ECS120 Introduction to the Theory of Computation
Fall Quarter 2007

Homework 8 Help
Due Friday November 30, 2007

Now that mapping reducibility proofs have been covered, you are expected to use them! For any problem asking to **prove or show** if a language is decidable, recognizable (but not decidable), co-recognizable (but not decidable), or neither recognizable nor co-recognizable, you **must** use mapping reducibility where appropriate!

Problem 8.1

Classify the following languages as decidable, recognizable (but not decidable), co-recognizable (but not decidable), or neither recognizable nor co-recognizable. Prove all your answers, giving decision procedures or reductions.

(a) $L = \{ \langle M \rangle \mid M \text{ accepts some even-length string} \}$
(b) $L = \{ \langle M \rangle \mid M \text{ accepts some palindrome} \}$
(c) $L = \{ \langle M \rangle \mid L(M) \text{ is Turing-decidable} \}$
(d) $L = \{ \langle M \rangle \mid L(M) \text{ is Turing-recognizable} \}$

See the discussion notes (especially the table at the end) and use mapping reducibility.

Clearly mark if the language is decidable, recognizable, co-recognizable, or neither. Hint: There is at least one language that is neither recognizable nor co-recognizable.

Problem 8.2

Show that the following language is not Turing-recognizable:

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

(Assume that the underlying alphabet has at least two characters.)

You can use any method described in the discussion notes under subsection "Claim: L is not Turing-recognizable." However, remember you do need to use formal mapping reducibility proofs where appropriate. I recommend showing $\overline{A_{TM}} \leq_m L_C$.

Problem 8.3

Show that all Turing-recognizable problems mapping reduce to A_{TM}.

Let L be a Turing-recognizable language. You are asked to show that $L \leq_m A_{TM}$.

Problem 8.4

Show that $L = \{ \langle M \rangle \mid M$ is a TM and M has no useless states $\}$ is not decidable.

Expect this problem to be difficult and be prepared to use mapping reducibility! Try building your function f based on a machine M' that has a useless state if (and only if) M does not accept string w. Be careful of what input your function versus M' is given!

Problem 8.5

Prove the following

(a) 2^n is $\mathcal{O}(n!)$.

(b) $n!$ is $\mathcal{O}(2^{n^2})$.

(c) $2^{n+100000000}$ is $\mathcal{O}(2^n)$.

You are on your own for this question. You should hopefully know how to do this from a previous class. Otherwise, please review the book (there are quite a few examples).