CSE 322: Formal Methods in Computer Science Spring 2003

PROBLEM SET 8

Due Friday, June 6, 2003, in class

---

1. Lewis and Papadimitriou, Problem 3.6.2.

2. Consider Lemma 3.7.1 on page 170 of Lewis and Papadimitriou which gives a bottom-up way of converting grammars into PDAs. Fill in the details of the formal inductive proof of the Claim in that Lemma, namely the statement: "For any $x \in \Sigma^*$ and $\gamma \in \Gamma^*$, $(p, x, \gamma) \vdash_M^* (p, \epsilon, S)$ if and only if $S \stackrel{R}{\Rightarrow}_G \gamma^R x$."

3. Show that recursively enumerable (i.e. Turing-recognizable) languages are closed under union, intersection, and concatenation.

4. Give an informal English description of a Turing machine that can *decide* membership in the language MUL $= \{a^m b^n c^{mn} \mid m, n \geq 1\}$. (Note that your Turing machine should be a decider, i.e., halt on all inputs.)

5. Let a k-PDA be a pushdown automaton that has k stacks. Thus a 0-PDA is an NFA and a 1-PDA is a conventional PDA. Since there are CFLs that are not regular, you already know that 1-PDAs are more powerful than 0-PDAs. Show that 2-PDAs are more powerful than 1-PDAs. (Hint: Argue how a 2-PDA can simulate a Turing machine.)

6. * (Extra credit) Assume that the language $A_{\text{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM that accepts } w\}$ is undecidable. Then, show that $ALL_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \Sigma^*\}$ is undecidable.