ソフトウェア基礎論配布資料 (8)
オブジェクト計算 (2): 部分型

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 12 月 8 日

1 subsumption と部分型

通常，オブジェクト指向言語では，あるメソッド群を持つオブジェクトは，より少ないメソッド群を持つオブジェクトとして振舞えることが許されている．例えば，get, put メソッドを持つ点オブジェクトは (少なくとも)get を持つオブジェクトが期待される場面で使用することができる．逆に言うと，get を持つオブジェクトが期待される場面では，それ以外のメソッドを持つオブジェクトが使われてもいい (少なくともメソッド起動に失敗することはない) はずである．

具体的には

$$\lambda x.x.get + 1$$

のような関数を

$$[get = \varsigma(x)\cdots, put = \varsigma(x)\cdots]$$

や

$$[get = \varsigma(x)\cdots, move = \varsigma(x)\cdots]$$

のようなオブジェクトに適用することができる．これは型無しの体系であれば全く問題ないが，単純型システムでは，そもそも，そのようなプログラムは型システムではじかれてしまう．なぜなら x の宣言された型と，引数として与える項の型は正確に一致している必要があるため，以下のいずれの項も，上記二種類のオブジェクト両方に適用することはできない．

- $\lambda x{:}[get : \mathtt{nat}].x.get + 1$
- $\lambda x{:}[get : \mathtt{nat}, put : T].x.get + 1$
- $\lambda x{:}[get : \mathtt{nat}, move : T'].x.get + 1$

ここで型システムに求められる柔軟性は，まさに，「あるメソッド群を持つオブジェクトに，より少ないメソッド群を持つオブジェクトとして振舞うことを許すこと」である．これは，例えば，$[get : \mathtt{nat}, put : T]$ 型の項が $[get : \mathtt{nat}]$ 型としても型付けされることを意味する．このようなある型の項を別の型の項としてみなす事を**サブサンプション**(*subsumption*) と呼ぶ[1]．もちろん，任意の型へ

[1] **型変換**(*coercion*) も類似の概念であり,「暗黙の型変換」と呼ばれるものに近い．

の subsumption をしていいわけではなく，型安全性を保つためには，何らかの「subsumption が安全に行える型 (の関係)」を定める必要がある．この関係を**部分型**(*subtype*) 関係と呼ぶ．

以下，部分型関係を $T_1 \leq T_2$ と書き，T_1 は T_2 の部分型である，とか T_2 は T_1 の supertype である，と言ったりする．上で述べた「より少ないメソッドを持つ型への変換」を実現するために考えられる規則を一般的に書くと

$$\frac{}{[l_i : T_i{}^{\,i\in 1..n+k}] \leq [l_i : T_i{}^{\,i\in 1..n}]} \qquad \text{(S-Object)}$$

となる．部分型関係はオブジェクトの型だけではなく，型構築子毎に考えることができる．

部分型関係を型付け規則とは独立に定めることによって，型付け規則は以下の subsumption 規則ひとつを導入するだけで，実現することができる．

$$\frac{\Gamma \vdash t : T \qquad T \leq T'}{\Gamma \vdash t : T'} \qquad \text{(T-Sub)}$$

単純型付き λ 計算を導入した時に自然数を原始的な概念として体系に導入したのと同様に，以下の形式的定義では関数を原始的な概念として扱うことにする．この主な理由は，関数型に対する自然な部分型関係がオブジェクト型の部分型関係として表現できないことがあげられる．また，全ての型の supertype であるような型 `Top` を導入する．

1.1 形式的定義

1.1.1 定義: 型 T，項 t，型環境 Γ は以下の文法で定義される．

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

1.1.2 定義: 部分型関係 $T_1 \leq T_2$ を以下の規則で定義する．

$$\frac{}{T \leq T} \qquad \text{(S-Refl)}$$

$$\frac{T_1 \leq T_2 \qquad T_2 \leq T_3}{T_1 \leq T_3} \qquad \text{(S-Trans)}$$

$$\frac{}{T \leq \texttt{Top}} \qquad \text{(S-Top)}$$

$$\frac{}{[l_i : T_i{}^{\,i\in 1..n+k}] \leq [l_i : T_i{}^{\,i\in 1..n}]} \qquad \text{(S-Object)}$$

$$\frac{T_3 \leq T_1 \qquad T_2 \leq T_4}{T_1 \to T_2 \leq T_3 \to T_4} \qquad \text{(S-Fun)}$$

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

$$\frac{}{\Gamma, x{:}T \vdash \#^0 x : T} \tag{T-Var}$$

$$\frac{\Gamma \vdash \#^n x : T}{\Gamma, x{:}T' \vdash \#^{n+1} x : T} \tag{T-Weak1}$$

$$\frac{\Gamma \vdash \#^n x : T \qquad x \not\equiv y}{\Gamma, y{:}T' \vdash \#^n x : T} \tag{T-Weak2}$$

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

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

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

$$\frac{\Gamma, x{:}T_1 \vdash t_0 : T_2}{\Gamma \vdash \lambda x{:}T_1.t_0 : T_1 \to T_2} \tag{T-Abs}$$

$$\frac{\Gamma \vdash t_1 : T_1 \to T_2 \qquad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1\ t_2 : T_2} \tag{T-App}$$

$$\frac{\Gamma \vdash t : T \qquad T \leq T'}{\Gamma \vdash t : T'} \tag{T-Sub}$$

1.2 例

以下，項として t `as` T という表記を導入する．これは $\lambda x{:}T.x\ t$ の略記であり，直感的には，項 t を T の項として使う，というプログラマの意図を示した注釈であると思える．これは，プログラム中の型の注釈というだけではなく，例えば，アクセスされたくないメソッドを隠蔽するのに用いることができる．

$$\begin{array}{lcl}
\mathit{RomCell} & = & [\mathit{get} : \mathtt{nat}] \\
\mathit{PromCell} & = & [\mathit{get} : \mathtt{nat}, \mathit{set} : \mathtt{nat} \to \mathit{RomCell}] \\
\mathit{Cell} & = & [\mathit{contents} : \mathtt{nat}, \mathit{get} : \mathtt{nat}, \mathit{set} : \mathtt{nat} \to \mathit{RomCell}] \\
 & & \\
\mathit{myCell} & = & [\mathit{contents} = 0, \\
 & & \ \mathit{get} = \varsigma(s{:}\ \mathit{Cell})s.\mathit{contents}, \\
 & & \ \mathit{put} = \varsigma(s{:}\ \mathit{Cell})\lambda n{:}\mathtt{nat}.s.\mathit{contents} := n] \ \ \mathtt{as} \ \ \mathit{PromCell}
\end{array}$$

1.3 諸性質

Uniqueness of Typing の性質は当然失われるが，型付け可能ならば「最小の」型が存在することがいえる．

1.3.1 定理 [**Minimal Typing**]**:** もし $\Gamma \vdash t : T$ が成立するならば，以下の性質を満たすような T_0 が存在する．

1. $\Gamma \vdash t : T_0$ かつ
2. 任意の $\Gamma \vdash t : T'$ なる T' に対し $T_0 \leq T'$.

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

1.3.3 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 簡約関係の定義

A.1 定義 [**値**]**:** 値 v の集合は以下の文法で定義される．

$$v ::= \quad \texttt{\#}^n x \mid \lambda x{:}T.t \mid [l_i = \varsigma(x_i{:}T_i)t_i \;^{i \in 1..n}]$$

A.2 定義 [**簡約関係** $\longrightarrow_v$]**:** (値呼びの) 簡約関係 $t \longrightarrow_v t'$ は以下の規則で定義される．

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

$$\frac{(t \equiv [l_i = \varsigma(x_i{:}T_i)t_i \;^{i \in 1..n}])}{t.l_j \Leftarrow \varsigma(x{:}T)t' \longrightarrow_v [l_i = \varsigma(x_i{:}T_i)t_i \;^{i \in 1..j-1,j+1..n}, l_j = \varsigma(x{:}T_j)t']} \qquad \text{(E-ObjUpdate)}$$

$$\frac{t \longrightarrow_v t'}{t.l \longrightarrow_v t'.l} \qquad \text{(E-Select)}$$

$$\frac{t_1 \longrightarrow_v t_1'}{t_1.l \Leftarrow \varsigma(x)t_2 \longrightarrow_v t_1'.l \Leftarrow \varsigma(x)t_2} \qquad \text{(E-Update1)}$$

$$\frac{}{(\lambda x.t_1)\; v_2 \longrightarrow_v [x \mapsto v_2]t_1} \qquad \text{(E-AppAbs)}$$

$$\frac{t_1 \longrightarrow_v t_1'}{t_1\; t_2 \longrightarrow_v t_1'\; t_2} \qquad \text{(E-App1)}$$

$$\frac{t_2 \longrightarrow_v t_2'}{v_1\; t_2 \longrightarrow_v v_1\; t_2'} \qquad \text{(E-App2)}$$