Homework 09

Due: Tuesday, April 18, 2006

Note: This assignment cannot be accepted late because solutions will be distributed at the April 18 class meeting when we review for Exam 3.

Instructions

1. Please review the homework grading policy outlined in the course information page.
2. On the *first page* of your solution write-up, you *must* make explicit which problems are to be graded for regular credit, which problems are to be graded for extra credit, and which problems you did not attempt. Use a table that looks like this:

Problem	1	2	3	4	5	6	7	8	9	...
Credit	RC	RC	RC	EC	RC	EC	NA	NA	EC	...

   where "RC" denotes "regular credit", "EC" denotes "extra credit", and "NA" denotes "not attempted". Failure to include such a table will result in an arbitrary set of problems being graded for regular credit, no problems being graded for extra credit, and a 5% penalty assessment.

3. You must also write down with whom you worked on the assignment. If this varies from problem to problem, write down this information separately with each problem.

Problems

Required: 5 of the following 7 problems
Points: 20 points per problem

1. (a) Show that P is closed under complement and concatenation.
   (b) Let A be a decidable language and let D be a polytime decider for it. Consider the following algorithm for deciding whether a given non-empty string s of length n belongs to A^*: For every possible way of splitting s into non-empty substrings $s = s_1 s_2 \ldots s_k$, run D on each substring s_i in that split and *accept* iff all substrings are accepted by D for some split. Derive an exact expression for how many possible such splits there are as a function of $n = |s|$. Use this to conclude that this algorithm does not run in polynomial time even though D does.
   (c) What does the result of part b imply about the closure of P under the star operation? Explain.
2. Do the following:
   (a) Example 3 of the TM-Examples.pdf handout gives a detailed description of a TM that decides the language $\{\mathtt{a}^k\mathtt{b}^k\mathtt{c}^k \mid k \geq 1\}$. Perform an asymptotic (big-O) analysis of this algorithm as a function of the length n of the input string and, in particular, determine the exponent of its highest-order term. Use this to conclude that this language is in P.
   (b) Do Exercise 7.11.
3. Do Problem 7.20(b).

4. In an undirected graph $G = (V, E)$, an *independent set* is a set of nodes $S \subseteq V$ such that for any pair of nodes $u, v \in S$ there does not exist an edge $(u, v) \in E$. In other words, S is an independent set in G if every node in S has no edge in G connecting it to any other node in S. Define the language

   $$\textit{INDEPENDENT-SET} = \{\langle G, k\rangle \mid G \text{ is a graph having an independent set of size } k\}\,.$$

   Prove that *INDEPENDENT-SET* is NP-complete.

5. A *Hamiltonian cycle* in a directed graph is a Hamiltonian path that forms a cycle in the graph. Define

   $$\textit{HAMCYCLE} = \{\langle G\rangle \mid G \text{ is a directed graph that has a Hamiltonian cycle}\}$$

   Prove that *HAMCYCLE* is NP-complete.

6. Do Problem 7.29. You may take for granted (without proving it) that *3COLOR* (defined in Problem 7.27) is NP-complete.

7. Suppose there is a (not-yet-discovered) polytime decider D for *HAMPATH*. Note that D itself can only give *yes/no* answers; it does not actually return such a path even if the answer is *yes*. Design an algorithm that actually generates a Hamiltonian path, if one exists, by using such a decider D as a subprocedure. Its input should be a directed graph G and a given start node s and end node t. Your algorithm should run in polynomial time (assuming, as we are, that D does).

For any of these problems where NP-completeness is to be proved, use an appropriate polytime reduction involving one of the NP-complete decision problems described in the book or in lectures.