Problem Set 7 — Due Wednesday, March 4, 1998

Problem 1. Recall that an *unrestricted grammar* $G = (V, \Sigma, R, S)$ is just like a context free grammar except that the rules are of the form $\alpha \to \beta$ for arbitrary sentential forms α and β — that is, R is a finite subset of $(V \cup \Sigma)^* \times (V \cup \Sigma)^*$. 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. The language of G, $L(G)$ is the set of terminal strings derivable from the start symbol S.

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

Part B. 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. Let $\Sigma = \{0, 1\}$. Let $\Gamma \supseteq \{0, 1, \#, \$, q_0, q_A\}$ be a larger alphabet. Fix a function *legal* : $\Gamma^4 \times \Gamma^4 \to \{0, 1\}$. We say that a string c is *well-formed* if $c = \$c_1\$c_2\$\cdots\$c_t\$$, for some number t, where $c_j \in (\Gamma - \$)^*$ (for every $1 \le j \le t$), and $|c_j| = |c_{j+1}|$ (for every $1 \le j < t$). Let $c = \$c_1\$c_2\$\cdots\$c_t\$$ be a a well-formed string, with $|c_1| = k$. We say that c is a *computation string* if $legal(c[i..i+3], c[i+k+1..i+k+4]) = 1$[1] for each $i \in \{1, \ldots, |c| - k - 4\}$. We say that c is *x-initial* if $x \in \Sigma^*$ and if $c_1 = q_0 x \#^i$ for some i. We say that c is an *accepting string* if c_t contains the character q_A.

Suppose that L is Turing acceptable. Define a function *legal* (depending on the Turing machine M that accepts L) such that $x \in L$ if and only if there is an x-initial accepting computation string, c_x.

Problem 3. 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 $M^{A_{\text{TM}}}$ can decide any acceptable language.

Part B. Prove that there is a natural language L that the black-box Turing machine $M^{A_{\text{TM}}}$ can *not* decide. *Hint:* Consider the "halting problem for black-box-A_{TM} Turing machines."

[1] By $x[l..r]$ we mean the substring of x which runs from positions l to r, where the leftmost character of x is numbered 1.

Problem 4. Please use the language of reductions for 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

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

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.