多段階 let 挿入を行うコード生成言語の型システムの設計

大石純平　亀山幸義

筑波大学 コンピュータ・サイエンス専攻

2016/9/9

日本ソフトウェア科学会第 33 回大会

アウトライン

1. 概要
2. 研究の目的
3. 研究の内容
4. まとめと今後の課題

アウトライン

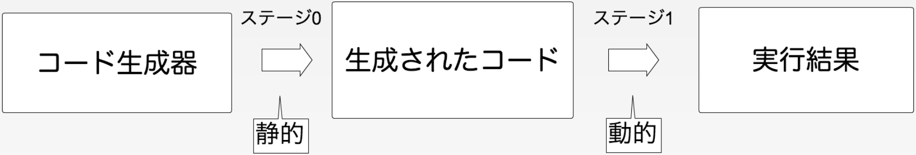
1. 概要
2. 研究の目的
3. 研究の内容
4. まとめと今後の課題

概要

プログラムを生成するプログラミング言語の安全性を保証する研究

⇒ **多段階 let 挿入**を効率的かつ安全に扱うための型システムを構築

コード生成



- **コード生成をサポートするプログラム言語 (=コード生成言語)**

コード生成言語による記述例

コード生成器	生成されるコード
$(\underline{\mathbf{int}}\ \ 3) \rightsquigarrow^*$	$\texttt{<}3\texttt{>}$
$(\underline{\mathbf{int}}\ \ 3)\ \underline{+}\ (\underline{\mathbf{int}}\ \ 5) \rightsquigarrow^*$	$\texttt{<}3+5\texttt{>}$
$\underline{\lambda} x.\ x\ \underline{+}\ (\underline{\mathbf{int}}\ \ 3) \rightsquigarrow^*$	$\texttt{<}\lambda x'.x'+3\texttt{>}$
$\underline{\mathbf{for}}\ x = \cdots\ \underline{\mathbf{to}}\ \cdots\ \underline{\mathbf{do}}\ \ \cdots \rightsquigarrow^*$	$\texttt{<}\mathbf{for}\ x' = \cdots\ \mathbf{to}\ \cdots\ \mathbf{do}\ \ \cdots\texttt{>}$

コード生成言語による記述例

コード生成器		生成されるコード
$(\underline{\textsf{int}}\ \ 3)$	$\rightsquigarrow^*$	$<3>$
$(\underline{\textsf{int}}\ \ 3)\ \underline{+}\ (\underline{\textsf{int}}\ \ 5)$	$\rightsquigarrow^*$	$<3+5>$
$\underline{\lambda} x.\ x\ \underline{+}\ (\underline{\textsf{int}}\ \ 3)$	$\rightsquigarrow^*$	$<\lambda x'.x' + 3>$
$\underline{\textsf{for}}\ x = \cdots\ \underline{\textsf{to}}\ \cdots\ \underline{\textsf{do}}\ \ \cdots$	$\rightsquigarrow^*$	$<\textsf{for}\ x' = \cdots\ \textsf{to}\ \cdots\ \textsf{do}\ \ \cdots>$

コードコンビネータ

- 下線つきの演算子
- コードを引数にとり，コードを返す

power 関数のコード生成器

普通の power 関数

$$
\begin{aligned}
\text{power} = \;& \lambda x.\textbf{fix}\ \lambda f.\lambda n. \\
& \quad \textbf{if}\ n = 0\ \textbf{then}\ 1 \\
& \quad \textbf{else}\ x\ \times\ (f\ (n-1))
\end{aligned}
$$

power 関数のコード生成器

`gen_power`: power **コード生成器**

$$
\begin{aligned}
\mathsf{gen\_power} = \ & \underline{\lambda} x.\mathbf{fix}\ \lambda f.\lambda n. \\
& \quad \mathbf{if}\ n = 0\ \mathbf{then}\ (\underline{\mathbf{int}}\ \ 1) \\
& \quad \mathbf{else}\ x\ \underline{\times}\ (f\ \ (n-1))
\end{aligned}
$$

power 関数のコード生成器

`gen_power`: power **コード生成器**

$$\begin{aligned}\text{gen\_power} = \ & \underline{\lambda} x.\mathbf{fix}\ \lambda f.\lambda n. \\ & \quad \mathbf{if}\ n = 0\ \mathbf{then}\ (\underline{\mathbf{int}}\ \ 1) \\ & \quad \mathbf{else}\ x\ \underline{\times}\ (f\ \ (n-1))\end{aligned}$$

$n = 5$ **に特化したコード生成**:

power 関数のコード生成器

`gen_power`: power **コード生成器**

$$
\begin{aligned}
\mathsf{gen\_power} = \ & \underline{\lambda} x.\textbf{fix}\ \lambda f.\lambda n. \\
& \quad \textbf{if}\ n = 0\ \textbf{then}\ (\underline{\textbf{int}}\ \ 1) \\
& \quad \textbf{else}\ x\ \underline{\times}\ (f\ \ (n-1))
\end{aligned}
$$

$n = 5$ **に特化したコード生成**:

$$\underline{\lambda} x.\ \mathsf{gen\_power}\ x\ 5 \rightsquigarrow^{*} <\lambda x'.\ x' \times x' \times x' \times x' \times x' \times 1>$$

- power **関数より高速**

二重 for ループのコード生成器

コード生成器

$$\underline{\textbf{for}}\ x = (\underline{\textbf{int}}\ \ 0)\ \underline{\textbf{to}}\ (\underline{\textbf{int}}\ \ n)\ \underline{\textbf{do}}$$
$$\quad \underline{\textbf{for}}\ y = (\underline{\textbf{int}}\ \ 0)\ \underline{\textbf{to}}\ (\underline{\textbf{int}}\ \ m)\ \underline{\textbf{do}}$$
$$\qquad \underline{\textbf{set}}\ a\ (x, y)\ \textsf{complex\_calculation}$$

二重 for ループのコード生成器

コード生成器

$$\underline{\textbf{for}}\ x = (\underline{\textbf{int}}\ \ 0)\ \underline{\textbf{to}}\ (\underline{\textbf{int}}\ \ n)\ \underline{\textbf{do}}$$
$$\quad \underline{\textbf{for}}\ y = (\underline{\textbf{int}}\ \ 0)\ \underline{\textbf{to}}\ (\underline{\textbf{int}}\ \ m)\ \underline{\textbf{do}}$$
$$\qquad \underline{\textbf{set}}\ a\ (x, y)\ \text{complex\_calculation}$$

$$\rightsquigarrow^{*}$$

生成されるコード

$$< \textbf{for}\ x' = 0\ \textbf{to}\ n\ \textbf{do}$$
$$\quad \textbf{for}\ y' = 0\ \textbf{to}\ m\ \textbf{do}$$
$$\qquad a[x', y'] \leftarrow \text{complex\_calculation} >$$

ループ不変式の移動

生成されるコード

< **for** $x' = 0$ **to** n **do**

for $y' = 0$ **to** m **do**

$a[x', y'] \leftarrow$ cc2

$b[x', y'] \leftarrow$ cc1>

ループ不変式の移動

生成されるコード

< **let** w = cc1 **in**
for $x' = 0$ **to** n **do**
let u = cc2 **in**
for $y' = 0$ **to** m **do**
$a[x', y'] \leftarrow u$
$b[x', y'] \leftarrow w$ >

ループ不変式の移動

生成されるコード

$$
\begin{array}{l}
< \textbf{let } \color{blue}{w} = \color{blue}{\mathsf{cc1}} \textbf{ in} \\
\quad \textbf{for } x' = 0 \textbf{ to } n \textbf{ do} \\
\quad\; \textbf{let } \color{red}{u} = \color{red}{\mathsf{cc2}} \textbf{ in} \\
\quad\; \textbf{for } y' = 0 \textbf{ to } m \textbf{ do} \\
\quad\quad a[x', y'] \leftarrow \color{red}{u} \\
\quad\quad b[x', y'] \leftarrow \color{blue}{w} \qquad >
\end{array}
$$

多段階 let 挿入

- 入れ子になった for ループなどを飛び越えた**複数のコード移動**を許す仕組み
- 効率的なコード生成に必要

危険な例

危険なコード生成の例

生成される危険なコード

< **for** $x' = 0$ **to** n **do**

for $y' = 0$ **to** m **do**

$a[x', y'] \leftarrow$ cc2

$b[x', y'] \leftarrow$ cc1>

危険なコード生成の例

生成される危険なコード

$$
\begin{array}{l}
< \textbf{let } w = \text{cc1} \textbf{ in} \\
\quad \textbf{for } x' = 0 \textbf{ to } n \textbf{ do} \\
\quad\; \textbf{let } u = \text{cc2} \textbf{ in} \\
\quad\; \textbf{for } y' = 0 \textbf{ to } m \textbf{ do} \\
\qquad a[x', y'] \leftarrow u \\
\qquad b[x', y'] \leftarrow w \qquad >
\end{array}
$$

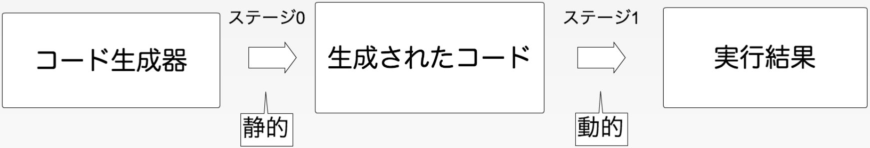
危険なコード生成の例

生成される危険なコード

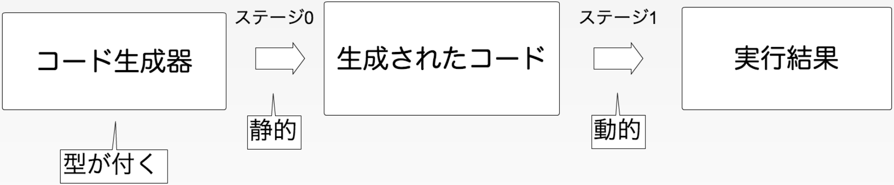
$$
\begin{array}{l}
< \textbf{let } w = \mathsf{cc1} \textbf{ in} \\
\quad \textbf{for } x' = 0 \textbf{ to } n \textbf{ do} \\
\quad\; \textbf{let } u = \mathsf{cc2} \textbf{ in} \\
\quad\; \textbf{for } y' = 0 \textbf{ to } m \textbf{ do} \\
\qquad a[x', y'] \leftarrow u \\
\qquad b[x', y'] \leftarrow w \qquad >
\end{array}
$$

- cc1 が安全な条件 $\Longleftrightarrow$ cc1 に x' か y' が含まれない
- cc2 が安全な条件 $\Longleftrightarrow$ cc2 に y' が含まれない

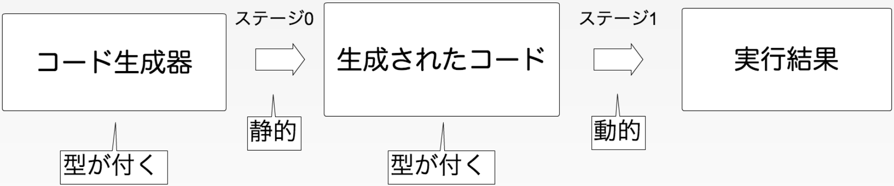
コード生成前に型付け，生成後のコードの型安全性を保証



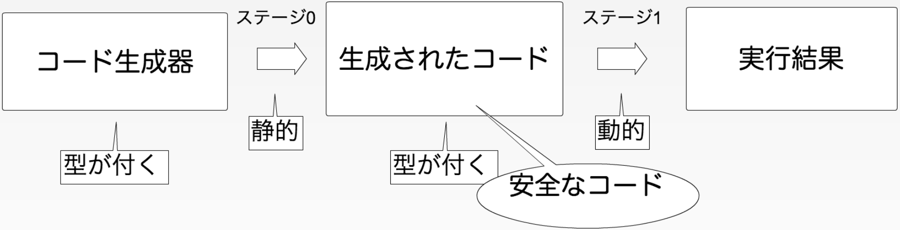
コード生成前に型付け，生成後のコードの型安全性を保証



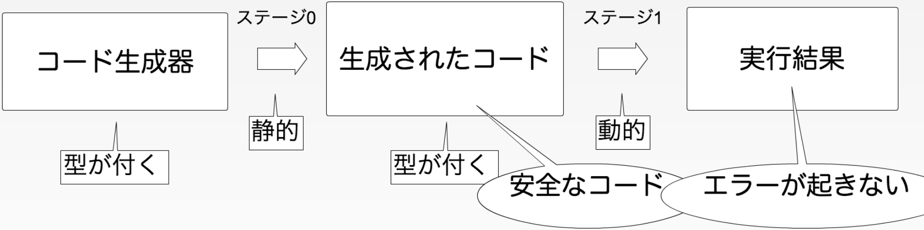
コード生成前に型付け，生成後のコードの型安全性を保証



コード生成前に型付け，生成後のコードの型安全性を保証



コード生成前に型付け，生成後のコードの型安全性を保証



アウトライン

研究の目的

表現力と安全性を兼ね備えたコード生成言語の構築

- **表現力**: **多段階** let **挿入，メモ化等の技法を表現**
- **安全性**: **生成されるコードの一定の性質を静的に検査**

研究の目的

表現力と安全性を兼ね備えたコード生成言語の構築

- 表現力: 多段階 let 挿入，メモ化等の技法を表現
- 安全性: 生成されるコードの一定の性質を静的に検査

本研究: 簡潔で強力なコントロールオペレータに基づくコード生成体系の構築

- コントロールオペレータ shift0/reset0 を利用し，let 挿入などのコード生成技法を表現
- 型システムを構築して型安全性を保証

アウトライン

表現力を上げ（コードレベルでの多段階let挿入），安全性も保証するためにどうすればよいのか

まず表現力について

let 挿入の実現方法

コード生成器

$\circ$ <u>**for**</u> $x = e1$ <u>**to**</u> $e2$ <u>**do**</u>

$\circ$ <u>**for**</u> $y = e3$ <u>**to**</u> $e4$ <u>**do**</u>

<u>**set**</u> <a> (x, y) $\circ$ cc

$\leadsto^*$

生成されるコード

<**let** u' $=$ cc$'$ **in**

for $x' = e1'$ **to** $e2'$ **do**

for $y' = e3'$ **to** $e4'$ **do**

$a[x', y'] \leftarrow u'$>

let 挿入の実現方法

コード生成器

○ **for** $x = e1$ **to** $e2$ **do**

  ○ **for** $y = e3$ **to** $e4$ **do**

    set <a> (x, y) ○ cc

$\rightsquigarrow^*$

生成されるコード

<**let** u' $=$ cc$'$ **in**

for $x' = e1'$ **to** $e2'$ **do**

  for $y' = e3'$ **to** $e4'$ **do**

    $a[x', y'] \leftarrow u'$>

shift0/reset0 の導入

○ のところに shift0/reset0 を用いることで，多段階 let 挿入を行う

shift0/reset0 による let 挿入

コード生成器:

$$\underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$$
$$\underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$$
$$\underline{\textbf{set}}\ a\ (x, y)\ cc$$

shift0/reset0 による let 挿入

コード生成器: **reset0** **for** $x = e1$ **to** $e2$ **do**

for $y = e3$ **to** $e4$ **do**

set $a\ (x, y)$ **shift0** $k \rightarrow$ **let** $u = cc$ **in** **throw** $k\ u$

shift0/reset0 による let 挿入

$$\textbf{reset0}\ (E[\textbf{shift0}\ k \to e]) \ \rightsquigarrow\ e\{k \Leftarrow E\}$$

コード生成器: $\textbf{reset0}\ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$

$\underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$

$\underline{\textbf{set}}\ a\ (x, y)\ \textbf{shift0}\ k \to \underline{\textbf{let}}\ u = cc\ \underline{\textbf{in}}\ \textbf{throw}\ k\ u$

$k \Leftarrow\ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$

$\underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$

$\underline{\textbf{set}}\ a\ (x, y)\ [\ ]$

shift0/reset0 による let 挿入

$$\textbf{reset0}\ (E[\textbf{shift0}\ k \to e]) \ \rightsquigarrow\ e\{k \Leftarrow E\}$$

コード生成器: $\textbf{reset0}\ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$

$\underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$

$\underline{\textbf{set}}\ a\ (x, y)\ \textbf{shift0}\ k \to \underline{\textbf{let}}\ u = cc\ \underline{\textbf{in}}\ \textbf{throw}\ k\ u$

$k \Leftarrow\ \ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$

$\underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$

$\underline{\textbf{set}}\ a\ (x, y)\ [\ ]$

生成コード: $<\ \textbf{let}\ u' = cc'\ \textbf{in}$

$\textbf{for}\ x' = e1'\ \textbf{to}\ e2'\ \textbf{do}$

$\textbf{for}\ y' = e3'\ \textbf{to}\ e4'\ \textbf{do}$

$a[x', y'] \leftarrow u'\ >$

shift0/reset0 による多段階 let 挿入

$$\textbf{reset0}\ (E[\textbf{shift0}\ k \to e]) \ \rightsquigarrow\ e\{k \Leftarrow E\}$$

コード生成器: $\textbf{reset0}\ \ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$

$\textbf{reset0}\ \ \underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$

$\underline{\textbf{set}}\ a\ (x, y)\ \textbf{shift0}\ k_1 \to \underline{\textbf{let}}\ u = cc1\ \underline{\textbf{in}}\ \textbf{throw}\ k_1\ u;$

$\underline{\textbf{set}}\ b\ (x, y)\ \textbf{shift0}\ k_1 \to\ \textbf{shift0}\ k_2 \to$

$\underline{\textbf{let}}\ w = cc2\ \underline{\textbf{in}}\ \textbf{throw}\ k_2(\textbf{throw}\ k_1\ w)$

shift0/reset0 による多段階 let 挿入

$$\textbf{reset0}\ (E[\textbf{shift0}\ k \to e]) \ \leadsto\ e\{k \Leftarrow E\}$$

コード生成器: $\textbf{reset0}\ \ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$

$\textbf{reset0}\ \ \underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$

$\underline{\textbf{set}}\ a\ (x, y)\ \textbf{shift0}\ k_1 \to \underline{\textbf{let}}\ u = cc1\ \underline{\textbf{in}}\ \textbf{throw}\ k_1\ u;$

$\underline{\textbf{set}}\ b\ (x, y)\ \textbf{shift0}\ k_1 \to\ \textbf{shift0}\ k_2 \to$

$\underline{\textbf{let}}\ w = cc2\ \underline{\textbf{in}}\ \textbf{throw}\ k_2(\textbf{throw}\ k_1\ w)$

生成コード: $<\ \textbf{let}\ w' = cc2'\ \textbf{in}$

$\textbf{for}\ x' = e1'\ \textbf{to}\ e2'\ \textbf{do}$

$\textbf{let}\ u' = cc1'\ \textbf{in}$

$\textbf{for}\ y' = e3'\ \textbf{to}\ e4'\ \textbf{do}$

$a[x', y'] \leftarrow u'$

$b[x', y'] \leftarrow w'\ >$

次に安全性

コード生成前の段階で，安全なコードかどうかを判断する

環境識別子 (EC) を利用したスコープ表現 [Sudo+2014]

$\gamma 0$ **for** $x = e1$ **to** $e2$ **do**

$\gamma 1$ **for** $y = e3$ **to** $e4$ **do**

$\gamma 2$ **set** $a\ (x, y)\ cc$

スコープ	使えるコード変数
$\gamma 0$	なし
$\gamma 1$	x
$\gamma 2$	x, y

環境識別子 (EC) を利用したスコープ表現 [Sudo+2014]

型システムでコード変数のスコープを表現:

$$\Gamma = \gamma 2 \geq \gamma 1,\ x : \langle \textbf{int} \rangle \hat{}\, \gamma 1,\ y : \langle \textbf{int} \rangle \hat{}\, \gamma 2$$

$\gamma 1$	$\gamma 2$
$\Gamma \vdash x : \langle \textbf{int} \rangle \hat{}\, \gamma 1$ OK	$\Gamma \vdash x : \langle \textbf{int} \rangle \hat{}\, \gamma 2$ OK
$\Gamma \vdash y : \langle \textbf{int} \rangle \hat{}\, \gamma 1$ NG	$\Gamma \vdash y : \langle \textbf{int} \rangle \hat{}\, \gamma 2$ OK
$\Gamma \vdash x \underline{+} y : \langle \textbf{int} \rangle \hat{}\, \gamma 1$ NG	$\Gamma \vdash x \underline{+} y : \langle \textbf{int} \rangle \hat{}\, \gamma 2$ OK

環境識別子 (EC) を利用したスコープ表現 [Sudo+2014]

型システムでコード変数のスコープを表現:

$$\Gamma = \gamma 2 \geq \gamma 1,\ x : \langle \textsf{int} \rangle\hat{}\gamma 1,\ y : \langle \textsf{int} \rangle\hat{}\gamma 2$$

$\gamma 1$	$\gamma 2$
$\Gamma \vdash x : \langle \textsf{int} \rangle\hat{}\gamma 1$ OK	$\Gamma \vdash x : \langle \textsf{int} \rangle\hat{}\gamma 2$ OK
$\Gamma \vdash y : \langle \textsf{int} \rangle\hat{}\gamma 1$ NG	$\Gamma \vdash y : \langle \textsf{int} \rangle\hat{}\gamma 2$ OK
$\Gamma \vdash x \underline{+} y : \langle \textsf{int} \rangle\hat{}\gamma 1$ NG	$\Gamma \vdash x \underline{+} y : \langle \textsf{int} \rangle\hat{}\gamma 2$ OK

コードレベルのラムダ抽象の型付け規則で固有変数条件を利用:

$$\frac{\Gamma,\ \gamma_2 \geq \gamma_1,\ x : \langle t_1 \rangle\hat{}\gamma_2 \vdash e : \langle t_2 \rangle\hat{}\gamma_2}{\Gamma \vdash \underline{\lambda} x.e : \langle t_1 \to t_2 \rangle\hat{}\gamma_1}\ (\gamma_2 \text{ is eigen var})$$

環境識別子 (EC) を利用したスコープ表現

先行研究:

- **局所的なスコープをもつ破壊的変数をもつコード生成の体系に対する (型安全な) 型システムの構築**
  [Sudo,Kiselyov,Kameyama 2014]
- **グローバルなスコープをもつ破壊的変数への拡張**
  [Kiselyov,Kameyama,Sudo 2016]

○ **コントロールオペレータには非対応**

環境識別子 (EC) を利用したスコープ表現

先行研究:

- **局所的なスコープをもつ破壊的変数をもつコード生成の体系に対する (型安全な) 型システムの構築** [Sudo,Kiselyov,Kameyama 2014]
- **グローバルなスコープをもつ破壊的変数への拡張** [Kiselyov,Kameyama,Sudo 2016]

○ **コントロールオペレータには非対応**

問題点:

shift0/reset0 **などのコントロールオペレータは，スコープの包含関係を逆転させてしまう．**

環境識別子(EC)の問題点

コード生成前・後でスコープの包含関係が逆転

コード生成器：

for $x = e1$ **to** $e2$ **do**

$\gamma 0$ **reset0 for** $y = e3$ **to** $e4$ **do**

$\gamma 1$ **shift0** $k \to$ **let** $u = cc$ **in**

$\gamma 2$ **throw** k

$\gamma 3$ **set** $a\ (x, y)\ u$

コード生成前・後でスコープの包含関係が逆転

コード生成器：

for $x = e1$ to $e2$ do
$\gamma 0$ reset0 for $y = e3$ to $e4$ do
$\gamma 1$ shift0 $k \to$ let $u = cc$ in
$\gamma 2$ throw k
$\gamma 3$ set $a\ (x, y)\ u$

生成コード：

for $x' = e1'$ to $e2'$ do
$\gamma 0$ let $u' = cc'$ in
$\gamma 2$ for $y' = e3'$ to $e4'$ do
$\gamma 1$
$\gamma 3$ $a[(x', y')] \leftarrow u'$

コード生成前・後でスコープの包含関係が逆転

コード生成器：

スコープ	使えるコード変数
$\gamma 0$	x
$\gamma 1$	x, y
$\gamma 2$	x, y, u
$\gamma 3$	x, y, u

生成コード：

スコープ	使えるコード変数
$\gamma 0$	x'
$\gamma 2$	x', u'
$\gamma 1$	x', y', u'
$\gamma 3$	x', y', u'

- 生成前と生成後の $\gamma 1$ と $\gamma 2$ の間には順序関係がない

コード生成前・後でスコープの包含関係が逆転

コード生成器：

スコープ	使えるコード変数
$\gamma 0$	x
$\gamma 1$	$x, y \Rightarrow x, y, u$
$\gamma 2$	$x, y, u \Rightarrow x, u$
$\gamma 3$	x, y, u

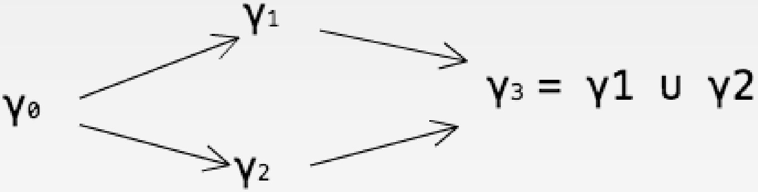
生成コード：

スコープ	使えるコード変数
$\gamma 0$	x'
$\gamma 2$	x', u'
$\gamma 1$	x', y', u'
$\gamma 3$	x', y', u'

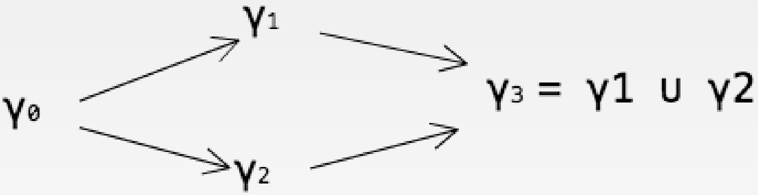
- **生成前と生成後の $\gamma 1$ と $\gamma 2$ の間には順序関係がない**

解決策

本研究の解決策 EC のジョイン

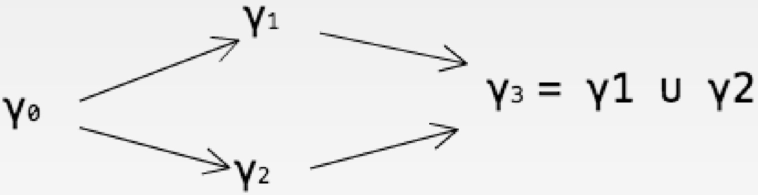


本研究の解決策 EC のジョイン



- $\gamma 1$ のコードレベル変数は $\gamma 2$ では使えない
- $\gamma 2$ のコードレベル変数は $\gamma 1$ では使えない

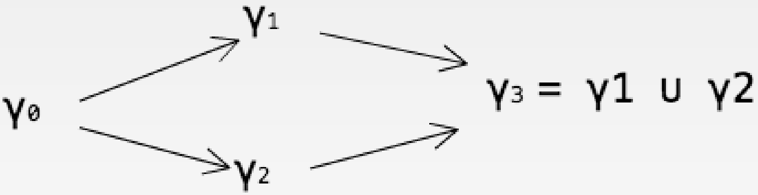
本研究の解決策 EC のジョイン



- $\gamma 1$ **のコードレベル変数は** $\gamma 2$ **では使えない**
- $\gamma 2$ **のコードレベル変数は** $\gamma 1$ **では使えない**

$\Rightarrow$ $\gamma 1$ **と** $\gamma 2$ **の間に順序を付けない**

本研究の解決策 EC のジョイン

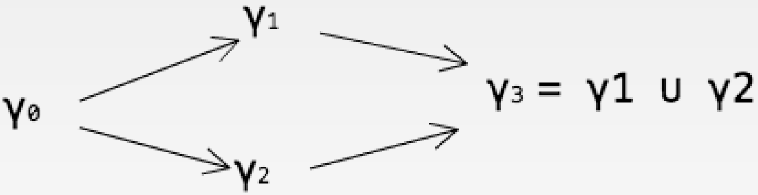


- $\gamma 1$ のコードレベル変数は $\gamma 2$ では使えない
- $\gamma 2$ のコードレベル変数は $\gamma 1$ では使えない

$\Rightarrow$ $\gamma 1$ と $\gamma 2$ の間に順序を付けない

- $\gamma 1, \gamma 2$ のコードレベル変数は $\gamma 3$ で使える

本研究の解決策 EC のジョイン



- $\gamma 1$ のコードレベル変数は $\gamma 2$ では使えない
- $\gamma 2$ のコードレベル変数は $\gamma 1$ では使えない

$\Rightarrow$ $\gamma 1$ と $\gamma 2$ の間に順序を付けない

- $\gamma 1, \gamma 2$ のコードレベル変数は $\gamma 3$ で使える

$\Rightarrow$ Sudo らの体系に $\cup$ (ユニオン) を追加

コード生成+shift0/reset0 の型システム (の一部)

reset0:

$$\frac{\Gamma \vdash e : \langle t \rangle\hat{}\gamma \;;\; \langle t \rangle\hat{}\gamma, \sigma}{\Gamma \vdash \textbf{reset0}\ e : \langle t \rangle\hat{}\gamma \;;\; \sigma}$$

shift0:

$$\frac{\Gamma,\ k : \langle t1 \rangle\hat{}\gamma 1 \Rightarrow \langle t0 \rangle\hat{}\gamma 0 \vdash e : \langle t0 \rangle\hat{}{\color{red}\gamma 0} \;;\; \sigma \quad \Gamma \models \gamma 1 \geq \gamma 0}{\Gamma \vdash \textbf{shift0}\ k \to e : \langle t1 \rangle\hat{}{\color{red}\gamma 1} \;;\; \langle t0 \rangle\hat{}\gamma 0, \sigma}$$

throw:

$$\frac{\Gamma \vdash v : \langle t1 \rangle\hat{}{\color{red}\gamma 1 \cup \gamma 2} \;;\; \sigma \quad \Gamma \models \gamma 2 \geq \gamma 0}{\Gamma,\ k : \langle t1 \rangle\hat{}\gamma 1 \Rightarrow \langle t0 \rangle\hat{}\gamma 0 \vdash \textbf{throw}\ k\ v : \langle t0 \rangle\hat{}\gamma 2 \;;\; \sigma}$$

型付けの例 (1)

$$e = \textbf{reset0}\ \ (\underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$$

$$\textbf{shift0}\ k\ \ \to\ \ \underline{\textbf{let}}\ u = \boxed{}\ \underline{\textbf{in}}\ \textbf{throw}\ k\ u)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma b \vdash u : \langle t\rangle\hat{}\gamma1 \cup \gamma2;\ \sigma}}{\Gamma b \vdash \textbf{throw}\ k\ u : \langle t\rangle\hat{}\gamma2;\ \epsilon} \quad \dfrac{\vdots}{\Gamma a \vdash \boxed{} : \langle t\rangle\hat{}\gamma0;\ \epsilon}}{\Gamma a \vdash \underline{\textbf{let}}\ u = \ldots : \langle t\rangle\hat{}\gamma0;\ \epsilon}\ (\gamma2^{*})}{\gamma1 \geq \gamma0,\ x : \langle t\rangle\hat{}\gamma1 \vdash \textbf{shift0}\ k\ \to\ \ldots : \langle t\rangle\hat{}\gamma1;\ \langle t\rangle\hat{}\gamma0}\ (\gamma1^{*})}{\dfrac{\vdash \underline{\textbf{for}}\ x = \ldots : \langle t\rangle\hat{}\gamma0;\ \langle t\rangle\hat{}\gamma0}{\vdash e : \langle t\rangle\hat{}\gamma0;\ \epsilon}}$$

$$\Gamma a = \gamma1 \geq \gamma0,\ x : \langle t\rangle\hat{}\gamma1,\ k : \langle t\rangle\hat{}\gamma_1 \Rightarrow \langle t\rangle\hat{}\gamma_0$$

$$\Gamma b = \Gamma a1,\ \gamma2 \geq \gamma0,\ u : \langle t\rangle\hat{}\gamma2$$

型付けの例 (1)

$$e = \textbf{reset0}\ \ (\underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$$

$$\textbf{shift0}\ k\ \to\ \underline{\textbf{let}}\ u = \boxed{\textcolor{green}{\underline{\textbf{int}}\ 3}}\ \underline{\textbf{in}}\ \textbf{throw}\ k\ u)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma b \vdash u : \langle t\rangle\,\hat{}\,\gamma1 \cup \gamma2;\ \sigma}}{\Gamma b \vdash \textbf{throw}\ k\ u : \langle t\rangle\,\hat{}\,\gamma2;\ \epsilon} \quad \begin{array}{c}\vdots\\ \Gamma a \vdash \boxed{\textcolor{green}{\underline{\textbf{int}}\ 3}} : \langle t\rangle\,\hat{}\,\textcolor{red}{\gamma0};\ \epsilon\end{array}}{\Gamma a \vdash \underline{\textbf{let}}\ u = \ldots : \langle t\rangle\,\hat{}\,\gamma0;\ \epsilon}\,(\gamma2^*)}{\gamma1 \geq \gamma0,\ x : \langle t\rangle\,\hat{}\,\gamma1 \vdash \textbf{shift0}\ k\ \to\ \ldots : \langle t\rangle\,\hat{}\,\gamma1;\ \langle t\rangle\,\hat{}\,\gamma0}}{\vdash \underline{\textbf{for}}\ x = \ldots : \langle t\rangle\,\hat{}\,\gamma0;\ \langle t\rangle\,\hat{}\,\gamma0}\,(\gamma1^*)}{\vdash e : \langle t\rangle\,\hat{}\,\gamma0;\ \epsilon}$$

$$\Gamma a = \gamma1 \geq \gamma0,\ x : \langle t\rangle\,\hat{}\,\textcolor{red}{\gamma1},\ k : \langle t\rangle\,\hat{}\,\gamma_1 \Rightarrow \langle t\rangle\,\hat{}\,\gamma_0$$

$$\Gamma b = \Gamma a1,\ \gamma2 \geq \gamma0,\ u : \langle t\rangle\,\hat{}\,\gamma2$$

型付けの例 (1)

$$e = \textbf{reset0}\ \ (\underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$$

$$\textbf{shift0}\ k\ \to\ \underline{\textbf{let}}\ u = \boxed{\ x \mathbin{\underline{+}} (\underline{\textbf{int}}\ 3)\ }\ \underline{\textbf{in}}\ \textbf{throw}\ k\ u)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma b \vdash u : \langle t\rangle\hat{}\gamma 1 \cup \gamma 2;\ \sigma}}{\Gamma b \vdash \textbf{throw}\ k\ u : \langle t\rangle\hat{}\gamma 2;\ \epsilon} \quad \dfrac{\vdots}{\Gamma a \vdash \boxed{\ x \mathbin{\underline{+}} (\underline{\textbf{int}}\ 3)\ } : \langle t\rangle\hat{}\gamma 0;\ \epsilon}}{\Gamma a \vdash \underline{\textbf{let}}\ u = \ldots : \langle t\rangle\hat{}\gamma 0;\ \epsilon}\ (\gamma 2^*}{\dfrac{\gamma 1 \geq \gamma 0,\ x : \langle t\rangle\hat{}\gamma 1 \vdash \textbf{shift0}\ k\ \to\ \ldots : \langle t\rangle\hat{}\gamma 1;\ \langle t\rangle\hat{}\gamma 0}{\dfrac{\vdash \underline{\textbf{for}}\ x = \ldots : \langle t\rangle\hat{}\gamma 0;\ \langle t\rangle\hat{}\gamma 0}{\vdash e : \langle t\rangle\hat{}\gamma 0;\ \epsilon}}\ (\gamma 1^*)}$$

$$\Gamma a = \gamma 1 \geq \gamma 0,\ x : \langle t\rangle\hat{}\gamma 1,\ k : \langle t\rangle\hat{}\gamma_1 \Rightarrow \langle t\rangle\hat{}\gamma_0$$

$$\Gamma b = \Gamma a 1,\ \gamma 2 \geq \gamma 0,\ u : \langle t\rangle\hat{}\gamma 2$$

型付けの例 (2)

$e' =$ **reset0** (**for** $x = e1$ **to** $e2$ **do** **reset0** (**for** $y = e3$ **to** $e4$ **do**
shift0 $k_2 \to$ **shift0** $k_1 \to$ **let** $u = X$ **in** **throw** k_1 (**throw** k_2 $e5$)))

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Gamma e \vdash e5 : \langle t\rangle\hat{}\gamma2 \cup \gamma1 \cup \gamma3;\quad \epsilon}}{\Gamma e \vdash \textbf{throw}\ k_2\ e5 : \langle t\rangle\hat{}\gamma1 \cup \gamma3;\quad \epsilon}}{\Gamma e = \Gamma d, \gamma3 \geq \gamma0, u : \langle t\rangle\hat{}\gamma3 \vdash \textbf{throw}\ k_1\ \ldots : \langle t\rangle\hat{}\gamma3;\quad \epsilon} \qquad \dfrac{\vdots}{\Gamma d \vdash X : \langle t\rangle\hat{}\gamma0; \epsilon}}{\Gamma d = \Gamma c, k_1 : \langle t\rangle\hat{}\gamma1 \Rightarrow \langle t\rangle\hat{}\gamma0 \vdash \textbf{let}\ u = \ldots : \langle t\rangle\hat{}\gamma0;\quad \epsilon}\,(\gamma3)}{\Gamma c = \Gamma b, k_2 : \langle t\rangle\hat{}\gamma2 \Rightarrow \langle t\rangle\hat{}\gamma1 \vdash \textbf{shift0}\ k_1 \ldots : \langle t\rangle\hat{}\gamma1;\quad \langle t\rangle\hat{}\gamma0}}{\Gamma b = \Gamma a, \gamma2 \geq \gamma1,\ y : \langle t\rangle\hat{}\gamma2 \vdash \textbf{shift0}\ k_2 \ldots : \langle t\rangle\hat{}\gamma2;\quad \langle t\rangle\hat{}\gamma1, \langle t\rangle\hat{}\gamma0}}{\Gamma a \vdash \textbf{for}\ y = \ldots : \langle t\rangle\hat{}\gamma1;\quad \langle t\rangle\hat{}\gamma1, \langle t\rangle\hat{}\gamma0}\,(\gamma2^*)}{\Gamma a = \gamma1 \geq \gamma0,\ x : \langle t\rangle\hat{}\gamma1 \vdash \textbf{reset0} \cdots : \langle t\rangle\hat{}\gamma1;\quad \langle t\rangle\hat{}\gamma0}}{\vdash \textbf{for}\ x = \ldots : \langle t\rangle\hat{}\gamma0;\quad \langle t\rangle\hat{}\gamma0}\,(\gamma1^*)}{\vdash e' = \textbf{reset0} \cdots : \langle t\rangle\hat{}\gamma0;\quad \epsilon}
$$

アウトライン

1. 概要
2. 研究の目的
3. 研究の内容
4. **まとめと今後の課題**

まとめと今後の課題

まとめ

- コード生成言語の型システムに shift0/reset0 を組み込んだ型システムの設計を完成させた．
- 安全なコードの場合に型が付くこと，安全でないコードの場合には型が付かないように意図通りに型システムが設計できていることをみた

今後の課題

- 設計した型システムの健全性の証明（Subject recudtion 等）
- 型推論アルゴリズムの開発
- 言語の拡張
  - グローバルな参照 (OCaml の let ref)
  - 生成したコードの実行 (MetaOCaml の run)

APPENDIX

アウトライン

5. **健全性の証明**

健全性の証明 (Subject Reduction)

型安全性 (型システムの健全性; Subject Reduction **等の性質) を厳密に証明する．**

Subject Redcution Property

$\Gamma \vdash M : \tau$ **が導ければ (プログラム** M **が型検査を通れば)，** M **を計算して得られる任意の** N **に対して，** $\Gamma \vdash N : \tau$ **が導ける (**N **も型検査を通り，** M **と同じ型，同じ自由変数を持つ)**