

## 1 Notational Convention

### 1.1 Set

値の集合を次のように記述することとする。

$$\langle\langle v_1, v_2, \dots \rangle\rangle$$

また、 $\langle\langle \rangle\rangle$  で空集合を表すこととする。

### 1.2 Map

キーと値の対応を示すマップを次のように定義する。

- $\{\{\}$  はマップである
- $\mu$  がマップならば  $\mu\{\{k : v\}\}$  はマップである

意味的には  $\{\{\}$  は空のマップを表し、 $\mu\{\{k : v\}\}$  で  $k$  に対応する値が  $v$  である以外は  $\mu$  と同じマップを表すこととする。また、 $\mu$  がマップであるとき、 $\mu$  の上でキー  $k$  に対応する値を  $\mu\{\{k\}$  と書くこととする。

従って  $\mu$  がマップであるとき次が成り立つ。

$$\begin{aligned}\mu\{\{k : v\}\}\{\{k\}\} &= v \\ \mu\{\{k : v\}\}\{\{l\}\} &= \mu\{\{l\}\}\end{aligned}$$

また、下記のような略記法を導入し、本文中で空でないマップを表す際に適宜用いることとする。

$$\{\{k_1 : v_1, k_2 : v_2, \dots k_n : v_n\}\} = \{\{\}\}\{\{k_1 : v_1\}\}\{\{k_2 : v_2\}\} \dots \{\{k_n : v_n\}\}$$

## 2 Code Translation

### 2.1 Labeled Statement

$$\begin{aligned}\mathbb{C}_s[\![label : s_1]\!](l, m, n, b, c, r, x, s) \\ = \mathbb{C}_s[\![s_1]\!](l, m \cup \langle\langle label \rangle\rangle, t, b\{\{label : n\}\}, c, r, x, s)\end{aligned}$$

### 2.2 Expression Statement

#### 2.2.1 Primary Expressions

$$\begin{aligned}\mathbb{C}_s[\![v;]\!](l, m, n, b, c, r, x, s) &= \mathbb{B}_{\text{exp}}(\text{Stack}(s), v)_{(l, n, x)} \\ \mathbb{C}_s[\![\text{true};]\!](l, m, n, b, c, r, x, s) &= \mathbb{B}_{\text{exp}}(\text{Stack}(s), \text{true})_{(l, n, x)} \\ \mathbb{C}_s[\![\text{false};]\!](l, m, n, b, c, r, x, s) &= \mathbb{B}_{\text{exp}}(\text{Stack}(s), \text{false})_{(l, n, x)} \\ \mathbb{C}_s[\![\text{null};]\!](l, m, n, b, c, r, x, s) &= \mathbb{B}_{\text{exp}}(\text{Stack}(s), \text{null})_{(l, n, x)} \\ &\dots\end{aligned}$$

$$\mathbb{C}_s[(e); \cdot] (l, m, n, b, c, r, x, s) = \mathbb{C}_s[e; \cdot] (l, m, n, b, c, r, x, s)$$

### 2.2.2 Member Expressions

$$\begin{aligned} & \mathbb{C}_s[e.p; \cdot] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e; \cdot] (l, m, t, b, c, r, x, s) \\ & \mathbb{B}_{\text{exp}}(\text{Stack}(s), \text{Stack}(s).p)(t, n, x) \\ \\ & \mathbb{C}_s[e_1[e_2]; \cdot] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; \cdot] (l, m, t_1, b, c, r, x, s) \\ & \mathbb{C}_s[e_2; \cdot] (t_1, m, t_2, b, c, r, x, s') \\ & \mathbb{B}_{\text{exp}}(\text{Stack}(s), \text{Stack}(s)[\text{Stack}(s')])(t_2, n, x) \end{aligned}$$

### 2.2.3 Function Call

$$\begin{aligned} & \mathbb{C}_s[e_1.p(e_2, e_3, \dots e_m); \cdot] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; \cdot] (l, m, t_1, b, c, r, x, s) \\ & \mathbb{C}_s[e_2; \cdot] (t_1, m, t_2, b, c, r, x, s_1) \\ & \mathbb{C}_s[e_3; \cdot] (t_2, m, t_3, b, c, r, x, s_2) \\ & \dots \\ & \mathbb{C}_s[e_m; \cdot] (t_{m-1}, m, t_m, b, c, r, x, s_{m-1}) \\ & \mathbb{B}_{\text{call}}(\text{Stack}(s), \text{Stack}(s).p, \text{Stack}(s_1), \text{Stack}(s_2), \dots \text{Stack}(s_{m-1}))(t_m, t_{m+1}, x) \\ & \mathbb{B}_{\text{recv}}(\text{Stack}(s))(t_{m+1}, n, x) \\ \\ & \mathbb{C}_s[e_1[e_2](e_3, e_4, \dots e_m); \cdot] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; \cdot] (l, m, t_1, b, c, r, x, s) \\ & \mathbb{C}_s[e_2; \cdot] (t_1, m, t_2, b, c, r, x, s_1) \\ & \mathbb{C}_s[e_3; \cdot] (t_2, m, t_3, b, c, r, x, s_2) \\ & \mathbb{C}_s[e_4; \cdot] (t_3, m, t_4, b, c, r, x, s_3) \\ & \dots \\ & \mathbb{C}_s[e_m; \cdot] (t_{m-1}, m, t_m, b, c, r, x, s_{m-1}) \\ & \mathbb{B}_{\text{call}}(\text{Stack}(s), \text{Stack}(s)[\text{Stack}(s_1)], \text{Stack}(s_2), \text{Stack}(s_3), \dots \text{Stack}(s_{m-1}))(t_m, t_{m+1}, x) \\ & \mathbb{B}_{\text{recv}}(\text{Stack}(s))(t_{m+1}, n, x) \\ \\ & \mathbb{C}_s[e_1(e_2, e_3, \dots e_m); \cdot] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; \cdot] (l, m, t_1, b, c, r, x, s) \\ & \mathbb{C}_s[e_2; \cdot] (t_1, m, t_2, b, c, r, x, s_1) \\ & \mathbb{C}_s[e_3; \cdot] (t_2, m, t_3, b, c, r, x, s_2) \\ & \dots \\ & \mathbb{C}_s[e_m; \cdot] (t_{m-1}, m, t_m, b, c, r, x, s_{m-1}) \\ & \mathbb{B}_{\text{call}}(\text{null}, \text{Stack}(s), \text{Stack}(s_1), \text{Stack}(s_2), \dots \text{Stack}(s_{m-1}))(t_m, t_{m+1}, x) \\ & \mathbb{B}_{\text{recv}}(\text{Stack}(s))(t_{m+1}, n, x) \end{aligned}$$

### 2.2.4 Assignment

$$\begin{aligned} & \mathbb{C}_s[v = e; ] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e; ] (l, m, t, b, c, r, x, s) \\ & \mathbb{B}_{\text{exp}}(v, \text{Stack}(s))(t, n, x) \end{aligned}$$

$$\begin{aligned} & \mathbb{C}_s[e_1.p = e_2; ] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; ] (l, m, t_1, b, c, r, x, s) \\ & \mathbb{C}_s[e_2; ] (t_1, m, t_2, b, c, r, x, s') \\ & \mathbb{B}_{\text{exp}}(\text{Stack}(s).p, \text{Stack}(s'))(t_2, n, x) \end{aligned}$$

$$\begin{aligned} & \mathbb{C}_s[e_1[e_2] = e_3; ] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; ] (l, m, t_1, b, c, r, x, s) \\ & \mathbb{C}_s[e_2; ] (t_1, m, t_2, b, c, r, x, s_1) \\ & \mathbb{C}_s[e_3; ] (t_2, m, t_3, b, c, r, x, s_2) \\ & \mathbb{B}_{\text{exp}}(\text{Stack}(s)[\text{Stack}(s_1)], \text{Stack}(s_2))(t_3, n, x) \end{aligned}$$

$$\begin{aligned} & \mathbb{C}_s[v \text{ op } = e; ] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e; ] (l, m, t, b, c, r, x, s) \\ & \mathbb{B}_{\text{exp}}(v, v \text{ op } \text{Stack}(s))(t, n, x) \end{aligned}$$

$$\begin{aligned} & \mathbb{C}_s[e_1.p \text{ op } = e_2; ] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; ] (l, m, t, b, c, r, x, s) \\ & \mathbb{C}_s[\text{Stack}(s).p; ] (l, m, t, b, c, r, x, s_1) \\ & \mathbb{C}_s[e_2; ] (l, m, t, b, c, r, x, s_2) \\ & \mathbb{B}_{\text{exp}}(\text{Stack}(s).p, \text{Stack}(s_1) \text{ op } \text{Stack}(s_2))(t, n, x) \end{aligned}$$

$$\begin{aligned} & \mathbb{C}_s[e_1[e_2] \text{ op } = e_3; ] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; ] (l, m, t_1, b, c, r, x, s) \\ & \mathbb{C}_s[e_2; ] (t_1, m, t_2, b, c, r, x, s_1) \\ & \mathbb{C}_s[\text{Stack}(s)[\text{Stack}(s_1)]; ] (t_2, m, t_3, b, c, r, x, s_2) \\ & \mathbb{C}_s[e_3; ] (t_2, m, t_3, b, c, r, x, s_3) \\ & \mathbb{B}_{\text{exp}}(\text{Stack}(s)[\text{Stack}(s_1)], \text{Stack}(s_2) \text{ op } \text{Stack}(s_3))(t_3, n, x) \end{aligned}$$

### 2.2.5 Special Forms

$$\begin{aligned} & \mathbb{C}_s[e_1 \ \&\& \ e_2; ] (l, m, n, b, c, r, x, s) \\ = & \mathbb{C}_s[e_1; ] (l, m, t_1, b, c, r, x, s) \\ & \mathbb{B}_{\text{cond}}(\text{Stack}(s), t_2)(t_1, n, x) \\ & \mathbb{C}_s[e_2; ] (t_2, m, n, b, c, r, x, s) \end{aligned}$$

$$\begin{aligned}
& \mathbb{C}_s[e_1 \parallel e_2;] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[e_1;] (l, m, t_1, b, c, r, x, s) \\
& \quad \mathbb{B}_{\text{cond}}(\text{Stack}(s), n)_{(t_1, t_2, x)} \\
& \quad \mathbb{C}_s[e_2;] (t_2, m, n, b, c, r, x, s) \\
\\
& \mathbb{C}_s[e_1 ? e_2 : e_3;] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[e_1;] (l, m, t_1, b, c, r, x, s) \\
& \quad \mathbb{B}_{\text{cond}}(\text{Stack}(s), t_2)_{(t_1, t_3, x)} \\
& \quad \mathbb{C}_s[e_2;] (t_2, m, n, b, c, r, x, s) \\
& \quad \mathbb{C}_s[e_3;] (t_3, m, n, b, c, r, x, s)
\end{aligned}$$

### 2.2.6 Other Binary Operators

$$\begin{aligned}
& \mathbb{C}_s[e_1 \text{ op } e_2;] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[e_1;] (l, m, t_1, b, c, r, x, s) \\
& \quad \mathbb{C}_s[e_2;] (t_1, m, t_2, b, c, r, x, s') \\
& \quad \mathbb{B}_{\text{exp}}(\text{Stack}(s) \text{ op } \text{Stack}(s'))_{(t_2, n, x)}
\end{aligned}$$

## 2.3 If Statement

$$\begin{aligned}
& \mathbb{C}_s[\text{if } (e) \ s_1] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[e;] (l, m, t_1, b, c, x, s) \\
& \quad \mathbb{B}_{\text{cond}}(\text{Stack}(s), t_2)_{(t_1, n, x)} \\
& \quad \mathbb{C}_s[s_1] (t_2, \ll, n, b, c, r, x, s) \\
\\
& \mathbb{C}_s[\text{if } (e) \ s_1 \text{ else } s_2] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[e;] (l, m, t_1, b, c, x, s) \\
& \quad \mathbb{B}_{\text{cond}}(\text{Stack}(s), t_2)_{(t_1, t_3, x)} \\
& \quad \mathbb{C}_s[s_1] (t_2, m, n, b, c, r, x, s) \\
& \quad \mathbb{C}_s[s_2] (t_3, m, n, b, c, r, x, s)
\end{aligned}$$

## 2.4 Iteration Statements

### 2.4.1 Do-While Statement

$$\begin{aligned}
& \mathbb{C}_s[\text{do } s_1 \text{ while } (e);] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[s_1] (l, \ll, t_1, b', c', r, x, s) \\
& \quad \mathbb{C}_s[e;] (t_1, \ll, t_2, \{\}, \{\}, r, x, s) \\
& \quad \mathbb{B}_{\text{cond}}(\text{Stack}(s), l)_{(t_2, n, x)}
\end{aligned}$$

where

$$\begin{aligned}
m &= \ll l_1, l_2, \dots, l_k \gg \\
b' &= b \{ l_1 : n, l_2 : n, \dots, l_k : n \} \{ "" : n \} \\
c' &= c \{ l_1 : t_1, l_2 : t_1, \dots, l_k : t_1 \} \{ "" : t_1 \}
\end{aligned}$$

### 2.4.2 While Statement

$$\begin{aligned}
& \mathbb{C}_s[\text{while } (e) \ s_1] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[e; ] (l, \llbracket e \rrbracket, t_1, \{\}, \{\}, r, x, s) \\
&\quad \mathbb{B}_{\text{cond}}(\text{Stack}(s), t_2)_{(t_1, n, x)} \\
&\quad \mathbb{C}_s[s_1] (t_2, \llbracket e \rrbracket, l, b', c', r, x, s)
\end{aligned}$$

where

$$\begin{aligned}
m &= \llbracket l_1, l_2, \dots, l_k \rrbracket \\
b' &= b\{\{l_1 : n, l_2 : n, \dots, l_k : n\} \}''' : n\} \\
c' &= c\{\{l_1 : l, l_2 : l, \dots, l_k : l\} \}''' : l\}
\end{aligned}$$

### 2.4.3 For Statement

$$\begin{aligned}
& \mathbb{C}_s[\text{for } (e_1; e_2; e_3) \ s_1] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[e_1; ] (l, \llbracket e_1 \rrbracket, t_1, \{\}, \{\}, r, x, s) \\
&\quad \mathbb{C}_s[e_2; ] (t_1, \llbracket e_2 \rrbracket, t_2, \{\}, \{\}, r, x, s) \\
&\quad \mathbb{B}_{\text{cond}}(\text{Stack}(s), t_3)_{(t_2, n, x)} \\
&\quad \mathbb{C}_s[s_1] (t_3, \llbracket e_3 \rrbracket, t_4, b', c', r, x, s) \\
&\quad \mathbb{C}_s[e_3] (t_4, \{\}, t_1, \{\}, \{\}, r, x, s)
\end{aligned}$$

where

$$\begin{aligned}
m &= \llbracket l_1, l_2, \dots, l_k \rrbracket \\
b' &= b\{\{l_1 : n, l_2 : n, \dots, l_k : n\} \}''' : n\} \\
c' &= c\{\{l_1 : t_4, l_2 : t_4, \dots, l_k : t_4\} \}''' : t_4\}
\end{aligned}$$

### 2.4.4 For-in Statement

$$\begin{aligned}
& \mathbb{C}_s[\text{for } (v \text{ in } e) \ s_1] (l, m, n, b, c, r, x, s) \\
&= \mathbb{C}_s[e; ] (l, \llbracket e \rrbracket, t_1, \{\}, \{\}, r, x, s) \\
&\quad \mathbb{B}_{\text{enum}}(\text{Stack}(s), \text{Stack}(s))_{(t_1, t_2, x)} \\
&\quad \mathbb{B}_{\text{exp}}(\text{Stack}(s'), 0)_{(t_2, t_3, x)} \\
&\quad \mathbb{B}_{\text{cond}}(\text{Stack}(s') < \text{Stack}(s).length, t_4)_{(t_3, n, x)} \\
&\quad \mathbb{C}_s[s_1] (t_4, \llbracket e \rrbracket, t_5, b', c', r, x, s'') \\
&\quad \mathbb{B}_{\text{exp}}(\text{Stack}(s'), \text{Stack}(s') + 1)_{(t_5, t_3, x)}
\end{aligned}$$

where

$$\begin{aligned}
m &= \llbracket l_1, l_2, \dots, l_k \rrbracket \\
b' &= b\{\{l_1 : n, l_2 : n, \dots, l_k : n\} \}''' : n\} \\
c' &= c\{\{l_1 : t_4, l_2 : t_4, \dots, l_k : t_4\} \}''' : t_3\}
\end{aligned}$$

## 2.5 Break Statement

$$\begin{aligned}
& \mathbb{C}_s[\text{break}; ] (l, m, n, b, c, r, x, s) \\
&= \mathbb{B}_{\text{cond}}(\text{true}, b\{\}''' : \})_{(l, n, x)} \\
& \\
& \mathbb{C}_s[\text{break } label; ] (l, m, n, b, c, r, x, s) \\
&= \mathbb{B}_{\text{cond}}(\text{true}, b\{\{label\} \}''' : \})_{(l, n, x)}
\end{aligned}$$

## 2.6 Continue Statement

$$\begin{aligned} & \mathbb{C}_s[\text{continue};] (l, m, n, b, c, r, x, s) \\ &= \mathbb{B}_{\text{cond}}(\text{true}, c\{\{\}\}) (l, n, x) \end{aligned}$$

$$\begin{aligned} & \mathbb{C}_s[\text{continue } \textit{label};] (l, m, n, b, c, r, x, s) \\ &= \mathbb{B}_{\text{cond}}(\text{true}, c\{\{\textit{label}\}\}) (l, n, x) \end{aligned}$$

## 2.7 Return Statement

$$\begin{aligned} & \mathbb{C}_s[\text{return } e;] (l, m, n, b, c, r, x, s) \\ &= \mathbb{C}_s[e;] (l, \ll, t_1, b, c, r, x, s) \\ & \quad \mathbb{B}_{\text{ret}}(\text{Stack}(s)) (t_1, r, x) \end{aligned}$$

$$\begin{aligned} & \mathbb{C}_s[\text{return};] (l, m, n, b, c, r, x, s) \\ &= \mathbb{B}_{\text{ret}}(\text{void0}) (l, r, x) \end{aligned}$$

## 2.8 Throw Statement

$$\begin{aligned} & \mathbb{C}_s[\text{throw } e;] (l, m, n, b, c, r, x, s) \\ &= \mathbb{C}_s[e;] (l, \ll, t, b, c, r, x, s) \\ & \quad \mathbb{B}_{\text{ret}}(\text{Stack}(s)) (t, x, x) \end{aligned}$$

## 2.9 Try Statement

$$\begin{aligned} & \mathbb{C}_s[\text{try } s_1 \text{ catch } (v) s_2] (l, m, n, b, c, r, x, s) \\ &= \mathbb{C}_s[s_1] (l, \ll, n, b, c, r, t_1, s) \\ & \quad \mathbb{B}_{\text{recv}}(v) (t_1, \ll, t_2, b, c, r, x, s) \\ & \quad \mathbb{C}_s[s_2] (t_2, \ll, n, b, c, r, x, s) \end{aligned}$$

$$\begin{aligned}
& \mathbb{C}_s[\text{try } s_1 \text{ finally } s_2] (l, m, n, b, c, r, x, s) \\
= & \mathbb{C}_s[s_1] (l, \ll, n', b', c', r_1, x_1, s) \\
& \mathbb{B}_{\text{recv}}(\mathbb{S}_{\text{stack}}(s'))_{(t_1, t_2, x)} \\
& \mathbb{C}_s[s_2] (t_2, \ll, t_3, b, c, r, x, s'') \\
& \mathbb{B}_{\text{cond}}(\text{true}, \mathbb{S}_{\text{stack}}(s'))_{(t_3, \mathbb{S}_{\text{stack}}(s'), x)} \\
& \mathbb{B}_{\text{ret}}(n)_{(n', t_1, x)} \\
& \mathbb{B}_{\text{ret}}(b_1)_{(b'_1, t_1, x)} \\
& \mathbb{B}_{\text{ret}}(b_2)_{(b'_2, t_1, x)} \\
& \dots \\
& \mathbb{B}_{\text{ret}}(b_n)_{(b'_n, t_1, x)} \\
& \mathbb{B}_{\text{ret}}(c_1)_{(c'_1, t_1, x)} \\
& \mathbb{B}_{\text{ret}}(c_2)_{(c'_2, t_1, x)} \\
& \dots \\
& \mathbb{B}_{\text{ret}}(c_m)_{(c'_m, t_1, x)} \\
& \mathbb{B}_{\text{recv}}(\mathbb{S}_{\text{stack}}(s))_{(r_1, r_2, x)} \\
& \mathbb{B}_{\text{ret}}(r_3)_{(r_2, t_1, x)} \\
& \mathbb{B}_{\text{ret}}(\mathbb{S}_{\text{stack}}(s))_{(r_3, r, x)} \\
& \mathbb{B}_{\text{recv}}(\mathbb{S}_{\text{stack}}(s))_{(x_1, x_2, x)} \\
& \mathbb{B}_{\text{ret}}(x_3)_{(x_2, t_1, x)} \\
& \mathbb{B}_{\text{ret}}(\mathbb{S}_{\text{stack}}(s))_{(x_3, x, x)}
\end{aligned}$$

where

$$\begin{aligned}
b &= \{ \{ lb_1 : b_1, lb_2 : b_2, \dots, lb_n : b_n \} \} \\
b' &= \{ \{ lb_1 : b'_1, lb_2 : b'_2, \dots, lb_n : b'_n \} \} \\
c &= \{ \{ lc_1 : c_1, lc_2 : c_2, \dots, lc_m : c_m \} \} \\
c' &= \{ \{ lc_1 : c'_1, lc_2 : c'_2, \dots, lc_m : c'_m \} \}
\end{aligned}$$

$$\begin{aligned}
& \mathbb{C}_s[\text{try } s_1 \text{ catch } (v) s_2 \text{ finally } s_3] (l, n, b, c, r, x, s) \\
= & \mathbb{C}_s[\text{try } \{ \text{try } s_1 \text{ catch } (v) s_2 \} \text{ finally } s_3] (l, n, b, c, r, x, s)
\end{aligned}$$

### 3 Code Generation

$$\mathbb{S}_{\text{stack}}(s) =$$

\$stack\_s

$$\mathbb{B}_{\text{exp}}(v, e)_{(l, n, x)} =$$

```

var l = {
  procedure: function ( ) {
    v = e;
    return { continuation: n };
  },

```

```

    exception:  $x$ 
};

```

$$\mathbb{B}_{\text{cond}}(e, t)_{(l, f, x)} =$$

```

var  $l$  = {
  procedure: function ( ) {
    if (  $e$  ) {
      return { continuation:  $t$  };
    } else {
      return { continuation:  $f$  };
    }
  },
  exception:  $x$ 
};

```

$$\mathbb{B}_{\text{ret}}(v)_{(l, n, x)} =$$

```

var  $l$  = {
  procedure: function ( ) {
    return {
      continuation:  $n$ ,
      ret_val      :  $v$ 
    };
  },
  exception:  $x$ 
};

```

$$\mathbb{B}_{\text{call}}(t, f, a_1, a_2, \dots)_{(l, n, x)} =$$

```

var  $l$  = {
  procedure: function ( ) {
    if ( [  $f$  is native function ] ) {
      return {
        continuation:  $n$ ,
        ret_val      :  $f$ .apply( $t$ , [ $a_1$ ,  $a_2$ , ...])
      };
    } else {
      return  $f(t, [a_1, a_2, \dots], n)$ ;
    }
  }
};

```



```

    },
    exception:  $x$ 
};

```

$$\mathbb{B}_{\text{recv}}(v)_{(l,n,x)} =$$

```

var  $l$  = {
  procedure: function (  $\$ret\_val$  ) {
     $v = \$ret\_val$ ;
    return {
      continuation:  $n$ 
    };
  },
  exception:  $x$ 
};

```

$$\mathbb{B}_{\text{enum}}(v, e)_{(l,n,x)} =$$

```

var  $l$  = {
  procedure: function ( ) {
    var  $a = \text{new Array}()$ ;
    for ( var  $i$  in  $e$  )  $a.push(i)$ ;
     $v = a$ ;
    return { continuation:  $n$  };
  },
  exception:  $x$ 
};

```