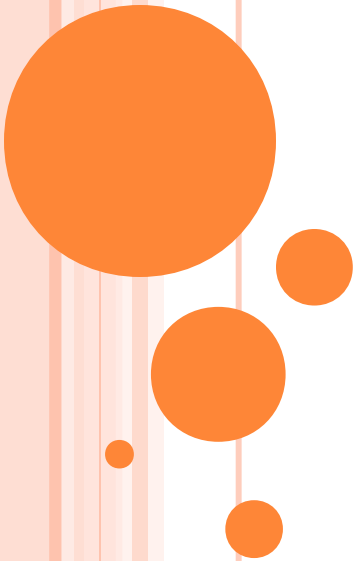

TRIPLE DI HOARE: SEQUENZE (ARRAY) E AGGIORNAMENTO SELETTIVO

Corso di Logica per la Programmazione

A.A. 2012/13

SEQUENZE: SINTASSI

- Estendiamo il linguaggio per usare *array* o *sequenze*
- Scriviamo **v : array [a,b) of T** per dire che **v** è una variabile di tipo:
  "sequenza di elementi di tipo **T** con dominio **[a,b)**",
  dove **T** può essere **int** o **bool**
- Il dominio di **v** viene indicato come *dom*(**v**)
- Scriviamo **v[i]** per denotare l'**i**-esimo elemento di **v**
- La **sintassi** delle espressioni diventa:

Exp ::= Const | Id | Ide[Exp] | (Exp) | Exp Op Exp | not Exp

SEQUENZE: SEMANTICA

- Ricordiamo che uno **stato** σ è una funzione

$$\sigma : \text{Ide} \rightarrow B \cup Z$$

- Estendiamo il concetto di stato: se **v** è un array di tipo **T**,

$\sigma(\mathbf{v})$: $dom(\text{v}) \rightarrow \mathbf{B}$ se **T = bool**

$\sigma(\mathbf{v})$: $dom(\text{v}) \rightarrow \mathbf{Z}$ se **T = int**

- Estendiamo la funzione di **interpretazione semantica**:

$E(\text{Ide[Exp]},\sigma) = E(\text{Ide},\sigma)(E(\text{Exp},\sigma))$ se $E(\text{Exp},\sigma) \in dom(\text{Ide})$

- Ide[Exp] non è sempre definito: estendiamo la funzione *def*

$$def(\text{Ide[Exp]}) = def(\text{Exp}) \wedge \text{Exp} \in dom(\text{Ide})$$

- Nota: le operazioni non possono essere applicate a sequenze, ma solo a singoli elementi di sequenze.
  Es: **a[2] < b[3]**, **a[2] * y + x** , ma non **a + b** !

AGGIORNAMENTO SELETTIVO

- Ogni elemento di una sequenza è una variabile, quindi può comparire a sinistra di un assegnamento. Es: **a[3] := 5**
- La semantica di questo comando (**assegnamento selettivo**) è data dal seguente assioma *(ass-sel):*

$$\{def(E) \wedge def(E') \wedge E \in dom(v) \wedge P[w/v]\} \quad v[E] := E' \quad \{P\}$$

dove $w = v[E'/E]$

- Si può usare anche la seguente regola derivata:

$$\frac{R \Rightarrow def(E) \wedge def(E') \wedge E \in dom(v) \wedge P[w/v] \qquad w = v[E'/E]}{\{R\}\ v[E] := E'\ \{P\}}$$

ESEMPIO DI AGGIORNAMENTO SELETTIVO

- Si verifichi la tripla:

  $\{k \in dom(v) \land (\forall\ i\ .\ i \in dom(v) \land i \neq k \Rightarrow v[i] > 0)\}$

  $v[k] := 3$

  $\{(\forall\ i\ .\ i \in dom(v) \Rightarrow v[i] > 0)\}$

- Applicando la regola, è sufficiente dimostrare:

  $k \in dom(v) \land (\forall\ i\ .\ i \in dom(v) \land i \neq k \Rightarrow v[i] > 0) \Rightarrow$
  $def(k) \land def(3) \land k \in dom(v) \land (\forall\ i\ .\ i \in dom(w) \Rightarrow w[i] > 0)$

  dove $w = v[3/k]$

- **Esercizio:** si completi la dimostrazione

ESERCIZIO: SCANSIONE DI SEQUENZA

- Si consideri il seguente programma annotato che conta il numero di elementi maggiori di zero in un array **a: array [0, n) of int**.

```
{a : array [0, n) of int }
 x, c := 0, 0;
{Inv : c = #{j : j ∈ [0, x) ∧ a[j] > 0 } ∧ x∈ [0, n] } {t : n - x}
while x < n do
        if (a[x] > 0) then  c := c+1   else skip   fi ;
        x := x + 1
endw
{Inv  ∧ ~(x < n)}
{ c = #{j : j ∈ [0, n) ∧ a[j] > 0 } }
```

- Scrivere e dimostrare la Condizione di Invarianza
- Scrivere e dimostrare la Condizione di Terminazione
- Scrivere e dimostrare la Condizione di Progresso