COSC 4111/5111 —Winter 2008

Posted: Feb 15, 2008
Due: TBA

Problem Set No. 2

NB. *All problems are equally weighted out of 5.* The problem set list for grad students is the entire list here. Undergrads should omit the problems marked "Grad". If they wish to do some of those for extra credit the extra credit will be applied on an "all or nothing" basis. That is, **no part marks will be given** for a "Grad" problem attempted by undergrads.

This is not a course on *formal* recursion theory. Your proofs should be informal (but $\neq$ sloppy), correct, and informative (and if possible short). Please do not trade length for correctness or readability.

Most problems are from "Computability", **Chapters 3, 7, 8**. All relate to material in said chapters.

(1) **Ch.3.** Nos. 3, 5, 18 (ignore the hint!), 22, 23, 26.

(2) (**Grad**) Prove that a recursively enumerable set of sentences $\mathcal{T}$ over a finitely generated language (e.g., like that of arithmetic) admits a recursive set of axioms, i.e., for some recursive Γ, $\mathcal{T} = \mathbf{Thm}_\Gamma$.

(*Hint.* Note that for any $\mathcal{A} \in \mathcal{T}$, any two sentences in the sequence

$$\mathcal{A}, \mathcal{A} \wedge \mathcal{A}, \mathcal{A} \wedge \mathcal{A} \wedge \mathcal{A}, \ldots$$

are logically equivalent. Now see if Theorem 1 and its corollaries (section 8.2 in "Computability") can be of any help.)

(3) Prove that $\{x : \phi_x \in \mathcal{PR}\}$ is not r.e., that is, you cannot computably enumerate all the ϕ-indices that happen to describe just the primitive recursive functions.

(4) (**Grad**) Show here that you can do the 2nd best than what 3 says: You can enumerate a proper subset of the set of all the ϕ-indices, and this subset happens to define all the primitive recursive functions.

That is, prove that there is a $F \in \mathcal{R}$ such that

(a) If $\lambda\vec{x}.g(\vec{x}) \in \mathcal{PR}$, then for some e, $F(e, \langle\vec{x}\rangle) = g(\vec{x})$ for all $\vec{x}$.

(b) For all e, $\lambda z.F(e,z) \in \mathcal{PR}$.

(*Hint.* Use off the shelf our arithmetisation tools from the "Appendix", but see what happens if you drop all the codes e with $(e)_0 = 3$ (and correspondingly drop the predicate "$U(u)$" from the definition of $Tree(u)$ and $T^{(n)}$.)

(5) **Ch.7.** Nos. 1, 8.

(6) (**Grad**) **Ch.7.** No. 9, 13.

(7) **Ch.8.** No. 7.

(8) (**Grad**) **Ch.8.** No. 11, 12.