2008 年度 計算数学 I レポート課題

部分再帰関数は, ラムダ計算で表現できる. 例を通して, その手法を調べてみよう. 自然数は, いわゆる Church numeral によって表現する.

$$\begin{aligned} 0 &= \lambda z \lambda y.\, z \\ 1 &= \lambda z \lambda y.\, yz \\ 2 &= \lambda z \lambda y.\, y(yz) \\ &\vdots \end{aligned}$$

一般的に, 自然数 n は,

$$n = \lambda z \lambda y.\, \overbrace{y(y(y(\cdots(y}^{n}\, z)\cdots)))$$

で表現されるが, この項は書きにくいので, $\lambda z \lambda y.\, y^n z$ と略記することにしよう. 部分再帰関数は, 原始帰納関数をベースに作るので, まずこちらを考えてみる. 原始帰納関数の本質は, 原始帰納法

$$\begin{aligned} f(0) &= a \\ f(n+1) &= g(f(n), n) \end{aligned}$$

にある. 一般の場合は後回しにして,

$$\begin{aligned} f(0) &= a \\ f(n+1) &= g(f(n)) \end{aligned}$$

のように, $f(n)$ の値だけで $f(n+1)$ が定まる場合を考える. つまり $f(n) = g(f(n-1)) = g(g(f(n-2))) = \cdots = g^n(f(0)) = g^n(a)$ ということである. このときは, 関数 f は単純に,

$$f = \lambda x.\, xag$$

で表現される. 右辺の g は, 原始帰納関数 g を表現するラムダ項のことであり, 本当なら関数 g とそれを表現するラムダ項は区別すべきだろうが, 簡単のため同一の記号 g を用いてしまっている. この f の表現の正当性をチェックしておくと,

$$\begin{aligned} fn &=_\beta (\lambda x.\, xag)(\lambda z \lambda y.\, y^n z) \\ &=_\beta (\lambda z \lambda y.\, y^n z)ag \\ &=_\beta g^n a \end{aligned}$$

となって, これは上の考察 $f(n) = g^n(a)$ と一致している.

問題 1. 足し算 $\mathsf{add}(m, n) = m + n$ を表現するラムダ項は, $\mathsf{S} = \lambda x \lambda z \lambda f.\, f($

xzf) (これは 1 を足す関数である) を用いて, add $= \lambda x \lambda y.\, yx\mathsf{S}$ と表現される. これを既知として, かけ算 mult$(m, n) = m \times n$ を表現するラムダ項を作れ (かけ算を原始帰納関数として表現し, それに上の議論を適用すればよい).

一般の $g(f(n), n)$ の形のときは工夫が必要である. その工夫とは, $f(n)$ と n のペアを常に考えることである. まずペアをラムダ項で表現する必要がある. 二つのラムダ項のペア $\langle M, N\rangle$ を

$$\langle M, N\rangle \;=\; \lambda x.\, xMN$$

で表現する. また, 第 1 成分への射影を $\pi = \lambda x \lambda y.\, x$ で表現する.

問題 2. 同様に第 2 成分への射影 π' をラムダ項で表現し, $\pi\langle M, N\rangle =_\beta M$ および $\pi'\langle M, N\rangle =_\beta N$ が成り立つことを示せ.

ペアを使って, 一般の原始帰納法

$$\begin{aligned} f(0) &= a \\ f(n+1) &= g(f(n), n) \end{aligned}$$

を書き直す. $f^\dagger(n) = \langle f(n), n\rangle$ で, 新たな関数 $f^\dagger$ を定めると, この関数は原始帰納的に

$$\begin{aligned} f^\dagger(0) &= \langle a, 0\rangle \\ f^\dagger(n+1) &= \langle g(\pi(f^\dagger(n)), \pi'(f^\dagger(n))),\, \pi'(f^\dagger(n))\rangle \end{aligned}$$

で与えられる. ポイントは, $f^\dagger(n+1)$ が, $f^\dagger(n)$ のみを用いて定義されているところである. もとは, $g(f(n), n)$ と n に依存した式だったが, この n が $\pi'(f^\dagger(n))$ で置き換えられている.

問題 3. 上の $f^\dagger(n)$ が確かに $\langle f(n), n\rangle$ に等しいことを示せ.

問題 4. $f^\dagger$ を表現するラムダ項を与えよ.

このとき元の関数 f は, 第 1 射影を用いて, $\lambda x.\, \pi(f^\dagger(x))$ とすればよい. 以上が, 一般の原始帰納法で与えられる関数 f をラムダ項として表現する方法である.

一つの例として, $\ulcorner n/2 \urcorner$ を表現するラムダ項を作ってみよう. ここで, $\ulcorner n/2 \urcorner$ とは, $n/2$ 以上の最小の自然数のことである.

まず真偽値として $\top = \lambda x \lambda y.\, x$ (真), および $\bot = \lambda x \lambda y.\, y$ (偽) と定める. 上に与えた射影 π 及び π' の式と同一だが, コンテキストによって使い分ける.

問題 5. if-then-else 節を表現するラムダ項を設計したい. if P then M else N

をラムダ項 L として表現して, $P =_\beta \top$ のときは $L =_\beta M$ になり, $P =_\beta \bot$ のときは, $L =_\beta N$ になるようにしたいということである. そのようなラムダ項 L を与えよ.

偶奇性を判定する関数 $\mathsf{even}(x)$ を次に作る. この関数は原始帰納法によって

$$\begin{aligned}\mathsf{even}(0) &= \top \\ \mathsf{even}(n+1) &= \neg\, \mathsf{even}(n)\end{aligned}$$

で定義できる. $\neg X$ とは, X の真偽を反対にする関数で, $\mathsf{if}\ X\ \mathsf{then}\ \bot\ \mathsf{else}\ \top$ で与えられる. 以上の準備のもとで, $\ulcorner n/2 \urcorner$ は原始帰納法

$$\begin{aligned}\ulcorner 0/2 \urcorner &= 0 \\ \ulcorner (n+1)/2 \urcorner &= \mathsf{if}\ \mathsf{even}(n)\ \mathsf{then}\ \ulcorner n/2 \urcorner + 1\ \mathsf{else}\ \ulcorner n/2 \urcorner\end{aligned}$$

で定義できる. この原始帰納法の 2 番目の式は, n にも依存していることに注意する.

問題 6. $\ulcorner n/2 \urcorner$ を表現するラムダ項を作れ. また小さい自然数 n に対して, 作られたラムダ項が正しく働くことを確認せよ.

部分帰納関数は, 原始帰納関数のスキームにもう一つのスキーム $f(x_1, \ldots, x_n) = \mu y.\, g(x_1, \ldots, x_n, y)$ を付け加えて作られる関数族のことであった. ここで, $\mu y.\, g(x_1, \ldots, x_n, y)$ とは, $g(x_1, \ldots, x_n, y) = 0$ を満たす最小の y のことである. しかしかわりに, 再帰的な関数定義が実現できれば, 本質的には同じことである. そこで, 再帰的な関数定義をラムダ計算で実現することを考えてみよう.

それには不動点演算子を用いる. いわゆる Curry の不動点演算子 $\mathsf{Y} = \lambda y.\, (\lambda x.\, y(xx))(\lambda x.\, y(xx))$ を用いてもよいのだが, 目先を変えて Turing の不動点演算子と呼ばれるラムダ項を用いてみる.

問題 7. Turing の不動点演算子は $\Theta = (\lambda x \lambda y.\, y(xxy))(\lambda x \lambda y.\, y(xxy))$ で定義される. 任意のラムダ項 M に対して, ΘM は $M(\Theta M)$ に β-簡約されることを示せ.

したがって, 特に, $\Theta M =_\beta M(\Theta M)$ である. Curry の不動点演算子に対しても同様の式 $\mathsf{Y}M =_\beta M(\mathsf{Y}M)$ は成り立つ. しかし, $\mathsf{Y}M$ を β-簡約して $M(\mathsf{Y}M)$ になる訳ではない.

部分帰納関数の例として, 素数かどうかを判定する関数 $\mathsf{prime}(x)$ を作ってみよう. $x \geq 2$ が素数ならば真 $\top$ を返し, そうでなければ偽 $\bot$ を返すような関数である. ちなみに, この関数は実は原始帰納関数であり, したがって, 本当

は不動点演算子を使わなくても表現できる．しかしだいぶ複雑になるであろう．ここでは, 再帰的定義を使って構成してみよう．

x が素数かどうか判定するには, 効率は悪いが, $y = 2, 3, \ldots, \ulcorner x/2 \urcorner - 1$ で順に割ってみればよい．どれかで割り切れれば素数ではない．

x を y で割った余りを返す関数 $\mathsf{rem}(x, y)$ は既にラムダ項として与えられているとしよう. 原始帰納関数として表せるので, ちょっとがんばれば具体的に構成することはできるが, 長くなるのでここでは与えられているとする．また, x と y が等しいとき真 $\top$ を返し, そうでないとき偽 $\bot$ を返す関数 $\mathsf{eq}(x, y)$ も与えられているとする．

次のように関数 g を再帰的に定義する:

$$g(y) \;=\; \mathsf{if}\ \mathsf{eq}(y, \ulcorner x/2 \urcorner)\ \mathsf{then}\ \top \\ \qquad\qquad \mathsf{else\ if}\ \mathsf{eq}(\mathsf{rem}(x, y), 0)\ \mathsf{then}\ \bot \\ \qquad\qquad\qquad \mathsf{else}\ g(y+1)$$

問題 8. 関数 g を表現するラムダ項を, Turing の不動点演算子を用いて与えよ．上に述べたように rem と eq は既知としてそのままでよい．また $\ulcorner x/2 \urcorner$ も上で作ったので, ここではラムダ項に無理に変換しなくてよい．

問題 9. 素数かどうか判定する関数 $\mathsf{prime}(x)$ をラムダ項で表現せよ．(g の意味を考えれば, すぐにわかるはずである.)