Linguaggio L

Sintassi (BNF)

(by value)

C ::= nil | **Id** = **E** | **C**;**C** | if (**E**) {**C**} *[*else {**C**}*]* | while (**E**) {**C**} | **D**;**C** | return **E**

E ::= v | **Id** | **uop E** | **E bop E** | (**E**) | **Id**(**ae**)

D ::= nil | const **Id***[*:**T***]* = **E** | var **Id***[*:**T***]* = **E** | **D**;**D** | function **Id**(**form**) -> **T** {**C**; return **E**} | **form = ae** | rec **D**

form ::= nil | const **Id: T, form** | var **Id: T, form**

ae ::= nil | **E, ae**

uop ::= + | - | !

bop ::= + | - | * | \ | % | == | != | > | >= | < | <= | && | ||

Id ::= insieme degli identificatori validi

Val_E ::= ℤ ∪ ℝ ∪ {true, false} ∪ {*s* | *s* ∈ *ASCII**}

T ::= Int | Double | Bool | String

`Int`	$n, n', n_1, ..$
`Double`	$d, d', d_1, ..$
`Bool`	$b, b', b_1, .. \in \{true, false\}$
`String`	$s, s', s_1, ..$

Metavariabili

$C, C', C'', C_0, C_1, ..$

$E, E', E'', E_0, E_1, ..$

$D, D', D'', D_0, D_1, ..$

$Id, Id', Id_1, x, x', x_1, ...$

$v, v', v'', v_0, v_1, ...$

$\tau, \tau', \tau'', \tau_0, \tau_1, ..$

(by reference)

form ::= nil | const **Id: T, form** | var **Id: T, form** | ref **Id: T, form**

ae ::= nil | **E, ae** | **I, ae**

Altri comandi:

C ::= do {**C**} while (**E**) | for (**D**; **E**; **C**) {**C**} | switch (**E**) {
case v1: **C**; break
case v2: **C**; break
:
*[*default: **C**; break*]*
}

C simbolo distinto della grammatica, quindi un **programma** è un comando

Semantica Statica

comando ben formato **C**: $\Delta \vdash_c C$
espressione ben formata **E**: $\Delta \vdash_e E : \tau$
dichiarazione ben formata **D**: $\Delta \vdash_d D : \Delta'$

ambiente statico Δ : $\text{Id} \cup \text{Val} \longrightarrow T \cup T\text{Loc}$

Semantica Statica Espressioni

Assiomi: $(A1)\, \varnothing \vdash_e \boldsymbol{i} : \text{Int}$, $(A2)\, \varnothing \vdash_e \boldsymbol{d} : \text{Double}$, $(A3)\, \varnothing \vdash_e \boldsymbol{b} : \text{Bool}$, $(A4)\, \varnothing \vdash_e \boldsymbol{s} : \text{String}$

Regole di inferenza:

$$(R1)\, \frac{(\Delta(Id) = \tau \vee \Delta(Id) = \tau\text{Loc})}{\Delta \vdash_e Id : \tau}, \quad (R2)\, \frac{\Delta \vdash_e E_1 : \tau_1, uop : \tau_1 \to \tau}{\Delta \vdash_e uop\, E_1 : \tau}, \quad (R3)\, \frac{\Delta \vdash_e E_1 : \tau_1, \Delta \vdash_e E_2 : \tau_2, bop : \tau_1 \times \tau_2 \to \tau}{\Delta \vdash_e E_1\, bop\, E_2 : \tau}$$

$$(R4)\, \frac{\Delta \vdash_e E : \tau}{\Delta \vdash_e (E) : \tau}$$

Semantica Statica Comandi

Assiomi: $(A5)\, \varnothing \vdash_c nil$

Regole di inferenza:

$$(R5)\, \frac{\Delta(Id) = \tau\text{Loc}, \Delta \vdash_e E : \tau}{\Delta \vdash_c Id = E}, \quad (R6)\, \frac{\Delta \vdash_c C_1, \Delta \vdash_c C_2}{\Delta \vdash_c C_1; C_2}, \quad (R7)\, \frac{\Delta \vdash_e E : \text{Bool}, \Delta \vdash_c C_1, \Delta \vdash_c C_2}{\Delta \vdash_c \text{if } (E)\{C_1\} \text{ else } \{C_2\}}$$

$$(R8)\, \frac{\Delta \vdash_e E : \text{Bool}, \Delta \vdash_c C}{\Delta \vdash_c \text{while } (E)\ \{C\}}, \quad (R9)\, \frac{\Delta \vdash_d D : \Delta', \Delta[\Delta'] \vdash_c C}{\Delta \vdash_c D; C}$$

$$\Delta[\Delta'](x) = \begin{cases} \Delta'(x), & \text{se } \Delta'(x) \text{ definito} \\ \Delta(x), & \text{altrimenti} \end{cases}$$

Semantica Statica Dichiarazioni

Assiomi: $(A6)\, \varnothing \vdash_d nil : \varnothing$

Regole di inferenza:

$$(R10)\, \frac{\Delta \vdash_e E : \tau, T == \tau}{\Delta \vdash_d \text{const } Id : T = E : [(Id, \tau)]}, \quad (R11)\, \frac{\Delta \vdash_e E : \tau, T == \tau}{\Delta \vdash_d \text{var } Id : T = E : [(Id, \tau Loc)]}, \quad (R12)\, \frac{\Delta \vdash_d D_1 : \Delta_1, \Delta[\Delta_1] \vdash_d D_2 : \Delta_2}{\Delta \vdash_d D_1; D_2 : \Delta_1[\Delta_2]}$$

Semantica Statica Funzioni

$$(FS1)\frac{\Delta \vdash_E E : \tau}{\Delta \vdash_C \text{return } E} \qquad \begin{cases} \mathcal{T}(nil) = nil \\ \mathcal{T}(\text{const Id} : \tau, \text{form}) = \tau, \mathcal{T}(\text{form}) \\ \mathcal{T}(\text{var Id} : \tau, \text{form}) = \tau, \mathcal{T}(\text{form}) \end{cases}$$

$$(FS2)\frac{\text{form} : \Delta_0, \Delta[\Delta_0] \vdash_C \text{var } res : \tau = E; C; \text{return } res, \Delta[\Delta_0][(res, \tau Loc)] \vdash_E E : \tau}{\Delta \vdash_D \text{function Id(form)} \ \to \tau \ \{\text{var } res : \tau = E; C; \text{return } res\} \ : \ [(\text{Id}, \mathcal{T}(\text{form}) \to \tau)]}$$

$$(FS3) \ \ nil : \varnothing, \quad \frac{\text{form} : \Delta_0, \text{Id} \notin \Delta_0}{\text{const Id} : \tau, \text{form} : \Delta_0[(\text{Id}, \tau)]} \quad \frac{\text{form} : \Delta_0, \text{Id} \notin \Delta_0}{\text{var Id} : \tau, \text{form} : \Delta_0[(\text{Id}, \tau loc)]}$$

$$(FS4)\frac{\Delta \vdash_{ae} ae : aet, \Delta(\text{Id}) = aet \to \tau}{\Delta \vdash_E \text{Id}(ae) : \tau} \qquad \begin{cases} \Delta \vdash_{ae} nil \\ \dfrac{\Delta \vdash_E E{:}\tau, \Delta \vdash_{ae} ae{:}aet}{\Delta \vdash_{ae} E, ae{:}\tau, aet} \end{cases}$$

Semantica Statica (dichiarazione) Funzioni Ricorsive

Creazione ambiente:

$$(FS2') \vdash_D \text{func Id(form)} \ \to \tau \ \{\text{var } res : \tau = E; C; \text{return } res\} \ : \ [(\text{Id}, \mathcal{T}(\text{form}) \to \tau)]$$

Validazione ambiente:

$$(FS2'')\frac{\text{form} : \Delta_0, \Delta[\Delta_0] \vdash_C \text{var } res : \tau = E; C; \text{return } res, \Delta[\Delta_0][(res, \tau)] \vdash_E E : \tau}{\Delta \vdash_D \text{func Id(form)} \ \to \tau \ \{\text{var } res : \tau = E; C; \text{return } res\}}$$

$$(RS1')\frac{\vdash_D D : \Delta}{\vdash_D rec\ D : \Delta}$$

$$(RS1'')\frac{\vdash_D D : \Delta', \Delta[\Delta'_{|I_0}] \vdash_D D}{\Delta \vdash_D rec\ D}, I_0 = FI(D) \cap BI(D)$$

Semantica Dinamica

esecuzione **C**: $\langle C, \rho, \sigma \rangle \longrightarrow_c \langle C', \rho', \sigma' \rangle$, $\mathbf{Exec}(C, \rho, \sigma) = \sigma' \iff \langle C, \rho, \sigma \rangle \longrightarrow_c^* \sigma'$

valutazione **E**: $\langle E, \rho, \sigma \rangle \longrightarrow_e \langle E', \rho, \sigma \rangle$, $\mathbf{Eval}(E, \rho, \sigma) = v \in Val \iff \langle E, \rho, \sigma \rangle \longrightarrow_e^* v$

elaborazione **D**: $\langle D, \rho, \sigma \rangle \longrightarrow_d \langle D', \rho', \sigma' \rangle$, $\mathbf{Elab}(D, \rho, \sigma) = \langle \rho', \sigma' \rangle \iff \langle D, \rho, \sigma \rangle \longrightarrow_d^* \langle \rho', \sigma' \rangle$

ambiente (dinamico) ρ : Id $\longrightarrow$ Loc ∪ Val   memoria σ : Loc $\longrightarrow$ Val

$\longrightarrow_c$, $\longrightarrow_e$, $\longrightarrow_d$ sono le funzioni di interpretazione semantica di C, E e D

SISTEMA DI TRANSIZIONI

stato i

Istruzioni da eseguire
$\rho = \{(Id_0, L_0), \ldots, (Id_i, v_i), \ldots\}$
$\sigma = \{(L_0, v_0), \ldots, (L_i, v_i), \ldots\}$

$\ldots \quad \substack{\longrightarrow_c \\ \longrightarrow_e \\ \longrightarrow_d} \quad \ldots$

stato i+1

Istruzioni da eseguire
$\rho' = \{(Id_0, L_0), \ldots, (Id_i, v_i), \ldots\}$
$\sigma' = \{(L_0, v_0'), \ldots, (L_i, v_i'), \ldots\}$

stato finale

nil
$\rho = \{(Id_0, L_0), \ldots, (Id_i, v_i), \ldots\}$
$\sigma = \{(L_0, v_0), \ldots, (L_i, v_i), \ldots\}$

Semantica Dinamica Espressioni

$$(Id1)\ \frac{\rho(Id) = v \ \vee\ (\rho(Id) = L \in Loc \wedge \sigma(L) = v)}{\langle Id, \rho, \sigma\rangle \longrightarrow_e v}$$

$$(uop1)\ \frac{\langle E, \rho, \sigma\rangle \longrightarrow_e \langle E', \rho, \sigma\rangle}{\langle uop\, E, \rho, \sigma\rangle \longrightarrow_e \langle uop\, E', \rho, \sigma\rangle}$$

$$(uop2)\ \langle uop\ v, \rho, \sigma\rangle \longrightarrow_e v' = \mathrm{uop}\ v$$

$$(bop1)\ \frac{\langle E_1, \rho, \sigma\rangle \longrightarrow_e \langle E_1', \rho, \sigma\rangle}{\langle E_1\, bop\, E_2, \rho, \sigma\rangle \longrightarrow_e \langle E_1'\, bop\, E_2, \rho, \sigma\rangle}$$

$$(bop2)\ \frac{\langle E_2, \rho, \sigma\rangle \longrightarrow_e \langle E_2', \rho, \sigma\rangle}{\langle v_1\, bop\, E_2, \rho, \sigma\rangle \longrightarrow_e \langle v_1\, bop\, E_2', \rho, \sigma\rangle}$$

$$(bop3)\ \langle v_1 bop\ v_2, \rho, \sigma\rangle \longrightarrow_e v = v_1\ \mathrm{bop}\ v_2$$

!!! bop è sintassi

bop è semantica

Semantica Dinamica Comandi

$$(id2)\ \frac{\langle E, \rho, \sigma\rangle \longrightarrow_e^* v}{\langle Id = E, \rho, \sigma\rangle \longrightarrow_c \langle Id = v, \rho, \sigma\rangle}$$

$$(id3)\ \langle Id = v, \rho, \sigma\rangle \longrightarrow_c \sigma[\rho(Id) = v]$$

$$(seq1)\ \frac{\langle C_1, \rho, \sigma\rangle \longrightarrow_c \langle C_1', \rho, \sigma'\rangle}{\langle C_1; C_2, \rho, \sigma\rangle \longrightarrow_c \langle C_1'; C_2, \rho, \sigma'\rangle}$$

$$(seq2)\ \frac{\langle C_1, \rho, \sigma\rangle \longrightarrow_c \sigma'}{\langle C_1; C_2, \rho, \sigma\rangle \longrightarrow_c \langle C_2, \rho, \sigma'\rangle}$$

$$(if1)\ \frac{\langle E, \rho, \sigma\rangle \longrightarrow_e^* true}{\langle \mathtt{if}\,(E)\{C_1\}\,\mathtt{else}\,\{C_2\}, \rho, \sigma\rangle \longrightarrow_c \langle C_1, \rho, \sigma\rangle}$$

$$(if2)\ \frac{\langle E, \rho, \sigma\rangle \longrightarrow_e^* false}{\langle \mathtt{if}\,(E)\{C_1\}\,\mathtt{else}\,\{C_2\}, \rho, \sigma\rangle \longrightarrow_c \langle C_2, \rho, \sigma\rangle}$$

$$(rep1)\ \frac{\langle E, \rho, \sigma\rangle \longrightarrow_e^* true}{\langle \mathtt{while}\,(E)\{C\}\, \rho, \sigma\rangle \longrightarrow_c \langle C; \mathtt{while}\,(E)\{C\}, \rho, \sigma\rangle}$$

$$(rep2)\ \frac{\langle E, \rho, \sigma\rangle \longrightarrow_e^* false}{\langle \mathtt{while}\,(E)\{C\}, \rho, \sigma\rangle \longrightarrow_c \sigma}$$

$$(b1)\ \frac{\langle D, \rho, \sigma\rangle \longrightarrow_d^* \langle \rho', \sigma'\rangle}{\langle D; C, \rho, \sigma\rangle \longrightarrow_c \langle C, \rho[\rho'], \sigma[\sigma']\rangle}$$

Semantica Dinamica Dichiarazioni

$$(const1)\ \frac{\langle E,\rho,\sigma\rangle \longrightarrow_e^* v}{< const\ Id : T = E,\rho,\sigma\rangle \longrightarrow_d \langle[(Id,v)],\sigma\rangle}$$

$$(var1)\ \frac{\langle E,\rho,\sigma\rangle \longrightarrow_e^* v}{\langle var\ Id : T = E,\rho,\sigma\rangle \longrightarrow_d \langle[(Id, new\ L)],[(L,v)]\rangle}$$

$$(dd1)\ \frac{\langle D_1,\rho,\sigma\rangle \longrightarrow_d \langle D_1',\rho',\sigma'\rangle}{\langle D_1;D_2,\rho,\sigma\rangle \longrightarrow_d \langle D_1';D_2,\rho',\sigma'\rangle} \qquad (dd2)\ \frac{\langle D_2,\rho[\rho_1],\sigma\rangle \longrightarrow_d \langle D_2',\rho[\rho_1]',\sigma'\rangle}{\langle \rho_1;D_2,\rho[\rho_1],\sigma\rangle \longrightarrow_d \langle \rho_1;D_2',\rho[\rho_1]',\sigma'\rangle}$$

$$(dd3)\ \langle \rho_1;\rho 2,\rho,\sigma\rangle \longrightarrow_d \langle \rho_1[\rho_2],\sigma\rangle$$

le regole (dd2) e (dd3) contengono configurazioni non ammissibili rispetto alla definizione di sistema di transizione

$(dd2)\ \langle \rho_1;D_2,\rho,\sigma\rangle, \langle \rho_1;D_2',\rho,\sigma'\rangle \quad (dd3)\ \langle \rho_1;\rho 2,\rho,\sigma\rangle$

la parte codice delle configurazioni di stato deve essere generabile dalla grammatiche che definisce D, e questo non vale per le configurazioni sopra

aggiungo gli ambienti alla sintassi

D ::= nil | let ID[:T] = E | var ID[:T] = E | D;D | ϱ

!!! solo il compilatore può generare gli ambienti della sintassi, non l'utente

Il sistema di transizione delle dichiarazioni è

$(\{\langle D,\rho,\sigma\rangle \cup \langle \rho', sigma'\rangle\}, \longrightarrow_d, \{\langle \rho', sigma'\rangle\},$ ⟨dichiarazione da elaborare, ambiente iniziale, memoria iniziale⟩)

Semantica Dinamica Funzioni (scoping statico e dinamico)

$$(FD1)\langle \text{func Id(form)} \rightarrow T\{C; \text{return } E\}, \rho, \sigma\rangle \rightarrow_d \langle (Id, \lambda\, \text{form} \,.\, \{\rho'; C; \text{return } E\}), \sigma\rangle$$

$$\begin{cases} \rho' = \rho_{|FI(C)-BI(form)} & scoping\ statico \\ \rho' = nil & scoping\ dinamico \end{cases}$$

qui C indica {ρ';C; return E}

$$(FD2)\frac{\rho(Id) = \lambda\, \text{form} \,.\, C}{\langle Id(ae), \rho, \sigma\rangle \rightarrow_e \langle \{form = ae; C\}, \rho, \sigma\rangle}$$

$$(FD3)\frac{\langle E, \rho, \sigma\rangle \rightarrow_e \langle E', \rho, \sigma\rangle}{\langle E, ae, \rho, \sigma\rangle \rightarrow_{ae} \langle E', ae, \rho, \sigma\rangle} \qquad (FD4)\frac{\langle ae, \rho, \sigma\rangle \rightarrow_{ae} \langle ae', \rho, \sigma\rangle}{\langle k, ae, \rho, \sigma\rangle \rightarrow_{ae} \langle k, ae', \rho, \sigma\rangle}$$

$$(FD5)\frac{\langle ae, \rho, \sigma\rangle \rightarrow_{ae} \langle ae', \rho, \sigma\rangle}{\langle \text{form} = ae, \rho, \sigma\rangle \rightarrow_d \langle \text{form} = ae', \rho, \sigma\rangle}$$

$$(FD6)\frac{av \vdash \text{form} : \rho_0, \sigma_0}{\langle \text{form} = av, \rho, \sigma\rangle \rightarrow_d \langle \rho_0, \sigma_0\rangle}$$

$$nil \vdash nil : \emptyset, \emptyset \qquad \frac{av \vdash \text{form} : \rho, \sigma}{v, av \vdash \text{let Id} : \tau, \text{form} : \rho[(\text{Id}, v)], \sigma} \qquad \frac{av \vdash \text{form} : \rho, \sigma}{v, av \vdash \text{var Id} : \tau, \text{form} : \rho[(\text{Id}, l_{(new)})], \sigma[(l_{(new)}, v)]}$$

$$(FD7)\langle \text{return } E, \rho, \sigma\rangle \rightarrow_c \langle E, \rho, \sigma\rangle$$

Semantica Dinamica (dichiarazione) Funzioni Ricorsive

Dichiarazione di funzione come se non fosse ricorsiva (effetto: ci sono identificatori liberi nel corpo - in sostanza il nome della funzione):

$$(RD1)\frac{\langle D, \rho - I_0, \sigma\rangle \rightarrow_d \langle D', \rho', \sigma'\rangle}{\langle rec\ D, \rho, \sigma\rangle \rightarrow_d \langle rec\ D', \rho', \sigma'\rangle}, I_0 = FI(D) \cap BI(D)$$

Quando finiamo con la applicazione della RD1, possiamo applicare la RD2:

$$(RD2)\ \langle rec\ \rho_0, \rho, \sigma\rangle \rightarrow \langle \{(f, \lambda form \,.\, (rec\ \rho_0) - form; C) \mid \rho_0(f) = \lambda form \,.\, C\}, \sigma\rangle$$

Identificatori Liberi

$FI_e : E \to \{\text{occorrenze } Id \text{ liberi}\}$

$FI_e(\text{v}) = \emptyset$

$FI_e(\text{Id}) = \{\text{Id}\}$

$FI_e(\text{uop E}) = FI_e(\text{E})$

$FI_e(\text{E1 bop E2}) = FI_e(\text{E1}) \cup FI_e(\text{E2})$

$FI_c : C \to \{\text{occorrenze } Id \text{ liberi}\}$

$FI_c(\text{nil}) = \emptyset$

$FI_c(\text{Id = E}) = \{\text{Id}\} \cup FI_e(\text{E})$

$FI_c(\text{C1;C2}) = FI_c(\text{C1}) \cup FI_c(\text{C2})$

$FI_c(\text{if (E) \{C1\} else \{C2\}}) =$
$\quad FI_e(\text{E}) \cup FI_c(\text{C1}) \cup FI_c(\text{C2})$

$FI_c(\text{while (E) \{C\}}) = FI_e(\text{E}) \cup FI_c(\text{C})$

$FI_c(\text{D;C}) = FI_d(\text{D}) \cup (FI_c(\text{C}) - BI_d(\text{D}))$

$FI_d : D \to \{\text{occorrenze } Id \text{ liberi}\}$

$FI_d(\text{nil}) = \emptyset$

$FI_d(\texttt{const Id:T = E}) = FI_e(\text{E})$

$FI_d(\text{var Id:T = E}) = FI_e(\text{E})$

$FI_d(\text{D1;D2}) = FI_d(\text{D1}) \cup (FI_d(\text{D2}) - BI_d(\text{D1}))$

$$BI_c = \overline{FI}_c$$
$$BI_e = \overline{FI}_e$$
$$BI_d = \overline{FI}_d$$

Identificatori Liberi Funzioni

$FI_c(\text{return E}) = FI_e(E)$

$FI_e(\text{Id(ae)}) = \{\text{Id}\} \cup FI_{ae}(ae)$

$FI_d(\text{function Id(form))-> T\{C\}}) = FI_c(\text{C}) - BI_{form}(\text{form})$

$FI_{form}(\text{form}) = \emptyset$

$FI_{ae}(\text{E,ae}) = FI_e(E) \cup FI_{ae}(ae)$

Anatomia delle funzioni