安全なコード移動が可能なコード生成言語の型システムの設計と実装

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

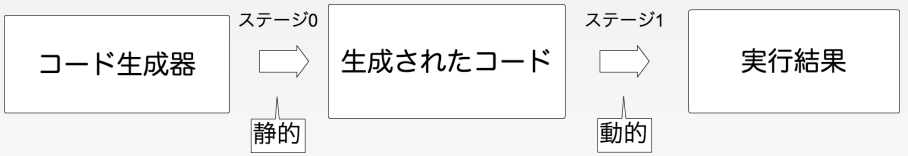
2017/1/27

筑波大学修論審査会

アウトライン

アウトライン

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
$(\underline{\textbf{int}}\ \ 3)$	$\rightsquigarrow^*$ <3>
$(\underline{\textbf{int}}\ \ 3)\ \underline{+}\ (\underline{\textbf{int}}\ \ 5)$	$\rightsquigarrow^*$ <3 + 5>
$\underline{\lambda} x.\ x\ \underline{+}\ (\underline{\textbf{int}}\ \ 3)$	$\rightsquigarrow^*$ <$\lambda x'.x' + 3$>
$\underline{\textbf{for}}\ x = \cdots\ \underline{\textbf{to}}\ \cdots\ \underline{\textbf{do}}\ \cdots$	$\rightsquigarrow^*$ <for $x' = \cdots$ to $\cdots$ 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

let 挿入 (コード移動) の実現方法

コード生成器

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

$\rightsquigarrow^*$

生成されるコード

<$\textbf{let}\ u' = cc'\ \textbf{in}$
for $x' = e1'$ to $e2'$ do
for $y' = e3'$ to $e4'$ do
$a[x', y'] \leftarrow u'$>

shift0/reset0 の導入

$\circ$ 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

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

コード生成器: $\textbf{reset0}\ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}\ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$
$\underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}\textbf{for}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$
$\underline{\textbf{set}}\ a\ (x, y)\ \underline{\textbf{set}}\ a\ (x, y)$ $cc\textbf{shift0}\ k \rightarrow$
$k \Leftarrow$ $\underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$
$\underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$
$\underline{\textbf{set}}\ a\ (x, y)\ [\ ]$

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

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

コード生成器: $\textbf{reset0}\ \underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$
$\textbf{reset0}\ \underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$
$\underline{\textbf{set}}\ a\ (x, y)\ \textbf{shift0}\ k_1 \rightarrow \underline{\textbf{let}}\ u = cc1\ \underline{\textbf{in}}\ \textbf{throw}\ k_1\ u;$
$\underline{\textbf{set}}\ b\ (x, y)\ \textbf{shift0}\ k_1 \rightarrow\ \textbf{shift0}\ k_2 \rightarrow$
$\underline{\textbf{let}}\ w = cc2\ \underline{\textbf{in}}\ \textbf{throw}\ k_2(\textbf{throw}\ k_1\ w)$

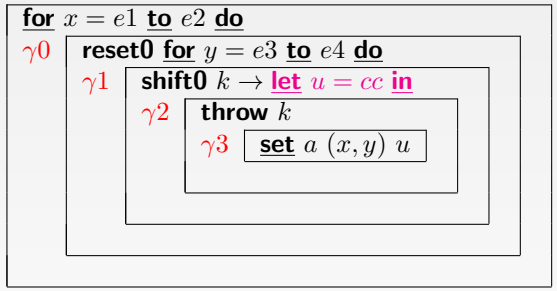
生成コード: $< \textbf{let}\ w' = cc2'\ \textbf{in}$
$\textbf{for}\ x' = e1'\ \textbf{to}\ e2'\ \textbf{do}$
$\textbf{let}\ u' = cc1'\ \textbf{in}$
$\textbf{for}\ y' = e3'\ \textbf{to}\ e4'\ \textbf{do}$
$a[x', y'] \leftarrow u'$
$b[x', y'] \leftarrow w' >$

アウトライン

問題点

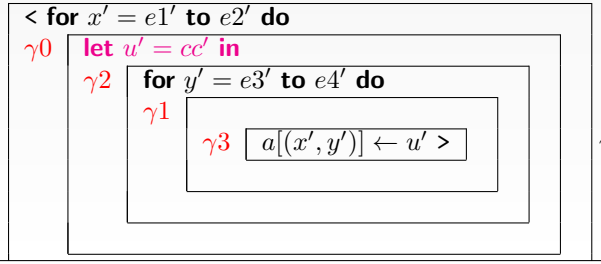
コード生成前・後でスコープの包含関係が逆転

コード生成器:



$\gamma 3 \geq \gamma 2 \geq \gamma$

生成コード:



$\gamma 3 \geq \gamma 1 \geq \gamma 2$

アウトライン

研究の目的，概要

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

概要

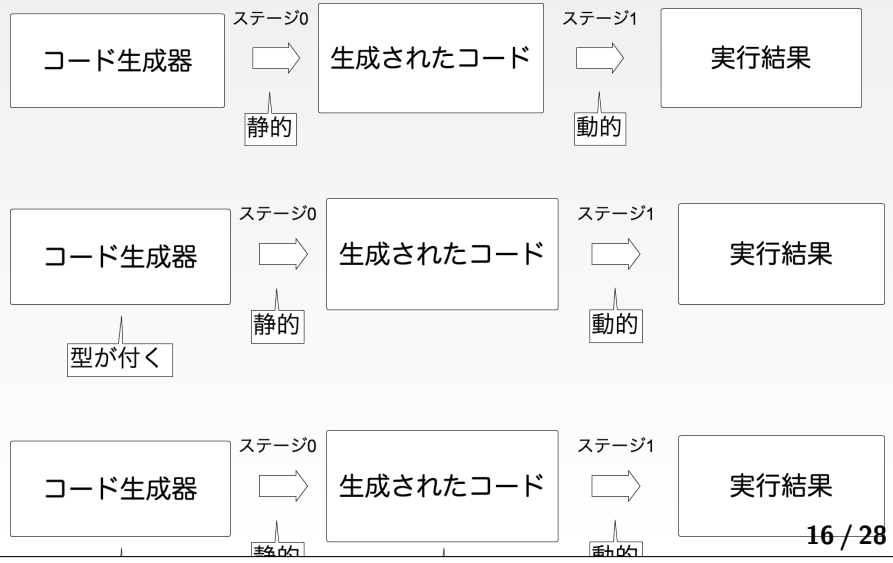
本研究: 簡潔で強力なコントロールオペレータに基づくコード生成体系の構築

- 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

解決策

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



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

$\gamma 2 \geq \gamma 1 \geq \gamma 0$

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

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

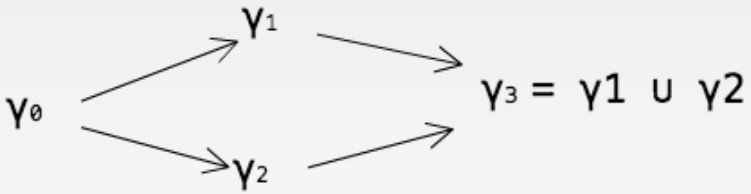
先行研究:

- **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**

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.**

本研究の解決策



- $\gamma 1$ **のコード変数は** $\gamma 2$ **では使ってはいけない**
- $\gamma 2$ **のコード変数は** $\gamma 1$ **では使ってはいけない**

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

- $\gamma 1, \gamma 2$ **のコード変数は** $\gamma 3$ **で使ってよい**

⇒ 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:

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

throw:

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

型付けの例 (1)

$e = \textbf{reset0}\ \ (\underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}$
$\textbf{shift0}\ k\ \rightarrow\ \underline{\textbf{let}}\ u =$ **int** 3 x + (**int** 3) $\underline{\textbf{in}}\ \textbf{throw}\ k\ u)$

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

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

型付けの例 (2)

$e' =$**reset0** $(\underline{\textbf{for}}\ x = e1\ \underline{\textbf{to}}\ e2\ \underline{\textbf{do}}\ \textbf{reset0}\ (\underline{\textbf{for}}\ y = e3\ \underline{\textbf{to}}\ e4\ \underline{\textbf{do}}$
$\textbf{shift0}\ k_2 \rightarrow\ \textbf{shift0}\ k_1 \rightarrow\ \underline{\textbf{let}}\ u = \square\ \underline{\textbf{in}}\ \textbf{throw}\ k_1\ (\textbf{throw}\ k_2\ e5)))$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Gamma e \vdash e5 : \langle t \rangle\hat{}\gamma 2 \cup \gamma 1 \cup \gamma 3;\quad \epsilon}}{\Gamma e \vdash \textbf{throw}\ k_2\ e5 : \langle t \rangle\hat{}\gamma 1 \cup \gamma 3;\quad \epsilon} \quad \dfrac{\vdots}{\Gamma d \vdash \square : \langle t \rangle\hat{}\gamma 0; \epsilon}}{\Gamma e = \Gamma d, \gamma 3 \geq \gamma 0, u : \langle t \rangle\hat{}\gamma 3 \vdash \textbf{throw}\ k_1\ \ldots : \langle t \rangle\hat{}\gamma 3;\ \epsilon}\ (\gamma 3^*)}{\Gamma d = \Gamma c, k_1 : \langle t \rangle\hat{}\gamma 1 \Rightarrow \langle t \rangle\hat{}\gamma 0 \vdash \underline{\textbf{let}}\ u = \ldots : \langle t \rangle\hat{}\gamma 0;\ \epsilon}}{\Gamma c = \Gamma b, k_2 : \langle t \rangle\hat{}\gamma 2 \Rightarrow \langle t \rangle\hat{}\gamma 1 \vdash \textbf{shift0}\ k_1 \ldots : \langle t \rangle\hat{}\gamma 1;\quad \langle t \rangle\hat{}\gamma 0}}{\Gamma b = \Gamma a, \gamma 2 \geq \gamma 1,\ y : \langle t \rangle\hat{}\gamma 2 \vdash \textbf{shift0}\ k_2 \ldots : \langle t \rangle\hat{}\gamma 2;\quad \langle t \rangle\hat{}\gamma 1, \langle t \rangle\hat{}\gamma 0}\ (\gamma 2^*)}{\Gamma a \vdash \underline{\textbf{for}}\ y = \ldots : \langle t \rangle\hat{}\gamma 1;\quad \langle t \rangle\hat{}\gamma 1, \langle t \rangle\hat{}\gamma 0}}{\Gamma a = \gamma 1 \geq \gamma 0,\ x : \langle t \rangle\hat{}\gamma 1 \vdash \textbf{reset0} \cdots : \langle t \rangle\hat{}\gamma 1;\quad \langle t \rangle\hat{}\gamma 0}\ (\gamma 1^*)}{\vdash \underline{\textbf{for}}\ x = \ldots : \langle t \rangle\hat{}\gamma 0;\quad \langle t \rangle\hat{}\gamma 0}}{\vdash e' = \textbf{reset0} \cdots : \langle t \rangle\hat{}\gamma 0;\quad \epsilon}$$

型推論アルゴリズム

- 制約生成
- 制約解消

制約生成

こういう制約が出てくる

制約解消

先の制約を解消して，型を決定する

アウトライン

まとめと今後の課題

まとめ

- コード生成言語にコード移動を許す仕組み (shift0/reset0) を導入し，その安全性を保証するための型システムの設計を行い
  - 安全性：Scope extrusion が起きないようにする
- 型推論アルゴリズムの開発を行った

今後の課題

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