コンピュータサイエンス基礎（2015 前期） レポート課題 解説

問 I (計算可能性と帰納的関数の理論)

(1) いろいろな解き方がありますが、以下はその一例（の略解）です。いずれについても、原始帰納的関数として具体的に定義し、それが求められた条件を満たしていることを示していれば十分です。まず、sub は原始帰納法を用いて簡単に定義できます。

$$\begin{array}{lclcl} \mathsf{sub}(x,0) & = & P^1_1(x) & = & x \\ \mathsf{sub}(x,y+1) & = & \mathsf{pre}(P^3_3(x,y,\mathsf{sub}(x,y))) & = & \mathsf{pre}(\mathsf{sub}(x,y)) \end{array}$$

eq は、sub を用いて、原始帰納的関数の合成として定義することができます。

$$\mathsf{eq}(x,y) \ = \ \mathsf{sub}(\mathsf{sub}(1,\mathsf{sub}(x,y)),\mathsf{sub}(y,x))$$

pair については、まず、$\mathsf{pair}(x,0) = \dfrac{1}{2}x(x+1) = \displaystyle\sum_{i=0}^{x} i$ を計算する原始帰納的関数 p を定義します。

$$\begin{array}{lcl} \mathsf{p}(0) & = & 0 \\ \mathsf{p}(y+1) & = & S(\mathsf{add}(y,\mathsf{p}(y))) \end{array}$$

pair は p と add を用いて原始帰納的関数の合成として表すことができます。

$$\mathsf{pair}(x,y) \ = \ \mathsf{p}(x+y)+y \ = \ \mathsf{add}(\mathsf{p}(\mathsf{add}(x,y)),y)$$

fst と snd は苦労した人が多かったようです。これは、講義資料のヒントに沿って解くのが一番やさしいように思います。講義資料に書いたように、$f : \mathbf{N}^{n+1} \to \mathbf{N}$ が原始帰納的であるとき、$f^+(x_1,\ldots,x_n,y) \ = \ \sum_{i=0}^{y} f(x_1,\ldots,x_n,i)$ で定まる $f^+ : \mathbf{N}^{n+1} \to \mathbf{N}$ も原始帰納的です（証明は簡単です）。特に、$f(x,z,y) = \mathsf{eq}(\mathsf{pair}(x,y),z)$ で定義される原始帰納的関数 f について $h(x,z) = f^+(x,z,z)$ とおけば、pair が全単射であること（要証明）と $\mathsf{pair}(x,y) \geq y$ がなりたつことから、

$$h(x,z) = \begin{cases} 1 & \text{ある } y \text{ について } \mathbf{pair}(x,y) = z \\ 0 & \text{それ以外} \end{cases}$$

が成り立ちます。さらに、$k(z,x) = \mathsf{mul}(x,h(x,z))$ で定まる原始帰納的関数 k を用いて、fst は

$$\mathsf{fst}(z) \ = \ k^+(z,z)$$

で与えられます。snd も同様に（f,h,k の定義で x と y を適宜入れ替えて）原始帰納的に与えることができます。

最後に、fib は、講義資料のヒントにある原始帰納的関数 g を用いて、$\mathsf{fib}(x) = \mathsf{fst}(g(x))$ と定義することができます。

(2) この問題は、講義の内容を参考に、素直に解いてもらえば簡単だと思います。

(a) もしも g が帰納的関数であるとすると、ある $i \in \mathbf{N}$ について $g(x) = \Phi^1(i,x)$ となるはずです。しかし、$g(i)$ を考えてみると、$\Phi^1(i,i) = g(i)$ が未定義ならば g の定義より $g(i) = 0$ でなくてはなりませんし、といって $\Phi^1(i,i) = g(i)$ が定義されているならば $g(i)$ は未定義でなくてはなりませんので、矛盾します。

(b) もしもこの集合が帰納的集合であるなら、集合

$$A = \{x \in \mathbf{N} \mid \Phi^1(x,x) \text{ が定義されている } \}$$

も帰納的集合であるはずです。しかし、(a) より、A（g の定義域です）は、帰納的可算集合ですらありません。

問 II (ラムダ計算)

(1) 自然数 m, n に対して $\overline{m}\ \overline{n} \to_\beta^* \overline{2m+n}$ となることを示せば十分。さて $\overline{m}$ の定義から

$$\overline{m}\ \overline{n} \to_\beta^* M\ M\ \lceil n \rceil\ \lceil m \rceil$$

となる。次に不動点コンビネータに関するベータ簡約より

$$\begin{aligned} M\ M\ \lceil n \rceil\ \lceil m \rceil \quad &\to_\beta^* \quad (\lambda bnmx\ .\ x\ M\ (\mathbf{Add}(\mathbf{Add}\ m\ m)\ n))\ M\ \lceil n \rceil\ \lceil m \rceil \\ &\to_\beta^* \quad (\lambda x\ .\ x\ M\ (\mathbf{Add}(\mathbf{Add}\ \lceil m \rceil\ \lceil m \rceil)\ \lceil n \rceil)) \\ &\to_\beta^* \quad \overline{2m+n} \end{aligned}$$

となり、$\overline{m}\ \overline{n} \to_\beta^* \overline{2m+n}$ が示せた。

(2) (a) $\lambda x\ .\ x\ \mathbf{\Omega}$、$\lambda x\ .\ y\ (\mathbf{I}\ \mathbf{I})$、$\lambda xy\ .\ y\ \mathbf{I}\ \mathbf{\Omega}$ などがある (括弧の付け方に注意)。

(b) 自由変数を持たない冠頭正規系 M は常に

$$\lambda x_1 \cdots x_m\ .\ x_i\ M_1 \cdots M_n \quad (1 \le i \le m, 0 \le n)$$

という形をしている。そこで $N = \lambda x_1 \cdots x_n\ .\ \mathbf{I}$ とし、$M\ \overbrace{N \cdots N}^{m}$ とすると以下を得る:

$$M\ \overbrace{N \cdots N}^{m} \to_\beta^* N\ M_1 \cdots M_n \to_\beta^* \mathbf{I}$$

問 III (型付ラムダ計算と表示的意味論)

(1) (a) と (b) ともに与えられた型判定を導出可能とするような A は存在しません。

(a) $\lambda w : A. M$ という形のラムダ項に対する型判定の導出規則は「$\Gamma, w : A \rhd M : B$ が導出可能なら $\Gamma \rhd \lambda w : A. M : A \Rightarrow B$ が導出可能」というものだけでした。従ってもし $\Gamma \rhd \lambda w : A. M : \beta$ が導出可能なら $\beta = A \Rightarrow C$ となる型 C の存在が示せますが明かに $\beta \neq A \Rightarrow C$ であり矛盾します。

(b) $x : \beta \Rightarrow \beta \rhd (\lambda y : A. x(yy))(\lambda y : A. x(yy)) : \beta \Rightarrow \beta$ が導出可能であるとするとその導出は最後に型判定の導出規則「$\Gamma \rhd M : X \Rightarrow Y$ と $\Gamma \rhd N : X$ が導出可能なら $\Gamma \rhd MN : Y$ が導出可能」を使っているはずなので $x : \beta \Rightarrow \beta \rhd \lambda y : A. x(yy) : B$ が導出可能となるような型 B の存在が示せます。同様の議論を繰り返すことで $x : \beta \Rightarrow \beta, y : A \rhd yy : C$ を導出可能とする型 C の存在が示せます。しかしこのような型判定を導出可能にする型 A と C が存在しないことは講義中の議論から分ります。

(2) 色々な答えが考えられますが、例えば $- \rhd \mathbf{fix}(\lambda x : \beta. x) : \beta$ が挙げられます。

(3) (a) と (b) は単純に定義に従って計算していけば十分です。(c) については $Fx \to_\beta^* x(Fx)$ という β 簡約列の存在から矛盾を導きます。もし $Fx \to_\beta^* x(Fx)$ が成り立つなら健全性から $[\![x : \beta \Rightarrow \beta \rhd Fx : \beta]\!] = [\![x : \beta \Rightarrow \beta \rhd x(Fx) : \beta]\!]$ が従います。よって (a) と (b) からどのような関数 $u : \mathbb{N} \to \mathbb{N}$ に対しても $(f(u))(u) = u((f(u))(u))$ が成り立つはずですが u として $u(n) = n + 1$ で与えられる関数をとると $(f(u))(u) \neq 1 + (f(u))(u) = u((f(u))(u))$ となり矛盾します。