

Small21: Programma e outermost inline block

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

Prog ::= [prog] Ide Block

Block ::= [block] Dcl Cmd

Sistema Y:

$$\begin{array}{c}
 \text{Bc} = [\text{block}] \text{dscs} \\
 \langle \text{ds}, Y_\rho^0 \rangle \rightarrow_Y ([\text{void}], Y_\rho^1) \\
 \langle \text{cs}, Y_\rho^1 \rangle \rightarrow_Y ([\text{void}], Y_\rho^F) \\
 \hline
 [\text{Y44}] \frac{}{\langle [\text{prog}] \text{I Bc}, Y_\rho^0 \rangle \rightarrow_Y ([\text{void}], Y_\rho^F)}
 \end{array}$$

Gestione Errori di Tipo:

$$\begin{array}{c}
 \langle \text{ds}, Y_\rho^0 \rangle \rightarrow_Y ([\text{terr}], Y_\rho^1) \\
 \hline
 [\text{E96}] \frac{}{\langle [\text{prog}] \text{I Bc}, Y_\rho^0 \rangle \rightarrow_Y ([\text{terr}], Y_\rho^1)}
 \end{array}
 \qquad
 \begin{array}{c}
 \langle \text{ds}, Y_\rho^0 \rangle \rightarrow_Y ([\text{void}], Y_\rho^1) \\
 \langle \text{cs}, Y_\rho^1 \rangle \rightarrow_Y ([\text{terr}], Y_\rho^F) \\
 \hline
 [\text{E97}] \frac{}{\langle [\text{prog}] \text{I Bc}, Y_\rho^0 \rangle \rightarrow_Y ([\text{terr}], Y_\rho^F)}
 \end{array}$$

Regole di Inferenza SEM_{CMD}

Introduciamo Un iteratore di stati per la computazione e lo utilizziamo nella regola [P1] per i programmi:

$$\begin{array}{c}
 \sigma \rightarrow_R ([\text{void}], \sigma_F) \\
 \sigma_F = (\Delta_F, \mu_F) \quad \text{---} \# \Delta_F = 1 \text{---} \\
 \Delta_F = \triangleright \{ \text{id}, \text{lc}, \text{fr}, [], v \} \\
 \hline
 [\text{R1}] \frac{}{\sigma \rightarrow_{R^*} \sigma_F}
 \end{array}
 \qquad
 \begin{array}{c}
 \Delta = \text{ar}_{\text{top}} + \Delta_R \quad \text{---} \# \Delta_F = 1 \text{---} \\
 \text{ar}_{\text{top}} = \{ \text{id}, \text{lc}, \text{fr}, c + \text{cnt}, v \} \\
 \text{ar}'_{\text{top}} = \{ \text{id}, \text{lc}, \text{fr}, \text{cnt}, v \} \\
 \langle c, (\text{ar}'_{\text{top}} + \Delta_R, \mu) \rangle \rightarrow ([\text{void}], \sigma_{\text{next}}) \\
 \hline
 [\text{R2}] \frac{}{(\Delta, \mu) \rightarrow_R ([\text{void}], \sigma_{\text{next}})}
 \end{array}$$

Osservazioni

- Y_ρ^0 : Contesto contenente i binding di tipo per tutti gli identificatori di primitive del Linguaggio;
- $\text{ar} + \Delta$: indica stack con ar come top record e Δ come coda;
- $c + \text{cnt}$: sta per SeqC c cnt , quando la continuazione, cnt , sia una sequenza di comandi;

Small21: Programma e outermost inline block - 2

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

Prog ::= [prog] Ide Block

Block ::= [block] Decl Cmd

Regole di Inferenza SEM_{CMD}

$$[P1] \frac{\begin{array}{l} \{I, 0, \rho^0, [], []\} = ar_0 \\ \triangleright ar_0 = \Delta_0 \quad [] = \mu_0 \\ (\Delta_0, \mu_0) = \sigma_0 \quad Bc = [block] ds cs \\ \langle ds, \sigma_0 \rangle \rightarrow ([void], (\Delta_1, \mu_1)) \\ ar_1 = \{id, lc, fr, cnt, v\} \\ \{id, lc, fr, cs, v\} = ar_2 \\ \triangleright ar_0 = \Delta_2 \quad (\Delta_2, \mu_1) = \sigma_2 \\ \sigma_2 \rightarrow_R (\Delta_F, \mu_F) \quad \# \Delta_F = 1 \end{array}}{\langle [prog] I Bc, \rangle \rightarrow ([void], (\Delta_F, \mu_F))}$$

Osservazioni

- ρ^0 : Contesto contenente i binding di tipo per tutti gli identificatori di primitive del Linguaggio;
- $\langle [prog] I Bc, \rangle \rightarrow h$: Introduce una differente inferenza $[prog] I Bc \rightarrow_p h$. Il nuovo sistema ha solo [P1].

Small21: inline block

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

$\text{BlockS} ::= [\text{blockS}] \text{Decl Cmd}$

Sistema Y:

[Y32] _____

Gestione Errori di Tipo:

[E51] _____

[E52] _____

Regole di Inferenza SEM_{CMD}

[S13] _____

Osservazioni

- $\text{ar} + \Delta$: indica l'estensione dello stack Δ con il record ar posto come top del nuovo stack;
- [S13] : Dovremo optare per questa data la presenza del "return" tra gli statements;

Small21. Proc-Fun block: Call

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

Stm ::= [call] Ide Avars

Sistema Y:

$$Y_\rho(\text{ide}) = [\text{abs}] [\text{void}] [::] t'$$
$$\langle \text{aps}, Y_\rho \rangle \rightarrow_Y (\text{ta}, Y_\rho)$$
$$t' = \text{ta}$$
$$[\text{Y25}] \frac{}{\langle [\text{call}] \text{ide aps}, Y_\rho \rangle \rightarrow_Y (\text{void}, Y_\rho)}$$

Gestione Errori di Tipo:

$$[\text{E45}] \frac{Y_\rho(\text{ide}) \neq [\text{abs}] [\text{void}] [::] t'}{\langle [\text{call}] \text{ide aps}, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

$$[\text{E46}] \frac{Y_\rho(\text{ide}) = \perp}{\langle [\text{call}] \text{ide aps}, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

$$Y_\rho(\text{ide}) \neq [\text{abs}] [\text{void}] [::] t'$$
$$\langle \text{aps}, Y_\rho \rangle \rightarrow_Y (\text{ta}, Y_\rho)$$
$$t' \neq \text{ta}$$
$$[\text{E47}] \frac{}{\langle [\text{call}] \text{ide aps}, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

Osservazioni

- **Controllo Incompleto:** Il controllo ignora il tipo di trasmissione utilizzato;

Small21. Proc-Fun block: Call - 2

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

$Stm ::= [call] Ide Avars$

Regole di Inferenza SEM_{CMD}

[S14] _____

Osservazioni

- \rightarrow_{TR1} : Introduce una nuova inferenza per trattare la trasmissione di parametri;
- epi_r : Le forme di trasmissione attualmente in Small21 non richiedono un epilogo, pertanto il valore di epi_r è irrilevante e la regola di call non provvede ad avviare alcun epilogo prima di rimuovere il corrente ar top ed uscire dal blocco procedura.
- Deallocazione di Memoria: All'uscita di un blocco la maggior parte dei Linguaggi provvedono a Deallocare la memoria allocata o comunque acceduta dal blocco e non più accessibile dal Programma. AM21 ha operazioni per farlo, ma la semantica dinamica data npn provvede ad alcuna forma di deallocazione.

Small21. Proc-Fun block: Apply

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

Exp ::= [apply] Ide Apars

Sistema Y:

[Y26] _____

Gestione Errori di Tipo:

[E32] _____

[E33] _____

[E34] _____

Osservazioni

- **Controllo Incompleto:** Il controllo ignora il tipo di trasmissione utilizzato;

Small21. Proc-Fun block: Apply - 2

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

Exp ::= [Apply] Ide Apars

Regole di Inferenza SEM_{EXP}

[S15] _____

Osservazioni

- \rightarrow_{TR1} : Introduce una nuova inferenza per trattare la trasmissione di parametri;
- epi_T : Le forme di trasmissione attualmente in Small21 non richiedono un epilogo, pertanto il valore di epi_T è irrilevante e la regola di call non provvede ad avviare alcun epilogo prima di rimuovere il corrente ar top ed uscire dal blocco procedura.
- Deallocazione di Memoria: All'uscita di un blocco la maggior parte dei Linguaggi provvedono a Deallocare la memoria allocata o comunque acceduta dal blocco e non più accessibile dal Programma. AM21 ha operazioni per farlo, ma la semantica dinamica data npn provvede ad alcuna forma di deallocazione.
- call ed apply: hanno semantica dinamica apparentemente identica: Il confronto tra le regole dice questo. In effetti, anche apply opera su una procedura (funzionale, o una funzione procedurale). Tuttavia, le 2 regole hanno comportamenti diversi nel valore calcolato, ovvero l'ultimo componente dell'Activation Record con cui le due invocazione sono gestite (vedi comportamento del comando return).

Small21: Trasmissione Parametri

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

$\text{FPars} ::= [\text{fp}] \text{PPF Type Ide} \mid [\text{emptyFP}]$

$\text{PPF} ::= [\text{value}] \mid [\text{ref}]$

$\text{APars} ::= [\text{ap}] \text{Exp} \mid [\text{emptyAP}]$

Sistema Y:

$$\begin{array}{c} \text{fps} = [\text{fp}][\text{value}] \text{ t ide} \\ \text{aps} = [\text{ap}] \text{ exp} \\ \langle \text{exp}, Y_\rho \rangle \rightarrow_Y (\text{ta}, Y_\rho) \\ \text{t} = \text{ta} \quad \text{t} \in \text{Simple} \\ \hline [\text{Y35}] \frac{}{\langle \text{fps} \triangleleft \text{aps}, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y_\rho)} \end{array} \quad \begin{array}{c} \text{fps} = [\text{fp}][\text{ref}] \text{ t ide} \\ \text{aps} = [\text{ap}] \text{ exp} \\ \langle \text{exp}, Y_\rho \rangle \rightarrow_{DY} ([\text{mut}] \text{ta}, Y_\rho) \\ \text{t} = \text{ta} \\ \hline [\text{Y36}] \frac{}{\langle \text{fps} \triangleleft \text{aps}, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y_\rho)} \end{array} \quad \begin{array}{c} \text{fps} = [\text{emptyFP}] \\ \text{aps} = [\text{emptyAP}] \\ \hline [\text{Y37}] \frac{}{\langle \text{fps} \triangleleft \text{aps}, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y_\rho)} \end{array}$$

Gestione Errori di Tipo:

$$\begin{array}{c} \dots \\ \langle \text{exp}, Y_\rho \rangle \rightarrow_{DY} (\text{terr}, Y_\rho) \\ \hline [\text{E59}] \frac{}{\langle \text{fps} \triangleleft \text{aps}, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)} \end{array} \quad \begin{array}{c} \dots \\ \langle \text{exp}, Y_\rho \rangle \rightarrow_{DY} (\text{terr}, Y_\rho) \\ \hline [\text{E60}] \frac{}{\langle \text{fps} \triangleleft \text{aps}, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)} \end{array}$$
$$\begin{array}{c} \dots \\ \text{t} \neq \text{ta} \\ \hline [\text{E61}] \frac{}{\langle \text{fps} \triangleleft \text{aps}, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y_\rho)} \end{array} \quad \begin{array}{c} \text{fps} = [\text{emptyFp}] \\ \text{aps} \neq [\text{emptyAp}] \\ \hline [\text{E62}] \frac{}{\langle \text{fps} \triangleleft \text{aps}, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y_\rho)} \end{array} \quad \begin{array}{c} \text{fps} \neq [\text{emptyFp}] \\ \text{aps} = [\text{emptyAp}] \\ \hline [\text{E63}] \frac{}{\langle \text{fps} \triangleleft \text{aps}, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y_\rho)} \end{array}$$

Osservazioni

- **Compatibilità:** degli attuali rispetto alla forma di trasmissione usata e al binding da produrre. Se le 2 regole sono applicabili allora la trasmissione è corretta ovvero ha tipo $[\text{void}]$.
- Quando la trasmissione è corretta rispetto ai tipi e alla compatibilità, allora possiamo applicarla durante l'invocazione di procedure e funzioni, utilizzando l'inferenza \rightarrow_{TR1} .

Small21: Trasmissione Parametri

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

$\text{FPars} ::= [\text{fp}] \text{PPF Type Ide} \mid [\text{emptyFP}]$

$\text{PPF} ::= [\text{value}] \mid [\text{ref}]$

$\text{APars} ::= [\text{ap}] \text{Exp} \mid [\text{emptyAP}]$

Regole di Inferenza SEM_{CMD}

$$\begin{array}{c} \text{fps} = [\text{fp}][\text{value}] \text{t ide} \\ \text{aps} = [\text{ap}] \text{exp} \\ \langle \text{exp}, (\Delta, \mu) \rangle \rightarrow [\text{ta}, \text{va}, (\Delta_1, \mu_1)] \\ \text{t} \in \text{Simple} \quad \text{t} = \text{ta} \quad \Delta_c |_0(\text{I}) = \perp \\ \quad \triangleright (\mu_1, \text{t}, 1) = (\text{loc}_t, \mu_2) \\ \quad \quad \mu_2[\text{loc}_t \leftarrow \text{va}] = \mu_3 \\ \quad \quad [\text{I}/([\text{Mut}] \text{t}, \text{loc}_t)] \otimes \Delta_c = \Delta_c^F \\ \quad \quad (\Delta_c^F, \mu_3) = \sigma_r \quad [] = \text{epi}_r \\ \hline [\text{S16}] \quad \langle \text{fps} \triangleleft \text{aps}, (\Delta, \Delta_c, \mu) \rangle \rightarrow_{\text{TR1}} (\sigma_r, \text{epi}_r) \end{array}$$

$$\begin{array}{c} \text{fps} = [\text{fp}][\text{Ref}] \text{t ide} \\ \text{aps} = [\text{ap}] \text{exp} \\ \langle \text{exp}, (\Delta, \mu) \rangle \rightarrow_{\text{DEN}} [[\text{Mut}] \text{ta}, \text{loca}, (\Delta_1, \mu_1)] \\ \text{t} \in \text{Simple} \quad \text{t} = \text{ta} \quad \Delta_c |_0(\text{I}) = \perp \\ \quad \quad [\text{I}/([\text{Mut}] \text{ta}, \text{loca})] \otimes \Delta_c = \Delta_c^F \\ \quad \quad (\Delta_c^F, \mu_3) = \sigma_r \quad [] = \text{epi}_r \\ \hline [\text{S17}] \quad \langle \text{fps} \triangleleft \text{aps}, (\Delta, \Delta_c, \mu) \rangle \rightarrow_{\text{TR1}} ([\text{void}], \gamma_\rho) \end{array}$$

Osservazioni

Small21: Return e Uscita da un Proc-Fun Block

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

Cmd ::= [return] Exp

Sistema Y:

$$[Y38] \frac{\langle e, Y_\rho \rangle \rightarrow_Y (t, Y_\rho) \quad t \in \text{Simple}}{\langle [\text{return}] e, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y_\rho)}$$

Gestione Errori di Tipo:

$$[E64] \frac{\langle e, Y_\rho \rangle \rightarrow ([\text{terr}], Y_\rho)}{\langle [\text{return}] e, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

$$[E65] \frac{t \notin \text{Simple}}{\langle [\text{return}] e, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

Regole di Inferenza SEM_{CMD}

$$[S18] \frac{\langle e, \sigma \rangle \rightarrow [t_e, v_e, (\Delta_e, \mu_e)] \quad t_e \in \text{Simple} \quad \Delta_e = \text{ar} + \Delta_r \quad \text{ar} = \{\text{id}, \text{ls}, \text{fr}, \text{cnt}, v\} \quad \{\text{id}, \text{ls}, \text{fr}, [], v_e\} = \text{ar1} \quad (\text{ar1} + \Delta_r, \mu_e) = \sigma_F}{\langle [\text{return}] e, \sigma \rangle \rightarrow ([\text{void}], \sigma_F)}$$

Osservazioni

- **Controllo Incompleto:** Occorre conoscere il tipo calcolato dalla prc/fun in cui occorre il comando e verificarne equivalenza con quello dell'argomento di return

Small21: Sequenza di Statements

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.

$\text{Stm} ::= [\text{seqS}] \text{Stm} \text{Stm}$

Sistema Y:

$$[\text{Y39}] \frac{\langle s_1, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y'_\rho) \quad \langle s_2, Y'_\rho \rangle \rightarrow_Y ([\text{void}], Y''_\rho)}{\langle [\text{seqS}] s_1 s_2, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y''_\rho)}$$

Gestione Errori di Tipo:

$$[\text{E66}] \frac{\langle s_1, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y'_\rho)}{\langle [\text{seqS}] s_1 s_2, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y'_\rho)} \quad [\text{E67}] \frac{\langle s_1, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y'_\rho) \quad \langle s_2, Y'_\rho \rangle \rightarrow_Y ([\text{terr}], Y''_\rho)}{\langle [\text{seqC}] s_1 s_2, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y''_\rho)}$$

Regole di Inferenza SEM_{CMD}

$$[\text{S19}] \frac{\begin{array}{l} \Delta = \text{ar} + \Delta_r \\ \text{ar} = \{\text{hd}, \text{cs}, \text{fr}, \text{cnt}, \text{v}\} \\ [\text{seqC}] s_2 \text{cnt} = \text{cnt1} \\ \{\text{hd}, \text{cs}, \text{fr}, \text{cnt1}, \text{v}\} = \text{ar1} \\ \text{ar1} + \Delta_r = \Delta_2 \end{array} \quad \langle s_1, (\Delta_2, \mu) \rangle \rightarrow ([\text{void}], \sigma_F)}{\langle [\text{seqS}] s_1 s_2, (\Delta, \mu) \rangle \rightarrow ([\text{void}], \sigma_F)}$$

Osservazioni

- Sottotipo: Nell'uso degli AST, le categorie della Sintassi Astratta sono assimilate ai Tipi. Pertanto la scrittura

$\text{Cmd} ::= \dots \mid \text{Stm} \mid \dots$,

utilizzata nella Sintassi Astratta, dice che in AM21 il tipo `Stm` è sottotipo di `Cmd` e i suoi valori (ed espressioni) possono essere utilizzati ovunque sia richiesto un tipo `Cmd`.

Attenzione. In Ocaml, questo non è vero e dobbiamo usare il costruttore `UnL` per fare boxing-unboxing.

Small21: Operazioni su Array

Il Sistema definisce il comportamento dei comandi sulla Macchina Astratta AM21.