Quiz 3 — CSE 105, Winter 2007

Name (print): SOLUTIONS
Student I.D.: ________________

- This quiz is closed book. You are only allowed to use one page of notes (double sided is fine)

- No form of collaboration is allowed during the quiz, including sharing notes, borrowing pencils, etc.

- Your solution will be evaluated both for correctness and clarity. A poorly written solution won’t get full credit even if correct.

- Read all the problems first before you start working on any of them, so you can manage your time wisely.

- Good luck!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1 [10 points]

Give the entire sequence of configurations (one per line) the above Turing machine goes through on input 011, starting from the initial configuration, until a final configuration is reached. (As a help, some of the configurations are already shown.)

```
<table>
<thead>
<tr>
<th>q</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>q1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>□</td>
<td>1</td>
<td>q1</td>
<td>1</td>
</tr>
<tr>
<td>□</td>
<td>1</td>
<td>1</td>
<td>q1</td>
</tr>
<tr>
<td>□</td>
<td>1</td>
<td>q2</td>
<td>1</td>
</tr>
<tr>
<td>□</td>
<td>q3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>q3</td>
<td>□</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>□</td>
<td>q4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>□</td>
<td>1</td>
<td>q4</td>
<td>0</td>
</tr>
<tr>
<td>□</td>
<td>1</td>
<td>q4</td>
<td>0</td>
</tr>
<tr>
<td>□</td>
<td>1</td>
<td>q4</td>
<td>0</td>
</tr>
</tbody>
</table>
```

Problem 2

Consider the language:

$$\text{DISJOINT}_{\text{CFG, DFA}} = \{ \langle G, M \rangle : G \text{ is a CFG, } M \text{ is DFA, and } \mathcal{L}(G) \cap \mathcal{L}(M) = \emptyset \}.$$
Prove that DISJOINT$_{CFG, DFA}$ is decidable giving an informal description of a TM that decides DISJOINT$_{CFG, DFA}$. You may use, as a subroutine, any of the algorithms studied in class (e.g., algorithms used to prove the decidability and closure properties of languages.)

Proof. Let $M'$ = "On input $⟨G, M⟩$,

1. Convert $G$ to a PDA $P$.
2. Construct a PDA $P'$ such that $L(P') = L(P) \cap L(M)$. This can be done by the construction given in class and section.
3. Convert $P'$ to a CFG $G'$.
4. Using the decider for $E_{CFG}$, accept if $L(G') = \emptyset$, reject otherwise."

By construction, $M'$ decides DISJOINT$_{CFG, DFA}$.

Problem 3

For any context free grammar $G$ and rule $R$ in it, let $G - R$ be the context free grammar obtained removing rule $R$ from $G$. For example, if $G$ is the grammar $S \rightarrow abS | T; T \rightarrow aT | bT | \epsilon$ and $R$ is the rule $S \rightarrow T$, then $G - R$ is the grammar $S \rightarrow abS; T \rightarrow aT | bT | \epsilon$. We say that rule $R$ is redundant in $G$ if it can be removed from grammar $G$ without affecting its language. E.g., in the example above $R$ is not redundant in $G$, while rule $S \rightarrow abS$ is redundant. Consider the computational problem of determining, given a context free grammar $G$ and rule $R$ in it, if $R$ is redundant in $G$. Formulate the problem as a language, and prove that it is undecidable.

We have several choices of undecidable languages from which we can reduce. Since we are dealing with CFGs, the two obvious choices are ALL$_{CFG}$ and EQ$_{CFG}$.

First, we need to formulate the problem as a language. Let

$$L = \{⟨G, R⟩ \mid G \text{ is a CFG, } R \text{ is a rule in } G, L(G) = L(G - R)\}.$$ 

First, let us reduce from ALL$_{CFG}$.

Proof. Assume that $M$ decides $L$. Let $M_1$ = "On input $⟨G⟩$,

1. Let $S$ be the initial nonterminal in $G$.
2. Let $S'$ and $T$ be new nonterminals—that is, the are not the same as those in $G$.
3. Construct a new grammar $G'$ that contains all of the rules of $G$ but with the additional rules $S' \rightarrow S | T$, for each $\sigma \in \Sigma$, the rule $T \rightarrow \sigma T$, and the rule $T \rightarrow \epsilon$. Let $S'$ be the initial nonterminal of $G'$.
4. Let $R$ be the rule $S \rightarrow T$ and run $M(⟨G', R⟩)$.
5. If $M$ accepts, accept. Otherwise reject.”

To be rigorous, we need to show that $M_1$ decides ALL$_{CFG}$. Note that the language of the grammar $G'$ constructed in step 3 is $L(G') = \Sigma^*$ and that $L(G' - R) = L(G)$. Therefore, $M((G', R))$ accepts if and only if $L(G) = \Sigma^*$, hence $M_1$ decides ALL$_{CFG}$. Since ALL$_{CFG}$ is undecidable, it must be the case that no such $M$ exists. 

Instead of ALL$_{CFG}$, we could have reduced from EQ$_{CFG}$ as follows.

Proof. Assume $M$ decides $L$. Let $M_2 =$ “On input $(G_1, G_2)$,

1. Rename the rules in $G_2$ to be different from the rules in $G_1$—this amounts to renaming the nonterminals.

2. Let $S_1$ and $S_2$ be the initial nonterminals in $G_1$ and (the modified) $G_2$, respectively.

3. Build a new grammar $G'$ with all of the rules from $G_1$ and (the modified) $G_2$. Let $S$ be the initial nonterminal of $G'$ and add the rules $S \rightarrow S_1 | S_2$.

4. Run $M((G', S \rightarrow S_2))$ and reject if $M$ rejects.

5. Run $M((G', S \rightarrow S_1))$ and reject if $M$ rejects.

6. Accept.”

Since $L(G') = L(G_1) \cup L(G_2)$, $L(G' - (S \rightarrow S_2)) = L(G_1)$, and $L(G' - (S \rightarrow S_1)) = L(G_2)$, if $M$ rejects in step 4, then $L(G_2) \not\subseteq L(G_1)$. Likewise, if $M$ rejects in step 5, then $L(G_1) \not\subseteq L(G_2)$. Therefore, if $M_2$ accepts, then $L(G_1) \subseteq L(G_2) \subseteq L(G_1)$ so $L(G_1) = L(G_2)$. Thus, $M_2$ decides EQ$_{CFG}$. Since EQ$_{CFG}$ is undecidable, no such $M$ can exist.

Note that giving either construction without the proof of correctness was acceptable and received full points.