型付き対称λ計算における論理和と論理積の導入

上田やよい　(指導教官:浅井健一)

1　はじめに

継続を扱う基礎言語体系の一つに、Filinski の提案した対称λ計算 (Symmetric λ-calculus, SLC) [1] がある。SLC では、項と継続が完全に対称な形をしており、項を扱うのと同様に継続を扱うことができる。

さらに昨年度の阪上紗里の修士論文 [2] にて、SLC の定式化等がなされ、その基礎理論が確立されてきた。

本研究では、この体系に組の概念を導入し、その型が論理和と論理積に対応していることを示す。

2　構文

型付き対称λ計算に組を導入した構文は以下の通り。

$$
\begin{array}{rrcl}
(\text{値}) & v & ::= & x \mid n \mid \lceil f \rceil \mid (e,\, e) \\
(\text{項}) & e & ::= & v \mid f \uparrow e \\
(\text{項関数}) & f_e & ::= & e \Leftarrow x \mid e \Leftarrow p_v \mid f_x \\
(\text{関数}) & f & ::= & f_e \mid f_c \mid \overline{e} \mid \underline{c} \\
(\text{継続関数}) & f_c & ::= & y \Rightarrow c \mid p_k \Rightarrow c \mid f_y \\
(\text{継続}) & c & ::= & k \mid c \downarrow f \\
(\text{値継続}) & k & ::= & y \mid \bullet \mid \lfloor f \rfloor \mid (c,\, c)
\end{array}
$$

$$
\begin{array}{rrcl}
(\text{項のパターン}) & p_v & ::= & (x_1,\, x_2) \mid \lceil f_x \rceil \\
(\text{継続のパターン}) & p_k & ::= & (y_1,\, y_2) \mid \lfloor f_y \rfloor
\end{array}
$$

$(e,\, e)$ が項の組、$(c,\, c)$ が継続の組である。

さらに、関数が受け取る値の型を示すため、項のパターン p_v と、継続のパターン p_k を導入した。

3　SLC 式の型

SLC 式の型は、論理式を用いて以下のように表す。

- 項 e の型 : $+X$ (X は任意)
- 継続 c の型 : $\neg Y$ (Y は任意)
- 関数 f の型 : $\left\{\begin{smallmatrix} +A \to +B \\ \neg B \to \neg A \end{smallmatrix}\right.$ (二つの型を同時に持つ)

$+X$ 型は X 型を返す項を表し、$\neg Y$ は Y 型を受け取る継続を表す。

4　項の組と継続の組の導入

組の型規則を以下に示す。

$$
(\text{項の組})\quad \frac{\Gamma \vdash e_1 : +A \quad \Gamma \vdash e_2 : +B}{\Gamma \vdash (e_1,\, e_2) : +(A \wedge B)}\ (\text{TAnd})
$$

$$
(\text{継続の組})\quad \frac{\Gamma \vdash c_1 : \neg A \quad \Gamma \vdash c_2 : \neg B}{\Gamma \vdash (c_1,\, c_2) : \neg(A \vee B)}\ (\overline{\text{TOr}})
$$

継続の組の型は、既存の継続の型と適合させるため、$\neg A \wedge \neg B = \neg(A \vee B)$ (ド・モルガン) と定義した。

継続の組は if 文の then 部分と else 部分に相当する。後述の inl 関数等を条件式に用いると、SLC で条件分岐を扱うことができる。

5　関数への論理和型、論理積型の導入

継続の組を引数に取る関数は、本研究で新たに導入したパターン $(y_1,\, y_2)$ を用いて、$(y_1,\, y_2) \Rightarrow c$ のように表す。$c\colon \neg T$ のとき、この関数の型は $\left\{\begin{smallmatrix} +T \to +(A \vee B) \\ \neg(A \vee B) \to \neg T \end{smallmatrix}\right.$ と表される。この関数に項 $e : +T$ を適用させると、

$$
\frac{\Gamma \vdash (y_1,\, y_2) \Rightarrow c : \left\{\begin{smallmatrix} +T \to +(A \vee B) \\ \neg(A \vee B) \to \neg T \end{smallmatrix}\right. \quad \Gamma \vdash e : +T}{\Gamma \vdash ((y_1,\, y_2) \Rightarrow c) \uparrow e : +(A \vee B)}\ (\overline{\text{TApp}})
$$

となり、$(A \vee B)$ 型の項を定義できる。特に $c = y_1$ であれば、この関数は inl と同じ働きをする。($c = y_2$ なら inr と同じ働きをする)

項関数についても同じことが言え、$\neg(A \wedge B)$ 型の継続は fst 関数等に適当な継続を適用すると得られる。

6　計算状態

SLC の計算状態は、$\langle\, c \mid e \,\rangle$ の二つ組か、$\langle\, c \mid f \mid e \,\rangle$ の三つ組で表す。対応する型規則は以下の通り。

$$
\frac{\vdash c : \neg T \quad \vdash e : +T}{\vdash \langle\, c \mid e \,\rangle}\ (\text{TProg1})
$$

$$
\frac{\vdash c : \neg B \quad \vdash f : \left\{\begin{smallmatrix} +A \to +B \\ \neg B \to \neg A \end{smallmatrix}\right. \quad \vdash e : +A}{\vdash \langle\, c \mid f \mid e \,\rangle}\ (\text{TProg2})
$$

7　簡約規則

SLC の非決定的簡約規則を図 1 に示す。$(*)$ は本研究で新たに加えた規則である。(β_p), $(\overline{\beta_p})$ で用いられている、パターン付き関数のβ簡約を以下に示す。

$$
\begin{array}{rcl}
\beta_v[\![(x_1,\, x_2),\, e,\, (e_1,\, e_2)]\!] & = & e[e_1/x_1,\, e_2/x_2] \\
\beta_v[\![\lceil f_x \rceil,\, e,\, \lceil f \rceil]\!] & = & e[f/f_x]
\end{array}
$$

$$
\begin{array}{rcl}
\beta_k[\![(y_1,\, y_2),\, c,\, (c_1,\, c_2)]\!] & = & c[c_1/y_1,\, c_2/y_2] \\
\beta_k[\![\lfloor f_y \rfloor,\, c,\, \lfloor f \rfloor]\!] & = & c[f/f_y]
\end{array}
$$

さらに、実際に SLC で if 文を記述し、簡約する例を併せて掲載した。inl 関数で継続の組から片方を選び取る様子に注目してほしい。慣用表現は以下の通り。

$$
\begin{array}{rcl}
inl & = & (y_1,\, y_2) \Rightarrow y_1 \\
[f_1,\, f_2] & = & y \Rightarrow (y \downarrow f_1,\, y \downarrow f_2)
\end{array}
$$

図 1: SLC の非決定的簡約規則 及び簡約例

$$
\begin{array}{rrcll}
(\mathit{begin}) & e & \rightsquigarrow & \langle \bullet \mid e \rangle & \\
(\mathit{left}) & \langle c \mid (e_1,\, e_2) \rangle & \rightsquigarrow & \langle c \mid ((x_1,\, e_2) \Leftarrow x_1) \uparrow e_1 \rangle & \cdots (*) \\
(\mathit{right}) & \langle c \mid (e_1,\, e_2) \rangle & \rightsquigarrow & \langle c \mid ((e_1,\, x_2) \Leftarrow x_2) \uparrow e_2 \rangle & \cdots (*) \\
(\overline{\mathit{pop}}) & \langle c \mid f \uparrow e \rangle & \rightsquigarrow & \langle c \mid f \mid e \rangle & \\
(\mathit{push}) & \langle c \mid f \mid e \rangle & \rightsquigarrow & \langle c \downarrow f \mid e \rangle & \\
(\mathit{exchange}) & \langle c \mid \overline{e'} \mid e \rangle & \rightsquigarrow & \langle c \mid f_x \uparrow e \Leftarrow \lceil f_x \rceil \mid e' \rangle & \\
(\beta) & \langle c \mid e' \Leftarrow x \mid e \rangle & \rightsquigarrow & \langle c \mid e'[e/x] \rangle & \\
(\beta_p) & \langle c \mid e \Leftarrow p_v \mid v \rangle & \rightsquigarrow & \langle c \mid \beta_v[\![p_v,\, e,\, v]\!] \rangle & \cdots (*) \\
(\overline{\beta_p}) & \langle k \mid p_k \Rightarrow c \mid e \rangle & \rightsquigarrow & \langle \beta_k[\![p_k,\, c,\, k]\!] \mid e \rangle & \cdots (*) \\
(\overline{\beta}) & \langle c \mid y \Rightarrow c' \mid e \rangle & \rightsquigarrow & \langle c'[c/y] \mid e \rangle & \\
(\overline{\mathit{exchange}}) & \langle c \mid \underline{c'} \mid e \rangle & \rightsquigarrow & \langle c' \mid \lfloor f_y \rfloor \Rightarrow c \downarrow f_y \mid e \rangle & \\
(\overline{\mathit{push}}) & \langle c \mid f \mid e \rangle & \rightsquigarrow & \langle c \mid f \uparrow e \rangle & \\
(\mathit{pop}) & \langle c \downarrow f \mid e \rangle & \rightsquigarrow & \langle c \mid f \mid e \rangle & \\
(\overline{\mathit{right}}) & \langle (c_1,\, c_2) \mid e \rangle & \rightsquigarrow & \langle c_2 \downarrow (y_2 \Rightarrow (c_1,\, y_2)) \mid e \rangle & \cdots (*) \\
(\overline{\mathit{left}}) & \langle (c_1,\, c_2) \mid e \rangle & \rightsquigarrow & \langle c_1 \downarrow (y_1 \Rightarrow (y_1,\, c_2)) \mid e \rangle & \cdots (*) \\
(\overline{\mathit{end}}) & \langle \bullet \mid v \rangle & \rightsquigarrow & v &
\end{array}
$$

例. $1 + (\mathit{if\ true\ then}\ 2\ \mathit{else}\ 3)$ を SLC で表現した場合

$$
\begin{array}{lll}
 & (x+1 \Leftarrow x) \uparrow [2 \Leftarrow x_1,\, 3 \Leftarrow x_2] \uparrow \mathit{inl} \uparrow n & \\
\rightsquigarrow & \langle \bullet \mid (x+1 \Leftarrow x) \uparrow [2 \Leftarrow x_1,\, 3 \Leftarrow x_2] \uparrow \mathit{inl} \uparrow n \rangle & (\mathit{begin}) \\
\rightsquigarrow & \langle \bullet \mid x+1 \Leftarrow x \mid [2 \Leftarrow x_1,\, 3 \Leftarrow x_2] \uparrow \mathit{inl} \uparrow n \rangle & (\overline{\mathit{pop}}) \\
\rightsquigarrow & \langle \bullet \downarrow (x+1 \Leftarrow x) \mid [2 \Leftarrow x_1,\, 3 \Leftarrow x_2] \uparrow \mathit{inl} \uparrow n \rangle & (\mathit{push}) \\
\rightsquigarrow & \langle \bullet \downarrow (x+1 \Leftarrow x) \mid [2 \Leftarrow x_1,\, 3 \Leftarrow x_2] \mid \mathit{inl} \uparrow n \rangle & (\overline{\mathit{pop}}) \\
\rightsquigarrow & \langle (\bullet \downarrow (x+1 \Leftarrow x) \downarrow (2 \Leftarrow x_1),\, \bullet \downarrow (x+1 \Leftarrow x) \downarrow (3 \Leftarrow x_2)) \mid \mathit{inl} \uparrow n \rangle & (\overline{\beta}) \\
\rightsquigarrow & \langle (\bullet \downarrow (x+1 \Leftarrow x) \downarrow (2 \Leftarrow x_1),\, \bullet \downarrow (x+1 \Leftarrow x) \downarrow (3 \Leftarrow x_2)) \mid \mathit{inl} \mid n \rangle & (\overline{\mathit{pop}}) \\
\rightsquigarrow & \langle \bullet \downarrow (x+1 \Leftarrow x) \downarrow (2 \Leftarrow x_1) \mid n \rangle & (\overline{\beta_p}) \\
\rightsquigarrow & \langle \bullet \downarrow (x+1 \Leftarrow x) \mid 2 \Leftarrow x_1 \mid n \rangle & (\mathit{pop}) \\
\rightsquigarrow & \langle \bullet \downarrow (x+1 \Leftarrow x) \mid 2 \rangle & (\beta) \\
\rightsquigarrow & \langle \bullet \mid x+1 \Leftarrow x \mid 2 \rangle & (\mathit{pop}) \\
\rightsquigarrow & \langle \bullet \mid 3 \rangle & (\beta) \\
\rightsquigarrow & 3 & (\overline{\mathit{end}})
\end{array}
$$

8 関連定理

非決定的 SLC について、以下が証明されている。

Theorem 8.1 (Progress) 型のついている SLC 式は、図 1 の規則で必ず簡約できる。

Theorem 8.2 (Preservation) 型のついた SLC 式を図 1 の規則で簡約した SLC 式には、必ず型がつく。

Lemma 8.3 (代入補題) 型のついた SLC 式は、変数に代入する前の型と代入した後の型は変わらない。

9 Call-By-Value

SLC の Call-By-Value 版を定義することができる。また、次の定理が証明されている。

Theorem 9.1 (簡約の一意性) 型のついた Call-By-Value SLC 式の簡約の仕方は一意的である。

10 まとめと今後の課題

既存の SLC に対して、非常に自然に組やその計算規則を導入できたと思われる。今後、Call-By-Value SLC の停止性の証明等をしていく予定である。

参考文献

[1] Filinski, A ”Declarative Continuations and Categorical Duality”, In Master’s thesis, DIKU Report 89/11, University of Copenhagen (August 1989).

[2] 阪上紗里「型付き対称λ計算の基礎理論」お茶の水女子大学大学院 博士前期課程 人間文化研究科 修士論文 (January 2008).