Introduction to Theory of Computing	**Problem Set 5**
CS 481 Fall 2004	**Due Friday, October 8, 2004**

Reading. To review the material for this past week read Sections 10, and 13-14 of Kozen.

Please turn in Problems 1-2 and Problem 3 separately with your name and cornell.edu email written on both.

(1) Consider the DFA on the Figure 1 below, where s is the start state, and the two circled states (3 and 6) are the accepting states. Show which pairs of states are equivalent, and which are not.

(2) Let L be a regular language over an alphabet Σ, and let $a, b \in \Sigma$ be two letters.

(a) Let $c \notin \Sigma$ be a new letter. For a word $w \in \Sigma^*$ let $f(w)$ be a word over $\Sigma \cup \{c\}$ where each occurrence of ab in the word is replaced by a c. For example, $f(aabbad) = acbad$, and $f(abbabeba) = cbceba$. Show that $f(L) = \{f(w) | w \in L\}$ is regular.

(b) Consider the previous definition for a letter $c \in \Sigma$. For example, we now would get $f(abcabc) = cccc$. Does $f(L)$ have to be regular whenever L is regular? Prove your answer.

(3) A *Finite State Transducer* is a type of deterministic finite automata that generates output not only an accept/reject decision. It has a start state s, and transitions, just like a DFA. See Figure 2 on the next page for an example. Each transition is labeled with two labels (separated by a "/" on the figure). The first label is the input symbol for that transition, and the second label is the output symbol. For example, the transition from state p to state q in the example, is followed on input symbol b, and it generates an output symbol 1. On reading a word $w = \sigma_1 \dots \sigma_k$, it starts in the start state s, and follows the appropriate transitions, just like a DFA, but at each transition it outputs the corresponding output. For example, the transducer on the figure, when reading input $aabbab$ goes through states s, q, s, p, q, a, p and outputs the word 100100.

(a) Give a formal definition of a Finite State Transducer similar to our formal definitions of DFA as a 5 tuple $(M, \Sigma, \delta, s, F)$. Assume that the input alphabet is Σ, and the output alphabet is

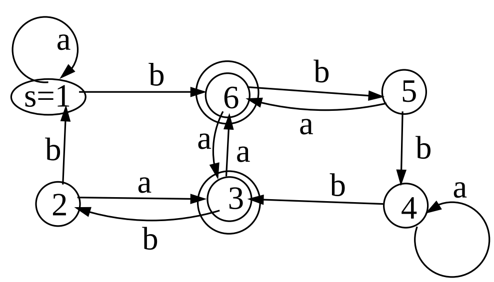

Figure 1: A DFA for Problem 1.

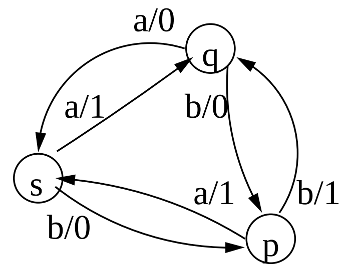

Figure 2: A Finite State Transducer.

Γ. Define also the output generated by the transducer, similar to our definition of $\hat{\delta}(s, w)$ for a DFA.

(b) Give a Finite State Transducer with input and output alphabet both $\{0, 1\}$, so that the machine swaps each 1 in an even position to a 0, but otherwise returns the same words. For example, on 001101 it would output 001000. Drawing the picture is enough (no proof or formal definition is required for this part.

(c) Consider a Finite State Transducer M which also has a set of accepting states F. Ignoring the output, we can view it as a regular DFA, and it defines a language $L(M)$ (those where following the transitions, the DFA ends in an accepting state). Let $O(M)$ be the set of words output by the transducer on words in L, that is the set of words generated by transition sequences ending in F. Prove that for all state transducers $O(M)$ is regular.