計算と論理

Jacques Garrigue, 2020 年 12 月 17 日

9　型付き λ 計算

λ 計算の関数は数学的関数と違い、定義域と値域を定めていない。例えば、前に見た c_+ は整数以外の物に適用しても意味がないが、そのような項は正しくないと判断できない。

9.1　型 (type) と項

型付き λ 計算では、そのような領域を**型**で表す。チャーチがラッセルの影響を受けて 1941 年に発表した単純型付 λ 計算では全ての値は一つだけの型に属する。型には二種類があり、基底型と関数・構造型に分けられる。

$$\begin{array}{rcl} b & ::= & \mathsf{nat} \mid \mathsf{bool} \mid \ldots \\ \tau & ::= & b \mid \tau \to \tau \mid \tau \times \tau \end{array}$$

項の中にも、型情報を入れ、定数も追加する。

$$M ::= x \mid c_\tau \mid \lambda x{:}\tau.M \mid (M\ M) \mid (M, M)$$

簡約規則には、β 簡約に加えて定数と δ 規則も導入される。例えば、対と自然数に関する計算を可能にするために、以下の定数を導入する。

$$\begin{array}{ll} \text{対} & \mathsf{fst}_{\tau\times\theta\to\tau}, \mathsf{snd}_{\tau\times\theta\to\tau} \\ \text{自然数} & \mathsf{0}_{\mathsf{nat}}, \mathsf{s}_{\mathsf{nat}}, \mathsf{iter}_{\mathsf{nat}\to(\tau\to\tau)\to\tau\to\tau} \end{array}$$

ここで 0 や s は固定の型を持っているが、fst, snd と iter は様々な型について定義しなければならない。τ や θ はそういう任意に決められる型を表している。なお、項の適用とは逆に、$\to$ の右辺の括弧を省略していい。

$$\mathsf{nat} \to (\tau \to \tau) \to \tau \to \tau = \mathsf{nat} \to ((\tau \to \tau) \to (\tau \to \tau))$$

上の定数に対して以下のような δ 規則を導入すれば計算が可能になる。

$$\begin{array}{rcl} (\lambda x{:}\tau.M)\ N & \to & [N/x]M \\ (\mathsf{fst}_{\tau\times\theta\to\tau}\ (M, N)) & \to & M \\ (\mathsf{snd}_{\tau\times\theta\to\theta}\ (M, N)) & \to & N \\ (\mathsf{iter}_{\mathsf{nat}\to(\tau\to\tau)\to\tau\to\tau}\ \mathsf{0}_{\mathsf{nat}}\ M\ N) & \to & N \\ (\mathsf{iter}_{\mathsf{nat}\to(\tau\to\tau)\to\tau\to\tau}\ (\mathsf{s}_{\mathsf{nat}\to\mathsf{nat}}\ P)\ M\ N) & \to & (\mathsf{iter}_{\mathsf{nat}\to(\tau\to\tau)\to\tau\to\tau}\ P\ M\ (M\ N)) \\ & \ldots & \end{array}$$

9.2 型推論

ある項の正しさ (well-formedness) を型判定式で表す。

$$\Gamma \vdash M : \tau$$

M は項であり、τ は型である。そして Γ は型宣言列と呼ばれる変数名と型の集合 $(x_1 : \tau_1, \ldots, x_n : \tau_n)$。

次の型推論規則で型判定式が証明された項を正しいとする。

変数 $\Gamma \vdash x : \tau \quad (x : \tau \text{は} \Gamma \text{に含まれる})$

定数 $\Gamma \vdash c_\tau : \tau$

直積 $$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \theta}{\Gamma \vdash (M, N) : \tau \times \theta}$$

抽象 $$\frac{\Gamma, x : \theta \vdash M : \tau}{\Gamma \vdash \lambda x{:}\tau.M : \theta \to \tau}$$

適用 $$\frac{\Gamma \vdash M : \theta \to \tau \quad \Gamma \vdash N : \theta}{\Gamma \vdash (M\ N) : \tau}$$

証明の例

$$\dfrac{\dfrac{\dfrac{x : \mathsf{nat} \vdash \mathsf{s}_{\mathsf{nat}\to\mathsf{nat}} : \mathsf{nat} \to \mathsf{nat} \quad x : \mathsf{nat} \vdash x : \mathsf{nat}}{x : \mathsf{nat} \vdash (\mathsf{s}_{\mathsf{nat}\to\mathsf{nat}}\ x) : \mathsf{nat}}}{\vdash \lambda x{:}\mathsf{nat}.(\mathsf{s}_{\mathsf{nat}\to\mathsf{nat}}\ x) : \mathsf{nat} \to \mathsf{nat}} \qquad \vdash 0_{\mathsf{nat}} : \mathsf{nat}}{\vdash ((\lambda x{:}\mathsf{nat}.(\mathsf{s}_{\mathsf{nat}\to\mathsf{nat}}\ x))\ 0_{\mathsf{nat}}) : \mathsf{nat}}$$

問題 1 以下の判定式の証明を書け。

$$\vdash \lambda x{:}\mathsf{nat}.\lambda y{:}\mathsf{nat}.(\mathsf{iter}_{\mathsf{nat}\to(\mathsf{nat}\to\mathsf{nat})\to\mathsf{nat}\to\mathsf{nat}}\ x\ \mathsf{s}_{\mathsf{nat}\to\mathsf{nat}}\ y) : \mathsf{nat} \to \mathsf{nat} \to \mathsf{nat}$$

9.3 性質

次の定理は fst, snd 以外の δ 規則を含まない計算系に関するものである。

定理 1 (主部簡約) $\Gamma \vdash M : \tau$ と $M \to N$ が成り立てば、$\Gamma \vdash N : \tau$ が成り立つ。

定理 2 (停止性) $\Gamma \vdash M : \tau$ ならば、無限な簡約列 $(M \to M_1 \to M_2 \to \ldots)$ は存在しない。

停止性が成り立てば、定義できないものがある。例えば、型のない λ 計算で停止しなかった $(\lambda x.(x\ x))(\lambda x.(x\ x))$ は定義できない。それを見せるために、$\lambda x{:}\tau.x\ x$ の証明を書いてみよう。

$$\dfrac{\dfrac{x : \tau \vdash x : \tau \to \theta \quad x : \tau \vdash x : \tau}{x : \tau \vdash (x\ x) : \theta}}{\vdash \lambda x{:}\tau.(x\ x) : \tau \to \theta}$$

項の構造から、以上の形の証明しかできないが、条件として $\tau = (\tau \to \theta)$ を満す必要がある。当然ながら、上記の型の定義では、こんな方程式を満すような τ は存在しない。

同様に Y を導入すると、止らない計算ができるので、型付 λ 計算では Y は定義できない。

9.4 万能性

以上のδ規則と停止性で分かるように、単体では型付λ計算は万能ではない。ただし、二種類の異る問題がある。

整数などに関するδ規則について、チャーチ数が型付けできないのは原因である。細かく言えば、型付けはできるが、汎用性が足りない。

$$\vdash \lambda f{:}\tau \to \tau.\lambda x{:}\tau.f^n\ x : (\tau \to \tau) \to \tau \to \tau$$

問題は、この関数はある特定の型τにおいてしか使えない。チャーチ数の計算は様々なものをxに代入することを利用していたので、これでは意味のあるエンコーディングはできない。この問題は、計算をδ規則に移すことで解決される。

同様に、再帰的な計算をしたければ、Yのδ規則を入れる必要がある。

$$\mathsf{Y}_{(\tau\to\tau)\to\tau}\ M \to M\ (\mathsf{Y}_{(\tau\to\tau)\to\tau}\ M)$$

整数とYだけを加えれば、型付λ計算は万能になる。実行にはδ規則を使うか、定数を型のないλ計算の項に変えて型のないλ計算で実行するかのどちらもができる。

型体系を強くすれば、汎用性のある定義ができるようになり、チャーチ数の定義がそのまま使える。**二次λ計算** (ジラールのシステムF) では型変数が用意されている。

$$\begin{array}{rcl} \tau & ::= & \ldots \mid \alpha \mid \forall\alpha.\tau \\ M & ::= & \ldots \mid \Lambda\alpha.M \mid M[\tau] \end{array}$$

型推論規則も追加する。

型抽象 $\dfrac{\Gamma \vdash M : \tau}{\Gamma \vdash \Lambda\alpha.M : \forall\alpha.\tau}$ **型適用** $\dfrac{\Gamma \vdash M : \forall\alpha.\tau}{\Gamma \vdash M[\theta] : [\theta/\alpha]\tau}$

型に関してもβ簡約が定義される。

$$(\Lambda\alpha.M)[\tau] \to [\tau/\alpha]M$$

二次λ計算では、チャーチ数は次のように表現できる。

$$\begin{array}{rcl} Nat & = & \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha \\ \mathsf{c}_n & = & \Lambda\alpha.\lambda f{:}\alpha \to \alpha.\lambda x{:}\alpha.f^n x \\ \mathsf{c}_+ & = & \lambda m{:}Nat.\lambda n{:}Nat.\Lambda\alpha.\lambda f{:}\alpha \to \alpha.\lambda x{:}\alpha.(m[\alpha]\ f\ x\ (n[\alpha]\ f\ x)) \\ \mathsf{c}_\times & = & \lambda m{:}Nat.\lambda n{:}Nat.\Lambda\alpha.\lambda f{:}\alpha \to \alpha.(m[\alpha]\ (n[\alpha]\ f)) \\ \mathsf{c}_{\mathsf{pow}} & = & \lambda m{:}Nat.\lambda n{:}Nat.\Lambda\alpha.n[\alpha \to \alpha]\ m[\alpha] \end{array}$$

そうすると、以下の型判定が証明できる。

$$\vdash \mathsf{c}_n : Nat \qquad\qquad \vdash \mathsf{c}_+, \mathsf{c}_\times, \mathsf{c}_{\mathsf{pow}} : Nat \to Nat \to Nat$$

しかし、二次λ計算は型付λ計算と同様に停止性を持っているので、相変わらずYは定義できない。それは欠点とは限らない。全ての計算が止まるということは、全ての項に計算できる意味があるということでもある。

さらに進んで、再帰的な型を持ったλ計算を使えば、Yも定義できるようになるが、停止性が失われる。