

# 計算と論理

Jacques Garrigue, 2023年12月20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。

$$\begin{aligned} b &::= \text{nat} \mid \text{bool} \mid \dots \\ \tau &::= b \mid \tau \rightarrow \tau \mid \tau \times \tau \end{aligned}$$

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。

$$\begin{aligned} \text{対} & \quad \text{fst}_{\tau \times \theta \rightarrow \tau}, \text{snd}_{\tau \times \theta \rightarrow \theta} \\ \text{自然数} & \quad 0_{\text{nat}}, \text{s}_{\text{nat} \rightarrow \text{nat}}, \text{iter}_{\text{nat} \rightarrow (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau} \end{aligned}$$

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。 $\tau$ や $\theta$ 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、 $\rightarrow$ の右辺の括弧を省略していい。

$$\text{nat} \rightarrow (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau = \text{nat} \rightarrow ((\tau \rightarrow \tau) \rightarrow (\tau \rightarrow \tau))$$

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。

$$\begin{aligned} (\lambda x:\tau. M) N &\rightarrow [N/x]M \\ (\text{fst}_{\tau \times \theta \rightarrow \tau} (M, N)) &\rightarrow M \\ (\text{snd}_{\tau \times \theta \rightarrow \theta} (M, N)) &\rightarrow N \\ (\text{iter}_{\text{nat} \rightarrow (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau} 0_{\text{nat}} M N) &\rightarrow N \\ (\text{iter}_{\text{nat} \rightarrow (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau} (\text{s}_{\text{nat} \rightarrow \text{nat}} P) M N) &\rightarrow (\text{iter}_{\text{nat} \rightarrow (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau} P M (M N)) \\ &\dots \end{aligned}$$

## 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、 $\tau$  は型である。そして  $\Gamma$  は型宣言列と呼ばれる変数名と型の集合  $(x_1 : \tau_1, \dots, x_n : \tau_n)$ 。

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	$\Gamma \vdash x : \tau$ ( $x : \tau$ は $\Gamma$ に含まれる)	抽象	$\frac{\Gamma, x : \theta \vdash M : \tau}{\Gamma \vdash \lambda x : \theta. M : \theta \rightarrow \tau}$
定数	$\Gamma \vdash c_\tau : \tau$	適用	$\frac{\Gamma \vdash M : \theta \rightarrow \tau \quad \Gamma \vdash N : \theta}{\Gamma \vdash (M N) : \tau}$
直積	$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \theta}{\Gamma \vdash (M, N) : \tau \times \theta}$		

### 証明の例

$$\frac{\frac{x : \text{nat} \vdash \text{s}_{\text{nat} \rightarrow \text{nat}} : \text{nat} \rightarrow \text{nat} \quad x : \text{nat} \vdash x : \text{nat}}{x : \text{nat} \vdash (\text{s}_{\text{nat} \rightarrow \text{nat}} x) : \text{nat}}}{\vdash \lambda x : \text{nat}. (\text{s}_{\text{nat} \rightarrow \text{nat}} x) : \text{nat} \rightarrow \text{nat}} \quad \vdash 0_{\text{nat}} : \text{nat}}{\vdash ((\lambda x : \text{nat}. (\text{s}_{\text{nat} \rightarrow \text{nat}} x)) 0_{\text{nat}}) : \text{nat}}$$

問題 1 以下の判定式の証明を書け。

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

## 9.3 性質

次の定理は  $\text{fst}$ ,  $\text{snd}$  以外の  $\delta$  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  $M \rightarrow N$  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  $(M \rightarrow M_1 \rightarrow M_2 \rightarrow \dots)$  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  $\lambda$  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。

$$\frac{x : \tau \vdash x : \tau \rightarrow \theta \quad x : \tau \vdash x : \tau}{x : \tau \vdash (x x) : \theta}$$

$$\vdash \lambda x : \tau. (x x) : \tau \rightarrow \theta$$

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  $\tau = (\tau \rightarrow \theta)$  を満たす必要がある。当然ながら、上記の型の定義では、こんな方程式を満たすような  $\tau$  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  $\lambda$  計算では  $Y$  は定義できない。

## 9.4 万能性

以上の $\delta$ 規則と停止性で分かるように、単体では型付入計算は万能ではない。ただし、二種類の異なる問題がある。

整数などに関する $\delta$ 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。

$$\vdash \lambda f:\tau \rightarrow \tau. \lambda x:\tau. f^n x : (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau$$

問題は、この関数はある特定の型 $\tau$ 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 $\delta$ 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 $\delta$ 規則を入れる必要がある。

$$Y_{(\tau \rightarrow \tau) \rightarrow \tau} M \rightarrow M (Y_{(\tau \rightarrow \tau) \rightarrow \tau} M)$$

整数と $Y$ だけを加えれば、型付入計算は万能になる。実行には $\delta$ 規則を使うか、定数を型のない入計算の項に変えて型のない入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。**二次入計算** (ジラルルのシステム F) では型変数が用意されている。

$$\begin{aligned} \tau &::= \dots \mid \alpha \mid \forall \alpha. \tau \\ M &::= \dots \mid \Lambda \alpha. M \mid M[\tau] \end{aligned}$$

型推論規則も追加する。

$$\text{型抽象} \quad \frac{\Gamma \vdash M : \tau}{\Gamma \vdash \Lambda \alpha. M : \forall \alpha. \tau} \quad \alpha \notin FV(\Gamma) \qquad \text{型適用} \quad \frac{\Gamma \vdash M : \forall \alpha. \tau}{\Gamma \vdash M[\theta] : [\theta/\alpha]\tau}$$

型に関しても $\beta$ 簡約が定義される。

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

二次入計算では、チャーチ数は次のように表現できる。

$$\begin{aligned} \text{Nat} &= \forall \alpha. (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha) \\ c_n &= \Lambda \alpha. \lambda f:(\alpha \rightarrow \alpha). \lambda x:\alpha. f^n x \\ c_+ &= \lambda m:\text{Nat}. \lambda n:\text{Nat}. \Lambda \alpha. \lambda f:(\alpha \rightarrow \alpha). \lambda x:\alpha. (m[\alpha] f (n[\alpha] f x)) \\ c_\times &= \lambda m:\text{Nat}. \lambda n:\text{Nat}. \Lambda \alpha. \lambda f:(\alpha \rightarrow \alpha). (m[\alpha] (n[\alpha] f)) \\ c_{\text{pow}} &= \lambda m:\text{Nat}. \lambda n:\text{Nat}. \Lambda \alpha. (n[\alpha \rightarrow \alpha] m[\alpha]) \end{aligned}$$

そうすると、以下の型判定が証明できる。

$$\vdash c_n : \text{Nat} \qquad \vdash c_+, c_\times, c_{\text{pow}} : \text{Nat} \rightarrow \text{Nat} \rightarrow \text{Nat}$$

しかし、二次入計算は型付入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入計算を使えば、 $Y$ も定義できるようになるが、停止性が失われる。