Math 114C, Winter 2019, Solutions to HW #4

---

x3A.1. Prove Lemma 3A.2.
Solution. Set $f(l,u,i) = \mathrm{seg}(u,i,i+l+1)$, so that

$$\mathrm{seg}(u,i,j) = f(j \dot{-} i \dot{-} 1, u, i) \quad (0 \le i < j \le \mathrm{lh}(u)),$$

and it is enough to prove that $f(l,u,i)$ is primitive recursive. For this we compute:

$$\begin{aligned} f(0,u,i) &= \mathrm{seg}(u,i,i+1) = \langle (u)_i \rangle = 2^{(u)_i+1}, \\ f(l+1,u,i) &= \mathrm{seg}(u,i,i+l+1) = \mathrm{seg}(u,i,l+1) * \langle (u)_{i+l} \rangle \\ &= \mathrm{seg}(u,i,l+1) \cdot p_{i+l}^{(u)_{i+l}+1}; \end{aligned}$$

so f can be defined by primitive recursion from primitive recursive functions and it is primitive recursive.

For the second claim, check that

$$\langle u_0, \dots, u_{n-1} \rangle = x \cdot p_i^{u_i+1} p_{i+1}^{u_{i+1}+1} \cdots p_{j-1}^{u_{j-1}+1} \cdot y \ge p_i^{u_i+1} p_{i+1}^{u_{i+1}+1} \cdots p_{j-1}^{u_{j-1}+1}$$

for some x and y, and obviously,

$$p_i^{u_i+1} p_{i+1}^{u_{i+1}+1} \cdots p_{j-1}^{u_{j-1}+1} \ge p_0^{u_i+1} p_1^{u_{i+1}+1} \cdots p_{j \dot{-} i}^{u_{j-1}+1} = \mathrm{seg}(u,i,j).$$

The function $\mathrm{seg}(u,i,j)$ is primitive recursive for every primitive recursive coding (by the proof above), but the inequality $\mathrm{seg}(u,i,j) \le u$ does not hold for the primitive recursive coding which simply reverses the first two prime numbers in the classical coding:

$$\langle x_0, \dots, x_{n-1} \rangle' = 3^{x_0+1} \cdots 2^{x_1+1} \cdot p_2^{x_2+1} \cdots,$$

because $\langle 0, 2 \rangle' = 3 \cdot 2^3 = 24 < 27 = 3^3 = \langle 2 \rangle'$.

x3A.2. Prove that every recursive partial function $f(\vec{x})$ has infinitely many codes, i.e., there exist infinitely many numbers e such that $f = \varphi_e^n$.
Solution. If E is a program which computes f and p_i^1 is any one-place function variable not occurring in E, then $\{\mathsf{p}_i^1(x) = \mathsf{p}_i^1(x)\} + E$ again computes f, so that if its code is e_i, then $f = \varphi_{e_i}^n$; and there are infinitely many symbols which do not occur in E.

The following alternative proof is a little better, as it does not refer to the specific coding of the recursive partial functions: the partial function

$$g(m, \vec{x}) = f(\vec{x})$$

is recursive; therefore, for some $\widehat{g}$ and all $m, \vec{x}$,

$$f(\vec{x}) = g(m, \vec{x}) = \varphi_{\widehat{g}}^{n+1}(m, \vec{x}) = \varphi_{S_n^1(\widehat{g}, m)}^n(\vec{x}),$$

so that for every m, $S_n^1(\widehat{g}, m)$ is a code of f; but S_n^1 is one-to-one, so f has infinitely many codes.

x3A.3. Prove that if $T_n(e, \vec{x}, y)$ is defined by (67) and $U(y)$ is defined by (70), then (68) holds.

Solution. This involves unravelling the coding. If y is the code of a convergent computation, then $y = \langle [s_0]_5, \dots, [s_n]_5 \rangle$ where s_n is the last stage, and so $[s_n]_5 = \mathrm{last}(y)$. Since it is a terminal computation, s_n is of the form $: w$ for the output w, and by the coding definition, $[s_n]_5 = \langle 1, \langle [w]_1 \rangle \rangle$, so $\langle [w]_1 \rangle = ([s_n]_5)_1 = \mathrm{last}(y)_1$ and $[w]_1 = \mathrm{last}(y)_{1,0}$. Finally, by the coding of numbers, $[w]_1 = \langle 0, 1, w \rangle$ and so $w = \mathrm{last}(y)_{1,0,2}$. (The full proof needs some argument here, but this is given on p. 60).

x3A.5. Prove that the relation $\mathrm{Transition}(e, s, s')$ is primitive recursive. (This computation has many details and it is not feasible to record them all. What is required in this Problem is to explain the architecture of the proof, and to work out some of the more interesting cases.)

Solution. There is no written solution of this problem: the best of the solutions which will be turned in will be added here.

x3A.6. Prove that some primitive recursive function $u(n)$ gives for each n a code of the Ackermann section $A_n(x)$.

Solution. Let a be some code of the Ackermann function, which is recursive; we observe that

$$A_n(x) = \varphi_a(n, x) = \varphi_{S^1_1(a,n)}(x),$$

so that for each n, the number $S^1_1(a, n)$ is a code of $A_n(x)$.

x3A.8. Prove that there is a recursive partial function $f(x)$ which does not have a total recursive extension, i.e., there is no total recursive function g such that $f \sqsubseteq g$.

Solution. Let $f(x) = 1 \dot{-} \varphi^1_x(x)$. If (towards a contradiction) $f \sqsubseteq \varphi^1_e$ for some total φ^1_e, then we have, from the definition and the hypothesis

$$1 \dot{-} \varphi^1_x(x)\downarrow \implies 1 \dot{-} \varphi^1_x(x) = \varphi^1_e(x),$$

which for $x = e$ gives

$$1 \dot{-} \varphi^1_e(e)\downarrow \implies 1 \dot{-} \varphi^1_e(e) = \varphi^1_e(e);$$

but the hypothesis of this implication is true (because φ^1_e is total), so, with $\varphi^1_e(e) = w$ we have $1 \dot{-} w = w$, which is absurd.

`Let me know of errors or better solutions.`