明示的環境計算体系への部分型の導入

澤田 康秀　(Yasuhide Sawada)
京都大学情報学研究科　(Graduate School of Informatics, Kyoto University)

1　はじめに

型付き λ 計算は、関数型プログラミング言語の基礎理論として位置づけられている体系である。この型付き λ 計算の表現力を拡充していくために様々な研究が行われているが、その中の一つに明示的環境の導入がある。それを実現した体系として挙げられるのが $\lambda\varepsilon$ [1] である。$\lambda\varepsilon$ は、合流性や強正規化性といった良い性質を保持するという点で優れた計算体系である。

環境という概念は、変数の値を保持する箱のようなものを表すが、一般的なプログラミング言語においては暗黙のうちに扱われてしまうことが多い。明示的環境とは、これを syntax の要素として明示的に扱えるようにしたものである。

$\lambda\varepsilon$ における明示的環境は、明示的代入とレコードをさらに一般化した概念でもある。明示的代入とは、単純型付き λ 計算などではメタレベルで行われる代入という操作を、明示的に扱えるようにするものである。一方レコードとは、型の異なる複数の項をまとめて扱うための概念であり、関数の引数や戻り値にもなり得る first class object である。$\lambda\varepsilon$ では、明示的環境を、これらの特徴を全て併せ持つ概念として用いることができる。

レコードの概念を型付き λ 計算に導入した体系はレコード計算と呼ばれており、それはオブジェクト指向言語のメカニズムを λ 計算上で表現することを目指したものでもある。このレコード計算において、いわゆるメソッドの継承に相当するしくみを表現するために部分型という概念が用いられるようになった [2, 3]。部分型を導入することで、より柔軟性の高い型システムを構築することが出来る。

明示的環境がレコードを一般化した概念であるならば、明示的環境計算に部分型を導入することは、レコード計算の場合と同様に有意義であると考えられた。そこで本研究では、$\lambda\varepsilon$ への部分型の導入を行った。

2　明示的環境

計算体系 $\lambda\varepsilon$ における明示的環境の扱いについて述べることにする。明示的環境は、変数の値を保持するためのものであり、例えば

$$\{2/x^{int}, 3/y^{int}\}$$

という形 [1] で記述する。この場合は、x^{int} の値を 2、y^{int} の値を 3 とするような環境を表している。

$\lambda\varepsilon$ は型付きの計算体系であるから、明示的環境にも型が付けられることになる。たとえば、$\{2/x^{int}, 3/y^{int}\}$ の型は $\{x^{int}, y^{int}\}$ である。環境の型は、環境が束縛する変数の集合として定義されることになっている。

また、$e[\![a]\!]$ と書いて、環境 e のもとで項 a を評価することを表す。すなわち、

$$\{2/x^{int}, 3/y^{int}\}[\![x^{int} + y^{int}]\!]$$

という項を書くことができ、これは $\{2/x^{int}, 3/y^{int}\}$ という環境のもとで x と y の値を足すことを表す。この場合は当然、$2 + 3$ という計算が行われることになる。

3　部分型

部分型とは、部分型関係 (型の包含関係にもとづいて定義される関係) を用いて型付けを行う仕組みのことである。$\lambda\varepsilon$ に対してこの部分型を導入することを考える。

3.1　部分型関係

$\lambda\varepsilon$ における型の定義は次のようになっている。

$$\begin{array}{rcll} A, B & ::= & K & \text{(基底型)} \\ & | & A \Rightarrow B & \text{(関数型)} \\ & | & E & \text{(環境型)} \end{array}$$

$$E \quad ::= \quad \{x_1^{A_1}, \ldots, x_n^{A_n}\} \qquad (n \geq 0)$$

この型に対する部分型関係 $<:$ は、以下の規則によって定めることが出来ると考えられる。

[1] $\lambda\varepsilon$ では、全ての変数は型を付けて記述される。

(**S ref**)
$$\frac{}{A <: A}$$

(**S trans**)
$$\frac{A <: B \quad B <: C}{A <: C}$$

(**S arrow**)
$$\frac{B <: A \quad A' <: B'}{A \Rightarrow A' <: B \Rightarrow B'}$$

(**S env**)
$$\frac{A_i <: B_i \ (i = 1, \ldots, m) \quad n \geq m}{\{x_1^{A_1}, \ldots, x_n^{A_n}\} <: \{x_1^{B_1}, \ldots, x_m^{B_m}\}}$$

規則 (**S-env**) は、環境型の大小関係を規定したものである。これは、レコード計算体系におけるレコード型の大小関係の規定と同様の形式をとっている。

3.2 Coercion

部分型を用いるためには、部分型関係を取り入れた型付け規則が必要になる。レコード計算体系などで最も一般的に用いられるのは、次の規則 (subsumption rule) である。

$$\frac{\Gamma \vdash a : A \qquad A <: B}{\Gamma \vdash a : B}$$

この規則は、文脈 Γ のもとで項 a が型 A を持ち、かつ $A <: B$ という部分型関係が成り立つとき、項 a の型を B に置き換えることを許すというものである。

しかし、この規則を $\lambda\varepsilon$ に導入すると不具合が生じてしまう。subsumption rule は、項を変化させずに型だけを置き換えることを許す規則であるから、項に対する型の唯一性が崩れてしまうという問題がある。そして $\lambda\varepsilon$ では、型をチェックしながら項の計算を行っていくという特徴があり、項に対して型が正しく決められなければ、それが誤った計算につながる恐れも生じてくるのである。

そこで、subsumption rule の導入は行わず、代わりに次の規則を追加することにする。

$$\frac{\Gamma \vdash a : A \qquad A <: B}{\Gamma \vdash a|_B \ : B}$$

つまり、部分型関係を用いて型を置き換える場合、その型の情報を項にも付加しておくことにするということである。$a|_B$ のように、型情報を明記しておいて、計算の際にその型に応じた項の変形を行う仕組みのことを coercion

[4, 5] という。この coercion を導入することによって、項の持つ型の情報を計算時に正しく反映させることが可能となる。

4 体系 $\lambda\varepsilon c$

本研究では、明示的環境計算の体系 $\lambda\varepsilon$ を拡張し、部分型を導入した $\lambda\varepsilon c$ という体系の構築を行った。$\lambda\varepsilon c$ は、coercion をサポートする仕組みを備えた体系であり、部分型を用いる際には coercion の適用を要求するという形をとっている。

4.1 定義

Definition 1 型

$$\begin{aligned} A, B &::= K \mid A \Rightarrow B \mid E \\ E &::= \{x_1^{A_1}, \ldots, x_n^{A_n}\} \end{aligned}$$

K は基底型を、E は環境の型を表す。E の定義において、$n \geq 0$ であり、$x_i^{A_i}$ は互いに相異なっていなければならない。

Definition 2 項 [2]

$$\begin{aligned} a, b, e ::= \; & x^A \\ \mid \; & \lambda x^A.\, b \\ \mid \; & ba \\ \mid \; & \{a_1/x_1^{A_1}, \ldots, a_n/x_n^{A_n}\} \\ \mid \; & e[\![a]\!] \\ \mid \; & a|_A \end{aligned}$$

$a|_A$ は coercion の項であり、a の型が A であることを明示するためのものである。明示的環境 $\{a_1/x_1^{A_1}, \ldots, a_n/x_n^{A_n}\}$ において、$x_1^{A_1}, \ldots, x_n^{A_n}$ $(n \geq 0)$ は互いに異なっていなければならない。

Definition 3 文脈、型判定

文脈は、宣言 (x^A という形の式) の集合である。また、文脈 Γ のもとで項 a が型 A を持つとき、$\Gamma \vdash a : A$ と書き、これを型判定という。

Definition 4 部分型関係 $<:$

$\lambda\varepsilon c$ における部分型関係 $<:$ は、以下の規則によって定義される、型の二項関係である。

[2] $\lambda\varepsilon c$ の項であることを強調するときには、$\lambda\varepsilon c$-term と呼ぶ。

(**S ref**)
$$\frac{}{A <: A}$$

(**S trans**)
$$\frac{A <: B \quad B <: C}{A <: C}$$

(**S arrow**)
$$\frac{B <: A \quad A' <: B'}{A \Rightarrow A' <: B \Rightarrow B'}$$

(**S env**)
$$\frac{A_i <: B_i \ (i = 1, \ldots, m) \quad n \geq m}{\{x_1^{A_1}, \ldots, x_n^{A_n}\} <: \{x_1^{B_1}, \ldots, x_m^{B_m}\}}$$

部分型関係 $A <: B$ が成り立っているとき、以下のいずれか 1 つが必ず成り立つことが明らかである。

- A, B はともに基底型で、かつ $A \equiv B$ 。
- A, B はともに関数型。
- A, B はともに環境型。

Definition 5　型付け規則

型判定の導出に用いられる型付け規則は、次のように定義される。

(**assume**)
$$\frac{}{x^A \vdash x^A : A}$$

$(\Rightarrow I)$
$$\frac{\Gamma \vdash b : B}{\Gamma - \{x^A\} \vdash \lambda x^A.\ b : A \Rightarrow B}$$

$(\Rightarrow E)$
$$\frac{\Gamma \vdash b : A \Rightarrow B \quad \Delta \vdash a : A}{\Gamma, \Delta \vdash ba : B}$$

(**env**I)
$$\frac{\Gamma_1 \vdash a_1 : A_1 \quad \cdots \quad \Gamma_n \vdash a_n : A_n}{\Gamma_1, \ldots, \Gamma_n \vdash \{a_1/x_1^{A_1}, \ldots, a_n/x_n^{A_n}\} : \{x_1^{A_1}, \ldots, x_n^{A_n}\}}$$

(**env**E)
$$\frac{\Gamma \vdash e : E \quad \Delta \vdash a : A}{\Gamma, (\Delta - E) \vdash e[\![a]\!] : A}$$

(**coercion**)
$$\frac{\Gamma \vdash a : A \quad A <: B}{\Gamma \vdash a|_B : B}$$

任意の項 a が型を持つならば、それは 1 つに決まる。その型のことを $\mathrm{TY}(a)$ と書くことにする。

Definition 6　自由変数、束縛変数

項 a 中の自由変数の集合を $\mathrm{FV}(a)$ で表す。$\mathrm{FV}(a)$ の定義は以下の通りである。項 a 中に現れる変数のうち、$\mathrm{FV}(a)$ に含まれないものは束縛変数である。

- $\mathrm{FV}(x^A) \ ::= \ \{x^A\}$
- $\mathrm{FV}(\lambda x^A.\, b) \ ::= \ \mathrm{FV}(b) - \{x^A\}$
- $\mathrm{FV}(ba) \ ::= \ \mathrm{FV}(b) \cup \mathrm{FV}(a)$
- $\mathrm{FV}(\{a_1/x_1^{A_1}, \ldots, a_n/x_n^{A_n}\}) \ ::= \ \mathrm{FV}(a_1) \ \cup \ldots \cup \ \mathrm{FV}(a_n)$
- $\mathrm{FV}(e[\![a]\!]) \ ::= \ \mathrm{FV}(e) \ \cup \ (\mathrm{FV}(a) - \ \mathrm{TY}(e))$
- $\mathrm{FV}(a\,|_A) \ ::= \ \mathrm{FV}(a)$

Definition 7　α 同値

- 項 a に含まれる自由変数 x^A をすべて y^A (a 中に現れない変数) で置き換えたものを b とおくと、$\lambda x^A.\, a$ と $\lambda y^A.\, b$ は α 同値である。
- a の部分項 b を、b と α 同値な b' で置き換えた項を a' とおくと、a と a' は α 同値である。

α 同値な項同士は、同一の項として扱われる。

Definition 8　簡約規則

$\lambda\varepsilon c$ における簡約規則の定義は以下のようになる。

$$(\lambda)\quad (\lambda x^A.b)a \ \mapsto_\lambda \ \{a/x^A\}[\![b]\!]$$

$$\begin{aligned}
(\mathbf{gc})\quad & e[\![a]\!] \ \mapsto_\varepsilon \ a \quad \text{if } \mathrm{TY}(e) \cap \mathrm{FV}(a) = \emptyset \\
(\mathbf{var})\quad & \{a_1/x_1^{A_1}, \ldots, a_n/x_n^{A_n}\}[\![x_i^{A_i}]\!] \ \mapsto_\varepsilon \ a_i \quad (1 \le i \le n) \\
(\mathbf{abs})\quad & e[\![\lambda x^A.b]\!] \ \mapsto_\varepsilon \ \lambda x^A.e[\![b]\!] \quad \text{if } x^A \notin \mathrm{TY}(e) \cup \mathrm{FV}(e) \\
(\mathbf{app})\quad & e[\![ba]\!] \ \mapsto_\varepsilon \ e[\![b]\!]e[\![a]\!] \\
(\mathbf{env})\quad & e[\![\{a_1/x_1^{A_1}, \ldots, a_n/x_n^{A_n}\}]\!] \ \mapsto_\varepsilon \ \{e[\![a_1]\!]/x_1^{A_1}, \ldots, e[\![a_n]\!]/x_n^{A_n}\} \\
(\mathbf{eval})\quad & e[\![f[\![x^A]\!]]\!] \ \mapsto_\varepsilon \ e[\![f]\!][\![x^A]\!] \quad \text{if } x^A \in \mathrm{TY}(f) \\
(\mathbf{coe})\quad & e[\![a\,|_A]\!] \ \mapsto_\varepsilon \ e[\![a]\!]\,|_A
\end{aligned}$$

$$\begin{aligned}
(\mathbf{e\ atom})\quad & a\,|_K \ \mapsto_c \ a \\
(\mathbf{e\ app})\quad & (b\,|_{A\Rightarrow B})\,a \ \mapsto_c \ (b\,(a\,|_C))\,|_B \quad (\mathrm{TY}(b) \equiv C \Rightarrow D) \\
(\mathbf{e\ env})\quad & \{a_1/x_1^{A_1}, \ldots, a_n/x_n^{A_n}\}\,|_{\{x_1^{B_1}, \ldots, x_m^{B_m}\}} \\
& \qquad \mapsto_c \ \{a_1\,|_{B_1}/x^{B_1}, \ldots, a_m\,|_{B_m}/x^{B_m}\} \qquad (n \ge m)
\end{aligned}$$

規則 (**abs**) における条件 $x^A \notin \mathrm{TY}(e) \cup \mathrm{FV}(e)$ は、項の α 同値性を利用して変数 x^A を適切に選ぶことで、常に成り立つようにすることが出来る。規則 (**c-atom**) は、基底型による coercion がそのまま外してしまえることを意味する。$a\,|_K$ という項において、a の持つ型は必ず K に一致するため、これは明らかである。

規則 (**c-app**) は関数に対する coercion を移動させるための規則である。たとえば、

$$(\lambda x^A.b)\,|_{C \Rightarrow D}\ c$$

という項があるとき、このままでは (λ) の規則が適用できないため、(**c-app**) のような規則が必要になると考えられる。

規則 (**c-env**) は環境に対する coercion の処理を記述したものである。その最も重要な役割は、不要な変数束縛の除去である。環境 e の中に a/x^A という変数束縛が含まれていても、$x^A \notin \mathrm{TY}(e)$ ならば、e は x^A を束縛しない[3] 。よって e 中の a/x^A は除去してしまうことが出来るのである。

4.2 計算例

従来の $\lambda\varepsilon$ では、たとえば $\lambda z^{\{x^A\}}.z^{\{x^A\}}$ という関数に $\{a/x^A, b/y^B\}$ という引数を与えることは許されなかった。関数の仮引数 z の型が $\{x^A\}$ なのに対し、実引数の型が $\{x^A, y^B\}$ であり、両者が一致しないためである。

しかし $\lambda\varepsilon c$ では、部分型の導入によって環境 $\{a/x^A, b/y^B\}$ の型を $\{x^A, y^B\}$ ではなく $\{x^A\}$ とすることも出来るようになるため、この環境を上の関数に渡すことが可能となる[4] 。

ただし、$\{a/x^A, b/y^B\}$ の型が $\{x^A\}$ になるということは、この環境が x^A しか束縛できなくなることを意味する。そのことが計算時に正しく反映されないと、誤って y^B も束縛してしまうなどの不具合が生じる可能性がある。よって $\{a/x^A, b/y^B\}$ の型を $\{x^A, y^B\}$ から $\{x^A\}$ に置き換える際には、coercion を適用して $\{a/x^A, b/y^B\}\,|_{\{x^A\}}$ と書き、型が $\{x^A\}$ であることを明記しておくことが重要となる。

以下に、

$$((\lambda z^{\{x^{int}\}}.\ z^{\{x^{int}\}})\{1/x^{int}, 10/y^{int}\}\,|_{\{x^{int}\}})\,[\![x^{int} + y^{int}]\!]$$

という項に対する計算の例を挙げる。

---

[3] 自由変数、束縛変数の定義からそれは明らかである。

[4] 勿論、型を置き換えるのは引数ではなく関数の方であってもよい。

$$
\begin{aligned}
&((\lambda z^{\{x^{int}\}}.\, z^{\{x^{int}\}})\{1/x^{int}, 10/y^{int}\}|_{\{x^{int}\}})\,[\![x^{int}+y^{int}]\!] \\
&\to\ \{\{1/x^{int}, 10/y^{int}\}|_{\{x^{int}\}}/z^{\{x^{int}\}}\}[\![z^{\{x^{int}\}}]\!][\![x^{int}+y^{int}]\!] \\
&\to\ \{1/x^{int}, 10/y^{int}\}|_{\{x^{int}\}}[\![x^{int}+y^{int}]\!] \\
&\to\ \{1/x^{int}, 10/y^{int}\}|_{\{x^{int}\}}[\![x^{int}]\!] + \{1/x^{int}, 10/y^{int}\}|_{\{x^{int}\}}[\![y^{int}]\!] \\
&\overset{+}{\to}\ 1 + \{1/x^{int}, 10/y^{int}\}|_{\{x^{int}\}}[\![y^{int}]\!] \\
&\to\ 1 + y^{int}
\end{aligned}
$$

この場合、$\{1/x^{int}, 10/y^{int}\}$ の型は $\{x^{int}\}$ として扱われている。よって $10/y^{int}$ という変数束縛は無効であり、y^{int} に 10 を代入してしまったとしたらそれは誤りである。$\lambda\varepsilon c$ では、coercion の適用によってそのような間違いを防ぐことが出来るようになっている。

4.3 性質

本研究では、$\lambda\varepsilon c$ が以下のような性質を満たすことが証明できた。(証明の詳細は [6] に記載。)

Theorem 9　Subject Reduction
$\Gamma \vdash a : A$ かつ $a \to_{\lambda\varepsilon c} b$ となるとき、$\Delta \subseteq \Gamma$ であるような Δ が存在し、$\Delta \vdash b : A$ が成り立つ。

Theorem 10　$\lambda\varepsilon$ に対する Conservativity
a, b は $\lambda\varepsilon$ の項とする。このとき、$a \overset{*}{\to}_{\lambda\varepsilon} b \ \Leftrightarrow\ a \overset{*}{\to}_{\lambda\varepsilon c} b$ 。

Theorem 11　強正規化性
任意の $\lambda\varepsilon c$-term に対する簡約は必ず有限回で停止する。

Theorem 12　合流性
$a \overset{*}{\to}_{\lambda\varepsilon c} b$ かつ $a \overset{*}{\to}_{\lambda\varepsilon c} c$ ならば、$b \overset{*}{\to}_{\lambda\varepsilon c} d$ かつ $c \overset{*}{\to}_{\lambda\varepsilon c} d$ となる d が必ず存在する。

5　結論

本研究では、明示的環境計算の体系 $\lambda\varepsilon$ を拡張し、部分型を持つ計算体系 $\lambda\varepsilon c$ を構築することが出来た。環境計算において部分型を用いる場合、項の持つ型の情報が計算に正しく反映されなくなる恐れが生じる、という問題点が浮かび上がったが、$\lambda\varepsilon c$ では coercion の導入によってそれを防ぐ仕組みが実現された。そしてこの体系が、合流性や強正規化性などの望ましい性質を満たすことも証明できた。

今後の課題としては、$\lambda\varepsilon c$ をさらに発展させ、coercion の黙示化 (自動補完) を可能にするような計算系を構築することが挙げられる。また、型抽象

などの有力な概念を組み込んでいくことも、興味深いテーマであると考えられる。

謝辞

丁寧な御指導により本研究を支えていただきました、京都大学情報学研究科佐藤雅彦教授、五十嵐淳講師、中澤巧爾助手には心から感謝致します。また、本研究について数々の助言を下さった皆様方にも深く御礼を申し上げます。

参考文献

[1] M. Sato, T. Sakurai, and R. Burstall. Explicit environments. *Fundamenta Informaticae*, Vol. 45, No. 1-2, pp. 79–115, 2001.

[2] L. Cardelli. A semantics of multiple inheritance. *Information and Computation*, Vol. 76, pp. 138–164, 1988.

[3] B. Pierce. *Types and Programming Languages*. The MIT Press, 2002.

[4] V. Breazu-Tannen, T. Coquand, C. Gunter, and A. Scedrov. Inheritance as implicit coercion. *Information and Computation*, Vol. 93, pp. 172–221, 1991.

[5] H Tsuiki. *A Record Calculus with a Merge Operator*. PhD thesis, Keio University, 1992.

[6] 澤田康秀. 部分型を持つ明示的環境計算. 修士論文, 京都大学情報学研究科知能情報学専攻, 2003.