

Demostraciones de corrección: Teorema del invariante

Algoritmos y Estructuras de Datos I

Guía 5. Ejercicio 2

Dadas la especificación y la implementación del problema `sumarParesHastaN`, escribir la precondition y la postcondición del ciclo, y demostrar formalmente su corrección.

```
proc sumarParesHastaN (in n:  $\mathbb{Z}$ , out result:  $\mathbb{Z}$ ) {
  Pre { $n \geq 0$ }
  Post { $result = \sum_{j=0}^{n-1} (\text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi})$ }
}
```

```
result := 0;
i := 0;
while (i < n) do
  result := result + i;
  i := i + 2
endwhile
```

$$I \equiv 0 \leq i \leq n+1 \wedge i \bmod 2 = 0 \wedge result = \sum_{j=0}^{i-1} (\text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi})$$

Guía 5. Ejercicio 5

Considere la siguiente especificación de la función `sumarPosicionesImpares`.

```
proc sumarPosicionesImpares (in s:  $seq(\mathbb{Z})$ , out result:  $\mathbb{Z}$ ) {
  Pre {true}
  Post { $result = \sum_{i=0}^{|s|-1} (\text{if } i \bmod 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})$ }
}
```

- Implementar un programa en SmallLang para resolver este problema, que incluya exactamente un ciclo con el siguiente invariante:

$$I \equiv 0 \leq j \leq |s| \wedge_L result = \sum_{i=0}^{j-1} (\text{if } i \bmod 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})$$
- Demostrar formalmente la corrección del ciclo propuesto.

Guía 5. Ejercicio 10

Sea el siguiente ciclo con su correspondiente precondition y postcondición:

```
while (i >= length(s) / 2) do
  suma := suma + s[length(s)-1-i];
  i := i - 1
endwhile
```

$$P_c : \{ |s| \bmod 2 = 0 \wedge i = |s| - 1 \wedge suma = 0 \}$$

$$Q_c : \{ |s| \bmod 2 = 0 \wedge i = |s|/2 - 1 \wedge_L suma = \sum_{j=0}^{|s|/2-1} s[j] \}$$

- Especificar un invariante de ciclo que permita demostrar que el ciclo cumple la postcondición.
- Especificar una función variante que permita demostrar que el ciclo termina.
- Demostrar formalmente la corrección y terminación del ciclo usando el Teorema del invariante.

Guía 5. Ejercicio 12

Demostrar que el siguiente programa es **correcto** respecto a la especificación dada.

```
proc indice (in s: seq( $\mathbb{Z}$ ), in e:  $\mathbb{Z}$ , out r:  $\mathbb{Z}$ ) {  
  Pre {True}  
  Post { $r = -1 \rightarrow (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \neq e) \wedge r \neq -1 \rightarrow (0 \leq$   
     $r < |s| \wedge_L s[r] = i)$ }  
}  
  
i := s.size() - 1;  
j := -1;  
while (i >= 0) do  
  if (s[i] = e) then  
    j := i  
  else  
    skip  
  endif;  
  i := i - 1  
endwhile;  
r := j;
```