## Homework 09

**Due:** Tuesday, December 5, 2006  
**Note:** This assignment cannot be accepted late because solutions will be distributed at the December 5 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:

<table>
<thead>
<tr>
<th>Problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>RC</td>
<td>RC</td>
<td>RC</td>
<td>EC</td>
<td>RC</td>
<td>EC</td>
<td>NA</td>
<td>NA</td>
<td>EC</td>
<td></td>
</tr>
</tbody>
</table>

   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 $s = s_1s_2 \ldots s_k$, 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:
   
   - Exercise 7.10. ($\text{ALL}_{\text{DFA}}$ is defined in Problem 4.3.)
   - Exercise 7.11.

3. Given any language $L$, define $\text{FirstHalf}(L)$ as follows:

   \[
   \text{FirstHalf}(L) = \{ w \mid \text{there exists } x \text{ such that } |x| = |w| \text{ and } wx \in L \}.
   \]

   In other words, a string $w$ is in $\text{FirstHalf}(L)$ if $w$ is the first half of some string in $L$. For example, if

   \[
   L = \{1, 00, 101, 1100, 101001\}
   \]

   then

   \[
   \text{FirstHalf}(L) = \{0, 11, 101\}.
   \]
a. Prove that the class of decidable languages is closed under \textsc{FirstHalf}.

b. Prove that NP is closed under \textsc{FirstHalf}.

c. Your proof for part (a) should involve constructing a TM (or TM variant) for \textsc{FirstHalf}(L) given a TM (or TM variant) for L. Does this construction also show that P is closed under \textsc{FirstHalf}? If so, explain how; if not, explain why not.

4. In an undirected graph \( G = (V, E) \), an \textit{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 \[ \textsc{Independent-Set} = \{ \langle G, k \rangle \mid G \text{ is a graph having an independent set of size } k \} . \]

Prove that \( \textsc{Independent-Set} \) is NP-complete.

5. Two Boolean formulas \( \phi_1 \) and \( \phi_2 \) with the same set of variables are defined to be (logically) equivalent if they evaluate to the same value for all possible 0/1 (i.e., \text{FALSE}/\text{TRUE}) assignments to these variables. Define the language \[ \textsc{Eq-Boolean} = \{ \langle \phi_1, \phi_2 \rangle \mid \phi_1 \text{ and } \phi_2 \text{ are equivalent Boolean formulas} \} . \]

Prove that \( \textsc{Eq-Boolean} \) is NP-complete.

6. Do Problem 7.29. You may take for granted (without proving it) that \textsc{3Color} (defined in Problem 7.27) is NP-complete.

7. Do Problem 7.36.

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 or online course handouts.