CPSC 326 Problem Set 1

Due February 19

1. Draw the state diagram of the DFA M whose formal definition is given below:
   $Q = \{q_1, q_2, q_3, q_4, q_5\}$
   $\Sigma = \{u, d\}$
   $q_0 = q_3$
   $F = \{q_3\}$
   δ is given by the table below:

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

2. Give state diagrams of finite automata recognizing the following languages. In all parts the alphabet is $\{0,1\}$.

   a. $\{w \mid w$ begins with a 1 and ends with a 0$\}$
   b. $\{w \mid w$ contains at least three 1s$\}$
   c. $\{w \mid w$ has length at least three and its third symbol is 0$\}$
   d. $\{w \mid w$ doesn't contain the substring 110$\}$
   e. $\{w \mid$ the length of w is at most 3$\}$
   f. $\{w \mid w$ is any string except 11 and 111$\}$
   g. $\{w \mid w$ contains at least two 0s and at most one 1$\}$
   h. $\{w \mid w$ contains exactly two 1s$\}$
   i. $\{w \mid w$ is not the empty string$\}$
   j. $\{\epsilon, 0\}$
   k. $\emptyset$

3. Give regular expressions that produce the same languages as in number 2 above.

4. Prove that any NFA can be converted into an equivalent one that has a single accept state.

5. For each of the following regular expressions, give two strings that are produced and two strings that are *not* produced – a total of four strings for each part. Assume $\Sigma = \{a, b\}$ for each.

a. a^*b^*
b. $a(ba)^*b$
c. $a^* \cup b^*$
d. $(aaa)^*$
e. $(aa \cup bb)^*$
f. $\Sigma^*a\Sigma^*b\Sigma^*a\Sigma^*$
g. $aba \cup bab$
h. $(\epsilon \cup a)b$
i. $(a \cup ba \cup bb)\Sigma^*$

6. In certain programming languages, comments appear between delimiters such as /# and #/. Let C be the language of all valid delimited comment strings. A member of C must begin with a /# and end with a #/ but have no intervening #/. For simplicity, assume that the alphabet for C is $\{a, b, c, /, \#\}$.

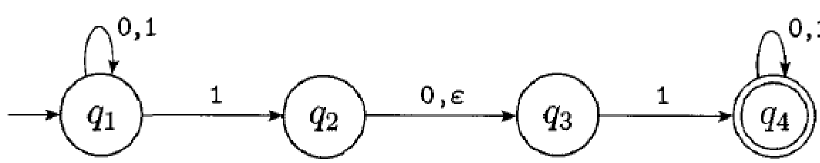
a. Give a finite automata that recognizes C.
b. Give a regular expression that generates C.

7. Convert the following regular expressions to NFAs. You do not need to use the algorithm discussed in class, as long as the NFA recognizes the same language the regular expressions generates.

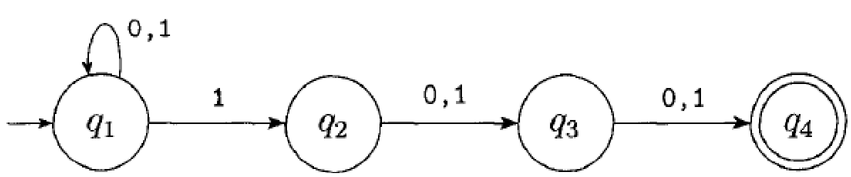
a. $a(abb)^* \cup b$
b. $a^+ \cup (ab)^+$
c. $(a \cup b^+)a^+b^+$

8. Convert the following NFAs to regular expressions. You do not need to use the algorithm discussed in class, as long as the regular expression generates the same language that the NFA recognizes.

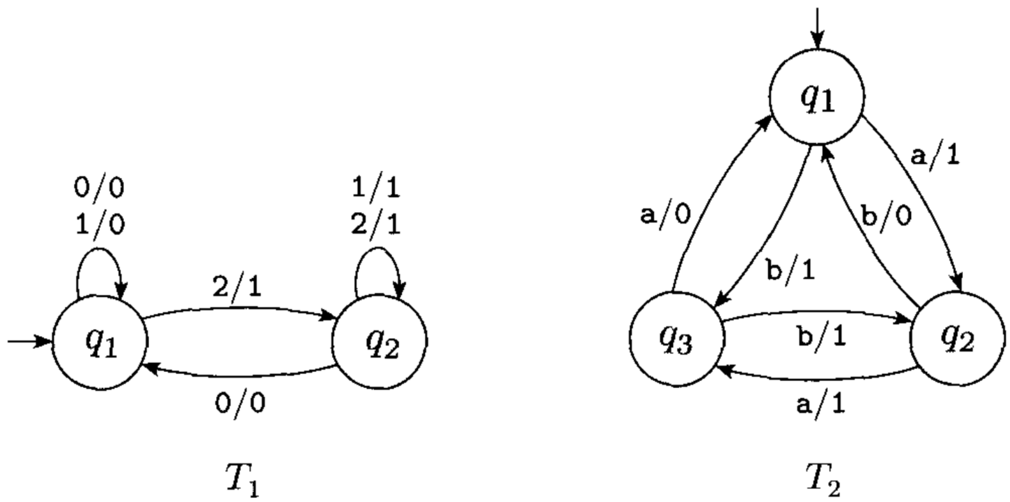
a.



b.



9. A *finite state transducer* (FST) is a type of deterministic finite automaton whose output is a string and not just *accept* or *reject*. The following are state diagrams of finite state



Each transition of an FST is labeled with two symbols, one designating the input symbol for that transition and the other designating the output symbol. The two symbols are written with a slash, /, separating them. In T_1, the transition from q_1 to q_2 has input symbol 2 and output symbol 1. Some transitions may have multiple input-output pairs, such as the transition in T_1 from q_1 to itself. When an FST computes on an input string w, it takes the input symbols $w_1, w_2 ... w_n$ on by one and, starting in the start state, follows the transitions by matching the input labels with the sequence of symbols in w. Every time it goes along a transition, it outputs the corresponding output symbol For example, on input 2212011, machine T_1 outputs 1111000. On input $abbb$, T_2 outputs 1011. Give the output produced by each of the following:

a. T_1 on 011
b. T_1 on 211
c. T_1 on 121
d. T_1 on 0202
e. T_2 on b
f. T_2 on bbab
g. T_2 on bbbbbb
h. T_2 on aaaa

10. Create a FST with the following behavior. Its input and output alphabets are $\{0, 1\}$. Its output string is identical to the input string on the even positions but inverted on the odd positions. For example, on the input 0000111 it should output 1010010.