The Facade Language

Peng Wang, MIT CSAIL

1 Syntax

The syntax of Facade is defined in Figure 2, drawing on notations listed in Figure 1.

2 Operational Semantics

The big-step operational semantics of Facade is defined in terms of two relations, $\Psi \vdash (\Sigma, s) \Downarrow \Sigma'$ (read as "runs to") and $\Psi \vdash (\Sigma, s) \downarrow$ (read as "safe"). They are shown in Figure 5 and Figure 6 respectively, drawing on notations and auxiliary functions listed in Figure 3 and Figure 4. The "safe" relation is defined coinductively, as reflected by the double horizontal lines in Figure 6. Coinductive definition allows potentially nonterminating programs to be also treated as safe, such as an empty forever-loop. Ψ is the list of available functions (specifications) the program can call, elided when irrelevant. "NoDup" means pairwise distinctive.

Optional	$\lfloor\cdot\rfloor$	List Of	$(\cdot)^*$
Ordered Pair	$\times$	Disjoint Sum	$+$
Machine Word	$\mathbb{W}$	String	$\mathbb{S}$

Figure 1: Notations used in this article

$$\begin{array}{rrcl}
\text{Constant} & w & \in & \mathbb{W} \\
\text{Label} & l & \in & \mathbb{S}_{\text{module}} \times \mathbb{S}_{\text{fun}} \\
\text{Variable} & x, y & \in & \mathbb{S} \\
\text{Binary Op} & o & ::= & + \mid - \mid \times \mid = \mid \neq \mid < \mid \leqslant \\
\text{Expression} & e & ::= & x \mid w \mid e\ o\ e \\
\text{Statement} & s & ::= & \textsf{skip} \mid s; s \mid \textsf{if}\ e\ \{s\}\ \textsf{else}\ \{s\} \\
 & & & \mid \textsf{while}\ e\ \{s\} \mid x := \textsf{call}\ l\ (x^*) \\
 & & & \mid x := e
\end{array}$$

Figure 2: Syntax of Facade

$$\begin{array}{rrcl}
\text{State}(\Sigma) & E & = & \mathbb{S} \to \lfloor V \rfloor \\
\text{Value} & V, I & = & \textsf{ADT}(A) + \textsf{SCA}(\mathbb{W}) \\
\text{ADT Domain} & A & = & [\text{parameter of theory}] \\
\text{Context } (\Psi) & & = & (\text{Label} \to \lfloor F \rfloor) \\
\text{Axiomatic Func. Spec} & F & = & P \times Q \\
\text{Precondition} & P & = & I^* \to \textsf{Prop} \\
\text{Postcondition} & Q & = & (I \times O)^* \times I \to \textsf{Prop} \\
\text{Output Value} & O & = & \lfloor A \rfloor
\end{array}$$

Figure 3: Notations used in operational semantics definition

$$\mathbf{upd}'(\Sigma, x, I, O) \equiv \begin{cases} \Sigma[x \to \textsf{ADT}(O)] & \text{if } I = \textsf{ADT}(\cdot) \wedge O \neq \bot \\ \Sigma - x & \text{if } I = \textsf{ADT}(\cdot) \wedge O = \bot \\ \Sigma & \text{if } I = \textsf{SCA}(\cdot) \end{cases}$$

$$\mathbf{upd}(\Sigma, x^*, I^*, O^*) \equiv \mathbf{fold}(\mathbf{upd}', \Sigma, x^*, I^*, O^*)$$

Figure 4: Auxiliary functions used in operational semantics definition

$\boxed{\Psi \vdash (\Sigma, s) \Downarrow \Sigma'}$

$$\frac{[\![e]\!]_\Sigma = \mathsf{SCA}(\cdot) \qquad [\![x]\!]_\Sigma \neq \mathsf{ADT}(\cdot)}{(\Sigma, x := e) \Downarrow \Sigma[x \to [\![e]\!]_\Sigma]}\ \textsc{Assign}$$

$$\frac{}{(\Sigma, \mathsf{skip}) \Downarrow \Sigma}\ \textsc{Skip} \qquad \frac{(\Sigma, s_1) \Downarrow \Sigma' \qquad (\Sigma', s_2) \Downarrow \Sigma''}{(\Sigma, s_1; s_2) \Downarrow \Sigma''}\ \textsc{Seq}$$

$$\frac{([\![e]\!]_\Sigma = \mathsf{ADT}(w) \land w \neq 0 \land (\Sigma, s_T) \Downarrow \Sigma') \ \lor\ ([\![e]\!]_\Sigma = \mathsf{ADT}(0) \land (\Sigma, s_F) \Downarrow \Sigma')}{(\Sigma, \mathsf{if}\ e\ \{s_T\}\ \mathsf{else}\ \{s_F\}) \Downarrow \Sigma'}\ \textsc{If}$$

$$\frac{(\Sigma, \mathsf{if}\ e\ \{s; \mathsf{while}\ e\ \{s\}\}\ \mathsf{else}\ \{\mathsf{skip}\}) \Downarrow \Sigma'}{(\Sigma, \mathsf{while}\ e\ \{s\}) \Downarrow \Sigma'}\ \textsc{While}$$

$$\frac{\begin{array}{c}\Psi(l) = (P, Q) \qquad \Sigma(x^*) = I^* \qquad P(I^*) \qquad [\![y]\!]_\Sigma \neq \mathsf{ADT}(\cdot) \\ \mathbf{NoDup}(x^*) \qquad |O^*| = |I^*| \qquad Q(I^*, O^*, R) \qquad \Sigma' = \mathbf{upd}(\Sigma, x^*, I^*, O^*)\end{array}}{\Psi \vdash (\Sigma, y := \mathsf{call}\ l\ (x^*)) \Downarrow \Sigma'[y \to R]}\ \textsc{Call}$$

Figure 5: Big-step operational semantics of Facade (the “RunsTo” relation)

$\boxed{\Psi \vdash (\Sigma, s) \downarrow}$

$$\frac{[\![e]\!]_\Sigma = \mathsf{SCA}(\cdot) \qquad [\![x]\!]_\Sigma \neq \mathsf{ADT}(\cdot)}{\overline{(\Sigma, x := e) \downarrow}}\ \textsc{Assign}$$

$$\frac{}{\overline{(\Sigma, \mathsf{skip}) \downarrow}}\ \textsc{Skip} \qquad \frac{(s_1, \Sigma) \downarrow \qquad \forall \Sigma'.\ (\Sigma, s_1) \Downarrow \Sigma' \Rightarrow (s_2, \Sigma') \downarrow}{\overline{(s_1; s_2, \Sigma) \downarrow}}\ \textsc{Seq}$$

$$\frac{([\![e]\!]_\Sigma = \mathsf{ADT}(w) \land w \neq 0 \land (\Sigma, s_T) \downarrow) \ \lor\ ([\![e]\!]_\Sigma = \mathsf{ADT}(0) \land (\Sigma, s_F) \downarrow)}{\overline{(\Sigma, \mathsf{if}\ e\ \{s_T\}\ \mathsf{else}\ \{s_F\}) \downarrow}}\ \textsc{If}$$

$$\frac{(\Sigma, \mathsf{if}\ e\ \{s; \mathsf{while}\ e\ \{s\}\}\ \mathsf{else}\ \{\mathsf{skip}\}) \downarrow}{\overline{(\Sigma, \mathsf{while}\ e\ \{s\}) \downarrow}}\ \textsc{While}$$

$$\frac{\Psi(l) = (P, Q) \qquad \Sigma(x^*) = I^* \qquad P(I^*) \qquad [\![y]\!]_\Sigma \neq \mathsf{ADT}(\cdot) \qquad \mathbf{NoDup}(x^*)}{\overline{\Psi \vdash (\Sigma, y := \mathsf{call}\ l\ (x^*)) \downarrow}}\ \textsc{Call}$$

Figure 6: Big-step operational semantics of Facade (the “Safe” predicate)