

# 全体ゼミ

## コード生成 + shift0/reset0 の体系

筑波大学 プログラム論理研究室  
大石 純平

平成 28 年 5 月 31 日

### 1 モチベーション

効率のよいコードを生成するために表現力を上げたい。ここで表現力をあげるために、プログラムに副作用を許すプリミティブを導入する。限定継続のうちのひとつ、s0/r0 を用いる。

### 2 s0/r0 とコード、それぞれの体系の説明

#### 2.1 s0/r0 の operation semantics 例を用いて説明

$$\langle_0 E[S_0 k.e] \lambda \rangle_0 \rightarrow e\{k := \lambda x. \langle_0 E[x] \rangle_0\} \quad (1)$$

$$\langle_0 v \rangle_0 \rightarrow v \quad (2)$$

#### 2.2 s/r との比較

$$\langle_1 E[S_1 k.e] \lambda \rangle_1 \rightarrow \langle_1 e\{k := \lambda x. \langle_1 E[x] \rangle_1\} \rangle_1 \quad (3)$$

$$\langle_1 v \rangle_1 \rightarrow v \quad (4)$$

#### 2.3 コードの operation semantics 例を用いて説明

コードの表現方法として、MetaOCaml のような、ブラケットとエスケープを用いるのではなく、コードコンビネータを使用している。なので、演算子や項などを構成する要素すべてに対して、通常レベルとコードレベルとの二種類のものを用意する必要がある。

#### 2.4 コードの中に s0/r0 がある場合とかの例を考える

#### 2.5 困ること

Scope extrusion

- 副作用とコード生成を組み合わせた時に発生する scope extrusion は、未束縛の変数が、元々の束縛されていた範囲を飛び出してしまうという問題

### 3 型システム

Scope extrusion が起きるのを型システムで制限する

Idea

- 副作用とコード生成を組み合わせた時に発生する scope extrusion は、未束縛の変数が、元々の束縛されていた範囲を飛び出してしまうという問題を、コードレベルのラムダ抽象  $\lambda x.e$  の  $x$  の方が固変数条件を満たさなければならないという制約によって解決する。
- 変数のスコープの範囲を型によって表す。

### 4 s0/r0 の型システム

#### 4.1 s0/r0 の型システムのダウンサイジングについて

### 5 コードの型システム

### 6 s0/r0 + コードの型システム