

工学部専門科目「計算と論理」配布資料

多相ラムダ計算 (System F)

五十嵐 淳

京都大学 大学院情報学研究科 通信情報システム専攻

cal18@fos.kuis.kyoto-u.ac.jp

<http://www.fos.kuis.kyoto-u.ac.jp/~igarashi/class/cal/>

November 13, 2018

1 System F: 多相ラムダ計算

教科書 Poly.v で見たような多相関数 (*polymorphic function*) (型引数を取り, 引数によって型が変化する関数) によって単純型付ラムダ計算を拡張した計算体系が多相ラムダ計算 (*polymorphic λ -calculus*) である. 多相ラムダ計算は System F とも呼ばれている.¹

System F の (単純型付ラムダ計算に対しての) 本質的な拡張は項 (プログラム) への多相関数²

$$\text{fun } X : \text{Type} \Rightarrow M$$

と型適用 $M T$ の導入, そして多相型 $\forall X : \text{Type}, T$ の導入である.

(ふつうの) 関数の β 簡約と同様に型抽象が型に適用された項は簡約によって計算が進む.

$$(\text{fun } X : \text{Type} \Rightarrow M[X]) T \longrightarrow M[T]$$

関数本体中の (型の) 仮引数 X が実引数 T で置き換わるところは β 簡約と全く同じである. 例えば, 与えられた関数 f を x に二回適用する多相高階関数

$$M \stackrel{\text{def}}{=} \text{fun } X : \text{Type} \Rightarrow \text{fun } f : X \rightarrow X \Rightarrow \text{fun } x : X \Rightarrow f (f x)$$

について,

$$M \text{ nat} \longrightarrow \text{fun } f : \text{nat} \rightarrow \text{nat} \Rightarrow \text{fun } x : \text{nat} \Rightarrow f (f x)$$
$$M \text{ bool} \longrightarrow \text{fun } f : \text{bool} \rightarrow \text{bool} \Rightarrow \text{fun } x : \text{bool} \Rightarrow f (f x)$$

となり, (機能は同じだが) 自然数, 真偽値に特化した関数が得られる.

¹アメリカの計算機科学者ジョン・レイノルズ (John Reynolds) とフランスの論理学者ジャン・イヴ・ジラルド (Jean-Yves Girard) が 1970 年代初めに独立に考え出した. System F の名前はジラルドによる.

²型抽象とも呼ぶ

型付け関係は単純型付ラムダ計算と同様に

$$\Gamma \vdash M : T$$

という形で表されるが、型環境には変数の型宣言 $x : T$ だけでなく、型変数の宣言 $X : \text{Type}$ も並ぶことになる。また (ふつうの) 関数や関数適用のための規則は変わらない。

多相関数についての型付け規則は

$$\frac{\Gamma, X : \text{Type} \vdash M : T \quad X \notin \text{dom}(\Gamma)}{\Gamma \vdash \text{fun } X : \text{Type} \Rightarrow M : \forall X : \text{Type}, T} \quad (\text{T-TFUN})$$

と与えられる。関数全体に与えられる型の違いはあるが、T-FUN 規則とよく似ている。この X は後で多相関数が型引数に適用された時に具体的な型に置き換えられるものである。実際、型適用の型付け規則は

$$\frac{\Gamma \vdash M : \forall X : \text{Type}, T[X]}{\Gamma \vdash M S : T[S]} \quad (\text{T-TAPP})$$

のようになり、多相関数型 $\forall X : \text{Type}, T[X]$ を持つ多相関数 M 型引数 S を与えると、 X を S に置き換えたような型 (すなわち $T[S]$) が得られる。

例えば、上で定義した二回適用する関数の例は以下のように型付け関係を導出することができる。

$$\begin{aligned} M &\stackrel{\text{def}}{=} \text{fun } X : \text{Type} \Rightarrow \text{fun } f : X \rightarrow X \Rightarrow \text{fun } x : X \Rightarrow f (f x) \\ \Gamma &\stackrel{\text{def}}{=} X : \text{Type}, f : X \rightarrow X, x : X \rightarrow X \end{aligned}$$

$$\frac{\frac{\frac{\frac{\Gamma \vdash f : X \rightarrow X}{\Gamma \vdash f (f x) : X} \text{T-VAR} \quad \frac{\frac{\Gamma \vdash f : X \rightarrow X}{\Gamma \vdash f x : X} \text{T-VAR} \quad \frac{\Gamma \vdash x : X}{\Gamma \vdash f x : X} \text{T-VAR}}{\Gamma \vdash f x : X} \text{T-APP}}{\Gamma \vdash f (f x) : X} \text{T-APP}}{X : \text{Type}, f : X \rightarrow X \vdash \text{fun } x : X \Rightarrow f (f x) : X \rightarrow X} \text{T-FUN}}{X : \text{Type} \vdash \text{fun } f : X \rightarrow X \Rightarrow \text{fun } x : X \Rightarrow f (f x) : (X \rightarrow X) \rightarrow X \rightarrow X} \text{T-TFUN}}{\vdash M : \forall X : \text{Type}, (X \rightarrow X) \rightarrow X \rightarrow X} \text{T-TAPP}} \text{T-TAPP}$$

以下に (自然数も真偽値も再帰関数も含まない) 純粋な System F の定義をまとめる。

1.1 型, ラムダ項, 型環境, 判断の構文

(types)	$ \begin{array}{l} S, T ::= X \\ \quad S \rightarrow T \\ \quad \forall X : \text{Type}, T \end{array} $
(terms)	$ \begin{array}{l} M, N ::= x \\ \quad \text{fun } x : T \Rightarrow M \\ \quad M_1 M_2 \\ \quad \text{fun } X : \text{Type} \Rightarrow M \\ \quad M T \end{array} $
(type environments) Γ	$ \begin{array}{l} ::= \bullet \\ \quad \Gamma, x : T \\ \quad \Gamma, X : \text{Type} \end{array} $

1.2 簡約

$(\text{fun } x : T \Rightarrow M[x]) N \longrightarrow M[N]$	(R-BETA)
$(\text{fun } X : \text{Type} \Rightarrow M[X]) T \longrightarrow M[T]$	(R-TBETA)
$\frac{M \longrightarrow M'}{\text{fun } x : T \Rightarrow M \longrightarrow \text{fun } x : T \Rightarrow M'}$	(RC-FUN)
$\frac{M \longrightarrow M'}{M N \longrightarrow M' N}$	(RC-APP1)
$\frac{N \longrightarrow N'}{M N \longrightarrow M N'}$	(RC-APP2)
$\frac{M \longrightarrow M'}{\text{fun } X : \text{Type} \Rightarrow M \longrightarrow \text{fun } X : \text{Type} \Rightarrow M'}$	(RC-TFUN)
$\frac{M \longrightarrow M'}{M T \longrightarrow M' T}$	(RC-TAPP)

1.3 型付け規則

$\frac{(x : T \in \Gamma)}{\Gamma \vdash x : T}$	(T-VAR)
--	---------

$$\frac{\Gamma, x : S \vdash M : T \quad (x \notin \text{dom}(\Gamma))}{\Gamma \vdash \text{fun } x : S \Rightarrow M : S \rightarrow T} \quad (\text{T-FUN})$$

$$\frac{\Gamma \vdash M : S \rightarrow T \quad \Gamma \vdash N : S}{\Gamma \vdash M N : T} \quad (\text{T-APP})$$

$$\frac{\Gamma, X : \text{Type} \vdash M : T \quad (X \notin \text{dom}(\Gamma))}{\Gamma \vdash \text{fun } X : \text{Type} \Rightarrow M : \forall X : \text{Type}, T} \quad (\text{T-TFUN})$$

$$\frac{\Gamma \vdash M : \forall X : \text{Type}, S[X]}{\Gamma \vdash M T : S[T]} \quad (\text{T-TAPP})$$

代入と変数キャプチャについて再び 単純型付ラムダ計算において、代入時には変数のキャプチャを避けなければならない、ということを経験したが、System F でも同じ問題が発生する。System F では、さらに、項への型の代入、型への型の代入があり、 $\text{fun } X : \text{Type} \Rightarrow M$ や $\forall X : \text{Type}, T$ という、型パラメータを宣言する項・型があるため、型変数のキャプチャを避ける必要がある。方法は単純型付ラムダ計算と同じで、型変数の名前替えを行うことになる。例えば、

$$Y \rightarrow (\forall X : \text{Type}, X \rightarrow Y)$$

の Y に $X \rightarrow \text{nat}$ を代入した場合、得られる結果は、

$$(X \rightarrow \text{nat}) \rightarrow (\forall X : \text{Type}, X \rightarrow X \rightarrow \text{nat})$$

ではなく、 $\forall X : \text{Type}, \dots$ の部分の X の名前替えをして

$$(X \rightarrow \text{nat}) \rightarrow (\forall Z : \text{Type}, Z \rightarrow X \rightarrow \text{nat})$$

となる。

引数の名前替えをした項同士 (例えば $\text{fun } x : \text{nat} \Rightarrow x$ と $\text{fun } y : \text{nat} \Rightarrow y$) を同一視したのと同様、多相関数の型変数、さらには、多相関数型の型変数の名前替えも自由に行うのが通例である。例えば、

$$\text{fun } X : \text{Type} \Rightarrow \text{fun } x : X \Rightarrow x \quad \text{と} \quad \text{fun } Y : \text{Type} \Rightarrow \text{fun } x : Y \Rightarrow x$$

や

$$\forall X : \text{Type}, X \rightarrow X \quad \text{と} \quad \forall Y : \text{Type}, Y \rightarrow Y$$

は同一視する。

1.4 (純粋な) System F の重要な性質

定理 1 (合流性) 任意の項 M_1, M_2, M_3 に対して、 $M_1 \rightarrow^* M_2$ かつ $M_1 \rightarrow^* M_3$ ならば、ある M_4 が存在し、 $M_2 \rightarrow^* M_4$ かつ $M_3 \rightarrow^* M_4$ が成り立つ。

定理 2 (型保存定理) $\Gamma \vdash M : T$ かつ、 $M \rightarrow M'$ ならば、 $\Gamma \vdash M' : T$ である。

定理 3 (強正規化性) $\Gamma \vdash M : T$ とする。 $M \rightarrow M_1 \rightarrow M_2 \rightarrow \dots$ なる項の無限列 M_1, M_2, \dots は存在しない。

2 System F + 自然数, 真偽値, リスト, 再帰関数

(types) $S, T ::= \dots$

- | nat
- | bool
- | list T

(terms) $M, N ::= \dots$

- | 0
- | S
- | match M with 0 => N_1 | S x => N_2 end
- | true
- | false
- | if M then N_1 else N_2
- | nil
- | cons
- | match M with nil => N_1 | cons $x y$ => N_2
- | fix $x(y : S) : T := M$

$\Gamma ::= \dots$

2.1 簡約

match 0 with 0 => N_1 | S x => N_2 end $\longrightarrow N_1$ (R-MATCHZ)

match S M with 0 => N_1 | S x => $N_2[x]$ end $\longrightarrow N_2[M]$ (R-MATCHS)

if true then N_1 else N_2 $\longrightarrow N_1$ (R-IFT)

if false then N_1 else N_2 $\longrightarrow N_2$ (R-IFF)

match nil T with nil => N_1 | cons $x y$ => $N_2[x, y]$ $\longrightarrow N_1$ (R-MATCHN)

match cons $T M_1 M_2$ with nil => N_1 | cons $x y$ => $N_2[x, y]$ $\longrightarrow N_2[M_1, M_2]$
(R-MATCHC)

(fix $x(y : S) : T := M[x, y]$) $N \longrightarrow M[\text{fix } x(y : S) : T := M[x, y], N]$ (R-FIX)

$$\frac{M \longrightarrow M'}{\text{match } M \text{ with } 0 \Rightarrow N_1 \mid S x \Rightarrow N_2 \text{ end} \longrightarrow \text{match } M' \text{ with } 0 \Rightarrow N_1 \mid S x \Rightarrow N_2 \text{ end}}$$
 (RC-MATCH1)

$$\frac{N_1 \longrightarrow N'_1}{\text{match } M \text{ with } 0 \Rightarrow N_1 \mid S x \Rightarrow N_2 \text{ end} \longrightarrow \text{match } M \text{ with } 0 \Rightarrow N'_1 \mid S x \Rightarrow N_2 \text{ end}}$$
 (RC-MATCH2)

$$\frac{N_2 \longrightarrow N'_2}{\text{match } M \text{ with } 0 \Rightarrow N_1 \mid S \ x \Rightarrow N_2 \text{ end} \longrightarrow \text{match } M \text{ with } 0 \Rightarrow N_1 \mid S \ x \Rightarrow N'_2 \text{ end}} \quad (\text{RC-MATCH3})$$

$$\frac{M \longrightarrow M'}{\text{if } M \text{ then } N_1 \text{ else } N_2 \longrightarrow \text{if } M' \text{ then } N_1 \text{ else } N_2} \quad (\text{RC-IF1})$$

$$\frac{N_1 \longrightarrow N'_1}{\text{if } M \text{ then } N_1 \text{ else } N_2 \longrightarrow \text{if } M \text{ then } N'_1 \text{ else } N_2} \quad (\text{RC-IF2})$$

$$\frac{N_2 \longrightarrow N'_2}{\text{if } M \text{ then } N_1 \text{ else } N_2 \longrightarrow \text{if } M \text{ then } N_1 \text{ else } N'_2} \quad (\text{RC-IF3})$$

$$\frac{M \longrightarrow M'}{\text{fun } x : T \Rightarrow M \longrightarrow \text{fun } x : T \Rightarrow M'} \quad (\text{RC-FUN})$$

$$\frac{M \longrightarrow M'}{\text{fix } x(y : S) : T := M \longrightarrow \text{fix } x(y : S) : T := M'} \quad (\text{RC-FIX})$$

2.2 型付け規則

$$\frac{}{\Gamma \vdash 0 : \text{nat}} \quad (\text{T-ZERO})$$

$$\frac{}{\Gamma \vdash S : \text{nat} \rightarrow \text{nat}} \quad (\text{T-SUCC})$$

$$\frac{\Gamma \vdash M : \text{nat} \quad \Gamma \vdash N_1 : T \quad \Gamma, x : \text{nat} \vdash N_2 : T}{\Gamma \vdash \text{match } M \text{ with } 0 \Rightarrow N_1 \mid S \ x \Rightarrow N_2 \text{ end} : T} \quad (\text{T-MATCH})$$

$$\frac{}{\Gamma \vdash \text{true} : \text{bool}} \quad (\text{T-TRUE})$$

$$\frac{}{\Gamma \vdash \text{false} : \text{bool}} \quad (\text{T-FALSE})$$

$$\frac{\Gamma \vdash M : \text{bool} \quad \Gamma \vdash N_1 : T \quad \Gamma \vdash N_2 : T}{\Gamma \vdash \text{if } M \text{ then } N_1 \text{ else } N_2 : T} \quad (\text{T-IF})$$

$$\frac{\Gamma, x : S \rightarrow T, y : S \vdash M : T}{\Gamma \vdash \text{fix } x(y : S) : T := M : S \rightarrow T} \quad (\text{T-FIX})$$

$$\frac{}{\Gamma \vdash \text{nil} : \forall X : \text{Type}, \text{list } X} \quad (\text{T-NIL})$$

$$\frac{}{\Gamma \vdash \text{cons} : \forall X : \text{Type}, X \rightarrow \text{list } X \rightarrow \text{list } X} \quad (\text{T-CONS})$$

$$\frac{\Gamma \vdash M : \text{list } S \quad \Gamma \vdash N_1 : T \quad \Gamma, x : S, y : \text{list } S \vdash N_2 : T}{\Gamma \vdash \text{match } M \text{ with nil} \Rightarrow N_1 \mid \text{cons } x \ y \Rightarrow N_2 : T} \quad (\text{T-MATCHL})$$

項 M が標準形 (*canonical form*) である, とは, M が $0, S N, \text{true}, \text{false}, \text{fun } x : T \Rightarrow N, \text{fix } x(y : S) : T := N$, もしくは $\text{fun } X : \text{Type} \Rightarrow M$ のいずれかの形をしていることをいう. 定理 4 (前進性) $\vdash M : T$ とする. 任意の M の部分項 M_0 について, 以下が成立する.

- M_0 が $M_1 M_2$ の形 (ただし M_1 は標準形) ならば, M_1 は $\text{fun } x : T' \Rightarrow M'$ もしくは $\text{fix } x(y : S') : T' := M'$ の形である.
- M_0 が $\text{match } M_1 \text{ with } 0 \Rightarrow M_2 \mid S \ x \Rightarrow M_3 \text{ end}$ の形 (ただし M_1 は標準形) ならば, M_1 は 0 または, $S \ M'$ の形である.
- M_0 が $\text{match } M_1 \text{ with nil} \Rightarrow M_2 \mid \text{cons } x \ y \Rightarrow M_3 \text{ end}$ の形 (ただし M_1 は標準形) ならば, M_1 は nil または, $\text{cons } M' \ M''$ の形である.
- M_0 が $\text{if } M_1 \text{ then } M_2 \text{ else } M_3$ の形 (ただし M_1 は標準形) ならば, M_1 は true もしくは false である.
- M_0 が $M_1 S$ の形 (ただし M_1 は標準形) ならば, M_1 は $\text{fun } X : \text{Type} \Rightarrow M'$ の形である.