Functions
(the first draft)

Dmitry Boulytchev

November 2, 2018

1 Functions in Expressions

At the syntax level function calls are introduced with the following construct:

$$\mathscr{E} \quad += \quad \mathscr{X}\,\mathscr{E}^*$$

In the concrete syntax function call looks conventional:

$$f\ \ (e_1, e_2, \ldots, e_k)$$

where f — a function name, e_i — its actual parameters.

Surpisingly, such a mild extension results in a complete redefinition of the semantics. Indeed, as function body can perform arbitrary actions on state, input and output streams, expressions with function calls can now have side effects. In order to express these side effects we need to redefine the semantics completely, this time using big-step operational style.

We extend the configuration, used in the semantic description for statements, with a fourth component — an optional integer value:

$$\mathscr{C} = \Sigma \times \mathbb{Z}^* \times \mathbb{Z}^* \times \mathbb{Z}^?$$

This component will correspond to an optional return value for function/procedure calls (either integer value n or "—", nothing).

The rules inself are summarized in the Fig. 1. Note the use of double environment for evaluating the body of function in the rule Call; note also, that now semantics of expressions and statements are mutually recursive.

2 Return Statement

In order to make it possible to return values from procedures we add a return statement to the language of statements:

$$\mathscr{S} += \text{return}\ \mathscr{E}^?$$

$$\Phi \vdash \langle \sigma, i, o, \_ \rangle \xrightarrow{n}_{\mathscr{E}} \langle \sigma, i, o, n \rangle \qquad [\text{Const}]$$

$$\Phi \vdash \langle \sigma, i, o, \_ \rangle \xrightarrow{x}_{\mathscr{E}} \langle \sigma, i, o, \sigma\, x \rangle \qquad [\text{Var}]$$

$$\frac{\Phi \vdash c \xrightarrow{A}_{\mathscr{E}} c' = \langle \_, \_, \_, a \rangle,\ \Phi \vdash c' \xrightarrow{B}_{\mathscr{E}} \langle \sigma'', i'', o'', b \rangle}{\Phi \vdash c \xrightarrow{A \otimes B}_{\mathscr{E}} \langle \sigma'', i'', o'', a \oplus b \rangle} \qquad [\text{Binop}]$$

$$\frac{\begin{array}{c}\Phi \vdash c_{j-1} \xrightarrow{e_j}_{\mathscr{E}} c_j = \langle \sigma_j, i_j, o_j, v_j \rangle, \\ \Phi\, f = \text{fun}\ f\ (\overline{a})\ \text{local}\ \overline{l}\ \{s\}, \\ \text{skip}, \Phi \vdash \langle \text{enter}\ \sigma_k\ (\overline{a}@\overline{l})\ [\overline{a_j \leftarrow v_j}], i_k, o_k, \_ \rangle \xrightarrow{s} \langle \sigma', i', o', n \rangle\end{array}}{\Phi \vdash c_0 = \langle \sigma_0, \_, \_, \_ \rangle \xrightarrow{f(\overline{e_k})}_{\mathscr{E}} \langle \text{leave}\ \sigma'\ \sigma_0, i', o', n \rangle} \qquad [\text{Call}]$$

Figure 1: Big-step Operational Semantics for Expressions

And, again, this small addition leads to redefinition of the semantics in continuation-passing style (CPS).

First, we define the following meta-operator "$\diamond$" on statements:

$$\begin{array}{rcl} s \diamond \text{skip} & = & s \\ s_1 \diamond s_2 & = & s_1; s_2 \end{array}$$

Then, we add another environment component, K, in the description of semantic relation "$\ldots \vdash \ldots \xrightarrow{\cdots} \ldots$". Informally speaking, now

$$K, \Phi \vdash c \xrightarrow{s} c'$$

is read "the execution of s, immediately followed by K, in the configuration c results in the configuration c'". Statement K is called continuation. The rules themselves are shown on Fig. 2. Note, the rule for the call statement is exactly the same, as for the call expression.

$$\mathrm{skip}, \Phi \vdash c \xrightarrow{\mathrm{skip}} c \qquad [\textsc{SkipSkip}]$$

$$\frac{\mathrm{skip}, \Phi \vdash c \xrightarrow{K} c'}{K, \Phi \vdash c \xrightarrow{\mathrm{skip}} c'} \qquad [\textsc{Skip}]$$

$$\frac{\Phi \vdash c \xrightarrow{e}_{\mathscr{E}} \langle \sigma, i, o, n \rangle;\ \mathrm{skip}, \Phi \vdash \langle \sigma[x \leftarrow n], i, o, — \rangle \xrightarrow{K} c'}{K, \Phi \vdash c \xrightarrow{x := e} c'} \qquad [\textsc{Assign}]$$

$$\frac{\Phi \vdash c \xrightarrow{e}_{\mathscr{E}} \langle \sigma, i, o, n \rangle;\ \mathrm{skip}, \Phi \vdash \langle \sigma, i, o@[n], — \rangle \xrightarrow{K} c'}{K, \Phi \vdash c \xrightarrow{\mathrm{write}\ \ (e)} c'} \qquad [\textsc{Write}]$$

$$\frac{\mathrm{skip}, \Phi \vdash \langle \sigma[x \leftarrow z], i, o, — \rangle \xrightarrow{K} c'}{K, \Phi \vdash \langle \sigma, z : i, o, — \rangle \xrightarrow{\mathrm{read}\ \ (x)} c'} \qquad [\textsc{Read}]$$

$$\frac{s_2 \diamond K, \Phi \vdash c \xrightarrow{s_1} c'}{K, \Phi \vdash c \xrightarrow{s_1; s_2} c'} \qquad [\textsc{Seq}]$$

$$\frac{\Phi \vdash c \xrightarrow{e}_{\mathscr{E}} \langle \sigma, i, o, n \rangle;\ K, \Phi \vdash \langle \sigma, i, o, — \rangle \xrightarrow{s_1} c'}{K, \Phi \vdash c \xrightarrow{\mathrm{if}\ e\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2} c'},\ n \neq 0 \qquad [\textsc{IfTrue}]$$

$$\frac{\Phi \vdash c \xrightarrow{e}_{\mathscr{E}} \langle \sigma, i, o, n \rangle;\ K, \Phi \vdash \langle \sigma, i, o, — \rangle \xrightarrow{s_2} c'}{K, \Phi \vdash c \xrightarrow{\mathrm{if}\ e\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2} c'},\ n = 0 \qquad [\textsc{IfFalse}]$$

$$\frac{\begin{array}{c}\Phi \vdash c \xrightarrow{e}_{\mathscr{E}} \langle \sigma, i, o, n \rangle \\ \mathrm{while}\ e\ \mathrm{do}\ s \diamond K, \Phi \vdash \langle \sigma, i, o, — \rangle \xrightarrow{s} c'\end{array}}{K, \Phi \vdash c \xrightarrow{\mathrm{while}\ e\ \mathrm{do}\ s} c'},\ n \neq 0 \qquad [\textsc{WhileTrue}]$$

$$\frac{\Phi \vdash c \xrightarrow{e}_{\mathscr{E}} \langle \sigma, i, o, n \rangle;\ \mathrm{skip}, \Phi \vdash \langle \sigma, i, o, — \rangle \xrightarrow{K} c'}{K, \Phi \vdash c \xrightarrow{\mathrm{while}\ e\ \mathrm{do}\ s} c'},\ n = 0 \qquad [\textsc{WhileFalse}]$$

$$\frac{\begin{array}{c}\Phi \vdash c_{j-1} \xrightarrow{e_j}_{\mathscr{E}} c_j = \langle \sigma_j, i_j, o_j, v_j \rangle \\ \Phi\ f = \mathrm{fun}\ f\ (\overline{a})\ \mathrm{local}\ \overline{l}\ \{s\} \\ \mathrm{skip}, \Phi \vdash \langle \mathrm{enter}\ \sigma_k\ (\overline{a}@\overline{l})\ [\overline{a_j \leftarrow v_j}], i_k, o_k, — \rangle \xrightarrow{s} \langle \sigma', i', o', n \rangle\end{array}}{K, \Phi \vdash c_0 = \langle \sigma_0, \_, \_, \_ \rangle \xrightarrow{f(\overline{e_k})} \langle \mathrm{leave}\ \sigma'\ \sigma_0, i', o', n \rangle} \qquad [\textsc{Call}]$$

$$K, \Phi \vdash c \xrightarrow{\mathrm{return}} c \qquad [\textsc{ReturnEmpty}]$$

$$\frac{\Phi \vdash c \xrightarrow{e}_{\mathscr{E}} c'}{K, \Phi \vdash c \xrightarrow{\mathrm{return}\ e} c'} \qquad [\textsc{Return}]$$

Figure 2: Continuation-passing Style Semantics for Statements