Posted: Nov. 20, 2021
Due: December 8, at 5pm in eClass.

Problem Set No. 3

NB. *All problems are equally weighted.*

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

All problems are from "Theory of Computation" or elaborations of such problems.

(1) **The graph theorem**. Prove that $y = f(\vec{x})$ is semi-recursive iff $f \in \mathcal{P}$.

Hint. For the left-to-right direction use the *strong projection theorem* to obtain

$$y = f(\vec{x}) \equiv (\exists z)Q(z, \vec{x}, y)$$

where Q is recursive.

Now find the *smallest* $w = \langle z, y \rangle$ such that $Q(z, \vec{x}, y)$ is true and return y.

(2) Do the *definition by positive cases* exercise (p.171–172).

Unlike Problem (1) of Assignment #2, this time you must **NOT** use **CT**. Rather use a *fully mathematical proof* (based on Problem (1) directly above).

(3) Use Rice's theorem to prove that every partial recursive f has infinitely many indices.

Hint. It suffices to show that the set $\{x : \phi_x = f\}$ is not recursive.

(4) (**Grad**)Use the recursion theorem to prove that K is NOT a complete index set, that is, there is no $\mathcal{C} \subseteq \mathcal{P}$ such that we have $K = \{x : \phi_x \in \mathcal{C}\}$.

(5) From the fact that the simulating functions for URMs are in $\mathcal{E}^2$ conclude that the Kleene predicate is in $\mathcal{E}^2_*$ and hence in $\mathcal{L}_{2,*}$.

Further conclude that the equivalence problem for L_2 programs is unsolvable.

Hint. In the text there is a proof via the Kleene predicate that the equivalence problem for $\mathcal{PR}$ functions is not semi-recursive (see Text!).

Imitate this argument for $\mathcal{L}_{2,*}$.