1 Dziedziny semantyczne

Oznaczmy:

$$\begin{aligned}
\mathrm{Val} &= \mathbb{N} \\
\mathrm{Loc} &= \mathbb{N} \\
\mathrm{Store} &= \mathrm{Loc} \to \mathrm{Val} \\
\mathrm{Env} &= \mathrm{Var} \to \mathrm{Loc} \\
\mathrm{Proc} &= \mathrm{Loc} \to \mathrm{Store} \to \mathrm{Store} \\
\mathrm{PEnv} &= \mathrm{PVar} \to \mathrm{Proc} \\
\mathrm{Mixins} &= \{\mathrm{commit}, \mathrm{on_cbv}\} \to \mathrm{Store} \to \mathrm{Store} \\
\mathrm{State} &= \mathrm{Mixins} \times \mathrm{Store}
\end{aligned}$$

Przy czym implicite, bez zaznaczania tego jawnie, zakładam, że $\bot$ jest najmniejszym elementem każdej z powyższych dziedzin.

Ideą rozwiązania jest to, że przy wywołaniu procedury będzie alokowane nowe miejsce w pamięci, w które będzie kopiowana wartość parametru procedury. Wszystkie operacje na parametrze procedury, niezależnie, czy dokonują się w trybie in-out czy przez wartość będą wykonywane na tej lokacji.

W momencie wychodzenia z trybu in-out (przez włączenie trybu przez wartość lub też wyjście z procedury) wartość tej pomocniczej zmiennej będzie kopiowana na parametr aktualny.

W tym celu, Mixins przechowuje (pod on_cbv) funkcję modyfikującą stan, która ma być na nim zaaplikowana przy przejściu do trybu przez wartość lub wyjściu z procedury oraz (pod commit) samą funkcję dokonującą modyfikacji stanu, tj. przepisującą zawartość lokalnej kopii parametru na parametr aktualny. W założeniu $\mathrm{m\,on\_cbv} \in \{\mathrm{m\,commit}, \mathrm{id}_{\mathrm{Store}}\}$ dla $\mathrm{m} : \mathrm{Mixins}$, przy czym pierwsza możliwość jest wtedy, gdy wykonanie jest w trybie in-out, a druga, gdy przez wartość.

Aby wytworzyć State mając dany jedynie $\mathrm{s} : \mathrm{Store}$ na początku wykonania, należy wziąć $(\lambda\_.\,\mathrm{id}_{\mathrm{Store}}, \mathrm{s}) : \mathrm{State}$, jeśli przyjmiemy semantykę, że instrukcje **cio** i **cbv** wykonane poza procedurą nie robią nic.

Zakładam, że dowolne użycie $\bot$ (np. aplikacja czegokolwiek do $\bot$) skutkuje $\bot$, tj. błędy propagują się.

2 Typy funkcji semantycznych

$$\begin{aligned}
\mathcal{E} &: \mathit{Expr} \to \mathrm{Env} \to \mathrm{Store} \to \mathrm{Val} \\
\mathcal{D} &: \mathit{Dec} \to \mathrm{Env} \to \mathrm{PEnv} \to \mathrm{Store} \to (\mathrm{Env} \times \mathrm{PEnv} \times \mathrm{Store}) \\
\mathcal{I} &: \mathit{Instr} \to \mathrm{Env} \to \mathrm{PEnv} \to \mathrm{State} \to \mathrm{State}
\end{aligned}$$

Zakładam, że dana jest funkcja $\mathrm{alloc} : \mathrm{Store} \to (\mathrm{Loc} \times \mathrm{Store})$.

3 Definicje

$$\mathcal{E}[\![n]\!]\ \_\ \_ = n_{\mathbb{N}} \qquad \text{[EXPR\_NUM]}$$

$$\mathcal{E}[\![x]\!]\ \rho\ s = s\ (\rho\ x) \qquad \text{[EXPR\_VAR]}$$

$$\mathcal{E}[\![E_1 + E_2]\!]\ \rho\ s = \mathcal{E}[\![E_1]\!]\ \rho\ s + \mathcal{E}[\![E_2]\!]\ \rho\ s \qquad \text{[EXPR\_PLUS]}$$

$$\mathcal{D}[\![\textbf{int } x := E]\!]\ \rho\ \rho_P\ s = (\rho[x \mapsto l], \rho_P, s'[l \mapsto \mathcal{E}[\![E]\!]\ \rho\ s]) \qquad \text{[DECL\_INT]}$$

where

$$(l, s') = \text{alloc}\ s$$

$$\mathcal{D}[\![D_1; D_2]\!]\ \rho\ \rho_P\ s = \mathcal{D}[\![D_2]\!]\ \rho'\ \rho'_P\ s' \qquad \text{[DECL\_SEMICOLON]}$$

where

$$(\rho', \rho'_P, s') = \mathcal{D}[\![D_1]\!]\ \rho\ \rho_P\ s$$

$$\mathcal{D}[\![\textbf{proc } p(x)I]\!]\ \rho\ \rho_P\ s = (\rho, \rho_P[p \mapsto \text{fix}\ \Phi], s) \qquad \text{[DECL\_PROC]}$$

where

$$\Phi : \text{Proc} \to \text{Proc}$$

$$\Phi\ F\ l\ s = \hat{m}\ \text{on\_cbv}\ \hat{s}$$

where

$$m : \text{Mixins}$$

$$m\ \text{commit}\ s = s[l \mapsto s\ l']$$

$$m\ \text{on\_cbv}\ s = s$$

$$(l', s') = \text{alloc}\ s$$

$$(\hat{m}, \hat{s}) = \mathcal{I}[\![I]\!]\ \rho[x \mapsto l']\ \rho_P[p \mapsto F]\ (m, s'[l' \mapsto s\ l])$$

$$\mathcal{I}[\![x := E]\!]\ \rho\ \_\ (m, s) = (m, s[\rho\ x \mapsto \mathcal{E}[\![E]\!]\ \rho\ s]) \qquad \text{[INSTR\_ASSIGN]}$$

$$\mathcal{I}[\![cbv]\!]\ \_\ \_\ (m, s) = (m[\text{on\_cbv} \mapsto \text{id}_{State}], m\ \text{on\_cbv}\ s) \qquad \text{[INSTR\_CBV]}$$

$$\mathcal{I}[\![cio]\!]\ \_\ \_\ (m, s) = (m[\text{on\_cbv} \mapsto m\ \text{commit}], s) \qquad \text{[INSTR\_CIO]}$$

$$\mathcal{I}[\![I_1; I_2]\!]\ \rho\ \rho_P = (\mathcal{I}[\![I_1]\!]\ \rho\ \rho_P); (\mathcal{I}[\![I_2]\!]\ \rho\ \rho_P) \qquad \text{[INSTR\_SEMICOLON]}$$

$$\mathcal{I}[\![skip]\!]\ \_\ \_ = \text{id}_{State} \qquad \text{[INSTR\_SKIP]}$$

$$\mathcal{I}[\![\textbf{if } E = 0 \textbf{ then } I_1 \textbf{ else } I_2 \textbf{ fi}]\!]\ \rho\ \rho_P\ (m, s) = \text{ifte}(\mathcal{E}[\![E]\!]\ \rho\ s = 0, \mathcal{I}[\![I_1]\!], \mathcal{I}[\![I_2]\!])\ \rho\ \rho_P\ (m, s) \qquad \text{[INSTR\_IF]}$$

$$\mathcal{I}[\![\textbf{call } p(x)]\!]\ \rho\ \rho_P\ (m, s) = (m, \rho_P\ p\ (\rho\ x)\ s) \qquad \text{[INSTR\_CALL]}$$

$$\mathcal{I}[\![\textbf{begin } D; I \textbf{ end}]\!]\ \rho\ \rho_P\ (m, s) = \mathcal{I}[\![I]\!]\ \rho'\ \rho'_P\ (m, s') \qquad \text{[INSTR\_BLOCK]}$$

where

$$(\rho', \rho'_P, s') = \mathcal{D}[\![D]\!]\ \rho\ \rho_P\ s$$