

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

指導教員 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

- ① 目的
- ② 準備
- ③ 問題点
- ④ 解決策
- ⑤ まとめと今後の課題

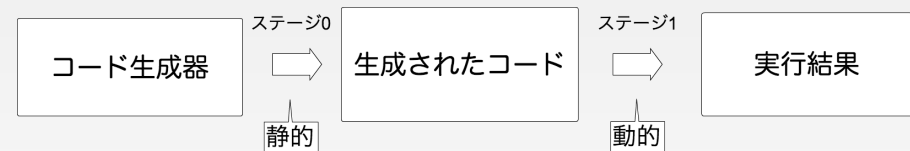
2 / 24

アウトライン

- ① 目的
- ② 準備
- ③ 問題点
- ④ 解決策
- ⑤ まとめと今後の課題

3 / 24

目的



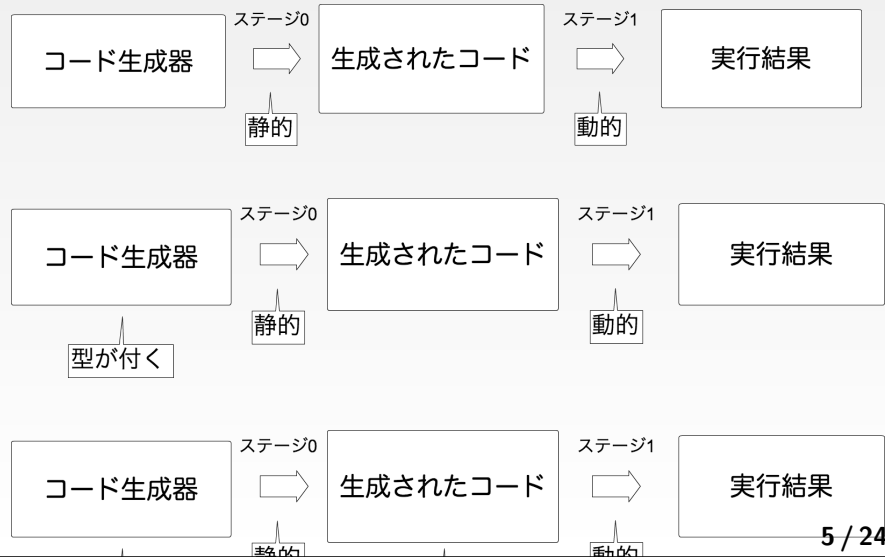
- コード生成をサポートするプログラム言語
(= **コード生成言語**)

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

4 / 24

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

- ① 目的
- ② 準備
- ③ 問題点
- ④ 解決策
- ⑤ まとめと今後の課題

6 / 24

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

$(\text{int } 3) \rightsquigarrow^* \langle 3 \rangle$

$(\text{int } 3) + (\text{int } 5) \rightsquigarrow^* \langle 3 + 5 \rangle$

$\lambda x. x + (\text{int } 3) \rightsquigarrow^* \langle \lambda x'. x' + 3 \rangle$

$\text{for } x = \dots \text{ to } \dots \text{ do } \dots \rightsquigarrow^* \langle \text{for } x' = \dots \text{ to } \dots \text{ do } \dots \rangle$

コードコンビネータ

- 下線付きの演算子
- コードを引数にとり, コードを返す

7 / 24

let 挿入 (コード移動) の実現方法

コード生成器

○ $\text{for } x = e1 \text{ to } e2 \text{ do}$

○ $\text{for } y = e3 \text{ to } e4 \text{ do}$

$\text{set } \langle a \rangle (x, y)$ ○

$\text{let } u = cc \text{ in } u$

生成されるコード

$\langle \text{let } u' = cc' \text{ in}$

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

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

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

shift0/reset0 の導入

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

8 / 24

アウトライン

- ① 目的
- ② 準備
- ③ 問題点
- ④ 解決策
- ⑤ まとめと今後の課題

9 / 24

コード生成前後でコードが移動する

コード生成器

生成されるコード

for $x = e1$ to $e2$ do

reset0 for $y = e3$ to $e4$ do \rightsquigarrow^*

shift0 $k \rightarrow$ **let** $u = y$ **in**

throw k **set** $a(x, y) u$

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

let $u' = y'$ **in**

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

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

Scope Extrusion

(コード移動により) 意図した束縛から、変数が抜け出てしまうこと

10 / 24

アウトライン

- ① 目的
- ② 準備
- ③ 問題点
- ④ 解決策
- ⑤ まとめと今後の課題

11 / 24

解決策

12 / 24

環境識別子 (EC) を利用したスコープ表現 [Sudo+2014]

```

 $\gamma_0$  for  $x = e_1$  to  $e_2$  do
   $\gamma_1$  for  $y = e_3$  to  $e_4$  do
     $\gamma_2$  set  $a(x, y)$   $cc$ 
    
```

スコープ	使えるコード変数
γ_0	なし
γ_1	x
γ_2	x, y

$$\gamma_2 \geq \gamma_1 \geq \gamma_0$$

13 / 24

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。

14 / 24

コード生成+shift0/reset0 の型システム (の一部)

reset0:

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

shift0:

$$\frac{\Gamma, k : \langle t_1 \rangle^{\gamma_1} \Rightarrow \langle t_0 \rangle^{\gamma_0} \vdash e : \langle t_0 \rangle^{\gamma_0} ; \sigma \quad \Gamma \models \gamma_1 \geq \gamma_0}{\Gamma \vdash \text{shift0 } k \rightarrow e : \langle t_1 \rangle^{\gamma_1} ; \langle t_0 \rangle^{\gamma_0}, \sigma}$$

throw:

$$\frac{\Gamma \vdash v : \langle t_1 \rangle^{\gamma_1} \cup \gamma_2 ; \sigma \quad \Gamma \models \gamma_2 \geq \gamma_0}{\Gamma, k : \langle t_1 \rangle^{\gamma_1} \Rightarrow \langle t_0 \rangle^{\gamma_0} \vdash \text{throw } k v : \langle t_0 \rangle^{\gamma_2} ; \sigma}$$

15 / 24

型付けの例 (1)

$e = \text{reset0 } (\text{for } x = e_1 \text{ to } e_2 \text{ do}$

$\text{shift0 } k \rightarrow \text{let } u = \boxed{\text{int } 3 \text{ } x \text{ } + \text{ (int } 3)} \text{ in throw } k u)$

$$\frac{\frac{\Gamma b \vdash u : \langle t \rangle^{\gamma_1} \cup \gamma_2 ; \sigma}{\Gamma b \vdash \text{throw } k u : \langle t \rangle^{\gamma_2} ; \epsilon} \quad \vdots \quad \Gamma a \vdash \boxed{\text{int } 3 \text{ } x \text{ } + \text{ (int } 3)} : \langle t \rangle^{\gamma_0} ; \epsilon}{\frac{\Gamma a \vdash \text{let } u = \dots : \langle t \rangle^{\gamma_0} ; \epsilon}{\gamma_1 \geq \gamma_0, x : \langle t \rangle^{\gamma_1} \vdash \text{shift0 } k \rightarrow \dots : \langle t \rangle^{\gamma_1} ; \langle t \rangle^{\gamma_0}} \quad (\gamma_1^*)}{\vdash \text{for } x = \dots : \langle t \rangle^{\gamma_0} ; \langle t \rangle^{\gamma_0} \vdash e : \langle t \rangle^{\gamma_0} ; \epsilon}$$

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

$$\Gamma b = \Gamma a, \gamma_2 \geq \gamma_0, u : \langle t \rangle^{\gamma_2}$$

16 / 24

型付けの例 (2)

$$\begin{array}{c}
 e' = \text{reset0} \text{ (for } x = e1 \text{ to } e2 \text{ do reset0 (for } y = e3 \text{ to } e4 \text{ do} \\
 \quad \text{shift0 } k_2 \rightarrow \text{shift0 } k_1 \rightarrow \text{let } u = \boxed{x \ y} \text{ in throw } k_1 \text{ (throw } k_2 \text{ } e5))) \\
 \vdots \\
 \frac{\Gamma e \vdash e5 : \langle t \rangle^{\gamma_2} \cup \gamma_1 \cup \gamma_3; \quad \epsilon}{\Gamma e \vdash \text{throw } k_2 \text{ } e5 : \langle t \rangle^{\gamma_1} \cup \gamma_3; \quad \epsilon} \quad \vdots \\
 \frac{\Gamma e = \Gamma d, \gamma_3 \geq \gamma_0, u : \langle t \rangle^{\gamma_3} \vdash \text{throw } k_1 \dots : \langle t \rangle^{\gamma_3}; \quad \epsilon \quad \Gamma d \vdash \boxed{x \ y} : \langle t \rangle^{\gamma_0}; \quad \epsilon}{\Gamma d = \Gamma c, k_1 : \langle t \rangle^{\gamma_1} \Rightarrow \langle t \rangle^{\gamma_0} \vdash \text{let } u = \dots : \langle t \rangle^{\gamma_0}; \quad \epsilon} \\
 \frac{\Gamma c = \Gamma b, k_2 : \langle t \rangle^{\gamma_2} \Rightarrow \langle t \rangle^{\gamma_1} \vdash \text{shift0 } k_1 \dots : \langle t \rangle^{\gamma_1}; \quad \langle t \rangle^{\gamma_0}}{\Gamma b = \Gamma a, \gamma_2 \geq \gamma_1, y : \langle t \rangle^{\gamma_2} \vdash \text{shift0 } k_2 \dots : \langle t \rangle^{\gamma_2}; \quad \langle t \rangle^{\gamma_1}, \langle t \rangle^{\gamma_0}} \quad (\gamma_2^*) \\
 \frac{\Gamma a \vdash \text{for } y = \dots : \langle t \rangle^{\gamma_1}; \quad \langle t \rangle^{\gamma_1}, \langle t \rangle^{\gamma_0}}{\Gamma a = \gamma_1 \geq \gamma_0, x : \langle t \rangle^{\gamma_1} \vdash \text{reset0} \dots : \langle t \rangle^{\gamma_1}; \quad \langle t \rangle^{\gamma_0}} \quad (\gamma_1^*) \\
 \frac{\vdash \text{for } x = \dots : \langle t \rangle^{\gamma_0}; \quad \langle t \rangle^{\gamma_0}}{\vdash e' = \text{reset0} \dots : \langle t \rangle^{\gamma_0}; \quad \epsilon}
 \end{array}$$

$\Gamma d = \dots, x : \langle t \rangle^{\gamma_1}, y : \langle t \rangle^{\gamma_2}, \gamma_1 \geq \gamma_0, \gamma_2 \geq \gamma_1, \dots$ 17 / 24

型推論アルゴリズム

18 / 24

型推論アルゴリズム

Γ, L, σ, e が与えられたとき, $\Gamma \vdash^L e : t; \sigma$ が成立するような t があるかどうか判定し, その型 t を返す

制約生成

与えられた項に対して, 型, EC, エフェクトに関する制約を返す

制約解消

その得られた制約を解消し, その制約を満たす代入 Θ を返す

19 / 24

制約生成

制約生成用の型システム T_2 の導入

subsumption 規則をあらゆる規則に付加させて型推論用 (制約を生成する) の型システムを作成. 型付け規則を一意に適用できるようにした

型に関する順序 $t_1 \geq t_2$ の導入

制約生成時において, コード型か普通の型か判断することができないためその 2 つを同時に表す \geq を導入した

20 / 24

制約解消

生成された制約 $\Delta \models C$
 仮定 Δ

EC に対する順序

$$d \geq e$$

制約 C

型	$t0 = t1 \quad t0 \geq t1$
EC	$\gamma0 = \gamma1 \quad \gamma0 \geq \gamma1$
エフェクト (型の列)	$\sigma0 = \sigma1$

制約に対する解の存在判定

型に対する単一化等をおこなう

ここでは、EC の不等式制約の解消について説明をする

21 / 24

制約解消：EC の不等式制約の解消

この時点で残る制約 $\Delta \models C$

仮定 Δ

$d \geq e$ の有限集合 (d は EC 定数)

制約 C

$e1 \geq e2$ の有限集合

$e, e1, e2, \dots$ EC を表す式

$e ::= d \mid x \mid e \cup e$

制約解消アルゴリズム (の一部)

$d1 \geq d1 \implies \Delta$ を使って判定

$e1 \geq e2 \cup e3 \implies e1 \geq e2$ かつ $e1 \geq e3$

$e1 \cup e2 \geq d \implies e1 \geq d$ または $e2 \geq d$

変数 x の除去 $[x := d1 \cup d2 \cup y]$

$$\begin{array}{llll} e1 \geq x & x \geq d1 & e1 \geq d1 & e2 \geq d1 \\ e2 \geq x & x \geq d2 & \implies e1 \geq d2 & e2 \geq d2 \\ & x \geq y & e1 \geq y & e2 \geq y \end{array}$$

22 / 24

アウトライン

- ① 目的
- ② 準備
- ③ 問題点
- ④ 解決策
- ⑤ まとめと今後の課題

23 / 24

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 (実装については制約生成まで)

今後の課題

- 設計した型システムの健全性の証明 (Subject reduction) の完成
- 型推論アルゴリズム (制約解消) の実装の完成

24 / 24