Semantics for Advanced Control Structures

Razieh Doregard *Fatemeh Nourian*

January 2025

If Rule

$$(\textsc{condtion True})\ \frac{\text{eval(BExpr,e)}}{e, \langle \text{if(BExpr)}\pi_1 \text{ else } \pi_2 | T\rangle \overset{\tau}{\hookrightarrow} e, \pi_1 \oplus \langle T\rangle}$$

$$(\textsc{condition False})\ \frac{\neg\ \text{eval(BExpr,e)}}{e, \langle \text{if(BExpr)}\pi_1 \text{ else } \pi_2 | T\rangle \overset{\tau}{\hookrightarrow} e, \pi_2 \oplus \langle T\rangle}$$

For Rule

The for loop structure can be simulated using a while loop; the only main difference is that the initialization from the for loop must be executed before the while loop. Additionally, the update step will become the update element of the while loop.

$$(\textsc{For})\ \frac{e, \langle \text{init}\rangle \to e', \pi' \quad \omega.\text{update} = \langle \text{update;}\rangle}{e, \langle \text{for(init; condition; update)}\{\text{body}\}\pi\rangle \overset{\tau}{\hookrightarrow} e', \langle \text{while(condition)}\{\text{body} \oplus \text{update}\}\pi\rangle}$$

While Rule

The update element in the loop's syntax is a statement, which is often equivalent to epsilon, except when converting a for loop to a while loop. In that case, the update section of the for loop is placed as the update element of the while loop, followed by a semicolon (;).

$$\text{While Syntax: While} = \text{condition} \times \text{Update}^* \times \text{Stmt}^*$$

$$(\textsc{While True})\ \frac{\text{eval(BExpr,e)}}{e, \langle \text{while (BExpr)}\ \{\pi\}\ | T\rangle \overset{\tau}{\hookrightarrow} e, \{\pi\} \oplus W \oplus \langle T\rangle}$$

$$(\textsc{While False})\ \frac{\neg\text{eval(BExpr,e)}}{e, \langle \text{while (BExpr)}\ \{\pi\}\ | T\rangle \overset{\tau}{\hookrightarrow} e, \langle T\rangle}$$

Break Rule

Regarding the conversion of for to while, it is sufficient to consider the BREAK statement only within the while loop.

$$(\textsc{Break})\ e, \langle \text{break} | T\rangle \overset{\tau}{\hookrightarrow} e, \epsilon$$

Continue Rule
Regarding the conversion of for to while, it is sufficient to consider the CONTINUE statement only within the while loop.

$$(\textsc{Continue})\ \frac{e, \langle \omega.\text{update} \rangle \to e', \pi'}{e, \langle \text{continue}, T \rangle \to e', \langle \omega \rangle}$$

Return Rule

$$(\textsc{Return})\ e, \langle \text{return} | T \rangle \overset{\tau}{\hookrightarrow} e, \epsilon$$

Local Variable Declaration Rule

$$\text{type id} = \text{Expr};$$

$$(\textsc{LocalVarDecl})\ e, \langle \text{type id} = \text{Expr}; \,|\, T \rangle \overset{\tau}{\hookrightarrow} e \cup \{\text{map}(id, \text{Expr})\}, \langle T \rangle$$

$$\text{type id};$$

$$(\textsc{LocalVarDecl})\ e, \langle \text{type id}; | T \rangle \overset{\tau}{\hookrightarrow} e \cup \{\text{map}(id, 0)\}, \langle T \rangle$$

Method Call Rule

$$(\textsc{MethodCall})\ (e, q, \pi) \overset{\tau}{\hookrightarrow} (e \cup map(params(n), p), \langle \text{body}(\text{eval}(p, e)) | T \rangle, \pi)$$

Assignment Rule

$$(\textsc{Assignment})\ e, \langle V = \text{Expr} | T \rangle \overset{\tau}{\hookrightarrow} e[\text{v} \to \text{eval}(\text{Expr})], \langle T \rangle$$

Send Message Rule

$$(\textsc{SendMessage})\ e, \text{k.m}(p) \xhookrightarrow{eval(self,e), eval(k,e), m, eval(p,e))!} e, \langle T \rangle$$

Switch Case Rule

$$\text{(SwitchCase)}\ \frac{\text{eval}(\text{SwitchExpr}, e) = \text{eval}(\text{CaseExpr}_i, e), \quad 0 \leq i < n}{\begin{array}{c} e, \text{switch}(\text{switchExpr}) \quad \text{case CaseExpr}_0 : \text{CaseStmt}_0 \quad \ldots \quad \text{case CaseExpr}_i : \pi \\ \text{case CaseExpr}_1 : \text{CaseStmt}_1 \quad \ldots \quad \text{case CaseExpr}_{n-1} : \text{CaseStmt}_{n-1} \quad \text{default} : \\ \text{CaseStmt}_i \quad \overset{\tau}{\hookrightarrow} \quad e, \pi \end{array}}$$

Default Rule

$$\text{(Default)}\ \frac{\forall i, 0 \leq i < n,\ \text{eval}(\text{SwitchExpr}, e) \neq \text{eval}(\text{CaseExpr}_i, e)}{\begin{array}{c} e, \text{switch}(\text{SwitchExpr}) \ \text{case CaseExpr}_i, \text{CaseStmt}_i, \ldots, \text{case CaseExpr}_{n-1} \\ \text{CaseStmt}_{n-1} \quad \text{default} : \pi \quad \overset{\tau}{\hookrightarrow} \quad e, \pi \end{array}}$$