ソフトウェア基礎論配布資料 (7)
オブジェクト計算 (1): Abadi-Cardelli の ς 計算

五十嵐 淳
京都大学 大学院情報学研究科知能情報学専攻
e-mail: igarashi@kuis.kyoto-u.ac.jp

平成 16 年 11 月 30 日

1 オブジェクト計算—概要

λ 計算が関数の概念を中心に備えた計算体系であり，Lisp, Scheme, ML などの関数型プログラミング言語の中核部分のモデルであるのと同様に，オブジェクト計算は (Smalltalk, Java, C++ などオブジェクト指向言語における) オブジェクトの概念をモデル化した計算体系である．

ここでのオブジェクトは，(隠蔽された) 内部状態と，内部状態を使って計算をするメソッドと呼ばれる操作 (群) から構成されるデータと考える．例えば，二次元点オブジェクトは，内部状態として座標値の組 x, y と，以下のメソッド

- x 座標値を知るためのメソッド *getx*
- y 座標値を知るためのメソッド *gety*
- x 座標値を変更するメソッド *putx*
- y 座標値を変更するメソッド *puty*
- 移動量 dx, dy を引数として受けとって点を $x + dx, y + dy$ に移動させるメソッド *move*

から構成することができる．また，数値などの基本的なデータでさえも，足し算やかけ算メソッドを持つオブジェクトと考えることもできる．内部状態は，通常，名前のついたデータの組 (例えば x という名前でアクセスされる数値と y という名前でアクセスされる数値) の集まりで表わされ，その各々をフィールドと呼ぶ．フィールドやメソッドについた名前をラベルということもある．

計算は，オブジェクトのメソッドを呼び出すことで発生する．オブジェクトの重要な機能のひとつとして「自分自身のメソッドを呼び出せる」ことが挙げられる．例えば，*move* メソッドは，自分自身の *putx*, *puty* メソッドを呼び出して，実装することができる．これは，this (Java, C++) や self と呼ばれる機構として実際のプログラミング言語にも備わっている．

ここで紹介する，Abadi と Cardelli による ς 計算は，このようなオブジェクトの概念を

- メソッドとフィールドの概念を統一し，
- 状態の隠蔽は (とりあえず) 扱わず，

- オブジェクトのアイデンティティは厳密に扱わない

というアプローチで体系化し単純な計算モデルを構築している．

例えば，一次元点オブジェクトは

$$\begin{array}{l}[x = \varsigma(s)5,\\ \ get = \varsigma(s)s.x,\\ \ put = \varsigma(s)\lambda x'.s.x \Leftarrow \varsigma(s')x',\\ \ move = \varsigma(s)\lambda dx.s.put\ (s.get + dx)]\end{array}$$

と記述することができる．(例をわかりやすくするために，メソッド引数の受渡しに λ 抽象・関数適用を使っていたり，数値や足し算を用いているが，これらは本来の体系には存在しない．) また計算はオブジェクトのメソッドを呼び出すことにより起こり，計算結果は，選ばれたメソッド本体内の (ς の直後に宣言されている) 自分自身を表わす変数にオブジェクト自身 (のコピー) を代入したものになる．(参考: λ 計算の β 簡約)

上のオブジェクトを p とすると，

$$\begin{array}{rcl} p.move\ 2 & \longrightarrow & ([s \mapsto p]\lambda dx.s.put(s.get + dx))\ 2\\ & \equiv & (\lambda dx.p.put\ (p.get + dx))\ 2\\ & \longrightarrow & p.put\ (p.get + 2)\\ & \longrightarrow & p.put\ (p.x + 2)\\ & \longrightarrow & p.put\ (5 + 2)\\ & \longrightarrow & p.put\ 7\\ & \longrightarrow & (\lambda x'.p.x \Leftarrow \varsigma(s')x')\ 7\\ & \longrightarrow & p.x \Leftarrow \varsigma(s')7\\ & \longrightarrow & [x = \varsigma(s')7,\ get = \varsigma(s)s.x, \ldots] \end{array}$$

2 型無し ς 計算

2.1 形式的定義

ラベルの集合を $\mathcal{L}$ とし，メタ変数として l を使用する．

2.1.1 定義: 項 t は以下の文法で定義される．

$$t ::= \quad \#^i x \mid [l = \varsigma(x)t, \ldots, l = \varsigma(x)t] \mid t.l \mid t.l \Leftarrow \varsigma(x)t$$

□

$[l_1 = \varsigma(x_1)t_1, \ldots, l_n = \varsigma(x_n)t_n]$ において $l_i = \varsigma(x_i)t_i$ の順序は重要ではないため，これらを入れ替えた項を ($\equiv$ の意味で) 同一視する．また略記として，$[l_i = \varsigma(x_i)t_i\ ^{i\in 1..n}]$ のような記法を用いる．

$\varsigma(x)t$ の x は宣言であり，その有効範囲は t である．

2.1.2 定義 [自由変数]: 項の自由変数集合 $FV(t)$ は以下のように定義される．

$$\begin{array}{lcl} FV(\#^i x) & = & \{\#^i x\}\\ FV([l_i = \varsigma(x_i)t_i\ ^{i\in 1..n}]) & = & \bigcup_{i\in 1..n} \left(\{\#^{j-1} x_i \mid \#^j x_i \in FV(t_i)\} \cup \{\#^j y \mid \#^j y \in FV(t_i)\ \&\ y \neq x_i\}\right)\\ FV(t.l) & = & FV(t)\\ FV(t_1.l \Leftarrow \varsigma(x)t_2) & = & FV(t_1) \cup \{\#^{j-1} x \mid \#^j x \in FV(t_2)\} \cup \{\#^j y \mid \#^j y \in FV(t)\ \&\ y \neq x\} \end{array}$$

$x \notin FV(t)$ のとき $l = \varsigma(x)t$ をフィールドと呼び，単に $l = t$ と略記する．また，同様に $t_1.l \Leftarrow \varsigma(x)t_2$ も $x \notin FV(t_2)$ のとき，$t_1.l := t_2$ と略記する．

代入操作 $[x \mapsto t]$ は λ 計算と同様に，shifting 操作 $\Uparrow_x t$, $\Downarrow_x t$, $[t/\#^n x]$ を経由して定義される．(定義は省略．)

2.1.3 定義 [簡約関係 $\longrightarrow_f$, $\longrightarrow_v$]: 簡約関係 $t \longrightarrow_f t'$ を以下の規則で定義する．また，簡約関係 $t \longrightarrow_v$ を規則 E-ObjSelect, E-ObjUpdate, E-Select, E-Update1 で定義する．

$$\frac{(t \equiv [l_i = \varsigma(x_i)t_i \ ^{i\in 1..n}])}{t.l_j \longrightarrow [x_j \mapsto t]t_j} \quad \text{(E-ObjSelect)}$$

$$\frac{(t \equiv [l_i = \varsigma(x_i)t_i \ ^{i\in 1..n}])}{t.l_j \Leftarrow \varsigma(x)t' \longrightarrow [l_i = \varsigma(x_i)t_i \ ^{i\in 1..j-1,j+1..n}, l_j = \varsigma(x)t'],} \quad \text{(E-ObjUpdate)}$$

$$\frac{t_j \longrightarrow t'_j}{[l_i = \varsigma(x_i)t_i \ ^{i\in 1..n}] \longrightarrow [l_i = \varsigma(x_i)t_i \ ^{i\in 1..j-1,j+1..n}, l_j = \varsigma(x)t'_j]} \quad \text{(E-Object)}$$

$$\frac{t \longrightarrow t'}{t.l \longrightarrow t'.l} \quad \text{(E-Select)}$$

$$\frac{t_1 \longrightarrow t'_1}{t_1.l \Leftarrow \varsigma(x)t_2 \longrightarrow t'_1.l \Leftarrow \varsigma(x)t_2} \quad \text{(E-Update1)}$$

$$\frac{t_2 \longrightarrow t'_2}{t_1.l \Leftarrow \varsigma(x)t_2 \longrightarrow t_1.l \Leftarrow \varsigma(x)t'_2} \quad \text{(E-Update2)}$$

2.2 諸性質

2.2.1 定理 [合流性, confluence]: $t_1 \longrightarrow_f^* t_2 \ \& \ t_1 \longrightarrow_f^* t_3 \Rightarrow \exists t_4.t_2 \longrightarrow_f^* t_4 \ \& \ t_3 \longrightarrow_f^* t_4$.

2.2.2 系 [正規形の唯一性, uniqueness of normal forms]: $t \longrightarrow_f^* t'$ かつ $t \longrightarrow_f^* t''$ かつ t', t'' が正規形ならば，$t' \equiv t''$ である．

2.2.3 定理: $t \longrightarrow_f^* [l_i = \varsigma(x_i)t_i \ ^{i\in 1..n}]$ ならば，$t \longrightarrow_v^* [l'_i = \varsigma(x'_i)t'_i \ ^{i\in 1..m}]$.

3 プログラミング in オブジェクト計算

3.1 λ 計算

$$\begin{array}{lcl} \langle\langle x \rangle\rangle & \equiv & x.arg \\ \langle\langle \lambda x.t \rangle\rangle & \equiv & [arg = \varsigma(x)x.arg, app = \varsigma(x)\langle\langle t \rangle\rangle] \\ \langle\langle t_1 \ t_2 \rangle\rangle & \equiv & (t_1.arg := t_2).app \end{array}$$

3.2 オブジェクト指向自然数

$$
\begin{array}{lcl}
o_0 & \equiv & [pred = \varsigma(x)x,\ add = \varsigma(x)\lambda y.y] \\
o_{n+1} & \equiv & [pred = o_n,\ add = \varsigma(x)\lambda y.x.pred := (x.pred.plus\ y)]
\end{array}
$$

4 単純型付 ς 計算

型付 λ 計算では「関数でないものを適用する」「真偽値に足し算をする」といった実行時エラーの発生を型システムを使って防ぐ (型安全性の確保) ことが，大きな目標のひとつであった．オブジェクト計算でも,「存在しないメソッドを呼び出す」といった実行時エラーを型システムを使って防ぐことを考えていく．そのための単純な方法として，オブジェクトの型を存在するメソッドの名前と，それが返す結果の型の組として表現することを考える．これを実現したのが，下の単純型付 ς 計算である．

4.1 形式的定義

4.1.1 定義: 型 T，項 t，値 v, 型環境 Γ は以下の文法で定義される．

$$
\begin{array}{ll}
T ::= & [l : T, \ldots, l : T] \\
t ::= & \#^i x \mid [l = \varsigma(x{:}T)t, \ldots, l = \varsigma(x{:}T)t] \mid t.l \mid t.l \Leftarrow \varsigma(x)t \\
v ::= & [l = \varsigma(x{:}T)t, \ldots, l = \varsigma(x{:}T)t] \\
\Gamma ::= & \bullet \mid \Gamma, x{:}T
\end{array}
$$

$[l_1 : T_1, \ldots, l_n : T_n]$ は $[l_i : T_i\ ^{i\in 1..n}]$ と略記する．

4.1.2 定義: 型付け関係 $\Gamma \vdash t : T$ を以下の規則で定義する．

$$\frac{}{\Gamma, x{:}T \vdash \#^0 x : T} \qquad \text{(T-Var)}$$

$$\frac{\Gamma \vdash \#^n x : T}{\Gamma, x{:}T' \vdash \#^{n+1} x : T} \qquad \text{(T-Weak1)}$$

$$\frac{\Gamma \vdash \#^n x : T \qquad x \neq y}{\Gamma, y{:}T' \vdash \#^n x : T} \qquad \text{(T-Weak2)}$$

$$\frac{\Gamma, x_1{:}T \vdash t_1 : T_1 \qquad \cdots \Gamma, x_n{:}T \vdash t_n : T_n \qquad (T \equiv [l_i : T_i\ ^{i\in 1..n}])}{\Gamma \vdash [l_i = \varsigma(x_i{:}T)t_i\ ^{i\in 1..n}] : T} \qquad \text{(T-Object)}$$

$$\frac{\Gamma \vdash t : [l_i : T_i\ ^{i\in 1..n}]}{\Gamma \vdash t.l_k : T_k} \qquad \text{(T-Select)}$$

$$\frac{(T \equiv [l_i : T_i\ ^{i\in 1..n}]) \qquad \Gamma \vdash t : T \qquad \Gamma, x{:}T \vdash t' : t_k}{\Gamma \vdash t.l_k \Leftarrow \varsigma(x{:}T)t' : T} \qquad \text{(T-Update)}$$

4.2 諸性質

4.2.3 定理 [Uniquness of Typing]: $\Gamma \vdash t : T$ かつ $\Gamma \vdash t : T'$ ならば $T \equiv T'$ である.

4.2.4 定理 [Type Preservation]: $\Gamma \vdash t : T$ かつ $t \longrightarrow_v t'$ ならば, $\Gamma \vdash t' : T$ である.

4.2.5 定理 [Progress]: $\bullet \vdash t : T$ かつ t が値でなければ, ある項 t' が存在して $t \longrightarrow_v t'$ である.

A 代入等の定義

$$
\begin{array}{lcll}
\Uparrow_x^j \#^i x & = & \#^{i+1} x & \text{if } i \geq j \\
\Uparrow_y^j \#^i x & = & \#^i x & \text{if } i < j \text{ or } x \neq y \\
\Uparrow_x^j [l_k = \varsigma(x_k) t_k \ ^{k \in 1..n}] & = & [l_k = \varsigma(x_k) \Uparrow_x^{j_k} (t_k) \ ^{k \in 1..n}] & j_k = \begin{cases} j+1 & \text{if } x = x_k \\ j & \text{if } x \neq x_k \end{cases} \\
\Uparrow_x^j (t.l) & = & (\Uparrow_x^j t).l & \\
\Uparrow_x^j (t_1.l \Leftarrow \varsigma(x) t_2) & = & (\Uparrow_x^j t_1).l \Leftarrow \varsigma(x)(\Uparrow_x^{j+1} t_2) & \\
\Uparrow_y^j (t_1.l \Leftarrow \varsigma(x) t_2) & = & (\Uparrow_y^j t_1).l \Leftarrow \varsigma(x)(\Uparrow_y^j t_2) &
\end{array}
$$

$$
\begin{array}{lcll}
\Downarrow_x^j \#^i x & = & \#^{i-1} x & \text{if } i \geq j \ \& \ i > 0 \\
\Downarrow_y^j \#^i x & = & \#^i x & \text{if } i < j \text{ or } x \neq y \\
\Downarrow_x^j [l_k = \varsigma(x_k) t_k \ ^{k \in 1..n}] & = & [l_k = \varsigma(x_k) \Downarrow_x^{j_k} (t_k) \ ^{k \in 1..n}] & j_k = \begin{cases} j+1 & \text{if } x = x_k \\ j & \text{if } x \neq x_k \end{cases} \\
\Downarrow_x^j (t.l) & = & (\Downarrow_x^j t).l & \\
\Downarrow_x^j (t_1.l \Leftarrow \varsigma(x) t_2) & = & (\Downarrow_x^j t_1).l \Leftarrow \varsigma(x)(\Downarrow_x^{j+1} t_2) & \\
\Downarrow_y^j (t_1.l \Leftarrow \varsigma(x) t_2) & = & (\Downarrow_y^j t_1).l \Leftarrow \varsigma(x)(\Downarrow_y^j t_2) &
\end{array}
$$

$$
\begin{array}{lcll}
[t/\#^i x]\#^i x & = & t & \\
[t/\#^i x]\#^j y & = & \#^j y & \text{if } i \neq j \text{ or } x \not\equiv y \\
[t/\#^i x][l_k = \varsigma(x_k) t_k^{k \in 1..n}] & = & [l_k = \varsigma(x_k)([t/\#^{j_k} x] t_k) \ ^{i \in 1..n}] & j_k = \begin{cases} i+1 & \text{if } x = x_k \\ i & \text{if } x \neq x_k \end{cases} \\
[t/\#^i x](t_0.l) & = & ([t/\#^i x] t_0).l & \\
[t/\#^i x](t_1.l \Leftarrow \varsigma(x) t_2) & = & ([t/\#^i x] t_1).l \Leftarrow \varsigma(x)([t/\#^{i+1} x] t_2) & \\
[t/\#^i x](t_1.l \Leftarrow \varsigma(y) t_2) & = & ([t/\#^i x] t_1).l \Leftarrow \varsigma(y)([t/\#^i y] t_2) & \text{if } x \neq y
\end{array}
$$

$$
[x \mapsto t_1] t_2 \quad = \quad \Downarrow_x^0 ([\Uparrow_x^0 t_1 / x] t_2)
$$