Semantica Operazionale Strutturata

Linguaggio

- definito tramite sintassi astratta (semplice, intuitiva ma ambigua)
- una stringa viene sempre accoppiata ad un albero sintattico (non ambiguo)

Insiemi di base

- booleani : $\{tt, ff\}$ metavariabili $t, t_1, t' \in \Pi$
- numeri naturali : $\{0, 1, 2, \ldots\}$ " $n, m, p \in \mathbb{N}$
- variabili : $\{a, b, c, \ldots z\}$ " $v \in Var$

  ← in numero finito

Insiemi derivati (in notazione BNF)

- Espressioni Aritmetiche : $e \in Exp$

  $e ::= m \mid v \mid e+e \mid e-e \mid e*e$

- Espressioni Booleane : $b \in Bexp$

  $b ::= t \mid e=e \mid b \text{ or } b \mid \sim b$

- Comandi : $c \in Com$

  $c ::= \text{skip} \mid v := e \mid c;c \mid \text{while } b \text{ do } c \mid \text{if } b \text{ then } c \text{ else } c$

Sintassi concreta non ambigua molto più complicata, ma non serve nel dare semantica

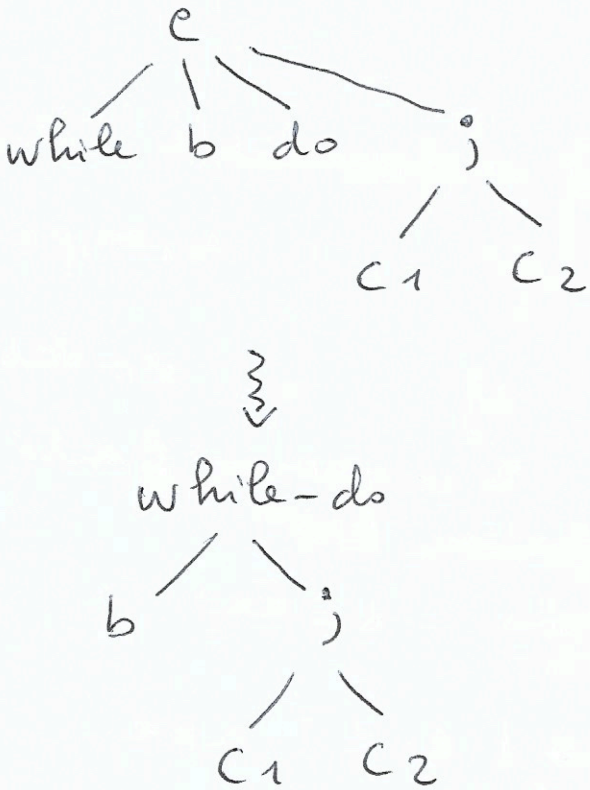
<u>Infatti</u>:

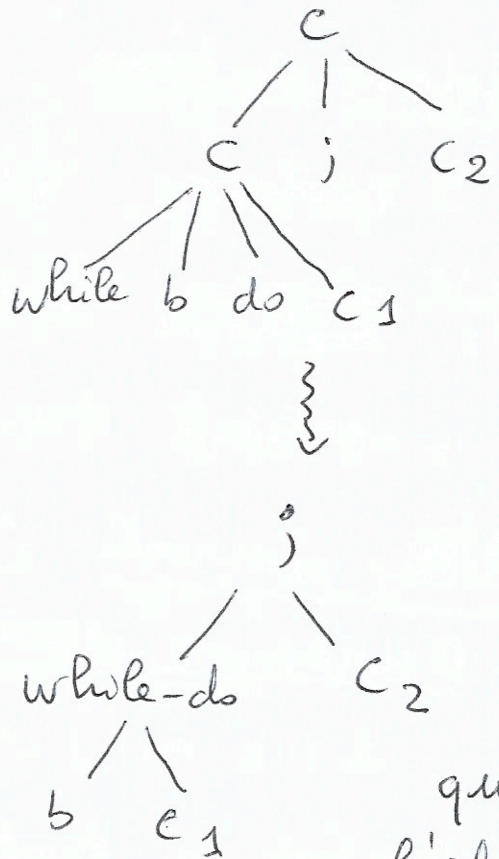
Parser: Programma scritto in sintassi concreta $\longrightarrow$ Albero sintattico di sintassi astratta

Quindi, nel dare semantica, possiamo partire dagli alberi di sintassi astratta

---

<u>while</u> b <u>do</u> $c_1 ; c_2$ è ambigua in sintassi astratta





questo è l'albero inteso in sintassi concreta

Se vogliamo ottenere questo albero, dobbiamo usare <u>begin</u> e <u>end</u>

<u>while</u> b <u>do</u> <u>begin</u>
$c_1 ; c_2$
<u>end</u>

$\longrightarrow$ zucchero sintattico

in sintassi concreta

Come dare semantica?

Definiamo, per ogni categoria sintattica (cioè per Exp, Bexp e Com) un modello detto "Sistema di Transizione"

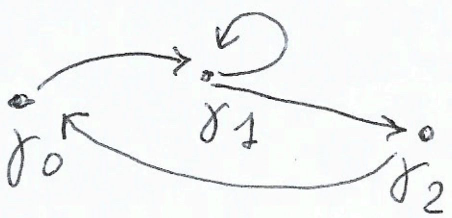
Def Un sistema di transizione è una tripla $\langle \Gamma, T, \rightarrow \rangle$ dove

- Γ è l'insieme (possibilmente infinito) di stati (o configurazioni)
- $T \subseteq \Gamma$ è l'insieme degli stati terminali
- $\rightarrow \subseteq \Gamma \times \Gamma$ è la relazione di transizione

- Una computazione a partire dallo stato γ_0 è una sequenza $\gamma_0 \rightarrow \gamma_1 \rightarrow \gamma_2 \rightarrow \dots$ che può essere finita o infinita
- Con $\rightarrow^*$ indichiamo la chiusura riflessiva e transitiva di $\rightarrow$, ovvero

$$\frac{}{\gamma \rightarrow^* \gamma} \qquad \frac{\gamma \rightarrow^* \gamma' \quad \gamma' \rightarrow \gamma''}{\gamma \rightarrow^* \gamma''}$$

Esempio (con rappresentazione grafica)



grafo in cui i nodi sono gli stati e gli archi le transizioni

Problemi

(1) Γ è di solito un insieme infinito contabile $\Rightarrow$ necessità di trovare una rappresentazione finita implicita attraverso grammatiche/BNF
Cioè Γ essenzialmente coincide con uno dei linguaggi delle 3 categorie sintattiche

<u>Es1:</u> $\Gamma_e = \{ \langle e, \sigma \rangle \mid e \in \text{Expr}, \sigma \in \text{Store} \}$

questo è definito in BNF

(2) $\rightarrow \subseteq \Gamma \times \Gamma$ è una relazione costituita, tipicamente, da infinite coppie $\gamma \rightarrow \gamma'$
$\Rightarrow$ necessità di trovare una rappresentazione finita implicita come minima relazione che soddisfa un certo <u>insieme finito di assiomi e regole d'inferenza</u>

(3) <u>Per dare significato alle variabili</u> (che possono solo assumere valore su $\mathbb{N}$) è necessario introdurre uno <u>store</u> $\sigma : \text{Var} \rightarrow \mathbb{N}$, come funzione che associa ad ogni variabile un valore

$$\sigma = \{ {}^{x_1}/_{n_1}, {}^{x_2}/_{n_2}, \dots {}^{x_k}/_{n_k} \}$$

se supponiamo che $\text{Var} = \{x_1, x_2, \dots x_k\}$
$\text{Store} = \{ \sigma \mid \sigma : \text{Var} \rightarrow \mathbb{N} \}$

Semantica delle Espressioni Aritmetiche

$\langle \Gamma_e, T_e, \rightarrow_e \rangle$ dove

- $\Gamma_e = \{ \langle e, \sigma \rangle \mid e \in Exp, \ \sigma \in Store \}$
- $T_e = \{ \langle n, \sigma \rangle \mid n \in \mathbb{N}, \ \sigma \in Store \}$
- la relazione $\rightarrow_e$ è definita come la minima relazione che soddisfa gli assiomi e le regole di inferenza qui sotto:

(Var) $$\frac{-}{\langle v, \sigma \rangle \rightarrow_e \langle \sigma(v), \sigma \rangle}$$ ← stato terminale

(Sum$_1$) $$\frac{\langle e_0, \sigma \rangle \rightarrow_e \langle e_0', \sigma' \rangle}{\langle e_0 + e_1, \sigma \rangle \rightarrow_e \langle e_0' + e_1, \sigma' \rangle}$$

(Sum$_2$) $$\frac{\langle e_1, \sigma \rangle \rightarrow_e \langle e_1', \sigma' \rangle}{\langle m + e_1, \sigma \rangle \rightarrow_e \langle m + e_1', \sigma' \rangle}$$

(Sum$_3$) $$\frac{-}{\langle m + m', \sigma \rangle \rightarrow_e \langle p, \sigma \rangle}$$ se $p = m + m'$

(Sub$_1$) $$\frac{\langle e_0, \sigma \rangle \rightarrow_e \langle e_0', \sigma' \rangle}{\langle e_0 - e_1, \sigma \rangle \rightarrow_e \langle e_0' - e_1, \sigma' \rangle}$$

(Sub$_2$) $$\frac{\langle e_1, \sigma \rangle \rightarrow_e \langle e_1', \sigma' \rangle}{\langle m - e_1, \sigma \rangle \rightarrow_e \langle m - e_1', \sigma' \rangle}$$

(Sub$_3$) $$\frac{}{\langle m - m', \sigma \rangle \rightarrow_e \langle p, \sigma \rangle}$$ se $m \geq m'$ e $p = m - m'$

Le 3 regole per la moltiplicazione fatele per esercizio (sono del tutto analoghe a quelle della somma).

Osservazioni:

(1) Gli assiomi e le regole SOS sono facilmente implementabili in PROLOG, definendo così un semplice (ma inefficiente) interprete per le espressioni aritmetiche

(2) Lo store σ non viene mai modificato durante la valutazione di una espressione!

- nessuno dei 3 assiomi (Var, Sum_3, Sub_3) va a modificare lo store σ
- se supponiamo che, per ogni regola, $\sigma = \sigma'$ nella premessa della regola, allora $\sigma = \sigma'$ anche nella conclusione!

Ovvero σ non è mai modificato

[esempio di dimostrazione per induzione sulla prova (di $\langle e, \sigma \rangle \to \langle e', \sigma' \rangle$)

(3) L'assioma Sub_3 permette di derivare la transizione $\langle m - m', \sigma \rangle \to_e \langle p, \sigma \rangle$ solo se $m \geq m'$. Altrimenti lo stato $\langle m - m', \sigma \rangle$ è bloccato, pur non essendo uno stato terminale

(è un caso di "errore", come vedremo)

(4) Le regole Sum_1 e Sum_2 riflettono una disciplina/regola di valutazione detta IS (Interna Sinistra):

- prima valutiamo l'argomento più a SX, ovvero e_0;
- quando terminiamo la valutazione di e_0, si passa a valutare e_1;
- quando termina anche la valutazione di e_1, allora si fa la somma (Sum_3)

Alternativamente ID (Interna Destra)

$$(Sum_1')\quad \frac{\langle e_1, \sigma\rangle \to_e \langle e_1', \sigma'\rangle}{\langle e_0 + e_1, \sigma\rangle \to_e \langle e_0 + e_1', \sigma'\rangle}$$

$$(Sum_2')\quad \frac{\langle e_0, \sigma\rangle \to_e \langle e_0', \sigma'\rangle}{\langle e_0 + m, \sigma\rangle \to_e \langle e_0' + m, \sigma'\rangle}$$

+ (Sum_3) invariata

- qui si valuta prima e_1, poi e_0 ed infine si fa la somma

Esempi di derivazione di transizioni

$$
\dfrac{\dfrac{\dfrac{}{\langle x, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle \longrightarrow \langle 5, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}\;(\mathrm{Var})}{\langle x+2, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle \longrightarrow \langle 5+2, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}\;(\mathrm{Sum}_1)}{\underbrace{\langle (x+2)-y, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}_{\gamma_0} \longrightarrow \underbrace{\langle (5+2)-y, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}_{\gamma_1}}\;(\mathrm{Sub}_1)
$$

$$
\dfrac{\dfrac{}{\langle 5+2, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle \longrightarrow \langle 7, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}\;(\mathrm{Sum}_3)}{\underbrace{\langle (5+2)-y, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}_{\gamma_1} \longrightarrow \underbrace{\langle 7-y, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}_{\gamma_2}}\;(\mathrm{Sub}_1)
$$

$$
\dfrac{\dfrac{}{\langle y, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle \longrightarrow \langle 3, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}\;(\mathrm{Var})}{\underbrace{\langle 7-y, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}_{\gamma_2} \longrightarrow \underbrace{\langle 7-3, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}_{\gamma_3}}\;(\mathrm{Sub}_2)
$$

$$
\dfrac{}{\underbrace{\langle 7-3, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}_{\gamma_3} \longrightarrow \underbrace{\langle 4, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle}_{\gamma_4}}\;(\mathrm{Sub}_3)
$$

$$\gamma_0 \to \gamma_1 \to \gamma_2 \to \gamma_3 \to \gamma_4 \in T_e$$

cioè $\langle (x+2)-y, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle \longrightarrow^{*} \langle 4, \{{}^{x}/_{5}, {}^{y}/_{3}\}\rangle$

ottenendo così il valore dell'espressione in $\{{}^{x}/_{5}, {}^{y}/_{3}\}$

Vogliamo ora dimostrare che $\rightarrow_e$ è <u>deterministica</u>, ovvero che

$$\text{se } \gamma \rightarrow_e \gamma' \text{ e } \gamma \rightarrow_e \gamma'' \text{, allora } \gamma' = \gamma'' \quad \forall \gamma, \gamma', \gamma''$$

Useremo una tecnica di prova che si chiama

<u>INDUZIONE</u> <u>STRUTTURALE</u>

Supponiamo di voler dimostrare una certa proprietà $P(e)$ vera per ogni $e \in Expr$

$$(e ::= n \mid v \mid e+e \mid e-e \mid e*e)$$

Se dimostriamo che:

1) $\forall n \in \mathbb{N}$ $\quad P(n)$ è vera

2) $\forall v \in Var$ $\quad P(v)$ è vera

3) $$\frac{P(e_0),\ P(e_1) \text{ vere}}{P(e_0+e_1) \text{ vera}}$$

4) $$\frac{P(e_0),\ P(e_1) \text{ vere}}{P(e_0-e_1) \text{ vera}}$$

5) $$\frac{P(e_0),\ P(e_1) \text{ vere}}{P(e_0*e_1) \text{ vera}}$$

allora concludiamo che <u>$\forall e \in Expr$ $P(e)$ vera</u>.

Nel nostro caso $P(e) = \forall \sigma, \gamma', \gamma''$

$$\left(\langle e, \sigma \rangle \rightarrow_e \gamma' \wedge \langle e, \sigma \rangle \rightarrow_e \gamma'' \right) \Rightarrow \gamma' = \gamma''$$

Dimostriamo ora che $P(e)$ è vera $\forall e \in Exp$ usando induzione strutturale.

1) $e = m \in \mathbb{N}$ Osserva che $\langle m, \sigma \rangle \not\to$ e allora $P(m)$ è vera perché la premessa dell'implicazione è falsa (caso "vacuo")

2) $e = v \in Var$ Per la regola (Var), l'unica transizione derivabile per $\langle v, \sigma \rangle$ è

$$\langle v, \sigma \rangle \to_e \langle \sigma(v), \sigma \rangle$$

Poiché σ è una funzione (cioè $\sigma(v)$ è univoco) e c'è solo la regola (Var) che è applicabile, la Tesi segue:

$$\langle v, \sigma \rangle \to_e \gamma' \wedge \langle v, \sigma \rangle \to_e \gamma'' \Rightarrow \gamma' = \gamma''$$

3) $e = e_0 + e_1$ Supponiamo che $\langle e_0 + e_1, \sigma \rangle \to \gamma'$, $\langle e_0 + e_1, \sigma \rangle \to \gamma''$

Vogliamo dimostrare che $\gamma' = \gamma''$.

Ci sono 3 sottocasi da esaminare in accordo al modo in cui derivo $\langle e_0 + e_1, \sigma \rangle \to \gamma'$

(a) $\langle e_0, \sigma \rangle \to \langle e_0', \sigma' \rangle$ e $\gamma' = \langle e_0' + e_1, \sigma' \rangle$

In questo caso, $e_0 \notin \mathbb{N}$ e l'unica regola che ho applicato è (Sum$_1$). Allora se $\langle e_0 + e_1, \sigma \rangle \to \gamma''$, è necessario che

$$\langle e_0, \sigma \rangle \to \langle e_0'', \sigma'' \rangle \quad \text{e} \quad \gamma'' = \langle e_0'' + e_1, \sigma'' \rangle$$

Ma, per ipotesi induttiva, $P(e_0)$ vale, per cui deve essere $e_0' = e_0''$ e $\sigma' = \sigma''$, da cui discende $\gamma' = \gamma''$ c.v.d.

(b) $e_0 = m \in \mathbb{N}$ ed $\langle e_1, \sigma \rangle \to \langle e_1', \sigma' \rangle$

In questo caso, $e_1 \notin \mathbb{N}$ e la regola che ho applicato è (Sum_2). Allora se $\langle e_0 + e_1, \sigma \rangle \to \gamma''$, è necessario che

$$\langle e_1, \sigma \rangle \to \langle e_1'', \sigma'' \rangle \text{ e } \gamma'' = \langle e_0 + e_1'', \sigma'' \rangle.$$

Ma, per ipotesi induttiva, $P(e_1)$ vale, per cui deve essere $e_1' = e_1''$ e $\sigma' = \sigma''$, da cui discende $\gamma' = \gamma''$ c.v.d

(c) $e_0 \in \mathbb{N}$ ed $e_1 \in \mathbb{N}$.

In questo caso, solo (Sum_3) è applicabile, ottenendo una sola possibile transizione

$$\langle e_0 + e_1, \sigma \rangle \to \langle p, \sigma \rangle \quad \text{dove } p = e_0 + e_1$$

Quindi la Tesi segue:

$$\langle e_0 + e_1, \sigma \rangle \to \gamma' \wedge \langle e_0 + e_1, \sigma \rangle \to \gamma'' \Rightarrow \gamma' = \gamma''$$

c.v.d

4) $e = e_1 - e_2$

5) $e = e_1 * e_2$

Questi 2 casi sono del tutto analoghi al caso $e = e_0 + e_1$

Fatelo per esercizio

Fine della prova

Poiché $\to_e$ è deterministica, a partire da $\langle e, \sigma \rangle$, arriveremo su una sola configurazione terminale $\langle n, \sigma \rangle$: "$n$ è il valore di e in σ"

È possibile perciò definire una funzione

$$eval : Expr \times Store \rightharpoonup \mathbb{N}$$

che dia semantica alle espressioni

funzione parziale!

$$eval(e, \sigma) = \begin{cases} n & \text{se } \langle e, \sigma \rangle \to^* \langle n, \sigma \rangle \\ \text{indefinita} & \text{altrimenti} \end{cases}$$

perché non sempre raggiunge uno stato terminale

<u>Ad esempio</u>:

- $eval((x+2)-y, \{x/5, y/3\}) = 4$ perché

  $$\langle (x+2)-y, \{x/5, y/3\} \rangle \to^* \langle 4, \{x/5, y/3\} \rangle$$

- $eval((x+2)-y, \{x/2, y/7\}) =$ indefinito perché

  $$\langle (x+2)-y, \{x/2, y/7\} \rangle \to^* \langle 4-7, \{x/2, y/7\} \rangle \not\to$$

  configurazione bloccata, ma non terminale

Equivalenza tra espressioni

$e \equiv e'$ sse $\forall \sigma \in Store \quad eval(e,\sigma) = eval(e',\sigma)$

(è equivalente a)

Ad es: $\quad v_1 + (v_2 + v_3) \equiv (v_1 + v_2) + v_3$

Oss: Eval è definita rispetto alla disciplina di valutazione IS. A rigore $eval_{IS}$.
Si può dimostrare che anche per ID, il risultato della valutazione è lo stesso:

$$eval_{IS} = eval_{ID}$$

dove $eval_{ID}(e,\sigma) = \begin{cases} n & \text{se } \langle e,\sigma\rangle \xrightarrow{*}_{ID} \langle n,\sigma\rangle \\ \text{indef.} & \text{altrimenti} \end{cases}$

Come vedremo, è possibile definire anche altre discipline di valutazione, come ES, ED, EP

E = esterna $\quad$ S = Sinistra $\quad$ D = Destra $\quad$ P = Parallel

ma queste sono diverse da IS = ID.

La regola IP prevede che le regole SOS per il + siano $(Sum_1^*) + (Sum_1') + (Sum_3)$

vedi foglio 5 $\quad$ vedi foglio 7 $\quad$ vedi fogli 5

La relazione $\rightarrow_{IP}$ risulta nondeterministica ma "confluente", cioè termina in modo univoco.

Semantica delle Espr. Booleane

$$b ::= t \mid e = e \mid b \text{ or } b \mid \sim b$$

$\langle \Gamma_b, T_b, \rightarrow_b \rangle$ dove $\Gamma_b = \{ \langle b, \sigma \rangle \mid b \in Bexp, \sigma \in Store \}$

$T_b = \{ \langle tt, \sigma \rangle, \langle ff, \sigma \rangle \mid \sigma \in Store \}$

e $\rightarrow_b$ è la minima relazione generata dai seguenti assiomi e regole d'inferenza

$$(Eq_1) \quad \frac{\langle e_0, \sigma \rangle \rightarrow_e \langle e_0', \sigma' \rangle}{\langle e_0 = e_1, \sigma \rangle \rightarrow_b \langle e_0' = e_1, \sigma' \rangle}$$

$$(Eq_2) \quad \frac{\langle e_1, \sigma \rangle \rightarrow_e \langle e_1', \sigma' \rangle}{\langle m = e_1, \sigma \rangle \rightarrow_b \langle m = e_1', \sigma' \rangle}$$

$$(Eq_3) \quad \frac{}{\langle m = n, \sigma \rangle \rightarrow_b \langle t, \sigma \rangle} \quad \text{dove } t = \begin{cases} tt & \text{se } m = n \\ ff & \text{se } m \neq n \end{cases}$$

IS

$$(Or_1) \quad \frac{\langle b_0, \sigma \rangle \rightarrow_b \langle b_0', \sigma' \rangle}{\langle b_0 \text{ or } b_1, \sigma \rangle \rightarrow_b \langle b_0' \text{ or } b_1, \sigma' \rangle}$$

$$(Or_2) \quad \frac{}{\langle tt \text{ or } b_1, \sigma \rangle \rightarrow_b \langle tt, \sigma \rangle}$$

$$(Or_3) \quad \frac{}{\langle ff \text{ or } b_1, \sigma \rangle \rightarrow_b \langle b_1, \sigma \rangle}$$

ES ← comincio con l'arg. a sx

↑ non valuto sempre tutti e due gli argomenti!

$$(Neg_1) \quad \frac{\langle b, \sigma \rangle \rightarrow_b \langle b', \sigma' \rangle}{\langle \sim b, \sigma \rangle \rightarrow_b \langle \sim b', \sigma' \rangle}$$

$$(Neg_2) \quad \frac{}{\langle \sim t, \sigma \rangle \rightarrow_b \langle t', \sigma \rangle} \quad \text{dove } t' = \begin{cases} tt & \text{se } t = ff \\ ff & \text{se } t = tt \end{cases}$$

Osservazioni:

(1) Lo store σ non viene mai modificato durante la valutazione di una bexp, perché:

- nessuno dei 4 assiomi (Eq_3, Or_2, Or_3, Neg_2) va a modificare lo store
- per le regole Eq_1 e Eq_2, sappiamo già che le transizioni nella premessa hanno $\sigma = \sigma'$, quindi anche nella conclusione $\sigma = \sigma'$
- Per le regole Or_1 e Neg_1, se supponiamo che $\sigma = \sigma'$ nella premessa, allora $\sigma = \sigma'$ anche nella conclusione

(Esempio di dimostrazione per induzione sulla prova:

"se $\langle b, \sigma \rangle \rightarrow_b \langle b', \sigma' \rangle$, allora $\sigma = \sigma'$")

Lo dimostro per induzione sulla prova di

(2) $\longrightarrow_b$ è deterministica, ovvero

se $\gamma \rightarrow_b \gamma'$ e $\gamma \rightarrow_b \gamma''$, allora $\gamma' = \gamma''$

(Si può dimostrare, come fatto per $\rightarrow_e$)

Per cui si può facilmente definire

$$eval_b(b, \sigma) = \begin{cases} t & \text{se } \langle b, \sigma \rangle \longrightarrow^* \langle t, \sigma \rangle \\ \text{indefinito} & \text{altrimenti} \end{cases}$$

$b \equiv b'$ sse $\forall \sigma \in Store.\ eval_b(b, \sigma) = eval_b(b', \sigma)$

<u>ad es:</u> $\sim((3 = v) \text{ or } (3 = 4)) \equiv \sim(v = 3)$

Si possono definire per b_0 or b_1 regole di valutazione diverse da ES. Ad esempio ED o IS, ma non sono tutte equivalenti.

ED

$$(Or_1') \quad \frac{\langle b_1, \sigma\rangle \to_b \langle b_1', \sigma'\rangle}{\langle b_0 \text{ or } b_1, \sigma\rangle \to_b \langle b_0 \text{ or } b_1', \sigma'\rangle}$$

$$(Or_2') \quad \frac{}{\langle b_0 \text{ or } tt, \sigma\rangle \to_b \langle tt, \sigma\rangle}$$

$$(Or_3') \quad \frac{}{\langle b_0 \text{ or } ff, \sigma\rangle \to_b \langle b_0, \sigma\rangle}$$

Esempio $\gamma = \langle (3-5) = 6 \text{ or } tt, \sigma\rangle$

$\gamma \not\to_{ES}$ perché $3-5$ è bloccato (Sub_3 non è applicabile)

$\gamma \to_{ED} \langle tt, \sigma\rangle$

In generale,

$$\overbrace{eval_{bES} \neq eval_{bED} \neq eval_{bIS}}^{\neq} = eval_{bID}$$

Esercizio: Provate a definire le regole per b_0 or b_1 secondo la disciplina IS.

Altri esercizi/esempi

1) Descrivere le regole ES per $\langle e_0 * e_1, \sigma \rangle$

$$\frac{}{\langle 0 * e_1, \sigma \rangle \longrightarrow \langle 0, \sigma \rangle}$$ ← non valuto e_1!

$$\frac{}{\langle 1 * e_1, \sigma \rangle \longrightarrow \langle e_1, \sigma \rangle}$$ ← ho già fatto il prodotto, ma ora devo valutare e_1 per avere il risultato

$$\frac{\langle e_0, \sigma \rangle \longrightarrow \langle e_0', \sigma' \rangle}{\langle e_0 * e_1, \sigma \rangle \longrightarrow \langle e_0' * e_1, \sigma' \rangle}$$

$$\frac{\langle e_1, \sigma \rangle \longrightarrow \langle e_1', \sigma' \rangle}{\langle m * e_1, \sigma \rangle \longrightarrow \langle m * e_1', \sigma' \rangle} \quad m \neq 0, 1$$

$$\frac{}{\langle m * n, \sigma \rangle \longrightarrow \langle p, \sigma \rangle} \quad \begin{array}{l} m \neq 0, 1 \\ p = m * n \end{array}$$

2) Dimostrare che $eval_{IS} \neq eval_{ES}$ per $e_0 * e_1$

$$\langle 0 * (3-5), \sigma \rangle \not\longrightarrow_{IS}$$

$$\langle 0 * (3-5), \sigma \rangle \longrightarrow_{ES} \langle 0, \sigma \rangle$$

3) Regole ED per $e_0 * e_1$

4) Regole ES per b_0 and b_1

Semantica dei Comandi

$c ::= \text{skip} \mid v := e \mid c;c \mid \text{if } b \text{ then } c \text{ else } c \mid$
$\text{while } b \text{ do } c$

$\langle \Gamma_c, T_c, \rightarrow_c \rangle$ dove $\Gamma_c = \{ \langle c, \sigma \rangle \mid c \in Com, \sigma \in store \}$
$\cup \{ \sigma \mid \sigma \in store \}$
$T_c = \{ \sigma \mid \sigma \in store \} = store$

e $\rightarrow_c \subseteq \Gamma_c \times \Gamma_c$ è la più piccola relazione generata dai seguenti assiomi e regole di inferenza

(Skip) $$\frac{}{\langle \text{skip}, \sigma \rangle \rightarrow_c \sigma}$$

(Ass) $$\frac{\langle e, \sigma \rangle \rightarrow_e^* \langle m, \sigma \rangle}{\langle v := e, \sigma \rangle \rightarrow_c \sigma[m/v]}$$

dove $\sigma[m/v](v') = \begin{cases} m & \text{se } v' = v \\ \sigma(v') & \text{altrimenti} \end{cases}$

"aggiornamento dello store"

Ad es: se $\sigma = \{x/1, y/1\}$, allora $\sigma[3/x] = \{x/3, y/1\}$

(Seq1) $$\frac{\langle c_0, \sigma \rangle \rightarrow_c \langle c_0', \sigma' \rangle}{\langle c_0; c_1, \sigma \rangle \rightarrow_c \langle c_0'; c_1, \sigma' \rangle}$$

(Seq2) $$\frac{\langle c_0, \sigma \rangle \rightarrow \sigma'}{\langle c_0; c_1, \sigma \rangle \rightarrow \langle c_1, \sigma' \rangle}$$

$$(\mathrm{If}_1)\quad \frac{\langle b,\sigma\rangle \xrightarrow{*}_b \langle tt,\sigma\rangle}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1,\sigma\rangle \rightarrow_c \langle c_0,\sigma\rangle}$$

$$(\mathrm{If}_2)\quad \frac{\langle b,\sigma\rangle \xrightarrow{*}_b \langle ff,\sigma\rangle}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1,\sigma\rangle \rightarrow_c \langle c_1,\sigma\rangle}$$

$$(\mathrm{Wh}_1)\quad \frac{\langle b,\sigma\rangle \xrightarrow{*}_b \langle tt,\sigma\rangle}{\langle \text{while } b \text{ do } c,\sigma\rangle \rightarrow_c \langle c;\text{while } b \text{ do } c,\sigma\rangle}$$

$$(\mathrm{Wh}_2)\quad \frac{\langle b,\sigma\rangle \xrightarrow{*}_b \langle ff,\sigma\rangle}{\langle \text{while } b \text{ do } c,\sigma\rangle \longrightarrow_c \sigma}$$

$$(\mathrm{Alt\text{-}Wh})\quad \frac{-}{\langle \text{while } b \text{ do } c,\sigma\rangle \rightarrow_c \langle \text{if } b \text{ then } c;\text{while } b \text{ do } c \text{ else skip},\sigma\rangle}$$

Esempi

$$(\mathrm{Wh}_1)\quad \frac{\langle tt,\sigma\rangle \xrightarrow{*}_b \langle tt,\sigma\rangle}{\underbrace{\langle \text{while } tt \text{ do skip},\sigma\rangle}_{\gamma_0} \rightarrow_c \underbrace{\langle \text{skip};\text{while } tt \text{ do skip},\sigma\rangle}_{\gamma_1}}$$

$$(\mathrm{Skip})\quad \frac{}{\langle \text{skip},\sigma\rangle \rightarrow \sigma}$$

$$(\mathrm{Seq}_2)\quad \frac{\langle \text{skip},\sigma\rangle \rightarrow \sigma}{\underbrace{\langle \text{skip};\text{while } tt \text{ do skip},\sigma\rangle}_{\gamma_1} \longrightarrow \underbrace{\langle \text{while } tt \text{ do skip},\sigma\rangle}_{\gamma_0}}$$

$$\begin{tikzcd} \gamma_0 \arrow[r, bend left] & \gamma_1 \arrow[l, bend left] \end{tikzcd}$$

Programma che non termina mai!

$$(Wh_1)\quad \frac{\langle tt,\sigma\rangle \to_b^* \langle tt,\sigma\rangle}{\underbrace{\langle \text{while } tt \text{ do } x:=x+1,\sigma\rangle}_{\gamma_0} \to \underbrace{\langle x:=x+1;\ \text{while}\ldots,\sigma\rangle}_{\gamma_0'}} \qquad \sigma=\{x/_0\}$$

$$\sigma=\{x/_0\}$$

$$(Var)\quad \frac{-}{\langle x,\sigma\rangle \to \langle 0,\sigma\rangle}$$

$$(Sum_1)\quad \frac{\langle x,\sigma\rangle \to \langle 0,\sigma\rangle}{\langle x+1,\sigma\rangle \to \langle 0+1,\sigma\rangle \to^{*} \langle 1,\sigma\rangle} \quad \text{usando } (Sum_3)$$

$$(Ass)\quad \frac{\langle x+1,\sigma\rangle \to^{*} \langle 1,\sigma\rangle}{\langle x:=x+1,\sigma\rangle \to \sigma[1/x]=\{x/_1\}}$$

$$(Seq_2)\quad \frac{\langle x:=x+1,\sigma\rangle \to \{x/_1\}}{\underbrace{\langle x:=x+1;\ \text{while } tt \text{ do } x:=x+1,\sigma\rangle}_{\gamma_0'} \to \underbrace{\langle \text{while}\ldots,\{x/_1\}\rangle}_{\gamma_1}}$$

$$\gamma_i = \langle \text{while } tt \text{ do } x:=x+1,\ \{x/_i\}\rangle$$

$$\gamma_i' = \langle x:=x+1;\ \text{while } tt \text{ do } x:=x+1,\ \{x/_i\}\rangle$$

$$\gamma_0 \to \gamma_0' \to \gamma_1 \to \gamma_1' \to \gamma_2 \to \gamma_2' \to \ldots$$

$$(Sub_3)\quad \frac{}{\langle 3-5,\sigma\rangle \not\to}$$

$$(Ass)\quad \frac{\langle 3-5,\sigma\rangle \not\to}{\underbrace{\langle x:=(3-5),\sigma\rangle}_{\gamma} \not\to}$$

$$\gamma$$

<u>Prop:</u> $\rightarrow_c$ è deterministica, cioè

"se $\gamma \rightarrow_c \gamma'$ e $\gamma \rightarrow_c \gamma''$, allora $\gamma' = \gamma''$"

---

parziale

$$exec : Com \times Store \multimap Store$$

$$exec(c, \sigma) = \begin{cases} \sigma' & \text{se } \langle c, \sigma \rangle \rightarrow_c^{*} \sigma' \\ \text{indefinita} & \text{altrimenti} \end{cases}$$

<u>Oss:</u> <u>Divergenza e deadlock sono equiparati</u>

$c_1 = \text{while } tt \text{ do skip}$ $\qquad$ $c_2 = X := (3-5)$

$\langle c_1, \sigma \rangle \longrightarrow \langle \text{skip}; c_1, \sigma \rangle$ $\qquad$ $\langle c_2, \sigma \rangle \not\rightarrow$

$exec(c_1, \sigma) = \text{indefinito} = exec(c_2, \sigma)$

Allora, se vogliamo semanticamente distinguere questi aspetti computazionali, dobbiamo raffinare la semantica, introducendo più dettagli.

Errori Dinamici (a run time)

$$\Gamma' = \Gamma \cup \{err\} \qquad T' = T \cup \{err\}$$

$$eval : Exp \times Store \xrightarrow[\text{(totale)}]{} \mathbb{N} \cup \{err\}$$

$$eval_b : Bexp \times Store \longrightarrow \mathbb{T} \cup \{err\}$$

$$exec : Com \times Store \multimap Store \cup \{err\}$$

(rimane parziale per divergenza)

Regola per generare "errore"

$$(Sub_4) \quad \frac{}{\langle m - m', \sigma \rangle \rightarrow err} \; m' > m$$

unico errore possibile nel nostro semplice ling.

Regole per propagare l'errore (in Exp)

$$(Sum_4) \quad \frac{\langle e_0, \sigma \rangle \rightarrow_e err}{\langle e_0 + e_1, \sigma \rangle \rightarrow_e err}$$

$$(Sum_5) \quad \frac{\langle e_1, \sigma \rangle \rightarrow_e err}{\langle m + e_1, \sigma \rangle \rightarrow_e err}$$

analoghe regole anche per − e *

Regole per propagare l'errore (in Bexp)

- analoghe a per Eq (Eq_4 e Eq_5)

$$(Or_4) \quad \frac{\langle b_0, \sigma \rangle \rightarrow_b err}{\langle b_0 \text{ or } b_1, \sigma \rangle \rightarrow_b err}$$

$$(Neg_3) \quad \frac{\langle b, \sigma \rangle \rightarrow_b err}{\langle \sim b, \sigma \rangle \rightarrow_b err}$$

Propagazione di errore (in Com)

$$(Ass_2)\quad \frac{\langle e,\sigma\rangle \to_e^* err}{\langle v := e,\sigma\rangle \to_c err}$$

$$(Seq_3)\quad \frac{\langle c_0,\sigma\rangle \to_c err}{\langle c_0; c_1,\sigma\rangle \to_c err}$$

$$(If_3)\quad \frac{\langle b,\sigma\rangle \to_b^* err}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1,\sigma\rangle \to_c err}$$

$$(Wh_3)\quad \frac{\langle b,\sigma\rangle \to_b^* err}{\langle \text{while } b \text{ do } c,\sigma\rangle \to_c err}$$

- Tutte queste regole per un solo tipo di errore dinamico ((Sub_4)).
- Altri errori dinamici in un ling. più esteso:
  - se ammettiamo la divisione
    - divisione per zero
  - se ammettiamo un tetto massimo (maxinteger)
    - overflow
  - se includiamo la radice quadrata
    - $\sqrt{\text{negativi}}$
  - se aggiungiamo gli array
    - superare i limiti della dimensione dell'array

⋮

(servono moltissime regole!)

Nondeterminismo e Parallelismo

$$c ::= \mathrm{skip} \mid v := e \mid \ldots \mid \underline{c \text{ or } c} \mid \underline{c \text{ par } c}$$

$$(Nd_1) \qquad \frac{-}{\langle c_0 \text{ or } c_1, \sigma\rangle \longrightarrow_c \langle c_0, \sigma\rangle}$$

$$(Nd_2) \qquad \frac{}{\langle c_0 \text{ or } c_1, \sigma\rangle \longrightarrow_c \langle c_1, \sigma\rangle}$$

$$(Par_1) \qquad \frac{\langle c_0, \sigma\rangle \longrightarrow_c \langle c_0', \sigma'\rangle}{\langle c_0 \text{ par } c_1, \sigma\rangle \longrightarrow_c \langle c_0' \text{ par } c_1, \sigma'\rangle}$$

$$(Par_2) \qquad \frac{\langle c_0, \sigma\rangle \longrightarrow_c \sigma'}{\langle c_0 \text{ par } c_1, \sigma\rangle \longrightarrow_c \langle c_1, \sigma'\rangle}$$

$$(Par_3) \qquad \frac{\langle c_1, \sigma\rangle \longrightarrow_c \langle c_1', \sigma'\rangle}{\langle c_0 \text{ par } c_1, \sigma\rangle \longrightarrow_c \langle c_0 \text{ par } c_1', \sigma'\rangle}$$

$$(Par_4) \qquad \frac{\langle c_1, \sigma\rangle \longrightarrow \sigma'}{\langle c_0 \text{ par } c_1, \sigma\rangle \longrightarrow \langle c_0, \sigma'\rangle}$$

- Verificare che $\longrightarrow_c$ non è più deterministica sul linguaggio esteso con "or" e "par"

Per farlo vedere, costruisco il sistema di transizione per $(X := 0;\ X := X + 1) \text{ par } X := 1$

$$
\begin{tikzcd}[column sep=small]
& \langle (x:=0;\, x:=x+1) \text{ par } x:=1,\ \{x/_0\}\rangle \arrow[dl] \arrow[drr] & & \\
\langle x:=x+1 \text{ par } x:=1,\ \{x/_0\}\rangle \arrow[d] \arrow[dr] & & & \langle x:=0;\, x:=x+1,\ \{x/_1\}\rangle \arrow[d] \\
\langle x:=x+1,\ \{x/_1\}\rangle \arrow[d] & \langle x:=1,\ \{x/_1\}\rangle \arrow[drr] & & \langle x:=x+1,\ \{x/_0\}\rangle \arrow[d] \\
\{x/_2\} & & & \{x/_1\}
\end{tikzcd}
$$

- Due possibili risultati finali: $\{x/_1\}$ o $\{x/_2\}$
- Per modellare questa situazione,

$$\text{exec}: \text{Com} \times \text{Store} -\!\!\circ\!\!\to \mathcal{P}(\text{Store})$$

$$\text{exec}\Big((x:=0;\, x:=x+1) \text{ par } x:=1,\ \{x/_0\}\Big) = \big\{\{x/_1\},\ \{x/_2\}\big\}$$

Gestione dell'errore dinamico

$$\frac{\langle c_0, \sigma\rangle \to_c \text{err}}{\langle c_0 \text{ par } c_1, \sigma\rangle \to_c \text{err}} \qquad\qquad \frac{\langle c_1, \sigma\rangle \to_c \text{err}}{\langle c_0 \text{ par } c_1, \sigma\rangle \to_c \text{err}}$$