University of Illinois at Urbana-Champaign
Department of Computer Science

Homework Assignment 5

CS 273 Introduction to Theoretical Computer Science
Fall Semester, 2002

Due: Tuesday, Dec. 3, at the beginnning of class

This is a group homework. Please staple your homework in four pairs. 1 and 2, 3 and 4, 5 and 6, 7 and 8. Hand in each of them to the corresponding stack in class.

1. **(10 pts) For the languages below, either prove that it is a regular language by giving an NFA or DFA that recognizes it; otherwise, argue that it is not a regular language. The alphabet is $\Sigma = \{0, 1\}$.**

   (a) **(10 pts)** $L = \{w \mid w$ **contains at least two 0's and at most one 1's**$\}$

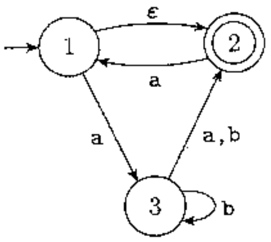
   (b) **(10 pts)** $L = \{w \mid$ **every block of 5 symbols of** w **contains at least two 1's**$\}$

2. **(10 pts) Use the same instructions and assumptions as for Problem 1, and solve the following problems.**

   (a) **(10 pts)** $L = \{w \mid w$ **contains an equal number of occurences of 0's and 1's**$\}$

   (b) **(10 pts)** $L = \{w \mid w$ **contains an equal number of occurences of the substring 01 and 10**$\}$

3. **(10 pts) Convert the following NFA into a DFA that recognizes the same language.**



4. **(10 pts) Determine whether the following statements are true in general. They might be for specific cases, but are they always true? Justify your answer.**

   (a) **If** $L_1 L_2$ **is regular and** $|L_1|$ **is finite, then** L_2 **is regular**

   (b) **If** $L_1 \cup L_2$ **is regular and** L_1 **is regular, then** L_2 **is regular**

5. **(10 pts) Prove that by allowing multiple start states, any NFA with** ϵ **moves can be converted into an NFA that has no** ϵ **moves. The new NFA must use the same states as the original NFA.**

6. (10 pts) Give a brief description (mostly or completely in English) of the language described by each of the following regular experessions. The alphabet is given by $\Sigma = \{0, 1\}$.

(a) $(1 + \epsilon)(00^*1)^*0^*$

(b) $(\Sigma 01 \Sigma)^* \cup \emptyset$

(c) $((01) \cup (10))^*(111 \cup 1^*) \cup \epsilon$

7. (10 pts) Design a Turing machine for the language $L = \{ww^R \mid w$ is any string of 0's and 1's$\}$, in which w^R denotes the reverse of w. Thus, the machine should decide whether the input is a palindrome. Describe the finite control of your machine in detail (possibly as a kind of pseudocode), but not necessarily with the specification of all the elements in the formal notation of a Turing machine.

8. (10 pts) Design a Turing machine (specified in the same way as in the previous problem) that decides in polynomial time membership in the language POW-PERM. The language is defined as follows. For the set $\{1, 2, 3, \ldots, k\}$, let p denote a permutation function. Furthermore, let p^t denote the composition of p with itself t times (for example, if p represents a circular left shift, then p^2 represents two circular left shifts). The language is defined as POW-PERM =

$$\{\langle p, q, t\rangle \mid p = q^t \text{ in which p and q are permutations on } \{1, 2, \ldots, k\} \text{ and } t \text{ is a binary integer}\}$$

Note that the most obvious algorithm does not run in polynomial time. Hint: First try it for the case in which t is a power of 2.