

Ricorsione

approfondimenti su ricorsione in coda e chiusure

Chiusure statiche

Negli esempi precedenti, le dichiarazioni di funzioni ausiliarie sono locali alle funzioni che le usano. Infatti il sistema di tipi non permette di riferire identificatori esterni alla dichiarazione di funzione.

Le **chiusure** permettono di gestire le variabili libere nel corpo delle funzioni.

Nel caso delle **chiusure statiche**, la rappresentazione della funzione che viene inserita nell'ambiente dovrà contenere a sua volta un piccolo ambiente che permette di risolvere tutti i riferimenti alle variabili libere presenti nel corpo della funzione.

A differenza delle **chiusure dinamiche**, tale ambiente collega le occorrenze delle variabili libere a quelle presenti quando la funzione è stata definita, e non dove è chiamata.

Chiusure statiche

```
// con chiusure, x può essere definita all'esterno  
  
int add3(y){  
    int x=3;  
    int add(z){  
        return x + z  
    }  
    return (add(y));  
}
```

add3(2)

Chiusure dinamica

// con chiusure, x può essere definita all'esterno

```
int add(z){  
    return z + y }
```

```
x = 10;  
stampa add(2)
```

Esercizi su ricorsione

Esercizio

Controllare ricorsivamente se un array di interi è ordinato

Esercizio

Controllare ricorsivamente se un array di interi è ordinato

```
bool sorted(int[] a, int from) {  
    bool ord = true;  
    if (from < a.length-1) {  
        if (a[from]<=a[from+1]) { ord := sorted(a, from+1); }  
        else { ord := false; }  
    }  
    return ord;  
}
```

Esercizio

Controllare ricorsivamente se un array di interi è ordinato

```
bool ordinato(int[] a) {  
    bool sorted(int[] a, int from) {  
        bool ord = true;  
        if (from < a.length-1) {  
            if (a[from]<=a[from+1]) { ord := sorted(a, from+1); }  
            else { ord := false; }  
        }  
        return ord;  
    }  
    return sorted(a, 0);  
}
```

Esercizio

Calcolare ricorsivamente il valore minimo in un array di interi

Esercizio

Calcolare ricorsivamente il valore minimo in un array di interi

```
int minimo(int[] a) {  
    int min(int[] a, int from, int m) {  
        if (from < a.length){  
            if (a[from] < m) { m := a[from]; }  
            m := min(a, from+1, m);  
        }  
        return m;  
    }  
    return min(a, 1, a[0]);  
}
```

Esercizio

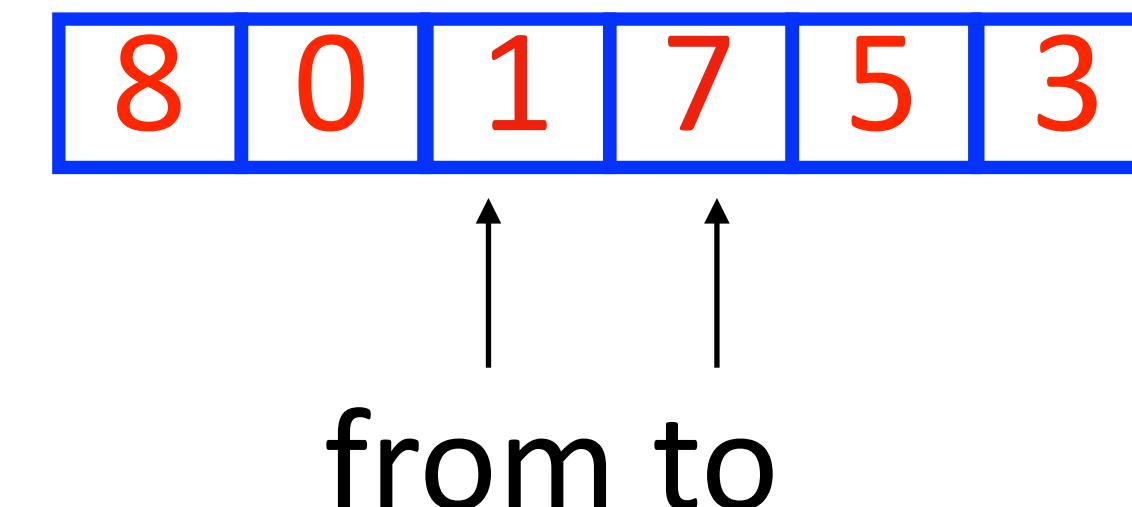
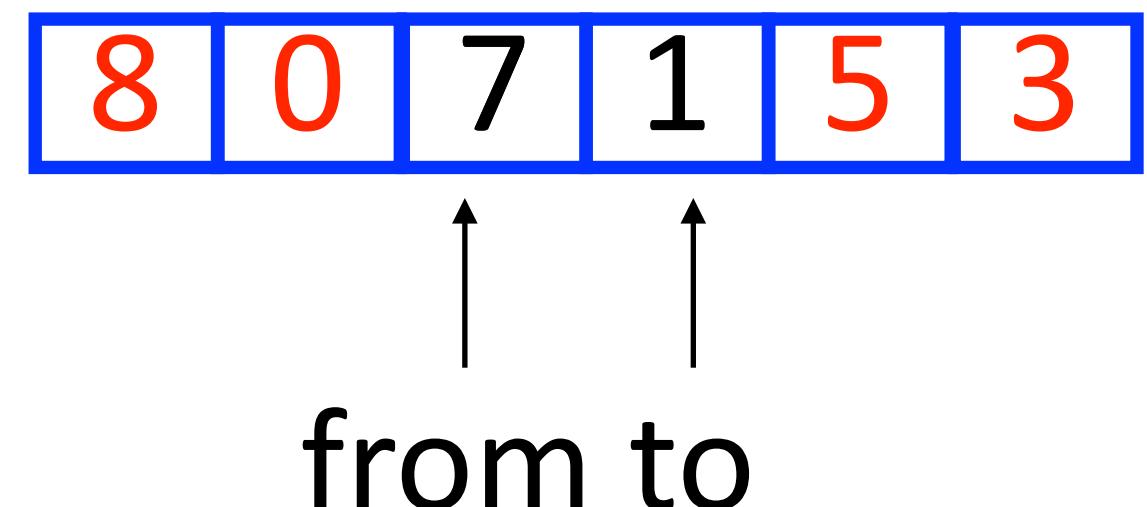
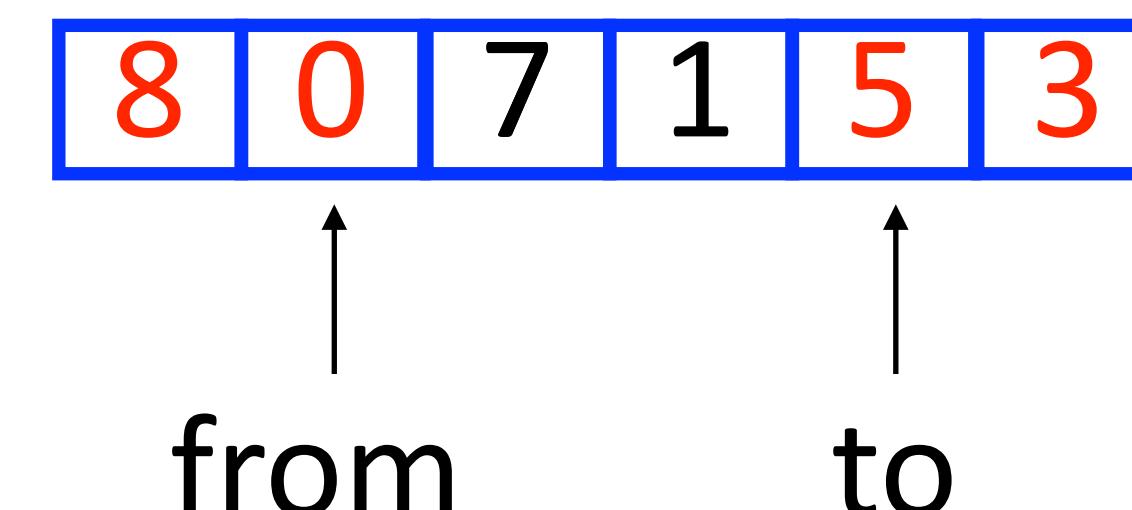
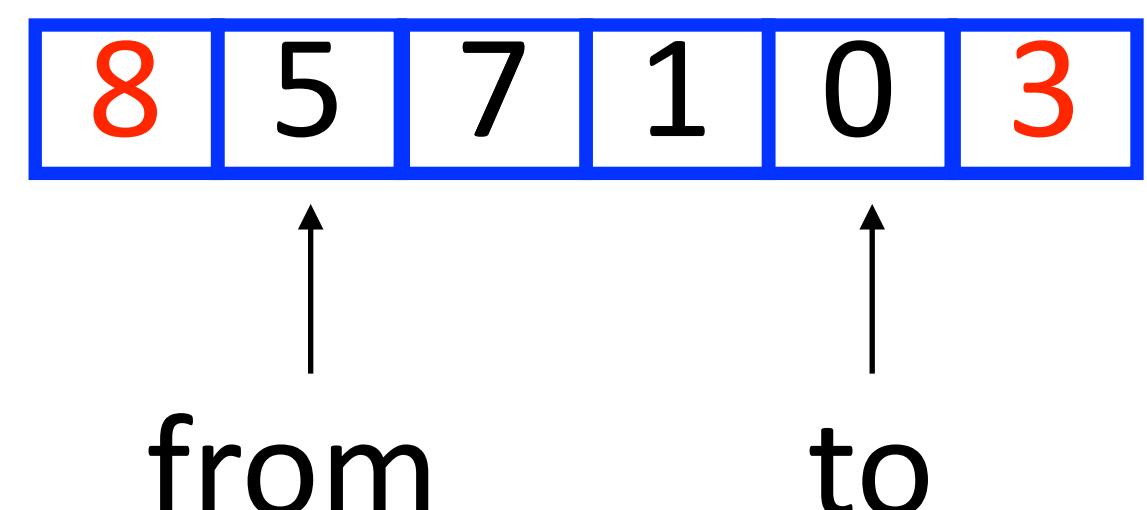
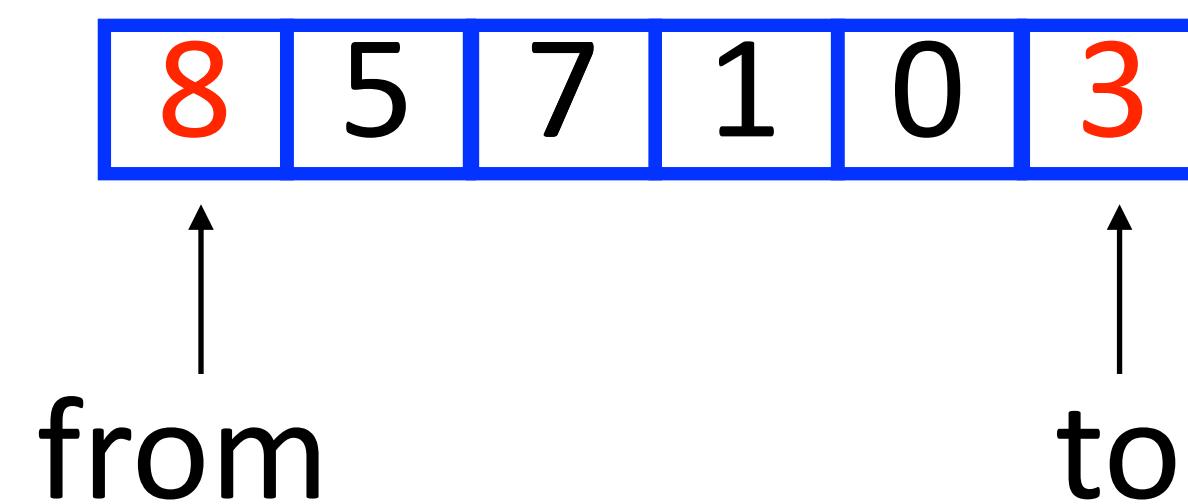
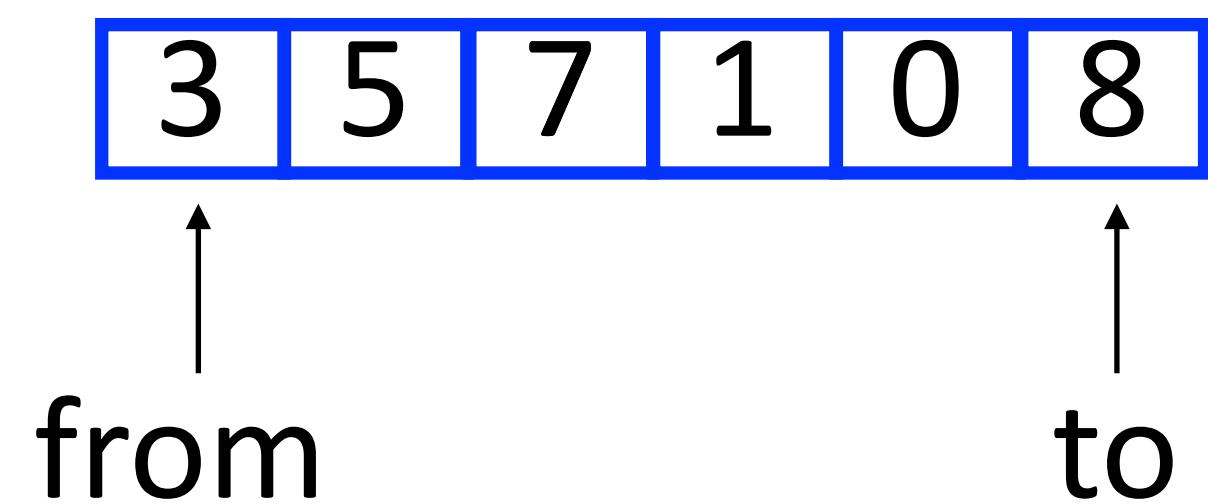
Invertire ricorsivamente gli elementi di un array di interi



Attenzione: il metodo appena visto si applica a tutti i problemi sugli array.
L'idea è di trovare la giusta decomposizione in:
parte corrente, che sfruttiamo nella chiamata corrente;
e resto dell'array, da far controllare alla invocazione ricorsiva.

Esercizio

Invertire ricorsivamente gli elementi di un array di interi



Esercizio

Invertire ricorsivamente gli elementi di un array di interi

```
bool inverti(int[] a) {  
    bool inv(int[] a, int from, int to) {  
        ...  
    }  
}
```

Esercizio

Invertire ricorsivamente gli elementi di un array di interi

```
bool inverti(int[] a) {  
    bool inv(int[] a, int from, int to) {  
        if (from < to) {  
            a[from],a[to] := a[to],a[from]; // non si potrebbe...  
            bool b := inv(a, from+1, to-1);  
        }  
        return true;  
    }  
    return inv(a, 0, a.length-1);  
}
```

Esercizio

Invertire ricorsivamente gli elementi di un array di interi

```
inverti(int[] a) { // se avessimo studiato anche le procedure...
    inv(int[] a, int from, int to) {
        if (from < to) {
            a[from],a[to] := a[to],a[from]; // non si potrebbe...
            inv(a, from+1, to-1);
        }
        // non usiamo return
    }
    inv(a, 0, a.length-1); // non usiamo return
}
```

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset

$$\emptyset \vdash C : \{ \text{somma} : (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$$

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$

$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$$
$$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$$
$$\Gamma \vdash \text{int } s = x; C1 : \Gamma_1? ??$$
$$\Gamma[\Gamma_1?] \vdash s : \text{int} ??$$

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$$
$$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$$
$$\Gamma \vdash \text{int } s = x; C1 : \Gamma_1? ?? \quad \text{con } \Gamma_1? = \Gamma_2?[\Gamma_3?]$$
$$\Gamma \vdash \text{int } s = x; : \Gamma_2? ??$$
$$\Gamma[\Gamma_2?] \vdash c1 : \Gamma_3? ??$$
$$\Gamma[\Gamma_1?] \vdash s : \text{int} ??$$

Esercizio

Controllare che il comando

$C = \text{int somma(int } x, \text{ int } y) \{ \text{ int } s = x; C1 \text{ return } s; \}$

$C1 = \text{if } (y > 0) \{ s := 1 + \text{somma}(x, y-1); \}$

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma: } (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$\emptyset \vdash C : \{ \text{somma: } (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$

$\Gamma \vdash \{ \text{ int } s = x; C1 \text{ return } s; \} : \text{int} ??$

$\Gamma \vdash \text{int } s = x; C1 : \Gamma_1? ?? \quad \text{con } \Gamma_1? = \{ s : \text{int} \}[\Gamma_3?]$

$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$

$\Gamma[s : \text{int}] \vdash C1 : \Gamma_3? ??$

$\Gamma[\Gamma_1?] \vdash s : \text{int} ??$

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$$
$$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$$
$$\Gamma \vdash \text{int } s = x; C1 : \Gamma_1? ?? \quad \text{con } \Gamma_1? = \{ s : \text{int} \}[\emptyset]$$
$$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$$
$$\Gamma[s : \text{int}] \vdash C1 : \emptyset ??$$
$$\Gamma[s : \text{int}] \vdash y > 0 : \text{bool} ??$$
$$\Gamma[s : \text{int}] \vdash s := 1 + \text{somma}(x, y-1) : \emptyset ??$$
$$\Gamma[\Gamma_1?] \vdash s : \text{int} ??$$

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$$
$$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$$
$$\Gamma \vdash \text{int } s = x; C1 : \{ s : \text{int} \} ??$$
$$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$$
$$\Gamma[s : \text{int}] \vdash C1 : \emptyset ??$$
$$\Gamma[s : \text{int}] \vdash y > 0 : \text{bool} \checkmark$$
$$\Gamma[s : \text{int}] \vdash s := 1 + \text{somma}(x, y-1) : \emptyset ??$$
$$\Gamma[s : \text{int}] \vdash s : \text{int} ??$$

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$

$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$

$\Gamma \vdash \text{int } s = x; C1 : \{ s : \text{int} \} ??$

$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$

$\Gamma[s : \text{int}] \vdash C1 : \emptyset ??$

$\Gamma[s : \text{int}] \vdash y > 0 : \text{bool} \checkmark$

$\Gamma[s : \text{int}] \vdash s := 1 + \text{somma}(x, y-1) : \emptyset \checkmark$

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

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$$
$$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$$
$$\Gamma \vdash \text{int } s = x; C1 : \{ s : \text{int} \} ??$$
$$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$$
$$\Gamma[s : \text{int}] \vdash C1 : \emptyset \checkmark$$
$$\Gamma[s : \text{int}] \vdash y > 0 : \text{bool} \checkmark$$
$$\Gamma[s : \text{int}] \vdash s := 1 + \text{somma}(x, y-1) : \emptyset \checkmark$$
$$\Gamma[s : \text{int}] \vdash s : \text{int} ??$$

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$

$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$

$\Gamma \vdash \text{int } s = x; C1 : \{ s : \text{int} \} \checkmark$

$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$

$\Gamma[s : \text{int}] \vdash C1 : \emptyset \checkmark$

$\Gamma[s : \text{int}] \vdash y > 0 : \text{bool} \checkmark$

$\Gamma[s : \text{int}] \vdash s := 1 + \text{somma}(x, y-1) : \emptyset \checkmark$

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

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$

$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} ??$

$\Gamma \vdash \text{int } s = x; C1 : \{ s : \text{int} \} \checkmark$

$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$

$\Gamma[s : \text{int}] \vdash C1 : \emptyset \checkmark$

$\Gamma[s : \text{int}] \vdash y > 0 : \text{bool} \checkmark$

$\Gamma[s : \text{int}] \vdash s := 1 + \text{somma}(x, y-1) : \emptyset \checkmark$

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

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} ??$

$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} \checkmark$

$\Gamma \vdash \text{int } s = x; C1 : \{ s : \text{int} \} \checkmark$

$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$

$\Gamma[s : \text{int}] \vdash C1 : \emptyset \checkmark$

$\Gamma[s : \text{int}] \vdash y > 0 : \text{bool} \checkmark$

$\Gamma[s : \text{int}] \vdash s := 1 + \text{somma}(x, y-1) : \emptyset \checkmark$

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

Esercizio

Controllare che il comando

```
C = int somma(int x, int y) { int s = x; C1 return s; }
```

```
C1 = if (y>0) { s := 1 + somma(x, y-1); }
```

sia ben tipato nell'ambiente di tipo \emptyset con $\Gamma = \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int}, x : \text{int}, y : \text{int} \}$

$\emptyset \vdash C : \{ \text{somma}: (\text{int} \times \text{int}) \rightarrow \text{int} \} \checkmark$

$\Gamma \vdash \{ \text{int } s = x; C1 \text{ return } s; \} : \text{int} \checkmark$

$\Gamma \vdash \text{int } s = x; C1 : \{ s : \text{int} \} \checkmark$

$\Gamma \vdash \text{int } s = x; : \{ s : \text{int} \} \checkmark$

$\Gamma[s : \text{int}] \vdash C1 : \emptyset \checkmark$

$\Gamma[s : \text{int}] \vdash y > 0 : \text{bool} \checkmark$

$\Gamma[s : \text{int}] \vdash s := 1 + \text{somma}(x, y-1) : \emptyset \checkmark$

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