Semantyka i Weryfikacja programów

Praca domowa 2.

Hubert Michalski hm438596

6 lutego 2024

1 Zadanie

Podaj semantykę denotacyjną dla języka o następującej gramatyce:

$\mathbf{Num} \ni n ::= \ 0 \mid 1 \mid \ -1 \mid 2 \mid \ -2 \mid \ldots$
$\mathbf{Var} \ni x ::= \ x \mid y \mid \ldots$
$\mathbf{PVar} \ni p ::= \ p \mid q \mid \ldots$
$\mathbf{Expr} \ni E ::= \ n \mid x \mid E_1 + E_2 \mid E_1 - E_2$
$\mathbf{Instr} \ni I ::= x := E \mid I_1; I_2 \mid \mathbf{skip} \mid \mathbf{if}\ E = 0\ \mathbf{then}\ I_1\ \mathbf{else}\ I_2 \mid \mathbf{begin}\ d\ I\ \mathbf{end} \mid \mathbf{call}\ p(x) \mid \mathbf{export}\ p \mid \mathbf{exit}\ p$
$\mathbf{Decl} \ni d ::= \mathbf{var}\ x := E \mid \mathbf{proc}\ p(x)\ \mathbf{is}\ (I) \mid d_1; d_2$

2 Rozwiązanie

Wszystko jest zdefiniowane standardowo jak na wykładzie z wyjątkiem dodatkowego "środowiska" przekazywanego do wykonywanej procedury. **XEnv** mapuje nazwy procedury na: komórkę pamięci argumentu formalnego procedury, komórkę pamięci argumentu aktualnego, którego używamy podczas wykonywania procedury oraz kontynuację zza momentu wywołania procedury. Pierwsza komórka służy oczywiście temu, żeby wiedzieć gdzie przypisać nową wartość podczas wykonywania **export** lub **exit**. Druga jest po to, żebyśmy wiedzieli jaką wartość chcemy przypisać podczas wykonywania **export**. Kontynuacja zaś jest potrzebna, aby procedura wiedziała co zrobić po wykonaniu instrukcji **exit**. Dla pełni jasności dodam jeszcze, że pamięć na nowa zmienną jest alokowana w momencie wywołania procedury a w deklaracji procedury jest ona odczytywana z **XEnv**, podobnie jest także z kontynuacją tzn. **PROC** nie przyjmuje bezpośrednio kontynuacji lecz także ją odczytuje ze środowiska.

2.1 Dziedziny semantyczne

$$\begin{aligned}
\mathbf{Int} &= \{0, 1, -1, 2, -2, \dots\} \\
\mathbf{Loc} &= \{0, 1, 2, 3, \dots\} \\
\\
\mathbf{VEnv} &= \mathbf{Var} \to \mathbf{Loc} \\
\mathbf{PEnv} &= \mathbf{PVar} \to \mathbf{PROC} \\
\mathbf{XEnv} &= \mathbf{PVar} \to (\mathbf{Loc} \times \mathbf{Loc} \times \mathbf{Cont}) \\
\mathbf{Store} &= \mathbf{Loc} \to \mathbf{Int} \\
\\
\mathbf{Cont} &= \mathbf{Store} \to \mathbf{Ans} \\
\mathbf{Cont_E} &= \mathbf{Int} \to \mathbf{Cont} \\
\mathbf{Cont_D} &= \mathbf{VEnv} \to \mathbf{PEnv} \to \mathbf{Cont} \\
\\
\mathbf{PROC} &= \mathbf{XEnv} \to \mathbf{Int} \to \mathbf{Cont} \\
\mathbf{EXPR} &= \mathbf{VEnv} \to \mathbf{Cont_E} \to \mathbf{Cont} \\
\mathbf{DECL} &= \mathbf{VEnv} \to \mathbf{PEnv} \to \mathbf{Cont_D} \to \mathbf{Cont} \\
\mathbf{INSTR} &= \mathbf{VEnv} \to \mathbf{PEnv} \to \mathbf{XEnv} \to \mathbf{Cont} \to \mathbf{Cont}
\end{aligned}$$

2.2 Funkcje semantyczne

$$\begin{aligned}
\mathcal{N} &: \mathbf{Num} \to \mathbf{Int} \\
\mathcal{E} &: \mathbf{Expr} \to \mathbf{EXPR} \\
I &: \mathbf{Instr} \to \mathbf{INSTR} \\
D &: \mathbf{Decl} \to \mathbf{DECL}
\end{aligned}$$

2.3 Równania semantyczne

$$\boxed{I : \mathbf{Instr} \to \mathbf{INSTR}}$$

$I[\![x{:=}E]\!]\rho_V \rho_P \rho_X \kappa = \mathcal{E}[\![E]\!]\rho_V(\lambda n.\lambda s.\ \kappa s[l \mapsto n])$

$\qquad$ **where** $l := \rho_V(x)$

$I[\![I_1;I_2]\!]\rho_V \rho_P \rho_X \kappa = I[\![I_1]\!]\rho_V \rho_P \rho_X (I[\![I_2]\!]\rho_V \rho_P \rho_X \kappa)$

$I[\![\mathbf{skip}]\!]\rho_V \rho_P \rho_X \kappa = \kappa$

$I[\![\mathbf{if}\ E = 0\ \mathbf{then}\ I_1\ \mathbf{else}\ I_2]\!]\rho_V \rho_P \rho_X \kappa = \mathcal{E}[\![E]\!]\rho_V(\lambda n.\ \mathbf{if}\ n = 0\ \mathbf{then}\ I[\![I_1]\!]\rho_V \rho_P \rho_X \kappa\ \mathbf{else}\ I[\![I_2]\!]\rho_V \rho_P \rho_X \kappa)$

$I[\![\mathbf{begin}\ d\ I\ \mathbf{end}]\!]\rho_V \rho_P \rho_X \kappa = D[\![d]\!]\rho_V \rho_P (\lambda \rho'_V.\lambda \rho'_P.\ I[\![I]\!]\rho'_V \rho'_P \rho_X \kappa)$

$I[\![\mathbf{call}\ p(x)]\!]\rho_V \rho_P \rho_X \kappa = \lambda s.\ \rho_P\big(p\big)(\rho_X[p \mapsto (l, l', \kappa)])(n)(s)$

$\qquad$ **where** $l := \rho_V(x), l' = newloc(s), n := s(l)$

$I[\![\mathbf{export}\ p]\!]\rho_V \rho_P \rho_X \kappa = \lambda s.\ \kappa s[l \mapsto s(l')]$

$\qquad$ **where** $(l, l', \kappa') := \rho_X(p)$

$I[\![\mathbf{exit}\ p]\!]\rho_V \rho_P \rho_X \kappa = \lambda s.\ \kappa' s[l \mapsto s(l) + 1]$

$\qquad$ **where** $(l, l', \kappa') := \rho_X(p)$

$$\boxed{D : \mathbf{Decl} \to \mathbf{DECL}}$$

$D[\![\mathbf{var}\ x{:=}E]\!]\rho_V \rho_P \kappa_D = \mathcal{E}[\![E]\!]\rho_V(\lambda n.\lambda s.\ \kappa_D \rho_V[x \mapsto l]\rho_P s[l \mapsto n])$

$\qquad \textbf{where}\ l := newloc(s)$

$D[\![\mathbf{proc}\ p(x)\ \mathbf{is}\ (I)]\!]\rho_V \rho_P \kappa_D = \kappa_D \rho_V \rho_P[p \mapsto Fix(\Phi)]$

$\qquad \textbf{where}\ \Phi(F) = \lambda\rho_X.\lambda n.\big(\lambda s.\ I[\![I]\!]\rho_V[x \mapsto l']\rho_P[p \mapsto F]\rho_X \kappa' s[l' \mapsto n]\big)$

$\qquad \textbf{where}\ (l, l', \kappa') := \rho_X(p)$

$D[\![d_1;d_2]\!]\rho_V \rho_P \kappa_D = D[\![d_1]\!]\rho_V \rho_P(\lambda\rho'_V.\lambda\rho'_P.\ D[\![d_2]\!]\rho'_V \rho'_P \kappa_D)$

$$\boxed{\mathcal{E} : \mathbf{Expr} \to \mathbf{EXPR}}$$

$\mathcal{E}[\![n]\!]\rho_V \kappa_E = \kappa_E\big(\mathcal{N}[\![n]\!]\big)$

$\mathcal{E}[\![x]\!]\rho_V \kappa_E = \lambda s.\ \kappa_E\big(s(\rho_V(x))\big)s$

$\mathcal{E}[\![E_1{+}E_2]\!]\rho_V \kappa_E = \mathcal{E}[\![E_1]\!]\rho_V\big(\lambda n_1.\ \mathcal{E}[\![E_2]\!]\rho_V(\lambda n_2.\ \kappa_E(n_1 + n_2))\big)$

$\mathcal{E}[\![E_1{-}E_2]\!]\rho_V \kappa_E = $ analogicznie...