

- **Punti Critici** trattiamo questi per primi
 - : Comandi etichettati;
 - : Sequenza di Comandi;
 - : Comando/Statement Goto;

- **Metodo** Seguire sempre:
 - : Relazione Funzione Semantica - Regole Inferenza Costrutto ;
 - : Regole Inferenza Semantica Statica Costrutto;
(Controllo dei tipi e proprietà contestuali)
 - : Regole di Inferenza Semantica Dinamica Costrutto;
(Comportamento Computazione)

- I costrutti dei punti critici sono tutti comandi

cmdSem : $\text{cmd} \rightarrow \text{State} \rightarrow \text{Type} * \text{State}$

$(\forall c, \sigma) \quad \text{cmdSem}(c, \sigma) = (t, \sigma') \quad \text{iff} \quad \langle c, \sigma \rangle \rightarrow (t, \sigma') \in \text{Sem}_{\text{CMD}}$

stmSem : $\text{stm} \rightarrow \text{State} \rightarrow \text{Type} * \text{State}$

$(\forall s, \sigma) \quad \text{stmSem}(s, \sigma) = (t, \sigma') \quad \text{iff} \quad \langle s, \sigma \rangle \rightarrow (t, \sigma') \in \text{Sem}_{\text{STM}}$

Sistema Y: Regole per CMD, STM, PROG

$$[Y19] \frac{\begin{array}{c} \#Y_\rho = 1 \\ Y_\rho |_0(L) = \perp \quad [L/[lab]] \otimes Y_\rho = Y'_\rho \\ \langle s, Y'_\rho \rangle \rightarrow_Y ([void], Y'_\rho) \end{array}}{\langle L [:] s, Y_\rho \rangle \rightarrow_Y ([void], Y'_\rho)}$$

$$[Y22] \frac{\begin{array}{c} \#Y_\rho = 1 \\ Y_\rho |_0(L) = [lab] \end{array}}{\langle [goto] L, Y_\rho \rangle \rightarrow_Y ([void], Y_\rho)}$$

$$[Y23] \frac{\begin{array}{c} \#Y_\rho = 1 \\ Y_\rho |_0(L) = \perp \\ \text{JumpAheadCheck} \end{array}}{\langle [goto] L, Y_\rho \rangle \rightarrow_Y ([void], Y_\rho)} \quad (***)$$

$$[Y24] \frac{\begin{array}{c} \#Y_\rho = 1 \\ \langle c_1, Y_\rho \rangle \rightarrow_Y ([void], Y'_\rho) \\ \langle s_2, Y'_\rho \rangle \rightarrow_Y ([void], Y''_\rho) \end{array}}{\langle [seqC] c_1 c_2, Y_\rho \rangle \rightarrow_Y ([void], Y''_\rho)}$$

Gestione Errori di Tipo:

$$[E35] \frac{\#Y_\rho \neq 1}{\langle L [:] s, Y_\rho \rangle \rightarrow_Y ([terr], Y'_\rho)}$$

$$[E36] \frac{Y_\rho |_0(L) \neq \perp}{\langle L [:] s, Y_\rho \rangle \rightarrow_Y ([terr], Y'_\rho)}$$

$$[E37] \frac{\begin{array}{c} \dots \\ \langle s, Y'_\rho \rangle \rightarrow_Y ([terr], Y'_\rho) \end{array}}{\langle L [:] s, Y_\rho \rangle \rightarrow_Y ([terr], Y'_\rho)}$$

Notazione e Osservazioni: Costruttori Aggiuntivi

- **JumpAheadCheck**: Richiede Controllo a run-time (trattabile in Semantica dinamica);
- *******: In [Y22] e [Y23] dove affermiamo che goto può occore solo nel blocco più esterno di un programma?;

Sistema Y: Regole per CMD, STM, PROG - 2

Gestione Errori di Tipo:

$$[E43] \frac{Y_\rho |_0(L) = t \quad t \neq [\text{lab}]}{\langle [\text{goto}] L, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)} \quad [E44] \frac{Y_\rho |_0(L) = \perp \quad \text{JumpAheadCheckFail}}{\langle [\text{goto}] L, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)} \quad [E45] \frac{\#Y_\rho \neq 1}{\langle [\text{goto}] L, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

$$[E56] \frac{\#Y_\rho \neq 1}{\langle [\text{seqC}] c_1 c_2, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y'_\rho)}$$

$$[E57] \frac{\langle c_1, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y'_\rho)}{\langle [\text{seqC}] c_1 c_2, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y'_\rho)}$$

$$[E58] \frac{\langle c_1, Y_\rho \rangle \rightarrow_Y ([\text{void}], Y'_\rho) \quad \langle s_2, Y'_\rho \rangle \rightarrow_Y ([\text{terr}], Y''_\rho)}{\langle [\text{seqC}] c_1 c_2, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y''_\rho)}$$

Notazione e Osservazioni: Costruttori Aggiuntivi

- [E45]:Perchè anche questa regola?

Comandi Small21 - Le Regole di Inferenza SEM_{CMD}

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

Controllo dei Tipi Dinamico: Il Sistema SEM_{CMD} è Integrato con il Sistema Y (ultime 2 slide sopra, per i comandi).

Cmd ::= Lab [:] Stm | Stm | [seqC] Cmd Cmd | ...

$$\begin{array}{c} \# \Delta = 1 \quad \Delta = \triangleright ar_1 \\ ar_1 = \{hd, cs, fr, cnt, v\} \\ [seqC] s \ cnt = c' \\ \Delta|_0(L) = \perp \quad [L/([lab], c')] \otimes \Delta = \Delta' \\ \langle s, (\Delta', \mu) \rangle \rightarrow ([void], \sigma_F) \\ [C1] \frac{}{\langle L [:] s, (\Delta, \mu) \rangle \rightarrow ([void], \sigma_F)} \end{array} \quad (**\text{ corretto13/05/21 } **)$$

$$\begin{array}{c} \# \Delta = 1 \quad \Delta = \triangleright ar_1 \\ ar_1 = \{hd, cs, fr, cnt, v\} \\ [seqC] c_2 \ cnt = cnt' \\ \{hd, cs, fr, cnt', v\} = ar'_1 \\ \triangleright ar'_1 = \Delta' \\ \langle c_1, (\Delta', \mu) \rangle \rightarrow ([void], \sigma_F) \\ [C2] \frac{}{\langle [seqC] c_1 c_2, (\Delta, \mu) \rangle \rightarrow ([void], \sigma_F)} \end{array}$$

Notazione e Osservazioni: Costruttori Aggiuntivi

- Denotazione: Estese con (t, C) per le label con $t = \text{Lab}$ e C una continuazione (rappresentata come sotto)
- Continuazione C : Rappresentata con AST di Small21e c in testa alla continuazione C ;
- $[seqC] c C$: Aggiunge c in testa alla continuazione C ;
- $(*\text{corretto13/05/21}*)$: Perché una volta incontrata e legata nel frame, una label deve essere ignorata?;
- Il predicato $\Delta = \triangleright ar_1$ implica $\Delta = \triangleright ar_1$: Perché ?;

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

Controllo dei Tipi Dinamico: Il Sistema SEM_{CMD} è Integrato con il Sistema Y.

Cmd ::= [goto] Lab | ...

$$\begin{array}{l}
 \Delta|_0(L) = ([lab], c) \\
 \Delta = >ar_1 \\
 ar_1 = \{hd, cs, fr, cnt, v\} \\
 \{hd, cs, fr, emptyC, v\} = ar'_1 \\
 (>ar'_1, \mu) = \sigma' \\
 [S3] \frac{\langle c, \sigma' \rangle \rightarrow ([void], \sigma_F)}{\langle [goto] L, (\Delta, \mu) \rangle \rightarrow ([void], \sigma_F)}
 \end{array}$$

$$\begin{array}{l}
 \Delta|_0(L) = \perp \quad \Delta = >ar_1 \\
 ar_1 = \{hd, cs, fr, cnt, v\} \\
 \text{JACL } cnt = [seqC] c \text{ cs} \\
 >ar'_1 = \Delta' \quad c = L [:] s \\
 \{hd, cs, fr, cs, v\} = ar'_1 \\
 [seqC] s \text{ cs} = c' \\
 [L/([lab], c')] \otimes \Delta' = \Delta'' \\
 [S3] \frac{\langle s, (\Delta'', \mu) \rangle \rightarrow ([void], \sigma_F)}{\langle [goto] L, (\Delta, \mu) \rangle \rightarrow ([void], \sigma_F)} \quad (***)
 \end{array}$$

Notazione ed Osservazioni: Costruttori Aggiuntivi

- [seqC] c C: Aggiunge c in testa alla continuazione C;
- JACL C = C': Vera quando C' è la sotto-sequenza terminale con primo comando etichettato con L. Falsa altr.
- emptyC: Indica "nessuna continuazione"
- (***) : Usa l'invariante di inserire il binding di una label e successivamente ignorarne la presenza ogni prima volta che si seleziona un comando etichettato e se ne considera la trasformazione definita.

Nucleo di Small21 per Macchina a Stato Minsky-Wang

Costrutti Minori: Semantica Statica e Dinamica -1

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

$Stm ::= [ifT] Exp Stm \mid \dots$

Sistema Y:

$$[Y21] \frac{\begin{array}{l} \langle e, Y_\rho \rangle \rightarrow_Y ([bool], Y_\rho) \\ \langle s_1, Y_\rho \rangle \rightarrow_Y ([void], Y_\rho) \end{array}}{\langle [ifT] e s_1, Y_\rho \rangle \rightarrow_Y ([void], Y_\rho)}$$

Gestione Errori di Tipo:

$$[E41] \frac{\begin{array}{l} \langle e, Y_\rho \rangle \rightarrow_Y (t, Y_\rho) \\ t \neq [bool] \end{array}}{\langle [ifT] e s_1, Y_\rho \rangle \rightarrow_Y ([terr], Y_\rho)}$$

$$[E42] \frac{\langle s_1, Y_\rho \rangle \rightarrow_Y ([terr], Y_\rho)}{\langle [ifT] e s_1, Y_\rho \rangle \rightarrow_Y ([terr], Y_\rho)}$$

Regole di Inferenza SEM_{CMD}

$$[S2] \frac{\begin{array}{l} \langle e, \sigma \rangle \rightarrow [[bool], true, \sigma_e] \\ \langle c, \sigma_e \rangle \rightarrow ([void], \sigma_1) \end{array}}{\langle [ifT] e c, \sigma \rangle \rightarrow ([void], \sigma_1)}$$

$$[S2'] \frac{\langle e, \sigma \rangle \rightarrow [[bool], false, \sigma_e]}{\langle [ifT] e c, \sigma \rangle \rightarrow ([void], \sigma_e)}$$

Osservazioni

- Definizione Completa dello Statement con guardia: `Costrutto condizionale if;`

Nucleo di Small21 per Macchina a Stato Minsky-Wang

Costrutti Minori: Semantica Statica e Dinamica -1

$\text{Stm} ::= \text{Exp} [=] \text{Exp} \mid \dots$

Sistema Y:

$$[\text{Y15}] \frac{\langle e_1, Y_\rho \rangle \rightarrow_{\text{DY}} (t_1, Y_\rho) \quad \langle e_r, Y_\rho \rangle \rightarrow_Y (t_r, Y_\rho) \quad \begin{array}{l} t_1 = [\text{Mut}] t \quad t = t_r \quad t \in \text{Simple} \end{array}}{\langle e_1 [=] e_r, Y_\rho \rangle \rightarrow_Y (t, Y_\rho)}$$

Gestione Errori di Tipo:

$$[\text{E17}] \frac{\langle e_1, Y_\rho \rangle \rightarrow_{\text{DY}} ([\text{mut}] t, Y_\rho) \quad t \notin \text{Simple}}{\langle e_1 [=] e_r, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)} \quad [\text{E18}] \frac{\langle e_r, Y_\rho \rangle \rightarrow_{\text{D}} (t, Y_\rho) \quad t \notin \text{Simple}}{\langle e_1 [=] e_r, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)} \quad [\text{E19}] \frac{\langle e_1, Y_\rho \rangle \rightarrow_{\text{DY}} (t_1, Y_\rho) \quad \langle e_r, Y_\rho \rangle \rightarrow_Y (t_r, Y_\rho) \quad t_1 = [\text{Mut}] t \quad t \neq t_r}{\langle e_1 [=] e_r, Y_\rho \rangle \rightarrow_Y (t, Y_\rho)}$$

Regole di Inferenza SEMCMD

$$\text{X11: } \frac{\langle e_r, \sigma \rangle \rightarrow [t_r, v_r, \sigma_r] \quad \langle e_1, \sigma_r \rangle \rightarrow_{\text{DEN}} [t_1, \text{loc}_t, \sigma_1] \quad \begin{array}{l} t_1 = [\text{mut}] t \quad t = t_r \\ t \in \text{Simple} \quad \sigma_1 = (\Delta_1, \mu_1) \\ \mu_1[\text{loc}_t \leftarrow v_r] = \mu_F \end{array}}{\langle e_1 [=] e_r, \sigma \rangle \rightarrow [t, v_r, (\Delta_1, \mu_F)]}$$

Notazione e Osservazioni

- \rightarrow_{DY} è l'inferenza di Tipo del Valore Denotabile di espressioni con doppio significato.
- \rightarrow_{DEN} è l'inferenza semantica dinamica di espressioni con doppio significato.
- $=$ valuta gli argomenti da destra a sinistra come in C/C++
- $\text{Simple} = \{\text{[int]}, \text{[bool]}\}$

Nucleo di Small21 per MaS: Espressioni con Stato

$Exp ::= [num] Num \mid [true] \mid [false] \mid [emptyE]$
 $\quad \mid DExp \mid Exp [+] Exp \mid Exp [-] Exp \mid Exp [=] Exp \mid \dots$
 $DExp ::= [val] Ide \mid \dots$

$$[Y8] \frac{}{\langle [num] N, Y_\rho \rangle \rightarrow_Y (\langle [int], Y_\rho \rangle)} \quad [Y9] \frac{}{\langle [true], Y_\rho \rangle \rightarrow_Y (\langle [bool], Y_\rho \rangle)} \quad [Y10] \frac{}{\langle [false], Y_\rho \rangle \rightarrow_Y (\langle [bool], Y_\rho \rangle)}$$

$$[Y11] \frac{}{\langle [emptyE], Y_\rho \rangle \rightarrow_Y (\langle [bool], Y_\rho \rangle)}$$

$$[Y12] \frac{Y_\rho(I) = [mut] t \quad t \in Simple}{\langle [val] I, Y_\rho \rangle \rightarrow_{DY} (\langle [mut] t, Y_\rho \rangle)} \quad [Y13] \frac{Y_\rho(I) = [mut] t \quad t \in Simple}{\langle [val] I, Y_\rho \rangle \rightarrow_Y (t, Y_\rho)} \quad [Y14] \frac{Y_\rho(I) = t \quad t \in Simple}{\langle [val] I, Y_\rho \rangle \rightarrow_Y (t, Y_\rho)}$$

$$[Y17] \frac{\langle e_1, Y_\rho \rangle \rightarrow_Y (t_1, Y_\rho) \quad \langle e_2, Y_\rho \rangle \rightarrow_Y (t_2, Y_\rho) \quad Y_\rho(op) = [abs] t[::]t'_1[::]t'_2 \quad op \in \mathcal{O}_2 \quad t'_1 = t_1 \quad t'_2 = t_2}{\langle e_1 [op] e_2, Y_\rho \rangle \rightarrow_Y (t, Y_\rho)}$$

Notazione

- $Exp [-] Exp$ Costruito da aggiungere a Small21 per ottenere un nucleo per i programmi MaS.
- \rightarrow_{DY} è l'inferenza di Tipo del Valore Denotabile di espressioni con doppio significato.
- $\mathcal{O}_2 = \{+, ==, >, <, or\}$; $\mathcal{O}_1 = \{\}$.
- $Simple = \{\langle [int], [bool] \rangle\}$

Sistema Y: Regole per EXP - Errori di tipo

$\text{Exp} ::= [\text{num}] \text{Num} \mid [\text{true}] \mid [\text{false}] \mid [\text{emptyE}]$
 $\mid \text{DExp} \mid \text{Exp} [+]\ \text{Exp} \mid \text{Exp} [==]\ \text{Exp} \mid \dots$

$\text{DExp} ::= [\text{val}] \text{Ide} \mid \dots$

$$[\text{E16}] \frac{Y_\rho(I) = [\text{mut}] t \quad t \notin \text{Simple}}{\langle [\text{val}] I, Y_\rho \rangle \rightarrow_{\text{DY}} ([\text{terr}], Y_\rho)} \quad [\text{E17}] \frac{Y_\rho(I) = t \quad t \neq [\text{mut}] t'}{\langle [\text{val}] I, Y_\rho \rangle \rightarrow_{\text{DY}} ([\text{terr}], Y_\rho)} \quad [\text{E18}] \frac{Y_\rho(I) \neq [\text{mut}] t}{\langle [\text{val}] I, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

$$[\text{E19}] \frac{Y_\rho(I) = \perp}{\langle [\text{val}] I, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

$$[\text{E25}] \frac{\langle e_1, Y_\rho \rangle \rightarrow_Y (t_1, Y_\rho) \quad Y_\rho(\text{op}) = [\text{abs}] t t'_1[x]t'_2 \quad t'_1 \neq t_1}{\langle e_1 [\text{op}] e_2, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)} \quad [\text{E26}] \frac{\langle e_2, Y_\rho \rangle \rightarrow_Y (t_2, Y_\rho) \quad Y_\rho(\text{op}) = [\text{abs}] t t'_1[x]t'_2 \quad t'_2 \neq t_2}{\langle e_1 [\text{op}] e_2, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)} \quad [\text{E27}] \frac{\langle e, Y_\rho \rangle \rightarrow_Y (t_e, Y_\rho) \quad Y_\rho(\text{op}) = [\text{abs}] t t' \quad t' \neq t_e}{\langle [\text{op}] e, Y_\rho \rangle \rightarrow_Y ([\text{terr}], Y_\rho)}$$

Notazione

\rightarrow_{DY} è l'inferenza di Tipo del Valore Denotabile di espressioni con doppio significato.

$\mathcal{O}_2 = \{+, -, *, \text{div}, ==, >, <, \text{or}, \text{and}\}; \quad \mathcal{O}_1 = \{-1, \text{not}\}.$

$\text{Simple} = \{[\text{int}], [\text{bool}]\}$

Espressioni con Stato: Le Transizioni SEM_{EXP} e SEM_{DEN}

Il Sistema di regole SEM_{EXP} (risp. SEM_{DEN}) definisce il comportamento delle r -espressioni (risp. l -espressioni) durante la computazione dei Programmi Small21 sulla Macchina Astratta AM21.

Controllo dei Tipi Dinamico: I Sistemi SEM_{EXP} e SEM_{DEN} sono Integrati con il Sistema Y (ultime 2 slides precedenti, per le espressioni).

$Exp ::= [val] Ide \mid [num] N \mid [true] \mid [false] \mid [emptyE] \mid \dots$

$$X1: \frac{}{\langle [num] N, \sigma \rangle \rightarrow \llbracket [int], N, \sigma \rrbracket}$$

$$X2: \frac{}{\langle [true], \sigma \rangle \rightarrow \llbracket [bool], [true], \sigma \rrbracket}$$

$$X2': \frac{}{\langle [false], \sigma \rangle \rightarrow \llbracket [bool], [true], \sigma \rrbracket}$$

$$X3: \frac{}{\langle [emptyE], \sigma \rangle \rightarrow \llbracket [unit], [emptyE], \sigma \rrbracket}$$

$$X4: \frac{\begin{array}{l} \sigma = (\Delta, \mu) \\ \Delta(I) = (t, loc_{t'}) \\ t = [mut] t' \\ t' \in Simple \end{array}}{\langle [val] I, \sigma \rangle \rightarrow_{DEN} \llbracket t, loc_{t'}, \sigma \rrbracket}$$

$$X5: \frac{\begin{array}{l} \sigma = (\Delta, \mu) \\ \Delta(I) = ([mut] t, loc_t) \\ t \in Simple \\ \mu(loc_t) = v_t \end{array}}{\langle [val] I, \sigma \rangle \rightarrow \llbracket t, v_t, \sigma \rrbracket}$$

$$X6: \frac{\begin{array}{l} \sigma = (\Delta, \mu) \\ \Delta(I) = (t, v_t) \\ t \in Simple \end{array}}{\langle [val] I, \sigma \rangle \rightarrow \llbracket t, v_t, \sigma \rrbracket}$$

Notazione e Osservazioni

- \rightarrow ha termine destro una tripla indicante tipo, valore (esprimibile/denotabile), stato, racchiusa da $\llbracket \]$.
- \rightarrow_{DEN} è la relazione per di Tipo-ValoreDenotato di espressioni con doppio significato.
- $N \in [0..N_k - 1]$ è il controllo a run-time sui bounds inferiore e superiore per i componenti di array statico.

Exp ::= Exp [op] Exp | ...

$$\begin{array}{c}
 \langle e_1, \sigma \rangle \rightarrow [t_1, v_1, \sigma_1] \\
 \langle e_2, \sigma_1 \rangle \rightarrow [t_2, v_2, \sigma_2] \\
 \Delta(\text{op}) = [\text{abs}] \ t[::]t'_1[::]t'_2 \\
 \begin{array}{cc}
 t'_1 = t_1 & t'_2 = t_2 \\
 \text{op} \in \mathcal{O}_2 & \overline{\text{op}}(v_1, v_2) = v
 \end{array} \\
 \hline
 \langle e_1 [\text{op}] e_2, \sigma \rangle \rightarrow [t, v, \sigma_2]
 \end{array}$$

X9:

Notazione e Osservazioni

- $\Delta(\text{op})$ fornisce il tipo di operatori primitivi (forniti dallo RTS di AM21)
- $\mathcal{O}_2 = \{+, -, *, \text{div}, ==, >, <, \text{or}, \text{and}\}$; $\mathcal{O}_1 = \{-1, \text{not}\}$.
- $\overline{\text{op}}$ è l'operazione dello RTS con cui implementiamo l'operatore op: Usiamo le corrispondenti op di Ocaml (se definite), implementazioni ad hoc, altrimenti.
- Simple = $\{\text{[int]}, \text{[bool]}\}$