Laboratorio de Programación

Programación de Ciclos con Invariantes

Algoritmos y Estructuras de Datos I

Departamento de Computación, FCEyN, Universidad de Buenos Aires.

Repaso - Ciclos

Sintaxis de un ciclo:

```
while(B) {
  // cuerpo del ciclo
}
```

- El ciclo se repite continuamente mientras la guarda B se cumpla. Cada repetición es una iteración.
- El ciclo termina cuando no se cumpla la guarda.
- Al salir, en caso de que el ciclo terminara, el estado resultante es el mismo que el del final de la última iteración.

Repaso - Teorema del Invariante

Sea P_c la precondición del ciclo, Q_c la postcondición, B la guarda e I un invariante del ciclo. Si se cumple:

1. $P_C \Rightarrow \text{I}$,
2. $\{I \land B\}$ `cuerpo del ciclo` $\{I\}$,
3. $I \land \neg B \Rightarrow Q_C$,

entonces el ciclo es **parcialmente** correcto (si termina, termina en Q_c).

```
// Vale I
while(B){
  // Vale I && B
  Cuerpo del ciclo
  // Vale I
}
// Vale Qc
```

Repaso: Teorema de corrección de un ciclo

- **Teorema.** Sean un predicado I y una función $fv : \mathbb{V} \to \mathbb{Z}$ (donde $\mathbb{V}$ es el producto cartesiano de los dominios de las variables del programa), y supongamos que $I \Rightarrow \mathrm{def}(B)$. Si

  1. $P_C \Rightarrow I$,
  2. $\{I \land B\}\ S\ \{I\}$,
  3. $I \land \neg B \Rightarrow Q_C$,
  4. $\{I \land B \land v_0 = fv\}$ **S** $\{fv < v_0\}$,
  5. $I \land fv \leq 0 \Rightarrow \neg B$,

  ... entonces la siguiente tripla de Hoare es válida:

  $\{P_C\}$ **while B do S endwhile** $\{Q_C\}$

Ejemplo

```
bool hayMayorACero(vector<int> v) {
  bool encontre = false;
  int i = 0;
  int n = v.size();
  while(i<n) {
    encontre = encontre || v[i] > 0;
    i = i+1;
  }
  return encontre;
}
```

- Sea v = {-1,-2,3,-3,4}

Principio de iteración

Iteración	i	v[i]	encontre
1	0	-1	**false**
2	1	-2	**false**
3	2	3	**false**
4	3	-3	**true**
5	4	4	**true**

Final de iteración

Iteración	i	v[i]	encontre
1	1	-2	**false**
2	2	3	**false**
3	3	-3	**true**
4	4	4	**true**
5	5	$\perp$	**true**

Ejemplo

Principio de iteración

Iteración	i	v[i]	encontre
1	0	-1	**false**
2	1	-2	**false**
3	2	3	**false**
4	3	-3	**true**
5	4	4	**true**

Final de iteración

Iteración	i	v[i]	encontre
1	1	-2	**false**
2	2	3	**false**
3	3	-3	**true**
4	4	4	**true**
5	5	$\perp$	**true**

Es un invariante?

- $I \equiv i \leq n$?
- $I \equiv i \leq n \wedge (encontre = \textbf{true} \vee encontre = \textbf{false})$?
- $I \equiv 0 \leq i \leq n \wedge_L encontre = \textbf{true} \leftrightarrow (\exists k : \mathbb{Z})\ 0 \leq k \leq i \wedge_L v[k] > 0$?
- $I \equiv 0 \leq i \leq n \wedge_L encontre = \textbf{true} \leftrightarrow (\exists k : \mathbb{Z})\ 0 \leq k < i \wedge_L v[k] > 0$?

No todos nos van a servir para poder demostrar la correctitud parcial del ciclo! (en particular $I \wedge \neg B \Rightarrow Q_c$)

Ejercicio

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 3 \leq i \leq n + 1 \wedge suma = \sum_{k=3}^{i-1} \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

```
int suma = 0;
int i = 3;
while(i <= n) {
  if (esPrimo(i)){
    suma = suma + i;
  }
  i++;
}
return suma;
```

Variante 1

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 3 \leq i \leq n + 2 \ \wedge \ i \ mod \ 2 = 1 \ \wedge$$
$$suma = \sum_{k=3}^{i-2} \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

```
int suma = 0;
int i = 3;
while(i <= n) {
  if (esPrimo(i)){
    suma = suma + i;
  }
  i = i + 2;
}
return suma;
```

Variante 2

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 1 \leq i \leq n \ \wedge \ i \ mod \ 2 = 1 \ \wedge$$

$$suma = \sum_{k=i+2}^{n} \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

```
int suma = 0;
int i = n;
if(i % 2 == 0 )
 i--;
while(i > 2) {
 if (esPrimo(i)){
   suma = suma + i;
 }
 i -= 2;
}
return suma;
```