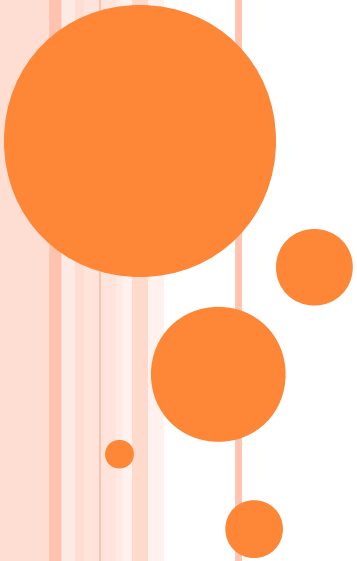

# ESERCIZI DI RICAPITOLAZIONE

**Corso di Logica per la Programmazione**

**A.A. 2010/11**

***Andrea Corradini, Paolo Mancarella***

# ESERCIZIO: FUNZIONE DI TERMINAZIONE

- Si consideri il seguente programma annotato

```
{x >0 ∧ y>0}
{Inv: T} {t : ?}
while not (x = y) do
        if x > y then x := x - 1; else x := x + 1; fi
endw
{x = y}
```

- Quale può essere una buona funzione di terminazione?
- Osserviamo che il corpo del ciclo fa sì che, ad ogni iterazione, diminuisca la *distanza* tra i valori delle variabili x e y
- Dunque, una buona *t* è

$$|x - y|$$

- Dimostrare progresso e terminazione

# ALTRI ESERCIZI DI FORMALIZZAZIONE

- Assumendo che **a,b : array [0, n) of nat**, si formalizzino i seguenti enunciati:

1. l'array **a** ha un solo minimo locale
2. **a** ha tutti elementi distinti
3. **b** è l'array **a** ordinato in senso crescente

# ALTRI ESERCIZI DI FORMALIZZAZIONE

- Utilizzando il calcolo del primo ordine si formalizzino i seguenti enunciati dichiarativi, indicando esplicitamente l'interpretazione intesa:

1) “Solo alcuni dei divisori di 60 sono anche divisori di 15”
2) “Ogni libro ha almeno un autore”
3) “Ci sono libri che hanno più di un autore”

# DIMOSTRAZIONI DEL PRIM'ORDINE

- Si provi che le seguenti formule sono valide:

$( \forall x . P \Rightarrow \sim Q ) \Rightarrow \sim ( \exists x . P \wedge Q )$

$( \forall x. P \Rightarrow Q ) \wedge \sim (\exists x. Q) \Rightarrow (\forall x. \sim P)$

$( \forall x . P ) \wedge \sim ( \exists x. P \wedge \sim R ) \Rightarrow ( \exists x . R )$

# SPECIFICA DI PROBLEMI DI PROGRAMMAZIONE

- Azzerare gli elementi di un array **a** con indice dispari
- Verificare che tutti gli elementi pari di un array **a** sono seguiti da un elemento dispari
- Calcolare la somma degli elementi dell'array **a** che sono uguali agli elementi di **b** nella medesima posizione
- Calcolare nella variabile **p** il numero degli elementi di **a** che non compaiono in **b**

# SPECIFICA DI PROBLEMI DI PROGRAMMAZIONE: soluzioni

- Azzerare gli elementi di un array **a** con indice dispari

  *Pre*: {a=V}

  *Post*: {($\forall$i. i$\in$ *dom*(a) $\Rightarrow$ ((*pari*(i) $\wedge$ a[i]=V[i]) $\vee$ (*dispari*(i) $\wedge$ a[i]=0))}

- Verificare che tutti gli elementi pari di un array **a** sono seguiti da un elemento dispari

  *Pre*: {a=V $\wedge$ *dom*(a)=[0,n)}

  *Post*: {check $\equiv$ (

  ($\forall$i. i$\in$ [0,n-1) $\Rightarrow$ (pari(a[i]) $\Rightarrow$ dispari(a[i+1])) $\wedge$ dispari(a[n-1]) )

# SPECIFICA DI PROBLEMI DI PROGRAMMAZIONE

- Calcolare la somma degli elementi dell'array **a** che sono uguali agli elementi di **b** nella medesima posizione

  *Pre*: $\{dom(a) = [0, n) \land dom(b) = [0, m)\}$

  *Post*: $\{sum = (\sum i.\ i \in [0,n) \land i \in [0,m) \land a[i]=b[i].\ a[i])$

- Calcolare nella variabile **p** il numero degli elementi di **a** che non compaiono in **b**

  *Pre*: $\{dom\ a = [0, n) \land dom\ b = [0, m)\}$

  *Post*: $\{p = (\#\{i: i \in [0,n) \mid \forall\ j.\ j \in [0,m) \Rightarrow a[i] \neq b[j]\})$

# ESERCIZIO: Calcolo MCD

- Si consideri il seguente programma annotato

$\{x = A \wedge y = B \wedge A>0 \wedge B>0\}$

$\{Inv : x>0 \wedge y>0 \wedge mcd(A,B) = mcd(x,y)\}\ \{t : x+y\}$

**while** x <> y **do**

    **if** x>y **then** x:= x-y; **else** y := y-x; **fi**

**endw**

$\{x=mcd(A,B)\}$

- Dimostrarne la correttezza, facendo uso delle seguenti note proprietà dell'operatore *mcd*:

$mcd\ (v,w) = v$ se v=w

$mcd\ (v,w) = mcd\ (v-w,w)$ se v>w

$mcd\ (v,w) = mcd\ (v,w-v)$ se v<w