Verification of Reactive Systems (Summer 2014) **Problem Set #1 (Due)**

Problem Set 1 is to test your knowledge of background material, and you need not turn it in. For material on graph algorithms, you can look at the Wikipedia book `http://en.wikipedia.org/wiki/Book:Graph_Algorithms` or in a standard textbook on algorithms, such as the one by Cormen and others. For material on formal languages and automata, you can look at the book by Sipser.

Problem 1.1. A graph is *semi-accessible* if for every pair of vertices $u, v \in V$, either $u \to^* v$ or $v \to^* u$. Give an algorithm to check if a graph is semi-accessible.

Problem 1.2. Show that every regular language can be recognized by an NFA (possibly with ϵ transitions) with exactly one initial state and exactly one final state. Does this property hold if you do not allow ϵ transitions?

Problem 1.3. Consider the language

$$L_n = \{w \in \{0,1\}^* \mid \text{the } n\text{th symbol from the right is a } 1\}$$

(a) describe an NFA with $O(n)$ states that recognizes this language. (b) Show that any DFA for this language requires at least 2^{n-1} states.

Problem 1.4. Give an algorithm to check if a regular language described as an NFA is non-empty. What is the time complexity of your algorithm? Give an algorithm to check if an NFA accepts all possible strings. What is the time complexity of your algorithm?

Problem 1.5. Give an algorithm that checks if a given numbering of nodes of a graph defines a topological order.

Problem 1.6. Show the following valid propositions in propositional logic.

1. $(\neg P \to P) \to P$
2. $\neg(P \to Q) \to \neg Q$

Problem 1.7. Using a truth table, determine which of the following are equivalent to $(p \wedge q) \to r$ and which are equivalent to $(p \vee q) \to r$:

1. $p \to (q \to r)$
2. $q \to (p \to r)$
3. $(p \to r)^(q \to r)$
4. $(p \to r) \vee (q \to r)$

Problem 1.8. The *diameter* of a tree $T = (V, E)$ is defiend as

$$\max_{u,v \in V} d(u, v)$$

that is, it is the longest of all pairs of shortest paths in the tree. Give an algorithm to compute the diameter of a tree and analyze the running time of your algorithm.

Problem 1.9. (a) Given two regular languages L_1 and L_2, how will you check if they have at least one string in common? (b) Can you use the algorithm in part (a) to check if all strings of L_1 also belong to L_2?

Problem 1.10. Given a DFA M and a number n, give an algorithm to count the number of strings of length n in $L(M)$.