

Mao (Modello Astratto Operazionale)

Tipi base aggiuntivi, multi-assegnamenti e return

Tipi base aggiuntivi

char e stringhe (char[])

Tipi base

Il linguaggio che abbiamo definito fino a questo momento prevedeva solo l'uso due tipi base interi e booleani

Possiamo facilmente estendere il linguaggio Mao con tipi base aggiuntivi, come i numeri in virgola mobile (vedi dispensa) e i caratteri

Vediamo come estendere il linguaggio Mao con il tipo base char che ci permetterà di introdurre anche il tipo di dato stringa come array di char

Caratteri

I caratteri sono utilizzati per rappresentare simboli, lettere e altri caratteri alfanumerici

In C++ utilizzeremo il tipo base `char` i cui valori possono essere rappresentati utilizzando una codifica tipo ASCII o Unicode

Le costanti di tipo `char` sono rappresentate tra apici, mentre caratteri speciali sono rappresentati da *sequenze di escape come* `'\t'` (*tabulazione*) e `'\n'` (*a capo*)

Esempio:

```
char lettera = 'R';  
char a_capo = '\n';
```

Caratteri

Estendiamo la sintassi dei tipi base e anche dei valori

Tb ::= ... un tipo base: **int** oppure **bool**
| **char** oppure un carattere

V ::= ...
| **c** dove **c** è un carattere della tastiera tra apici come 'a' o ';' o una sequenza di escape come '\n'

Il valore di default per il tipo **char** è il carattere '\0' nullo

Possiamo estendere le operazioni di confronto come **<** o **==** anche ai caratteri: che ci dicono se un carattere precede o è uguale ad un altro

Attenzione: in molti linguaggi il confronto tra caratteri confronta la rappresentazione ASCII o Unicode che coincide solo in parte con l'ordine alfabetico

Stringhe

Le stringhe sono utilizzate per rappresentare sequenze di caratteri

In C le rappresentiamo come array di caratteri, con il tipo `char []`

Invece di scrivere espressioni come

```
['C', 'i', 'a', 'o']
```

scriviamo direttamente la stringa racchiusa tra doppi apici

```
"Ciao"
```

Esercizio

Scrivere un programma che data una stringa `s` controlla se la lunghezza di `s` è maggiore di 2 e in caso affermativo crea una copia di `s` eliminando il primo e l'ultimo carattere, altrimenti non fa niente

Esercizio

Scrivere un programma che data una stringa *s* controlla se la lunghezza di *s* è maggiore di 2 e in caso affermativo crea una copia di *s* eliminando il primo e l'ultimo carattere, altrimenti non fa niente

```
char[] copia = new char[s.length - 2];  
if (s.length > 2) {  
    int i = 1;  
    while (i < s.length-1) {  
        copia[i-1] := s[i];  
        i := i + 1;  
    }  
}
```


Esercizio

Scrivere un programma che data una stringa `s` conti il numero di caratteri minuscoli

Supponiamo di avere l'operatore `<` che confronta i caratteri

Esercizio

Scrivere un programma che data una stringa *s* conti il numero di caratteri minuscoli

```
int occ = 0;
int i = 0;
while (i < s.length) {
    if (('a' <= s[i]) && (s[i] <= 'z'))
        { occ := occ + 1; }
    i := i + 1;
}
```

Esercizio

Scrivere un programma che data una stringa s controlli se la stringa è palindroma

Usiamo un noto schema di verifica di proprietà:
Assumiamo la proprietà vera all'inizio e
cerchiamo il primo caso che non verifica la proprietà, se esiste

Esercizio

Scrivere un programma che data una stringa *s* controlli se la stringa è palindroma

```
bool palindroma = true;

int i = 0;

while (i < s.length/2) {
    if (s[i] != s[s.length-1 - i])
        { palindroma := false; }

    i := i + 1;
}
```

possiamo fare meglio?

Esercizio

Scrivere un programma che data una stringa *s* controlli se la stringa è palindroma

```
bool palindroma = true;
```

```
int i = 0;
```

```
while (i < s.length/2 && palindroma) {
```

```
    if (s[i] != s[s.length-1 - i])
```

```
        { palindroma := false; }
```

```
    i := i + 1;
```

```
}
```

possiamo semplificare
ancora?

Esercizio

Scrivere un programma che data una stringa *s* controlli se la stringa è palindroma

```
bool palindroma = true;
int i = 0;
while (i < s.length/2 && palindroma) {
    palindroma := (s[i] == s[s.length-1 - i]);
    i := i + 1;
}
```

Assegnamenti multipli

Assegnamento multiplo

Molti linguaggi permettono l'uso di **assegnamenti multipli** che permettono di inizializzare più variabili contemporaneamente.

A sinistra del simbolo di assegnamento (o di inizializzazione) ammettiamo una lista di variabili e a destra una lista di espressioni della stessa lunghezza

In JavaScript, possiamo scrivere ad esempio:

```
let x, y, z = 6, 7, 42;
```

per dichiarare tre variabili x, y e z, inizializzandole rispettivamente a 6, 7 e 42

Assegnamento multiplo

Attenzione: il comportamento di un assegnamento multiplo può differire dalla composizione sequenziale dei singoli assegnamenti

Perché?

perché le espressioni vengono tutte valutate prima che i valori delle variabili vengano aggiornati

Esempio: `x, y := y, x;` scambia i valori di `x` e `y`

in generale è diverso da

```
x := y;
```

```
y := x;
```

alla fine `x` e `y` hanno il valore originale di `y`

Assegnamento multiplo

In Mao l'assegnamento multiplo sarà utile per trattare in modo diretto il passaggio dei parametri attuali alle funzioni

Dobbiamo introdurre le categorie sintattiche corrispondenti a liste non vuote di:

- identificatori

Esempio: `x, y, z`

- espressioni

Esempio: `true, 1+z`

- dichiarazioni

Esempio: `int i, char c`

Sintassi dell'assegnamento multiplo

Nuove categorie sintattiche per le liste di Id, espressioni e dichiarazioni:

$$\text{Ids} ::= \text{Id} \mid \text{Id}, \text{Ids}$$
$$\text{Es} ::= \text{E} \mid \text{E}, \text{Es}$$
$$\text{Ds} ::= \text{T Id} \mid \text{T Id}, \text{Ds}$$

Sfruttando le nuove categorie sintattiche, generalizziamo i comandi atomici:

$$\text{C} ::= \dots \mid \text{Ds} = \text{Es}; \mid \text{Ids} := \text{Es};$$

Variabili libere nell'assegnamento multiplo

Quali variabili devono essere considerate nell'ambiente di tipo?

$$fv(T_1 Id_1, \dots, T_n Id_n = E_1, \dots, E_n;) = fv(E_1) \cup \dots \cup fv(E_n)$$

$$fv(Id_1, \dots, Id_n := E_1, \dots, E_n;) = \{ Id_1, \dots, Id_n \} \cup fv(E_1) \cup \dots \cup fv(E_n)$$

$$dv(T_1 Id_1, \dots, T_n Id_n = E_1, \dots, E_n;) = \{ Id_1, \dots, Id_n \}$$

$$dv(Ids := Es;) = \emptyset$$

Controllo dei tipi nell'assegnamento multiplo

$C ::= \dots$
 $| T_1 Id_1, \dots, T_n Id_n = E_1, \dots, E_n;$

$| Id_1, \dots, Id_n := E_1, \dots, E_n;$

La dichiarazione multipla aggiunge nuove associazioni locali.

Prima dobbiamo però controllare che:

- tutte le inizializzazioni siano tipate correttamente

stesso Γ

$$\frac{\Gamma \vdash E_1 : T_1 \quad \Gamma \vdash Ds = Es; : \Gamma'}{\Gamma \vdash T_1 Id_1, Ds = E_1, Es; : \Gamma'[Id_1 : T_1]}$$

Controllo dei tipi nell'assegnamento multiplo

$C ::= \dots$
 $| T_1 Id_1, \dots, T_n Id_n = E_1, \dots, E_n;$

$| Id_1, \dots, Id_n := E_1, \dots, E_n;$

Per l'assegnamento multiplo dobbiamo controllare che:

- tutte gli assegnamenti siano tipati correttamente

stesso Γ

Γ' sarà sempre,
necessariamente \emptyset

$$\frac{\Gamma(Id_1) = T_1 \quad \Gamma \vdash E_1 : T_1 \quad \Gamma \vdash Ids := Es; : \Gamma'}{\Gamma \vdash Id_1, Ids := E_1, Es; : \emptyset}$$

Semantica operativa della dichiarazione multipla

$C ::= \dots$

| $T_1 \text{ Id}_1, Ds = E_1, Es;$

Per eseguire una dichiarazione multipla nello stato (ρ, σ) , dobbiamo:

- calcolare (v_1, σ_1) valutando l'espressione E_1 nello stato (ρ, σ)
- calcolare (ρ', σ') prodotto dalle dichiarazioni successive nello stato (ρ, σ_1)
- preparare una nuova cella di memoria $l_1 \notin \sigma'$
- estendere l'ambiente ρ' con l'associazione $\text{Id}_1 \mapsto l_1$
- estendere la memoria σ' con l'associazione $l_1 \mapsto v_1$

$$\frac{\langle E_1, \rho, \sigma \rangle \Downarrow (v_1, \sigma_1) \quad \langle Ds = Es;, \rho, \sigma_1 \rangle \rightarrow (\rho', \sigma') \quad l_1 \notin \sigma'}{\langle T_1 \text{ Id}_1, Ds = E_1, Es;, \rho, \sigma \rangle \rightarrow \langle \rho'[\text{Id}_1 \mapsto l_1], \sigma'[l_1 \mapsto v_1] \rangle}$$

Semantica operativa dell'assegnamento multiplo

$C ::= \dots$

| $Id_1, Ids = E_1, Es;$

Per eseguire un assegnamento multiplo nello stato (ρ, σ) , dobbiamo:

- calcolare (v_1, σ_1) valutando l'espressione E_1 nello stato (ρ, σ)
- calcolare (ρ', σ') prodotto dagli assegnamenti successivi nello stato (ρ, σ_1)
- trovare la locazione associata a Id_1 nell'ambiente ρ' (che sarà ρ)
- estendere la memoria σ' con l'associazione $l_1 \mapsto v_1$

$$\frac{\langle E_1, \rho, \sigma \rangle \Downarrow (v_1, \sigma_1) \quad \langle Ids := Es; , \rho, \sigma_1 \rangle \rightarrow (\rho', \sigma') \quad \rho'(Id_1) = l_1}{\langle Id_1, Ids = E_1, Es; , \rho, \sigma \rangle \rightarrow \langle \rho', \sigma'[l_1 \mapsto v_1] \rangle}$$

$$\langle Id_1, Ids = E_1, Es; , \rho, \sigma \rangle \rightarrow \langle \rho', \sigma'[l_1 \mapsto v_1] \rangle$$

Esempio

Verifichiamo che l'assegnamento multiplo non si comporta sempre come una sequenza di assegnamenti

```
x, y := x+100, x;
```

```
x := x + 100;  
y := x;
```

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio $x, y := x+100, x;$

$$\frac{\langle E_1, \rho, \sigma \rangle \Downarrow (v_1, \sigma_1) \quad \langle Ids := Es; , \rho, \sigma_1 \rangle \rightarrow (\rho', \sigma') \quad \rho'(Id_1) = l_1}{\langle Id_1, Ids = E_1, Es; , \rho, \sigma \rangle \rightarrow \langle \rho', \sigma'[l_1 \mapsto v_1] \rangle}$$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle \text{ (non ci possono essere modifiche all'ambiente)}$$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio $x, y := x+100, x;$

$$\frac{\langle E_1, \rho, \sigma \rangle \Downarrow (v_1, \sigma_1) \quad \langle Ids := Es; , \rho, \sigma_1 \rangle \rightarrow (\rho', \sigma') \quad \rho'(Id_1) = l_1}{\langle Id_1, Ids = E_1, Es; , \rho, \sigma \rangle \rightarrow \langle \rho', \sigma'[l_1 \mapsto v_1] \rangle}$$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle \quad \text{con } \sigma_? = \sigma_2?[l_x \mapsto v_1]$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle v_1, \sigma_1 \rangle \quad \text{con } \sigma_1 = \sigma \text{ (valutazione di espressione pura)}$$

$$\langle y := x; , \rho, \sigma_1 \rangle \rightarrow \langle \rho, \sigma_2? \rangle \quad \text{(non ci possono essere modifiche all'ambiente)}$$

Esempio

$x, y := x+100, x;$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

$$\frac{\langle E_1, \rho, \sigma \rangle \Downarrow (v_1, \sigma_1) \quad \langle Ids := Es; , \rho, \sigma_1 \rangle \rightarrow (\rho', \sigma') \quad \rho'(Id_1) = l_1}{\langle Id_1, Ids = E_1, Es; , \rho, \sigma \rangle \rightarrow \langle \rho', \sigma'[l_1 \mapsto v_1] \rangle}$$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle \quad \text{con } \sigma_? = \sigma_{2?}[l_x \mapsto 102]$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle 102, \sigma \rangle \quad \checkmark$$

$$\langle y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_{2?} \rangle$$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio $x, y := x+100, x;$

$$\frac{\langle E_1, \rho, \sigma \rangle \Downarrow (v_1, \sigma_1) \quad \langle Ids := Es; , \rho, \sigma_1 \rangle \rightarrow (\rho', \sigma') \quad \rho'(Id_1) = l_1}{\langle Id_1, Ids = E_1, Es; , \rho, \sigma \rangle \rightarrow \langle \rho', \sigma'[l_1 \mapsto v_1] \rangle}$$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle \quad \text{con } \sigma_? = \sigma_2?[l_x \mapsto 102]$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle 102, \sigma \rangle \quad \checkmark$$

$$\langle y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_2? \rangle \quad \text{con } \sigma_2? = \sigma[l_y \mapsto v_2]$$

$$\langle x, \rho, \sigma \rangle \Downarrow \langle v_2, \sigma_3 \rangle \quad \text{con } \sigma_3 = \sigma \text{ (valutazione di espressione pura)}$$

$$\frac{\langle E, \rho, \sigma \rangle \Downarrow v \quad \rho(Id) = l}{\langle Id := E; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l \mapsto v] \rangle}$$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio $x, y := x+100, x;$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle \quad \text{con } \sigma_? = \sigma_{2?}[l_x \mapsto 102]$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle 102, \sigma \rangle \checkmark$$

$$\langle y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_{2?} \rangle \quad \text{con } \sigma_{2?} = \sigma[l_y \mapsto 2]$$

$$\langle x, \rho, \sigma \rangle \Downarrow \langle 2, \sigma \rangle \checkmark$$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio $x, y := x+100, x;$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle \quad \text{con } \sigma_? = \sigma[l_y \mapsto 2][l_x \mapsto 102]$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle 102, \sigma \rangle \checkmark$$

$$\langle y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_{2?} \rangle \quad \text{con } \sigma_{2?} = \sigma[l_y \mapsto 2] \checkmark$$

$$\langle x, \rho, \sigma \rangle \Downarrow \langle 2, \sigma \rangle \checkmark$$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio $x, y := x+100, x;$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l_y \mapsto 2][l_x \mapsto 102] \rangle \checkmark$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle 102, \sigma \rangle \checkmark$$

$$\langle y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l_y \mapsto 2] \rangle \checkmark$$

$$\langle x, \rho, \sigma \rangle \Downarrow \langle 2, \sigma \rangle \checkmark$$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio

```
x := x + 100;
y := x;
```

$\langle x := x + 100; y := x; , \rho , \sigma \rangle \rightarrow \langle \rho , \sigma' \rangle$ (non ci sono modifiche all'ambiente)

$\langle x, y := x + 100, x; , \rho , \sigma \rangle \rightarrow \langle \rho , \sigma[l_y \mapsto 2][l_x \mapsto 102] \rangle$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio

$x := x + 100;$
 $y := x;$

$$\frac{\langle E, \rho, \sigma \rangle \Downarrow v \quad \rho(\text{Id}) = l}{\langle \text{Id} := E; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l \mapsto v] \rangle}$$

$$\langle x := x + 100; y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle$$

$$\langle x := x + 100; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_1 \rangle$$

$$\langle y := x; , \rho, \sigma_1 \rangle \rightarrow \langle \rho, \sigma_? \rangle$$

$$\langle x, y := x + 100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l_y \mapsto 2][l_x \mapsto 102] \rangle$$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio

$x := x + 100;$
 $y := x;$

$$\frac{\langle E, \rho, \sigma \rangle \Downarrow v \quad \rho(l) = l}{\langle l := E; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l \mapsto v] \rangle}$$

$$\langle x := x+100; y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle$$

$$\langle x := x+100; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_1 \rangle \quad \text{con } \sigma_1 = \sigma[l_x \mapsto 102]$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle 102, \sigma \rangle \quad (\text{valutazione di espressione pura})$$

$$\langle y := x; , \rho, \sigma_1 \rangle \rightarrow \langle \rho, \sigma_? \rangle$$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l_y \mapsto 2][l_x \mapsto 102] \rangle$$

Esempio

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

$x := x + 100;$
 $y := x;$

$$\frac{\langle E, \rho, \sigma \rangle \Downarrow v \quad \rho(\text{Id}) = l}{\langle \text{Id} := E; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l \mapsto v] \rangle}$$

$$\langle x := x+100; y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma? \rangle$$

$$\langle x := x+100; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l_x \mapsto 102] \rangle \checkmark$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle 102, \sigma \rangle \checkmark$$

$$\langle y := x; , \rho, \sigma[l_x \mapsto 102] \rangle \rightarrow \langle \rho, \sigma? \rangle$$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l_y \mapsto 2][l_x \mapsto 102] \rangle$$

Esempio

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

$x := x + 100;$
 $y := x;$

$$\frac{\langle E, \rho, \sigma \rangle \Downarrow v \quad \rho(\text{Id}) = l}{\langle \text{Id} := E; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l \mapsto v] \rangle}$$

$$\langle x := x+100; y := x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma_? \rangle$$

$$\langle x := x+100; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l_x \mapsto 102] \rangle \checkmark$$

$$\langle x+100, \rho, \sigma \rangle \Downarrow \langle 102, \sigma \rangle \checkmark$$

$$\langle y := x; , \rho, \sigma[l_x \mapsto 102] \rangle \rightarrow \langle \rho, \sigma_? \rangle \quad \text{con } \sigma_? = \sigma[l_x \mapsto 102][l_y \mapsto 102]$$

$$\langle x, \rho, \sigma[l_x \mapsto 102] \rangle \Downarrow \langle 102, \sigma \rangle \quad (\text{valutazione di espressione pura})$$

$$\langle x, y := x+100, x; , \rho, \sigma \rangle \rightarrow \langle \rho, \sigma[l_y \mapsto 2][l_x \mapsto 102] \rangle$$

$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio

$x := x + 100;$
 $y := x;$

$$\langle x := x+100; y := x; , \rho , \sigma \rangle \rightarrow \langle \rho , \sigma[l_x \mapsto 102, l_y \mapsto 102] \rangle \checkmark$$

$$\langle x := x+100; , \rho , \sigma \rangle \rightarrow \langle \rho , \sigma[l_x \mapsto 102] \rangle \checkmark$$

$$\langle x+100 , \rho , \sigma \rangle \Downarrow \langle 102 , \sigma \rangle \checkmark$$

$$\langle y := x; , \rho , \sigma[l_x \mapsto 102] \rangle \rightarrow \langle \rho , \sigma[l_x \mapsto 102, l_y \mapsto 102] \rangle \checkmark$$

$$\langle x , \rho , \sigma[l_x \mapsto 102] \rangle \Downarrow \langle 102 , \sigma \rangle \checkmark$$

$$\langle x, y := x+100, x; , \rho , \sigma \rangle \rightarrow \langle \rho , \sigma[l_y \mapsto 2][l_x \mapsto 102] \rangle$$

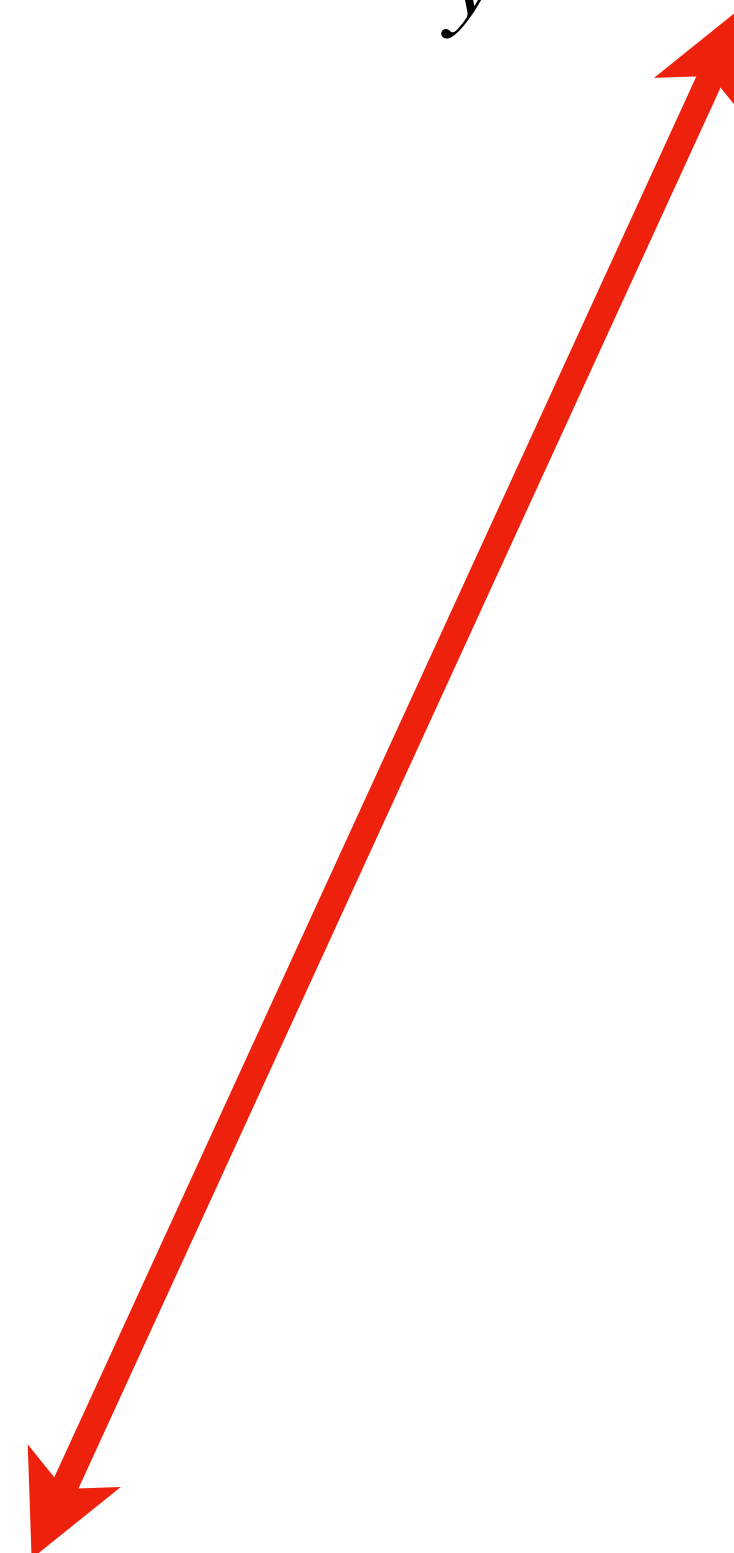
$$\rho = [x \mapsto l_x, y \mapsto l_y,]$$

$$\sigma = [l_x \mapsto 2, l_y \mapsto 20]$$

Esempio

$x := x + 100;$
 $y := x;$

$$\langle x := x + 100; y := x; , \rho , \sigma \rangle \rightarrow \langle \rho , \sigma[l_x \mapsto 102, l_y \mapsto 102] \rangle$$



$$\langle x, y := x + 100, x; , \rho , \sigma \rangle \rightarrow \langle \rho , \sigma[l_y \mapsto 2][l_x \mapsto 102] \rangle$$

Direttiva return

espressioni con effetti collaterali

Direttiva return

Vogliamo poter definire un'espressione che possa apportare qualsiasi tipo di modifiche alla memoria prima di restituire un risultato

Introduciamo un blocco di comandi che alla fine dell'esecuzione restituisce il valore aggiornato di un'espressione qualsiasi

Esempio: Supponiamo di voler incrementare un contatore e restituirne il valore aggiornato

```
{ count := count+1; return count; }
```

Sintassi della direttiva return

$E ::= \dots$

$| \{ C \text{ return } E; \}$

Attenzione!! questa espressione cambia la memoria quindi introduce effetti collaterali!!!

Consente di valutare l'espressione E nello stato prodotto dall'esecuzione del comando C per poi restituire il valore di E come risultato e la memoria modificata

L'ambito di visibilità di eventuali dichiarazioni introdotte dal comando C è confinato localmente al comando stesso e alla valutazione dell'espressione E

Controllo dei tipi della direttiva return

Data la seguente definizione di variabili libere

$$fv(\{ \textcolor{blue}{C} \text{ return } \textcolor{red}{E}; \}) = fv(\textcolor{red}{C}) \cup (fv(\textcolor{red}{E}) \setminus dv(\textcolor{red}{C}))$$

Per controllare il tipo di questa espressione ci serve:

- $\textcolor{red}{E} ::= \dots$
- un giudizio di tipo $\Gamma \vdash \textcolor{red}{C} : \Gamma_1$ per il comando;
 - verificare che l'espressione sia tipabile nell'ambiente di tipo modificato $\Gamma[\Gamma_1]$

$$\frac{\Gamma \vdash \textcolor{red}{C} : \Gamma_1 \quad \Gamma[\Gamma_1] \vdash \textcolor{red}{E} : T}{\Gamma \vdash \{ \textcolor{blue}{C} \text{ return } \textcolor{red}{E}; \} : T}$$

Attenzione!! il nuovo ambiente di tipo serve **solo** per tipare $\textcolor{red}{E}$ perchè la direttiva è racchiusa in un blocco

Semantica operativa della direttiva return

$E ::= \dots$

| $\{ C \text{ return } E; \}$

Per eseguire questa espressione nello stato (ρ, σ) :

- eseguiamo il comando C in (ρ, σ) ottenendo un nuovo (ρ_1, σ_1)
- valutiamo E nello stato (ρ_1, σ_1) ottenendo un valore v e una
- nuova memoria σ_2

$$\frac{\langle C, \rho, \sigma \rangle \rightarrow (\rho_1, \sigma_1) \quad \langle E, \rho_1, \sigma_1 \rangle \Downarrow (v, \sigma_2)}{\langle \{ C \text{ return } E; \}, \rho, \sigma \rangle \Downarrow (v, \sigma_2)}$$

Esercizio

Scrivere un'espressione che restituisce il massimo tra due variabili a e b

Esercizio

Scrivere un'espressione che restituisce il massimo tra due variabili a e b

```
{  
    int max = a;  
    if (b>max) { max := b; }  
    return max;  
}
```

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T?$

$$\frac{\Gamma \vdash C : \Gamma_1 \quad \Gamma[\Gamma_1] \vdash E : T}{\Gamma \vdash \{ C \text{ return } E; \} : T}$$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_?$

$\Gamma \vdash C : \Gamma_1 \quad \Gamma[\Gamma_1] \vdash E : T$
$\Gamma \vdash \{ C \text{ return } E; \} : T$

$\Gamma[\Gamma_?] \vdash \text{max} : T_?$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_?$ con $\Gamma_? = \Gamma_{1?}[\Gamma_{2?}]$

$\Gamma \vdash \text{int max} = a; : \Gamma_{1?}$

$\Gamma[\Gamma_{1?}] \vdash \text{if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_{2?}$

$$\frac{\Gamma \vdash C_1 : \Gamma_1 \quad \Gamma[\Gamma_1] \vdash C_2 : \Gamma_2}{\Gamma \vdash C_1 C_2 : \Gamma_1[\Gamma_2]}$$

$\Gamma[\Gamma_?] \vdash \text{max} : T_?$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_?$ con $\Gamma_? = \Gamma_{1?}[\Gamma_{2?}]$

$\Gamma \vdash \text{int max} = a; : \Gamma_{1?}$ con $\Gamma_{1?} = \{ \text{max} : \text{int} \}$ ✓

$\Gamma[\Gamma_{1?}] \vdash \text{if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_{2?}$

$\Gamma[\Gamma_?] \vdash \text{max} : T_?$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_?$ con $\Gamma_? = \{ \text{max} : \text{int} \} [\Gamma_{2?}]$

$\Gamma \vdash \text{int max} = a; : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_{2?}$

$$\frac{\Gamma \vdash E : \text{bool} \quad \Gamma \vdash C_1 : \Gamma_1 \quad \Gamma \vdash C_2 : \Gamma_2}{\Gamma \vdash \text{ if } (E) \{ C_1 \} \text{ else } \{ C_2 \} : \emptyset}$$

$\Gamma[\Gamma_?] \vdash \text{max} : T_?$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_?$ con $\Gamma_? = \{ \text{max} : \text{int} \}$

$\Gamma \vdash \text{int max} = a; : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{if } (b > \text{max}) \{ \text{max} := b; \} : \emptyset$

$\Gamma[\text{max} : \text{int}] \vdash b > \text{max} : \text{bool}$

$\Gamma[\text{max} : \text{int}] \vdash \text{max} := b; : \Gamma_3$

$\Gamma[\Gamma_?] \vdash \text{max} : T_?$

$$\frac{\Gamma \vdash E : \text{bool} \quad \Gamma \vdash C_1 : \Gamma_1 \quad \Gamma \vdash C_2 : \Gamma_2}{\Gamma \vdash \text{if } (E) \{ C_1 \} \text{ else } \{ C_2 \} : \emptyset}$$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int } \text{max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int } \text{max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_?$ con $\Gamma_? = \{ \text{max} : \text{int} \}$

$\Gamma \vdash \text{int } \text{max} = a; : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{if } (b > \text{max}) \{ \text{max} := b; \} : \emptyset$

$\Gamma[\text{max} : \text{int}] \vdash b > \text{max} : \text{bool} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{max} := b; : \Gamma_3$

$\Gamma[\Gamma_?] \vdash \text{max} : T_?$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int } \text{max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int } \text{max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_?$ con $\Gamma_? = \{ \text{max} : \text{int} \}$

$\Gamma \vdash \text{int } \text{max} = a; : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{if } (b > \text{max}) \{ \text{max} := b; \} : \emptyset$

$\Gamma[\text{max} : \text{int}] \vdash b > \text{max} : \text{bool} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{max} := b; : \emptyset \checkmark$

$\Gamma[\Gamma_?] \vdash \text{max} : T_?$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \Gamma_?$ con $\Gamma_? = \{ \text{max} : \text{int} \}$

$\Gamma \vdash \text{int max} = a; : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{if } (b > \text{max}) \{ \text{max} := b; \} : \emptyset \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash b > \text{max} : \text{bool} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{max} := b; : \emptyset \checkmark$

$\Gamma[\Gamma_?] \vdash \text{max} : T_?$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : T_?$

$\Gamma \vdash \text{int max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma \vdash \text{int max} = a; : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \emptyset \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash b > \text{max} : \text{bool} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{max} := b; : \emptyset \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{max} : T_?$

Esercizio

Controllare che l'espressione

$E1 = \{ \text{int } \text{max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} \text{ return max; } \}$

sia ben tipata nell'ambiente di tipo $\Gamma = \{ a : \text{int}, b : \text{int} \}$

$\Gamma \vdash E1 : \text{int} \checkmark$

$\Gamma \vdash \text{int } \text{max} = a; \text{ if } (b > \text{max}) \{ \text{max} := b; \} : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma \vdash \text{int } \text{max} = a; : \{ \text{max} : \text{int} \} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{if } (b > \text{max}) \{ \text{max} := b; \} : \emptyset \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash b > \text{max} : \text{bool} \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{max} := b; : \emptyset \checkmark$

$\Gamma[\text{max} : \text{int}] \vdash \text{max} : \text{int} \checkmark$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v?, \sigma? \rangle$

$$\frac{\langle \text{C}, \rho, \sigma \rangle \rightarrow (\rho_1, \sigma_1) \quad \langle \text{E}, \rho_1, \sigma_1 \rangle \Downarrow (v, \sigma_2)}{\langle \{ \text{C return E; } \}, \rho, \sigma \rangle \Downarrow (v, \sigma_2)}$$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v?, \sigma? \rangle$

$\langle \text{int max=a; if (b>max) {max := b;}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$$\frac{\langle \text{C}, \rho, \sigma \rangle \rightarrow (\rho_1, \sigma_1) \quad \langle \text{E}, \rho_1, \sigma_1 \rangle \Downarrow (v, \sigma_2)}{\langle \{ \text{C return E; } \}, \rho, \sigma \rangle \Downarrow (v, \sigma_2)}$$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v?, \sigma? \rangle$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v_?, \sigma_? \rangle$

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho_2, \sigma_2 \rangle$

$\langle \text{if (b>max) \{max := b;\}}, \rho_2, \sigma_2 \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v_?, \sigma_? \rangle$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v?, \sigma? \rangle$

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho_2, \sigma_2 \rangle$ con $\rho_2 = \rho[\text{max} \mapsto l_m]$ e $\sigma_2 = \sigma[l_m \mapsto 12]$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle$ ✓ // espressione pura

$\langle \text{if (b>max) \{max := b;\}}, \rho_2, \sigma_2 \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v?, \sigma? \rangle$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v?, \sigma? \rangle$

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle \checkmark$ // espressione pura

$\langle \text{if (b>max) \{max := b;\}}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v?, \sigma? \rangle$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v_?, \sigma_? \rangle$

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle \checkmark$ // espressione pura

$\langle \text{if (b>max) \{max := b;\}}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{b>max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \Downarrow \langle \text{true}, \sigma[l_m \mapsto 12] \rangle$

$\langle \text{max := b;}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v_?, \sigma_? \rangle$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v?, \sigma? \rangle$

$\langle \text{int max=a; if (b>max) {max := b;}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle \checkmark$ // espressione pura

$\langle \text{if (b>max) {max := b;}}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{b>max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \Downarrow \langle \text{true}, \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle \text{max := b;}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v?, \sigma? \rangle$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v_?, \sigma_? \rangle$

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle \checkmark$ // espressione pura

$\langle \text{if (b>max) \{max := b;\}}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{b>max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \Downarrow \langle \text{true}, \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle \text{max := b;}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v_?, \sigma_? \rangle$ con $\rho_1 = \rho[\text{max} \mapsto l_m]$ e $\sigma_1 = \sigma[l_m \mapsto 18]$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v_?, \sigma_? \rangle$

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle \checkmark$ // espressione pura

$\langle \text{if (b>max) \{max := b;\}}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{b>max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \Downarrow \langle \text{true}, \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle \text{max := b;}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v_?, \sigma_? \rangle$ con $\rho_1 = \rho[\text{max} \mapsto l_m]$ e $\sigma_1 = \rho[l_m \mapsto 18]$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v_?, \sigma_? \rangle$

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle \checkmark$ // espressione pura

$\langle \text{if (b>max) \{max := b;\}}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{b>max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \Downarrow \langle \text{true}, \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle \text{max := b;}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{max}, \rho_1, \sigma_1 \rangle \Downarrow \langle v_?, \sigma_? \rangle$ con $\rho_1 = \rho[\text{max} \mapsto l_m]$ e $\sigma_1 = \sigma[l_m \mapsto 18]$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle v_?, \sigma_? \rangle$

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 18] \rangle \checkmark$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle \checkmark$ // espressione pura

$\langle \text{if (b>max) \{max := b;\}}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{b>max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \Downarrow \langle \text{true}, \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle \text{max := b;}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 18] \rangle \Downarrow \langle v_?, \sigma_? \rangle$ con $\rho_1 = \rho[\text{max} \mapsto l_m]$ e $\sigma_1 = \sigma[l_m \mapsto 18]$

Esercizio

Valutare l'espressione

E1 = { int max=a; if (b>max) {max := b;} return max; }

nello stato (ρ, σ) con $\rho = [a \mapsto l_a, b \mapsto l_b]$ e $\sigma = [l_a \mapsto 12, l_y \mapsto 18]$

$\langle \text{E1}, \rho, \sigma \rangle \Downarrow \langle 18, \sigma[l_m \mapsto 18] \rangle \checkmark$ // l_m non accessibile da ρ

$\langle \text{int max=a; if (b>max) \{max := b;\}}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 18] \rangle \checkmark$

$\langle \text{int max=a;}, \rho, \sigma \rangle \rightarrow \langle \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle a, \rho, \sigma \rangle \Downarrow \langle 12, \sigma \rangle \checkmark$ // espressione pura

$\langle \text{if (b>max) \{max := b;\}}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{b>max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \Downarrow \langle \text{true}, \sigma[l_m \mapsto 12] \rangle \checkmark$

$\langle \text{max := b;}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 12] \rangle \rightarrow \langle \rho_1, \sigma_1 \rangle \checkmark$

$\langle \text{max}, \rho[\text{max} \mapsto l_m], \sigma[l_m \mapsto 18] \rangle \Downarrow \langle v_? = 18, \sigma_? = \sigma[l_m \mapsto 18] \rangle \checkmark$



COMING SOON

Le funzioni

Esercizio

Scrivere un'espressione che restituisce il massimo tra due variabili a e b

```
{  
    int max = a;  
    if (b>max) { max := b; }  
    return max;  
}
```


Esercizio

Scrivere **una funzione** che restituisce il massimo tra due **valori interi**

```
{ // a e b diventano parametri formali
    int max = a;
    if (b>max) { max := b; }
    return max;
}
```

Esercizio

Scrivere **una funzione** che restituisce il massimo tra due **valori interi**

```
(int a, int b) { // diamo un nome alla funzione
    int max = a;
    if (b > max) { max := b; }
    return max;
}
```

Esercizio

Scrivere **una funzione** che restituisce il massimo tra due **valori interi**

```
max(int a, int b) {  
    int m = a; // evitiamo conflitti di nomi  
    if (b>m) { m := b; }  
    return m;  
} // qual è il tipo del risultato?
```

Esercizio

Scrivere **una funzione** che restituisce il massimo tra due **valori interi**

```
int max(int a, int b) {  
    int m = a;  
    if (b>m) { m := b; }  
    return m;  
} // questa è una dichiarazione di funzione
```

Esercizio

Scrivere **una funzione** che restituisce il massimo tra due **valori interi**

```
int max(int a, int b) {  
    int m = a;  
    if (b>m) { m := b; }  
    return m;  
}
```

// questa è una **dichiarazione di funzione**

max(12,18) // questa è un'**invocazione di funzione**

max(x,18) // un'altra **invocazione di funzione**