ECS 120: Theory of Computation — Handout 8
UC Davis — Phillip Rogaway — May 18, 1999

Problem Set 8 — Due Wednesday, May 26, 1999

Problem 1. [**10 points**] Recall that an *unrestricted grammar* $G = (V, \Sigma, R, S)$ is just like a context-free grammar except that the rules are a finite subset of $(\Sigma \cup \Gamma)^* \Gamma (\Sigma \cup \Gamma)^* \times (\Sigma \cup \Gamma)^*$. Derivations in an unrestricted grammar are just like derivations in a CFG: if there is a rule $\alpha \to \beta$ and you see α in a sentential form, you can replace α by β (possibly resulting in the erasure or change of terminals). The language of G, $L(G)$ is the set of terminal strings derivable from the start symbol S.

Part A. Exhibit an unrestricted grammar for $L = \{ww : w \in \{a, b\}^*\}$

Part B. Prove that every Turing-acceptable language is generated by an unrestricted grammar.

Part C. Prove that there is no algorithm which takes an unrestricted grammar G and a word w and decides if $w \in L(G)$.

Problem 2 [**10 points**] Consider a **black-box-L Turing machine** M^L. Such a machine is just like an ordinary Turing machine, except that it has an "oracle" for the language L. That means that the Turing machine has a special *query tape* and, if the TM writes on that tape a string x and goes into a special **query state**, q_{query}, then the next state will be either q_{yes} or q_{no}, depending on whether or not x is in L.

Part A. Show that for every Turing-acceptable language L there exists a TM M such that $M^{A_{\text{TM}}}$ decides L.

Part B. Exhibit a language H such that for every TM M, $M^{A_{\text{TM}}}$ does not accept H. Prove your language has this property.

Part C. Exhibit a language B such that B is not Turing-acceptable but for which, for some TM M, $M^{A_{\text{TM}}}$ accepts B? Prove that your language has this property.

Problem 3. [**20 points**] Use the language of reductions in answering the problems below.

Part A. Show that

$$L_A = \{\langle M, k\rangle : M \text{ is a TM which accepts at least one string of length } k\}$$

is not decidable.

Part B. Prove that

$$L_B = \{\langle M, k\rangle : M \text{ is a TM that loops on at least one string of length } k\}$$

is *not* decidable.

Part C. Let

$$L_C = \{\langle M, k\rangle : M \text{ is a TM which accepts some string of length } k, \text{ but } M \text{ loops on some (other) string of length } k\}.$$

Show that L_C is not r.e.. (Assume the underlying alphabet has at least two characters.)

Part D. Show that L_C is not co-r.e., either.