

# Capitolo 1.6

Lorenzo Vaccarecci

5 Marzo 2024

## 1 Esempi di progettazione e analisi di algoritmi iterativi

### 1.1 Esempio didattico

// Pre:  $x = x_0 \wedge y = y_0$  con  $x_0, y_0 \in \mathbb{N}$

```
while(x!=0) {  
    x = x-1;  
    y = y+1;  
}
```

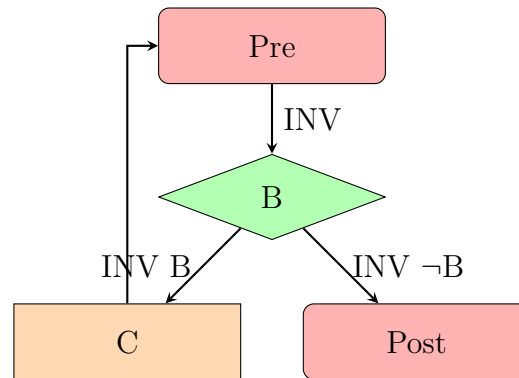
// Post:  $y = x_0 + y_0$  (viene eseguito  $x_0$  volte)

Se vale Pre all'inizio deve valere Post alla fine. (condizioni sullo stato)  
Come si pone? Trovo *INV* (invariante) t.c.

1. vale all'inizio  $pre \Rightarrow INV$
2. alla fine garantire la Post  $INV \wedge \neg B \Rightarrow Post$
3.  $// INV \wedge B$   
    C  
     $// INV$
4. Esiste una funzione di terminazione cioè una possibilità (dipende dallo stato) tale che sugli stati per cui vale  $INV \wedge B$  (cioè se entro nel ciclo)
  - decresce eseguendo *C* strettamente
  - è limitata inferiormente

*INV approssimazione di Post*

Proviamo che 1+2+3 valgono.



|                     |  |                      |
|---------------------|--|----------------------|
| 1+2+3 $\Rightarrow$ | se vale Pre prima di while(B)C e il ciclo termina, poi vale Post   | correttezza parziale |
|                     | se vale Pre prima di while(B)C allora il ciclo termina e vale Post | correttezza totale   |

+4 = quantità dipendente dallo stato che decresce strettamente a ogni iterazione  $\rightarrow$  permette di determinare il tipo di correttezza