Zadanie 2

Bartosz Kucypera, bk439964

Dziedziny syntaktyczne

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

Dziedziny semantyczne

$\mathbf{Int} = \ldots$
$\mathbf{Loc} = \ldots$
$\mathbf{Store} = \mathbf{Loc} \to \mathbf{Int}$
$\mathbf{VEnv} = \mathbf{Var} \to \mathbf{Loc}$
$\mathbf{PEnv} = \mathbf{PVar} \to \mathbf{Proc}$
$\mathbf{AEnv} = \mathbf{PVar} \to \mathbf{Loc}$
$\mathbf{EEnv} = \mathbf{PVar} \to \mathbf{Loc}$
$\mathbf{Cont} = \mathbf{Store} \to \mathbf{Store}$
$\mathbf{Cont}_\mathrm{E} = \mathbf{Int} \to \mathbf{Cont}$
$\mathbf{Cont}_\mathrm{D} = \mathbf{VEnv} \to \mathbf{PEnv} \to \mathbf{Cont}$
$\mathbf{Cont}_\mathrm{P} = \mathbf{VEnv} \to \mathbf{PEnv} \to \mathbf{AEnv} \to \mathbf{EEnv} \to \mathbf{Loc} \to \mathbf{Cont}$
$\mathbf{Proc} = \mathbf{Cont} \to \mathbf{Cont}_\mathrm{P}$

AEnv dla każdej procedury pamięta lokację zmiennej ”z której eksportujemy”.

EEnv dla każdej procedury pamięta lokację zmiennej ”do której eksportujemy”.

Funkcje semantyczne

$\mathcal{N}$: $\mathbf{Num} \to \mathbf{Int}$

$\mathcal{E}$: $\mathbf{Expr} \to \underbrace{\mathbf{VEnv} \to \mathbf{Cont}_\mathrm{E} \to \mathbf{Cont}}_{\textbf{EXPR}}$

$\mathcal{D}$: $\mathbf{Decl} \to \underbrace{\mathbf{VEnv} \to \mathbf{PEnv} \to \mathbf{Cont}_\mathrm{D} \to \mathbf{Cont}}_{\textbf{DECL}}$

$\mathcal{I}$: $\mathbf{Instr} \to \underbrace{\mathbf{VEnv} \to \mathbf{PEnv} \to \mathbf{AEnv} \to \mathbf{EEnv} \to \mathbf{Cont} \to \mathbf{Cont}}_{\textbf{INSTR}}$

Klauzule semantyczne

Wyrażenia

$\mathcal{E}[\![\mathbf{n}]\!] = n : \mathbf{Int}$ where $n = \mathcal{N}[\![\mathbf{n}]\!]$

$\mathcal{E}[\![x]\!]\ \rho_V \kappa_E = \lambda s : \mathbf{Store}.\kappa_E\ n\ s$ where $l = \rho_V\ x, n = s\ l$

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

$\mathcal{E}[\![E_1 - E_2]\!]\ \rho_V \kappa_E = \mathcal{E}[\![E_1]\!]\rho_V\ \lambda n_1 : \mathbf{Int}.\ \mathcal{E}[\![E_2]\!]\rho_V \lambda n_2 : \mathbf{Int}.\kappa_E(n_1 - n_2)$

Deklaracje

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

$\mathcal{D}[\![\mathbf{proc}\ p(x)\ \mathbf{is}\ (I)]\!]\rho_V \rho_P \kappa_D = \kappa_D \rho_V \rho_P [p \mapsto P]$
where $P\kappa\rho_V\ \rho_P\ \rho_A\ \rho_E\ l\ s = \mathcal{I}[\![I]\!]\ \rho_V[x \mapsto j]\rho_P\rho_A[p \mapsto j]\rho_E[p \mapsto l]s[j \mapsto s\,l]$ where $j = newloc(s)$

$\mathcal{D}[\![d_1; d_2]\!]\ \rho_V \rho_P \kappa_D = \mathcal{D}[\![d_1]\!]\ \rho_V \rho_P \lambda \rho'_V . \lambda \rho'_P .\ \mathcal{D}[\![d_2]\!] \rho'_V \rho'_P \kappa_D$

Instrukcje

$\mathcal{I}[\![x := E]\!]\rho_V \rho_P \rho_A \rho_E \kappa = \mathcal{E}[\![E]\!]\rho_V \lambda n : \mathbf{Int}.\lambda s : \mathbf{Store}.\kappa\ s[l \mapsto n]$ where $l = \rho_V\ x$

$\mathcal{I}[\![I_1; I_2]\!]\rho_V \rho_P \rho_A \rho_E \kappa = \mathcal{I}[\![I_1]\!]\rho_V \rho_P \rho_A \rho_E (\mathcal{I}[\![I_2]\!]\rho_V \rho_P \rho_A \rho_E \kappa)$

$\mathcal{I}[\![\mathbf{skip}]\!]\rho_V \rho_P \rho_A \rho_E \kappa = \kappa$

$\mathcal{I}[\![\mathbf{if}\ E = 0\ \mathbf{then}\ I_1\ \mathbf{else}\ I_2]\!]\rho_V \rho_P \rho_A \rho_E \kappa = \mathcal{E}[\![E]\!]\ \rho_V \lambda n : \mathbf{Int}.ifte_{\mathbf{Cont}}(n = 0, \mathcal{I}[\![I_1]\!]\rho_V \rho_P \rho_A \rho_E \kappa, \mathcal{I}[\![I_2]\!]\rho_V \rho_P \rho_A \rho_E \kappa)$

$\mathcal{I}[\![\mathbf{begin}\ d\ I\ \mathbf{end}]\!]\rho_V \rho_P \rho_A \rho_E \kappa = \mathcal{D}[\![d]\!]\ \rho_V \rho_P \lambda \rho'_V . \lambda \rho'_P . \mathcal{I}[\![I]\!]\rho'_V \rho'_P \rho_A \rho_E \kappa$

$\mathcal{I}[\![\mathbf{call}\ p(x)]\!]\ \rho_V \rho_P \rho_A \rho_E \kappa = P \kappa \rho_V \rho_P \rho_A \rho_E l$ where $P = \rho_P p$, $l = \rho_V x$

$\mathcal{I}[\![\mathbf{export}\ p]\!]\rho_V \rho_P \rho_A \rho_E \kappa\ s = \kappa\ s[l \mapsto n]$ where $l = \rho_E p$, $n = s(\rho_A\ p)$

$\mathcal{I}[\![\mathbf{exit}\ p]\!]\ \rho_V \rho_P \rho_A \rho_E \kappa\ s = s[l \mapsto n + 1]$ where $l = \rho_E p$, $n = s\ l$