

Lezione 07

invarianti e cicli

la prova di correttezza di un ciclo

PRE while (B) corpo; POST

invariante R vale sempre qui



La prova si struttura in 3 parti:

- 1) Condizione iniziale: $PRE \Rightarrow R$ (R vale la prima volta)
- 2) Invarianza: $R \ \&\& \ B \ \langle \text{corpo} \rangle \ R$
- 3) Terminazione: $R \ \&\& \ ! B \Rightarrow POST$

Esempio 1: lettura da cin di 10 valori interi

dobbiamo contare le letture : `int j=0;` e

e abbiamo bisogno di un ciclo il cui corpo viene eseguito 10 volte

```
while (j < 10) // j conta le letture fatte, se j<10 dobbiamo continuare
{
    fa una lettura in più e aumenta j
}
R= (0<= j <= 10) && ( letti j valori)
```

```
//PRE=(cin contiene almeno 10 valori)
int j=0, x;
while (j < 10) // R=(0<=j<=10) && (letti j valori)
{
    cin >>x;
    j=j+1;
}
//POST= (letti 10 valori da cin)
```

$R = (0 \leq j \leq 10) \ \&\& \text{ (letti } j \text{ valori)}$

1) Condizione iniziale:

con $j=0$, R diventa $0 \leq 0 \leq 10$ ok
letti 0 valori ok

2) invarianza:

$R \ \&\& \ j < 10 \Rightarrow$ che abbiamo letto $j < 10$ valori e quindi dobbiamo eseguire il corpo che fa la lettura $(j+1)$ -esima e poi aumenta j di 1 e da $j < 10$ segue che $j+1 \leq 10$ e quindi vale R

3) $R \ \&\& \ ! (j < 10) = R \ \&\& \ (j \geq 10) \Rightarrow j = 10 \Rightarrow$ letti 10 interi \Rightarrow
POST = (letti 10 valori)

Esempio 2: somma di 10 interi letti

//PRE=(cin contiene almeno 10 valori)

int j=0, x, sum=0;

while (j < 10) // R=($0 \leq j \leq 10$) && (letti j valori) && (sum è la loro
somma)

{

cin >> x;

j=j+1;

sum=sum+x;

}

//POST= (letti 10 valori da cin e sum è la loro somma)

$R = (0 \leq j \leq 10) \ \&\& \text{ (letti } j \text{ valori e sum è la loro somma)}$

Correttezza

1) Condizione iniziale:

con $j=0$, R diventa $0 \leq 0 \leq 10$ ok

letti 0 valori, somma=0 ok

2) invarianza:

$R \ \&\& \ j < 10 \Rightarrow$ che abbiamo letto $j < 10$ valori e quindi dobbiamo eseguire il corpo, che fa la lettura $(j+1)$ -esima, lo somma a sum e aumenta j di 1 \Rightarrow vale R

3) $R \ \&\& \ ! (j < 10) \Rightarrow R \ \&\& \ (j = 10) \Rightarrow$ (letti 10 valori e sommati in sum) = POST

Esempio 3: leggere e sommare fino ad una sentinella -1

//PRE=(cin contiene -1)

int x, sum=0; bool sent=false;

while (!sent) // **R**

{

 cin >>x;

 if(x != -1)

 sum=sum+x;

 else sent=true;

}

//POST= (letti i valori da cin fino a -1 e sum è la somma dei valori prima del -1)

invariante R

(per $n \geq 0$ letti n valori)

&&

(! sent \Rightarrow tutti n valori sono diversi da -1 e sum è la loro somma)

&&

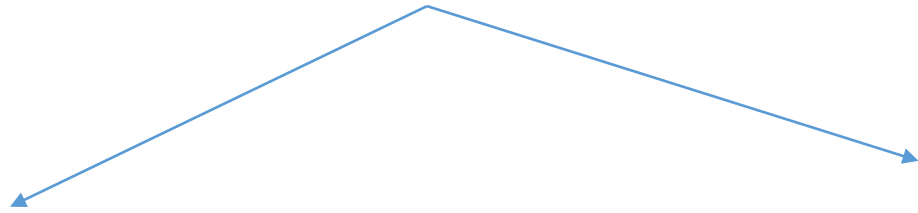
(sent \Rightarrow l' n -esimo valore è -1 e i precedenti no e sum è la loro somma)

correttezza

1) Condizione iniziale: come prima

2) invarianza: nel corpo c'è un condizionale, 2 strade :

$R \ \&\& \ !sent \Rightarrow$ letti n valori tutti diversi da -1 e sum è la somma,
nuova lettura x



$x = -1$

$sent = true$

vale R

$x \neq -1$

$sum = sum + x$

vale R

3) terminazione

$R \ \&\& \ ! \ (!sent) = R \ \&\& \ sent$

(sent \Rightarrow l'n-esimo valore è -1 e i precedenti no e sum è la somma dei precedenti)

\Rightarrow POST

Esempio 3 bis : sentinella -1-1

//PRE=(cin contiene -1-1)

int x, sum=0; bool sent1=false, sent2=false;

while (!sent2) **// R**

{

 cin >>x;

 if(x != -1)

 {sum=sum+x; if (sent1) {sum=sum-1; sent1=false;} }

 else if (sent1) sent2=true;

 else sent1=true;

}

//POST= (letti i valori da cin fino a -1-1 e sum è la somma dei valori prima del -1-1)

invariante R

(per $n \geq 0$ letti n valori)

&&

(!sent1 && !sent2 \Rightarrow tra gli n valori letti non c'è -1-1, l'ultimo non è -1 e sum è la somma degli n valori letti)

&&

(sent 1 && !sent2 \Rightarrow non 'è -1-1 tra gli n valori letti e l' n -esimo valore è -1 e sum è la somma dei primi $n-1$ valori)

&&

(sent1 && sent2 \Rightarrow gli ultimi 2 valori sono -1-1 e non c'è -1-1 che precede quello trovato e sum è la somma dei primi $n-2$ valori)