Assignment I

Note: Solve as many problems from Michel Sipser's book *Theory of Computation*. You will find similar problems in your examination.

Q1. Construct minimum DFA and write down regular expressions for the following languages over $\{0, 1\}$: a) All strings which contain either the pattern "$11''$ or "$010''$. b) All strings which contain exactly one of the patterns above. c) Set of all strings ending with "$011''$. d) Set of all strings not ending with "$011''$.

Q2. (This question is to be attempted after context free grammars are discussed in class)
A context free grammar $G = (V, \Sigma, P, S)$ is a **right linear grammar** if the right hand side of every production in P has at most one non-terminal and the non-terminal (if any) shall occur only as the *last* symbol in the production. (For example ($S \longrightarrow 0A|1S$, $A \longrightarrow 00A|11$) is right linear, but ($S \longrightarrow 0S0|1S1|\epsilon$) is not right linear). Show that a language L is regular if and only if L has a right linear grammar. (Prove that given any DFA M, you can construct a right linear grammar G such that $L(G) = L(M)$ and given any right linear grammar G', you can construct a DFA M' such that $L(M') = L(G')$). Construct right linear grammar for each of the languages specified in the previous question.

Q3. Given two finite state machines M_1 and M_2, how will you solve the following decision questions? a) $L(M_1) \subseteq L(M_2)$? b) $L(M_1) = L(M_2)$? c) $L(M_1) \neq L(M_2)$.

Q4. Let L be a regular language. Show that the following languages are regular: a) Let L' be any language (not necessary regular), $L \setminus L' = \{x | \exists y \in L', xy \in L\}$. (Hint: Start with a DFA for L and change the final states to meet the specification.) b) $\frac{1}{2}L = \{x | \exists y, |x| = |y|, xy \in L\}$.

Q5. Given any map $h : \Sigma \longrightarrow \Sigma^*$, we can extend the map to act from Σ^* to Σ^* as follows: $h(a_1a_2..a_n) = h(a_1)h(a_2)...h(a_n)$. (For example consider $h : \{0, 1\} \longrightarrow \{0, 1\}^*$ defined by $h(0) = 11$ and $h(1) = 00$. Then $h(010) = 110011$). The extended map is called a *homomorphism* from Σ^* to Σ^*. Suppose L is any regular language and h any homomorphism, define $h(L) = \{h(x) | x \in L\}$. Show that $h(L)$ is regular whenever L is regular. (Hint: Think in terms of regular expression representation for regular languages).

Q6. A *substitution* f is a map from Σ to 2^{Σ^*}. That is, a substitution maps each symbol in Σ to a language (i.e., set of strings) in Σ^*. We can extended f into a map from Σ^* to Σ^* in a way similar to what was done with homomorphisms – $f(a_1a_2..a_n) = f(a_1)f(a_2)...f(a_n)$, where the concatenation operation on the right side is the concatenation of languages. (Recall that if L and L' are languages, $LL' = \{xy|x \in L, y \in L'\}$.) Show that if $f(a)$ is regular for each $a \in \Sigma$, then $f(L)$ is regular whenever L is regular.

Q7. Design a finite state machine that finds the least significant bit in the sum of two binary numbers. (Consider $\Sigma = \{00, 01, 10, 11\}$ to capture all combinations of two input bits. Assume that the bits of the input numbers are fed serially into your FSM. When LSB of the present sum is 1, your machine must be in a final state). Note that this technique can be extended to find any bit of the sum, essentially showing that finite automata can perform addition.

Q8. Find a language L such that L^* is not regular.

Q9. Show that the language L_k consisting of all binary strings with the k^{th} last position having a 1 can be accepted by an NFA with k states, but requires a DFA with at least 2^k states. This shows that the size of the optimal DFA for a language can be exponential in the size of the optimal NFA for the language.

Q10. Find all the Myhill Nerode equivalence classes for the language $\{xx : x \in \{0,1\}^*\}$. (Hence conclude that the language is not regular).

Q11. Show that the language $L = \{a^ib^jc^k|$ if $i = j$ then $j = k\}$ is non regular. Show that there exists a $k > 0$ such that for all strings $x \in L$ with $|x| \geq k$, it is possible to split x into parts uvw such that $|uv| \leq k$, $|v| \geq 1$ and for all integers $t \geq 0$, $uv^tw \in L$. (That is, conditions in the pumping lemma are satisfied, but the language is not regular). Why is this not a contradiction to the pumping lemma?

Q12. Establish the following stronger for of pumping lemma: If L is regular, then there exists a positive integer k such that for all strings $x \in L$ which can be written in the form $x = z_1z_2z_3$ for any strings z_1, z_2 and z_3 with $|z_2| \geq k$, it is possible to further split z_2 into parts uvw such that $|uv| \leq k$, $|v| \geq 1$ and for all integers $t \geq 0$, $z_1uv^twz_3 \in L$. Use the above result to prove that $L = \{a^ib^mc^m|i \geq 1, m \geq 1\}$ is not regular. Do you see the difficulty in using the standard pumping lemma in proving this result?

Q13. Consider the CFG ($S \longrightarrow aSb|SS|\epsilon$). Find all the Myhill Nerode equivalence classes of the language specified by the grammar.