Homework Assignment #3

(revised on March 18, 2013)

Note

This assignment is due 2:10PM Wednesday, March 20, 2013. Please write or type your answers on A4 (or similar size) paper. Drop your homework by the due time in Yih-Kuen Tsay's mail box on the first floor of Management College Building II, or put it on the instructor's desk before the class on the due date starts. Late submission will be penalized by 20% for each working day overdue. You may discuss the problems with others, but copying answers is strictly forbidden.

Note for this revision

Problem 8 (Exercise 1.18) is now made complete. You may skip Problem 10 (Exercise 1.21, where the needed lemma is yet to be covered in class) and do it later in the next homework assignment.

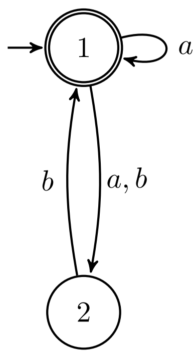
Problems

(Note: problems marked with "Exercise X.XX" or "Problem X.XX" are taken from [Sipser 2006] with probable adaptation.)

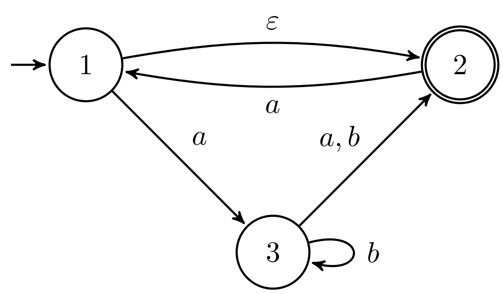
1. (Exercise 1.7; 10 points) Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. In all parts, the alphabet is $\{\mathtt{0}, \mathtt{1}\}$.
   (a) The language $\{w \mid w$ contains the substring $\mathtt{0101}$, i.e., $w = x\mathtt{0101}y$ for some x and $y\}$ with five states
   (b) The language $0^*1^*0^+$ with three states
2. (Exercise 1.8; 5 points) Use the construction given in the proof (using NFAs) of Theorem 1.45 (closedness of regular languages under union) to give the state diagram of an NFA recognizing the union of $\{w \mid w$ begins with a $\mathtt{1}$ and ends with a $\mathtt{0}\}$ and $\{w \mid w$ contains at least three $\mathtt{1}$s$\}$. The alphabet is $\{\mathtt{0}, \mathtt{1}\}$.
3. (Exercise 1.9; 5 points) Use the construction given in the proof (using NFAs) of Theorem 1.47 (closedness of regular languages under concatenation) to give the state diagram of an NFA recognizing the concatenation of $\{w \mid$ the length of w is at most five$\}$ and $\{w \mid$ every odd position of w is a $\mathtt{1}\}$. The alphabet is $\{\mathtt{0}, \mathtt{1}\}$.
4. (Exercise 1.10; 5 points) Use the construction given in the proof (using NFAs) of Theorem 1.49 (closedness of regular languages under the Kleene star) to give the state diagram of an NFA recognizing the star of $\{w \mid w$ contains at least three $\mathtt{1}$s$\}$. The alphabet is $\{\mathtt{0}, \mathtt{1}\}$.
5. (Exercise 1.12; 10 points) Let $D = \{w \mid w$ contains an even number of $\mathtt{a}$'s and an odd number of $\mathtt{b}$'s and does not contain the substring $\mathtt{ab}\}$. Give a DFA with five states that recognizes D and a regular expression that generates D. (Suggestion: describe D in a simpler way.)

6. (Exercise 1.14; 10 points) Show by giving an example that, if M is an NFA that recognizes language C, swapping the accept and nonaccept states in M doesn't necessarily yield a new NFA that recognizes the complement of C. Is the class of languages recognized by NFAs closed under complement? Explain you answer.

7. (Exercise 1.16; 10 points) Use the construction given in Theorem 1.39 (every NFA has an equivalent DFA) to convert the following two nondeterministic finite automata to equivalent deterministic finite automata.



(a)



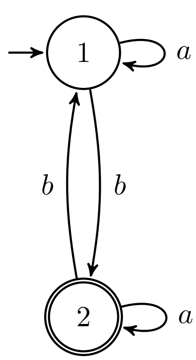
(b)

8. (Exercise 1.18; 10 points) Use the procedure described in Lemma 1.55 to convert the regular expression $(0 \cup 1)^*000(0 \cup 1)^*$ to a nondeterministic finite automaton.

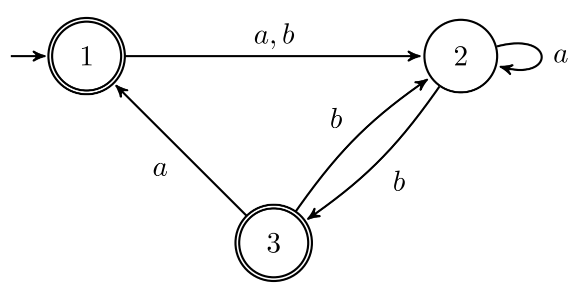
9. (Exercise 1.20; 10 points) Give regular expressions generating the following languages:

   (a) $\{w \mid w \text{ begins with a } \texttt{1} \text{ and ends with a } \texttt{0}\}$
   (b) $\{w \mid w \text{ contains the substring } \texttt{0101}, \text{ i.e., } w = x\texttt{0101}y \text{ for some } x \text{ and } y\}$
   (c) $\{w \mid w \text{ doesn't contain the substring } \texttt{110}\}$
   (d) $\{w \mid \text{every odd position of } w \text{ is a } \texttt{1}\}$ (Note: see w as $w_1 w_2 \cdots w_n$, where $w_i \in \{\texttt{0}, \texttt{1}\}$)

10. (Exercise 1.21; 10 points) Use the procedure described in Lemma 1.60 to convert the following finite automata to regular expressions.

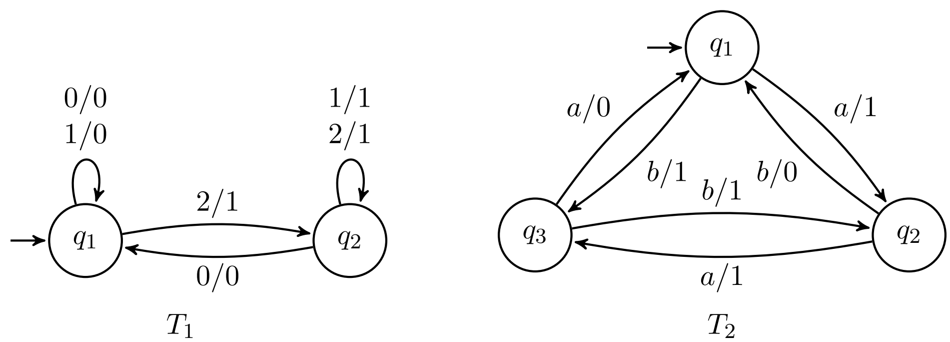


(a)



(b)

11. (Exercise 1.24; 5 points) A *finite-state transducer* (FST) is a type of deterministic finite automaton whose output is a string rather than *accept* or *reject*. The following are state diagrams of finite state transducers T_1 and T_2.



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 conditions 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 $w_1 \cdots w_n$ one by one and, starting from the start state, follows the transitions by matching the input labels with the sequence of symbols $w_1 \cdots w_n = w$. Every time it goes along a transition, it outputs the corresponding output symbol. For example, on input 2212011, machine T_1 enters the sequence of states $q_1, q_2, q_2, q_2, q_2, q_1, q_1, q_1$ and produces output 1111000. On input `abbb`, T_2 outputs 1011. Give the sequence of states entered and the output produced in each of the following parts.

(a) T_1 on input 211

(b) T_2 on input `bbab`

12. (Exercise 1.25; 10 points) Read the informal definition of the finite state transducer given in Exercise 1.24. Give a formal definition of this model, following the patterns in Definition 1.5 (Page 35). Assume that an FST has an input alphabet Σ and an output alphabet Γ but not a set of accept states. Include a formal definition of the computation of an FST. (Hint: an FST is a 5-tuple. Its transition function is of the form $\delta : Q \times \Sigma \longrightarrow Q \times \Gamma$.)