Demostraciones de corrección

Algoritmos y Estructuras de Datos I

Ejercicio 1

Indicar como se debería completar la precondición para poder demostrar corrección en cada caso:

```
proc transformarEnPar (inout n: ℤ) {
  Pre {n = N0 ∧ ??}
  Post {esPar(n) ∧ n > N0}
}
```

Programa 1	Programa 2
S1: `n := 2*n`	**S2:** `n := n + 1`

Ejercicio 2

Demostrar que el programa dado es correcto respecto a la siguiente especificación:

```
proc restaYpromedio (inout a: ℤ, inout b: ℤ) {
  Pre {a = A0 ∧ b = B0}
  Post {a = A0 − B0 ∧ b = (A0 + B0)/2}
}
```

S1: `aux := a`
S2: `a := a-b`
S3: `b := (aux + b) / 2`

Ejercicio 3

Demostrar que el programa dado es correcto respecto a la siguiente especificación:

```
proc swap (inout a: ℤ, inout b: ℤ) {
  Pre {a = A0 ∧ b = B0 ∧ a ≠ 0 ∧ b ≠ 0}
  Post {a = B0 ∧ b = A0}
}
```

S1: `a := a*b`
S2: `b := a/b`
S3: `a := a/b`

Ejercicio 4

Completar la especificación de diferenciaPositiva de forma tal que se pueda demostrar que ambos programas son correctos.

`proc diferenciaPositiva` (in a: $\mathbb{Z}$, in b: $\mathbb{Z}$, out res: $\mathbb{Z}$) {
  `Pre` $\{??\}$
  `Post` $\{res = |a - b|\}$
}

- Programa 1
  S1: `res := a-b`
- Programa 2
  S2: `if (a > b) then res := a - b else res := b - a endif`

Ejercicio 5

Completar la especificación de sumarTodos de forma tal que se pueda demostrar que el siguiente programa es correcto.

`proc sumarTodos` (in s: $seq\langle\mathbb{Z}\rangle$, in n: $\mathbb{Z}$, inout suma: $\mathbb{Z}$) {
  `Pre` $\{...\}$
  `Post` $\{suma = \sum_{i=0}^{|s|-1} s[i]\}$
}

S: `suma := suma + s[n-1]`

Ejercicio 6

Completar la especificación de todoPositivos de forma tal que se pueda demostrar que el siguiente programa es correcto.

`proc todoPositivos` (inout s: $seq\langle\mathbb{Z}\rangle$) {
  `Post` $\{|s| = |S_0| \wedge_L (\forall i : \mathbb{Z}) 0 \leq i < |s| \rightarrow_L s[i] > 0\}$
}

S: `s[0] := -1 * s[0]`

Ejercicio 7

Especificación

`proc sumar` (in s: $seq\langle\mathbb{Z}\rangle$, out result: $\mathbb{Z}$) {
  `Pre` $\{\text{true}\}$
  `Post` $\{result = \sum_{j=0}^{|s|-1} s[j]\}$
}

Implementación en SmallLang

```
result := 0;
i := 0;
while (i < s.size()) do
  result := result + s[i];
  i := i + 1;
endwhile
```

Invariante de Ciclo

$$I \equiv 0 \leq i \leq |s| \wedge_L result = \sum_{j=0}^{i-1} s[j]$$

a) Escribir la precondición y la postcondición del ciclo.
b) ¿Qué punto falla en la demostración de corrección si el primer término del invariante se reemplaza por "$0 \leq i < |s|$"?
c) ¿Qué punto falla en la demostración de corrección si el límite superior de la sumatoria (que es "$i - 1$") se reemplaza por "i"?
d) ¿Qué punto falla en la demostración de corrección si se invierte el orden de las dos instrucciones del cuerpo del ciclo?
e) Demostrar formalmente la corrección parcial del ciclo, usando el teorema del invariante.