

『プログラム言語論』 資料

(2013/05/14, 亀山)

以前に定義した「小さなプログラム言語 (L_0)」に、いくつかの機能を追加して、それに対する意味論 (抽象機械による意味論のみ) を見てみよう。

- L_0 に、変数、変数に対する束縛 (let 式) を追加した言語 L_1 : この言語に対する抽象機械は「変数がどういう値を持っているか」の情報が必要であり、これを環境 E であらわす。(CEK 機械)
- L_1 に、 λ 抽象、関数適用を追加した言語 L_2 : この言語に対する抽象機械のコンポーネントは C,E,K のままでよいが、関数を実行時データとして表すための「関数クロージャ」が必要がある。

なお、将来、C 言語の $x = x + 1;$ のように、変数の値が変更できる機能を追加する。

1 構文の定義

言語 L_0 の具体構文 (再掲、曖昧さを持つもの):

$$n ::= 0 \mid 1 \mid 2 \mid \dots \quad (\text{自然数定数})$$
$$s ::= n \mid (s) \mid s + s \mid s * s \quad (\text{式})$$

言語 L_1, L_2 の具体構文:

$$s_1 ::= \dots \mid x \mid \text{let } x = s_1 \text{ in } s_1$$
$$s_2 ::= \dots \mid x \mid \text{let } x = s_2 \text{ in } s_2 \mid \lambda x. s_2 \mid s_2 s_2$$

言語 L_1, L_2 の抽象構文:

$$e_1 ::= \dots \mid x \mid \text{Let}(x, e_1, e_1)$$
$$e_2 ::= \dots \mid x \mid \text{Let}(x, e_2, e_2) \mid \text{Lam}(x, e_2) \mid \text{App}(e_2, e_2)$$

2 抽象機械

CK 機械に環境 (E) を加えた CEK 機械を見てみよう。

- C ... control string (その抽象機械の機械語で書かれたプログラム; ここでは「(抽象構文の) 式」)
- E ... environment (「環境」, 変数がどういう値を持つか決める)
- K ... continuation (一般には「継続」と呼ばれるものだが、ここでは単にスタックを表す)

実装上は、E と K は、まとめて1つのスタックに積まれることが多いが、ここでは理解のしやすさのため、E と K に分けて考える。

抽象機械の状態: $\langle eval, C, E, K \rangle$ と $\langle apply, K, v \rangle$ の2種類の状態がある。

値 (value) を以下で定義する。

$$v_1 ::= n$$
$$v_2 ::= n \mid \text{Closure}(x, e_2, E)$$

v_1 は L_1 に対応する機械での値 v であり、 v_2 は L_2 に対応する機械での値 v である。 $\text{Closure}(x, e_2, E)$ が関数クロージャを表し、関数の引数が x 、本体が e_2 で、その本体を評価するときの環境が E であるものである。

環境 E に対する操作:

- $[]$: 空の環境 (どの変数も値を束縛されていない環境)。
- $E[x = v]$: 環境 E に、 $x = v$ という束縛を追加した環境。
- $\text{lookup}(x, E)$: 環境 E における変数 x の値。

L_1 に対する抽象機械の動作 (状態遷移):

$$\begin{aligned} \langle \text{eval}, \text{Var}(x), E, K \rangle &\rightarrow \langle \text{apply}, K, v \rangle \quad (v = \text{lookup}(x, E)) \\ \langle \text{eval}, \text{Int}(n), E, K \rangle &\rightarrow \langle \text{apply}, K, n \rangle \\ \langle \text{eval}, \text{Plus}(e, e'), E, K \rangle &\rightarrow \langle \text{eval}, e, E, \text{push}((\text{plus1}, e', E), K) \rangle \\ \langle \text{eval}, \text{Times}(e, e'), E, K \rangle &\rightarrow \langle \text{eval}, e, E, \text{push}((\text{times1}, e', E), K) \rangle \\ \langle \text{eval}, \text{Let}(x, e, e'), E, K \rangle &\rightarrow \langle \text{eval}, e, E, \text{push}((\text{letin}, x, e', E), K) \rangle \end{aligned}$$

$$\begin{aligned} \langle \text{apply}, \text{push}((\text{plus1}, e, E), K), v \rangle &\rightarrow \langle \text{eval}, e, E, \text{push}((\text{plus2}, v), K) \rangle \\ \langle \text{apply}, \text{push}((\text{plus2}, n_1), K), n_2 \rangle &\rightarrow \langle \text{apply}, K, n \rangle \quad (n_1 + n_2 = n) \\ \langle \text{apply}, \text{push}((\text{times1}, e, E), K), v \rangle &\rightarrow \langle \text{eval}, e, E, \text{push}((\text{times2}, v), K) \rangle \\ \langle \text{apply}, \text{push}((\text{times2}, n_1), K), n_2 \rangle &\rightarrow \langle \text{apply}, K, n \rangle \quad (n_1 * n_2 = n) \\ \langle \text{apply}, \text{push}((\text{letin}, x, e, E), K), v \rangle &\rightarrow \langle \text{eval}, e, E[x = v], K \rangle \\ \langle \text{apply}, \text{init}, v \rangle &\rightarrow v \quad (\text{最終結果}) \end{aligned}$$

計算は $\langle \text{eval}, e, [], \text{init} \rangle$ という状態からはじめる。

L_2 に対する抽象機械の動作 (追加分のみ):

$$\begin{aligned} \langle \text{eval}, \text{Lam}(x, e), E, K \rangle &\rightarrow \langle \text{apply}, K, \text{Closure}(x, e, E) \rangle \\ \langle \text{eval}, \text{App}(e, e'), E, K \rangle &\rightarrow \langle \text{eval}, e, E, \text{push}((\text{apply1}, e', E), K) \rangle \\ \langle \text{apply}, \text{push}((\text{apply1}, e, E), K), v \rangle &\rightarrow \langle \text{eval}, e, E, \text{push}((\text{apply2}, v), K) \rangle \\ \langle \text{apply}, \text{push}((\text{apply2}, \text{Closure}(x, e, E)), K), v \rangle &\rightarrow \langle \text{eval}, e, E[x = v], K \rangle \end{aligned}$$

計算は $\langle \text{eval}, e, [], \text{init} \rangle$ という状態からはじめる。

3 宿題

(この問題で、印刷・配布したバージョンで「 L_2 に対応する CEK 機械で、、、」と書いてあったのは、「 L_1 に対応する CEK 機械で、、、」の間違いです。実際には、 L_2 は L_1 を含むので、 L_2 と思っても、全く支障はないのですが、この段階では、 L_2 全体に対する CEK 機械 (特にクロージャ処理) を理解しなくてよいところだったので、誤植によって、解くのが大変になった人も多いかもしれません。お詫びします。)

L_1 に対応する CEK 機械で、以下のものの状態遷移を書きなさい。(計算は $\langle \text{eval}, e, [], \text{init} \rangle$ という状態からはじめる。また、ステップ数が非常に多い場合は、多少省略して書いてもよい。)

- e が $\text{let } x = 1 \text{ in } x + 2 * 3$ (具体構文での記述) のとき。
- e が $(\text{let } x = 1 \text{ in let } y = x + 4 \text{ in } x * y) * 3$ (具体構文での記述) のとき。
- e が $(\text{let } x = 1 \text{ in let } x = x + 4 \text{ in } x * 3) * 3$ (具体構文での記述) のとき。