 symbol seen after a 1 symbol. This row must count off enough 1 symbols to reach acceptance for M_b.

Question 5. [10 pts.]

In the lecture, Prof. Black gave you the formal definition of DFA, and also defined how to construct a DFA that represents the union of 2 DFAs. However, in the lecture he did not prove that this construction is correct. Prove that $w \in L(M_a) \Rightarrow w \in L(M_a \cup M_b)$. (This is one piece of the proof of the closure of regular languages under union.)

Answer.

To prove that $w \in L(M_a) \Rightarrow w \in L(M_a \cup M_b)$,

Let

$$\begin{aligned} M_a &= \langle Q_a, \Sigma_a, \delta_a, q_{0a}, F_a \rangle \\ M_b &= \langle Q_b, \Sigma_b, \delta_b, q_{0b}, F_b \rangle \end{aligned}$$

By the definition of DFA acceptance

$$\begin{aligned} \text{If } w &= w_1 w_2 w_3 \cdots w_n \\ \text{then } w \in L(M_a) &\Rightarrow \exists R, n. R = \{r_0, r_1, r_n, \ldots, r_n\} \\ \text{where } r_0 &= q_{0a}, \\ \forall i. 0 < i \leq n. & \\ r_i &= \delta_a(r_{i-1}, w_i), \\ r_i &\in Q_a. \\ r_n &= \delta_a(r_{n-1}, w_n), \text{ and} \\ r_n &\in F_a \end{aligned}$$

Now let

$$M_u = M_a \cup M_b.$$

By the definition of the union of DFA's we get

$$
\begin{aligned}
M_u &= \langle Q_u, \Sigma, \delta_u, q_{0u}, F_u \rangle \\
\text{where } Q_u &= \{(r, s) \mid r \in Q_a \wedge s \in Q_b\} \\
\delta_u((r, s), w) &= (\delta_a(r, w), \delta_b(s, w)) \\
q_{0u} &= (q_{0a}, q_{0b}) \\
F_u &= \{(x, y) \mid x \in F_a \wedge y \in F_b\}
\end{aligned}
$$

And with

$$w = w_1 w_2 w_3 \cdots w_n$$

then by the definition of DFA acceptance we get

$$
\begin{aligned}
t_0 &= q_{0u} \\
\forall i. 0 < i \leq n & \\
t_i &= \delta_u((r_{i-1}, s_{i-1}), w_i) \\
&= (\delta_a(r_{i-1}, w_i), \delta_b(s_{i-1}, w_i)). \\
\text{Thus, } t_n &= \delta_u((r_{n-1}, s_{n-1}), w_n) \\
&= (\delta_a(r_{n-1}, w_n), \delta_b(s_{n-1}, w_n)) \\
&= (r_n, \delta_b(s_{n-1}, w_n))
\end{aligned}
$$

Since $r_n \in F_a$, then by the definition of F_u, $t_n \in F_u$.