Problem Set 6 — Due Friday, February 28, 1997

Problem 1. Page 170, Problem 4.18.

Problem 2. 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 enumerable 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."

Problem 3.

Part A. Prove that

$$L_A = \{\langle M, w\rangle : M \text{ is a TM which uses at most 17 tape squares when run on } w\}$$

is decidable.

Part B1. Prove that

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

is Turing-acceptable.

Part B2. Prove that L_B is *not* decidable.

Part C1. Prove that

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

is *not* decidable.

Part C2. Show that L_C co-Turing-acceptable (co-r.e.).

Part D1. Let

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

Show that L_D is not decidable. (Assume the underlying alphabet has at least two characters.)

Part D2. Show that $\overline{L_D}$ is not decidable, either.