Locally Nameless手法を使った型システムの健全性の証明

廣田知子　(指導教官: 浅井健一)

1　はじめに

型システムの重要な性質として健全性（つまり、型チェックを通ったら、実行時には型エラーが決して起らない）を挙げることができる。本研究では、定理証明系システム Coq を用いて行われた Locally Nameless 手法を用いた単純λ計算の型システムの健全性の既存の証明 [2] を拡張し、元のシステムに shift/reset を付加した多相の型システム [1] における健全性を検証する。

2　Locally Nameless 手法

Locally Nameless 手法は、α-equality の問題を解消するための変数の名前付けの手法であり、自由変数には $x, y, ...$ 等の名前を付けるが、局所変数（束縛変数）には名前を付けず、de Bruijn index を採用する。局所変数名を 0 以上の整数で表し、その式における一番内側の binder の変数名を 0 と考え、外にいくに従って 0, 1, 2,... と大きくしていく。例えば、λ 計算式の $\lambda x.\lambda y.\lambda z.y\,x\,z$ は $\lambda.\lambda.\lambda.1\,2\,0$ となる。

3　継続と shift / reset

shift/reset は継続を扱うための命令であり、これらを使ってプログラムの例外処理等を実現することが出来る。継続はいわば現在の計算が終了した後にする仕事であり、shift は現在の継続を取得し、reset は取得する継続の範囲を限定する命令である。

4　型付き言語の構文

本研究で使用する型付き関数型言語の構文は以下の通り。

$$
\begin{array}{llrll}
\text{値}: & v & := & n \mid x \mid \lambda.e & \\
\text{式}: & e & := & v & (\text{値}) \\
 & & \mid & e_1\,e_2 & (\text{関数適用}) \\
 & & \mid & S.e & (shift\ \text{式}) \\
 & & \mid & \langle e \rangle & (reset\ \text{式}) \\
 & & \mid & let\ e_1\ e_2 & (let\ \text{式}) \\
\text{単相型}: & \alpha,\beta,\gamma,\delta & := & tb \mid tf \mid (\alpha/\gamma \to \beta/\delta) & \\
\text{多相型}: & A & := & \alpha \mid \forall.A &
\end{array}
$$

n は局所変数、x は自由変数、tb は (n と同じく de Bruijn index で表された) 局所型変数、tf は自由型変数を表す。$(\alpha/\gamma \to \beta/\delta)$ は $\alpha \to \beta$ 型の関数で、実行すると現在の継続の返す型が γ から δ に変化するような式の型である。shift や reset が出てこなければ、γ と δ は常に等しくなる。$\forall.A$ は多相の型を表す。$\forall$ で束縛される型変数や $shift$ 式 $S.e$ は λ 式と同じく de Bruijn index で表される。

5　評価規則

式の評価規則は $e \rightsquigarrow e'$ で表現する。これは「e は e' へと評価できる」という意味である。以下に本研究における型システムの評価規則を示す。

$$
\begin{array}{rcll}
e_1\,e_2 & \rightsquigarrow & e_1'\,e_2 & \text{if } e_1 \rightsquigarrow e_1' \\
v\,e_2 & \rightsquigarrow & v\,e_2' & \text{if } e_2 \rightsquigarrow e_2' \\
(\lambda.e)\,v & \rightsquigarrow & e^v & \\
let\ e_1\ e_2 & \rightsquigarrow & let\ e_1'\ e_2 & \text{if } e_1 \rightsquigarrow e_1' \\
let\ v\ e & \rightsquigarrow & e^v & \\
(S.e_1)\,e_2 & \rightsquigarrow & S.let\ \lambda.\langle 1\,(0\,e_2)\rangle\ e_1 & \\
v\,(S.e_1) & \rightsquigarrow & S.let\ \lambda.\langle 1\,(v\,0)\rangle\ e_1 & \\
\langle e \rangle & \rightsquigarrow & \langle e' \rangle & \text{if } e \rightsquigarrow e' \\
\langle S.e \rangle & \rightsquigarrow & \langle let\ \lambda.0\ e \rangle & \\
\langle v \rangle & \rightsquigarrow & v &
\end{array}
$$

ここで e^v は e の中に含まれる局所変数 0 に v を代入したものを表す。

6　型規則

式の型規則は以下の様に表記する。

$$\Gamma;\ \alpha \vdash e : T;\ \beta$$

これは「型環境 Γ において、式 e は T 型を持ち、その実行によって継続の返す型が α から β に変化する」ことを表す。又、どんな型 α においても $\Gamma;\ \alpha \vdash e : T;\ \alpha$ が得られる時は以下の様に書く。

$$\Gamma \vdash_p e : T$$

7　型の推論規則

上の表記法を使い、式を型付けするために必要な型の推論規則を以下に示す。

$$\frac{ok\ \Gamma \quad (x : M) \in \Gamma}{\Gamma \vdash_p x : M^{X_s}}\ (var)$$

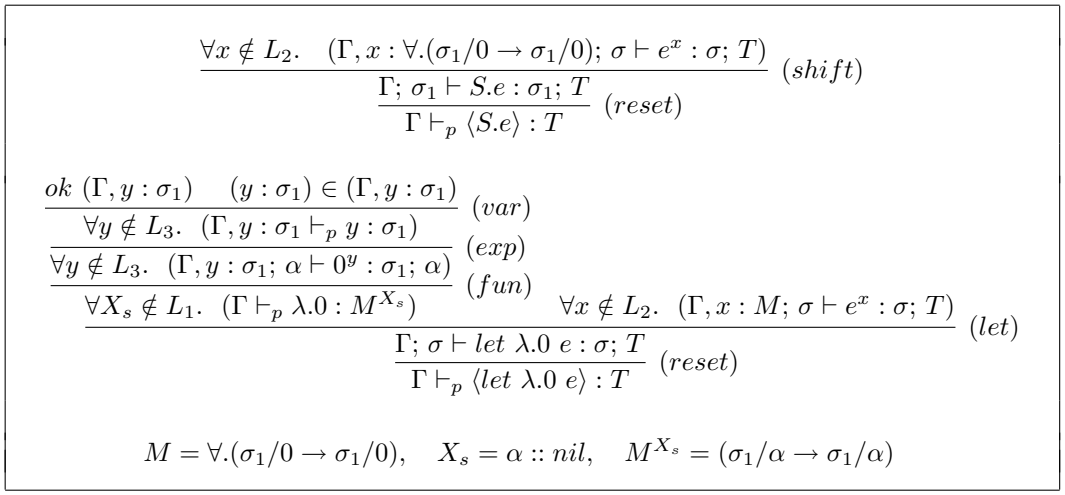
$$\frac{\forall x \notin L.\quad (\Gamma, x : \sigma;\ \alpha \vdash e^x : T;\ \beta)}{\Gamma \vdash_p \lambda.e : (\sigma/\alpha \to T/\beta)}\ (fun)$$

$$\frac{\Gamma;\ \gamma \vdash e_1 : (\sigma/\alpha \to T/\beta);\ \delta \quad \Gamma;\ \beta \vdash e_2 : \sigma;\ \gamma}{\Gamma;\ \alpha \vdash e_1\,e_2 : T;\ \delta}\ (app)$$

$$\frac{\forall X_s \notin L.\ (\Gamma \vdash_p e_1 : M^{X_s}\ \ \Gamma, x : M;\ \alpha \vdash e_2^x : T;\ \beta)}{\Gamma;\ \alpha \vdash let\ e_1\ e_2 : T;\ \beta}\ (let)$$

$$\frac{\forall x \notin L.\quad (\Gamma, x : \forall.(T/0 \to \alpha/0);\ \sigma \vdash e^x : \sigma;\ \beta)}{\Gamma;\ \alpha \vdash S.e : T;\ \beta}\ (shift)$$

$$\frac{\Gamma;\ \sigma \vdash e : \sigma;\ T}{\Gamma \vdash_p \langle e \rangle : T}\ (reset) \qquad \frac{\Gamma \vdash_p e : T}{\Gamma;\ \alpha \vdash e : T;\ \alpha}\ (exp)$$



$$\frac{\forall x \notin L_2.\quad (\Gamma, x : \forall.(\sigma_1/0 \to \sigma_1/0);\ \sigma \vdash e^x : \sigma;\ T)}{\Gamma;\ \sigma_1 \vdash S.e : \sigma_1;\ T}\ (shift)$$
$$\frac{}{\Gamma \vdash_p \langle S.e \rangle : T}\ (reset)$$

$$\frac{\dfrac{\dfrac{\dfrac{ok\ (\Gamma, y : \sigma_1) \quad (y : \sigma_1) \in (\Gamma, y : \sigma_1)}{\forall y \notin L_3.\ (\Gamma, y : \sigma_1 \vdash_p y : \sigma_1)}\ (var)}{\forall y \notin L_3.\ (\Gamma, y : \sigma_1;\ \alpha \vdash 0^y : \sigma_1;\ \alpha)}\ (exp)}{\forall X_s \notin L_1.\ (\Gamma \vdash_p \lambda.0 : M^{X_s})}\ (fun) \qquad \forall x \notin L_2.\ (\Gamma, x : M;\ \sigma \vdash e^x : \sigma;\ T)}{\Gamma;\ \sigma \vdash let\ \lambda.0\ e : \sigma;\ T}\ (let)$$
$$\frac{}{\Gamma \vdash_p \langle let\ \lambda.0\ e \rangle : T}\ (reset)$$

$$M = \forall.(\sigma_1/0 \to \sigma_1/0), \quad X_s = \alpha :: nil, \quad M^{X_s} = (\sigma_1/\alpha \to \sigma_1/\alpha)$$

図 1: 型の推論規則の適用例（証明木）

$\frac{A}{B}$

この規則の読み方は、においては「B が成り立つためには A が成り立つ必要がある」である。例えば (var) の規則は「$\Gamma \vdash_p x : M^{X_s}$ が成り立つためには型環境 Γ が ok であり、かつ x が M 型を持つという情報が Γ に含まれていなくてはならない」と読める。Γ が ok であるとは、Γ 中に現れる変数名が全て異なることを意味している。又、M^{X_s} は M 中に現れる局所変数を全て自由変数のリスト X_s 中の要素で置き換えたものを表す。(fun) や $(shift)$ に現れる $\forall x \notin L.$ は L に含まれない x(自由変数) を、(let) での $\forall X_s \notin L.$ は自由変数のリスト X_s のどの要素も L に含まれていないことを表している。L には何も制限を付けていないので、後述の健全性の証明に際して自分で自由に L を定めることが出来る。e^x は上述の評価規則の場合と同じである。

8 Coq による定式化

型システムとして、前述した言語の構文、式の評価規則、型規則と型の推論規則を定式化するが、その前にこれら諸規則の定式化に必要な、局所 (型) 変数を自由 (型) 変数に置き換える関数を導入する。その後さらに健全性の証明に必要となる諸々の補題を導入する。なお、型規則が二種類存在するため、Coq においては型の推論規則を相互再帰的に宣言する。

9 型システムの健全性

型システムの健全性 (Type Soundness) を証明するには、以下の二つの定理が成立することを示せばよい。

定理 1 (type preservation)

$\Gamma;\ \alpha \vdash e : T;\ \beta \bigwedge e \rightsquigarrow e' \Rightarrow \Gamma;\ \alpha \vdash e' : T;\ \beta$

$\Gamma \vdash_p e : T \bigwedge e \rightsquigarrow e' \Rightarrow \Gamma \vdash_p e' : T$

定理 2 (type progress)

$\vdash_p \langle e \rangle : T \Rightarrow e \in v \bigvee \exists e', \langle e \rangle \rightsquigarrow e'$

定理 1 の証明は評価規則について場合分けを行い、評価前の式に推論規則を適用して証明木を作り、それを用いて評価後の式の証明木が作れることを示していく。その例が図 1 である。図 1 では評価規則 $\langle S.e \rangle \rightsquigarrow \langle let\ \lambda.0\ e \rangle$ に関して $\langle S.e \rangle$ と $\langle let\ \lambda.0\ e \rangle$ のそれぞれに対して証明木を作成し、両式が同じ型を持つことを示した。評価前の式の証明木に現れる x は $\Gamma;\ \sigma_1 \vdash S.e : \sigma_1;\ T$ に現れない（つまり Γ に現れない）変数なので、その条件を満たすように L_2 を定めることが出来、評価後の式における証明木の (let) に現れる x も又 Γ に含まれない変数であるから、前者で定義した L_2 を持ってくることが出来る。ここで例えば x を他のどこにも現れない変数と規定すると、左式で現れる変数（X_s 中の要素や y）とも異なる必要があるので、(let) の時点で値を定めることが出来ず、証明が複雑になる。これが推論規則で $x \notin L$ と定義した利点である。又、定理 2 の証明については型の推論規則で場合分けを行い、それぞれについて定理が成立することを示していく。

10 まとめと今後の課題

本研究では単純 λ 計算に shift/reset を付加した多相の型システムにおける健全性を Locally Nameless 手法を用いて Coq 上で証明した。今後はこの型システムにさらに部分評価器を導入したシステムに関する健全性を Coq を用いて検証する予定である。

参考文献

[1] Asai, K., and Y. Kameyama “Polymorphic Delimited Continuations,” *APLAS'07, LNCS 4807*, pp. 239–254 (November 2007).

[2] Aydemir. B., A. Charguéraud, B. C. Pierce, R. Pollack, and S. Weirich “Engineering Formal Metatheory,” *POPL'08*, pp. 3–15 (January 2008).