

Programación Concurrente en Java

Práctico 1: Verificación de Programas Secuenciales

J. Blanco, N. Wolovick

1. Expresar en palabras las siguientes aserciones sobre los estados.

- | | |
|--|--|
| <p>a) $s \geq 0$</p> <p>b) $r = 0 \vee w = 0$</p> <p>c) $(r > 0 \Rightarrow w = 0) \wedge (r = 0 \Leftarrow w > 0)$</p> <p>d) $done \equiv x > 0$</p> <p>e) $done \Rightarrow x > 0$</p> <p>j) $\langle \forall i : 0 \leq i < N : x.i \Rightarrow \langle \forall j : 0 \leq j < N : e.j \rangle \rangle$</p> <p>k) $n = \langle \#i : 0 \leq i < N : e.i \rangle$</p> <p>l) $n = \langle \sum i : 0 \leq i < N \wedge a.i : 2^i \rangle$</p> <p>m) $\langle \forall i : 0 \leq i < N : \langle \exists j : 0 \leq j < N \wedge i \neq j : a.i = a.j \rangle \rangle$</p> <p>n) $0 \leq x \wedge (found \Rightarrow (0 < x \wedge f.x = 0) \vee (y \leq 0 \wedge f.y = 0)) \wedge (\neg found \wedge 0 < x \Rightarrow f.x \neq 0)$</p> <p>$\tilde{n}$) $s_0 \neq 0 \Rightarrow (B_0 \wedge 0 < b_0 \wedge I)$</p> <p>o) $m \neq 0 \Rightarrow ((B_0 \Rightarrow 0 = b_0) \wedge (B_1 \Rightarrow 0 = b_1) \wedge I)$</p> | <p>f) $\langle \forall i : 0 \leq i < N : a.i = i^2 \rangle$</p> <p>g) $\langle \exists i : 0 \leq i < N : b.i = 0 \rangle$</p> <p>h) $\langle \forall i : 0 \leq i < N : a.i = a.(N - (i + 1)) \rangle$</p> <p>i) $\langle \forall i, j : 0 \leq i, j < N \wedge i + j = N - 1 : a.i = a.j \rangle$</p> |
|--|--|

2. Expresar lógica de primer orden (FOL) los siguientes predicados sobre los estados.

- a) El arreglo a tiene al menos un elemento que es 0 o 1.
- b) El arreglo cvw tiene a cero todos los elementos $[0, i)$.
- c) El arreglo a tiene al menos un máximo que es igual a M .
- d) Si la bandera $done$ está prendida, entonces n tiene la suma de los elementos de a .
- e) La bandera $found$ indica exactamente si en el índice i está el valor M .
- f) Los elementos de arreglo c están ordenados de manera creciente.
- g) Todos los elementos del arreglo b son iguales (se puede hacer con un solo cuantificador).
- h) Todos los elementos del arreglo b son distintos.
- i) El arreglo a es un diente de sierra, es decir crece desde el principio hasta un máximo y luego decrece.
- j) El arreglo a es la Bandera Nacional Holandesa.

3. ¿Cuáles de las siguientes fórmulas de la lógica de Hoare son verdaderas en el sentido de la *corrección parcial*? Demostrar con la lógica o dar un contraejemplo en el modelo de ejecución.

- | | |
|---|--|
| <p>a) $\{true\} x := 100 \{true\}$</p> <p>b) $\{true\} x := 100 \{x = 100\}$</p> <p>c) $\{x = 50\} x := 100 \{x = 50\}$</p> <p>d) $\{y = 50\} x := 100 \{y = 50\}$</p> <p>e) $\{true\} x := 100 \{false\}$</p> <p>f) $\{false\} x := 100 \{x = 50\}$</p> <p>g) $\{y = 14\} \mathbf{skip} \{y = 14\}$</p> <p>h) $\{y = 12 \wedge x \geq y\} \mathbf{skip} \{x > 12\}$</p> | <p>i) $\{x = X \wedge y = Y\}$
 $z := x; x := y; y := z$
 $\{x = Y \wedge y = X\}$</p> <p>j) $\{true\} \mathbf{if} \mathbf{false} \rightarrow S \mathbf{fi} \{Q\}$</p> <p>k) $\{true\}$
 $\mathbf{if} x \leq y \rightarrow \mathbf{skip}$
 $\square y \leq x \rightarrow x, y := y, x$
 $\mathbf{fi} \{x \leq y\}$</p> <p>l) $\{x = X \wedge y = Y\}$
 $\mathbf{if} x \leq y \rightarrow \{x \leq y\} \mathbf{skip}$
 $\square y \leq x \rightarrow \{y \leq x\} x, y := y, x$
 $\mathbf{fi} \{X \leq Y\}$</p> |
|---|--|

4. Demostrar la corrección parcial de los siguientes *programas con ciclos* utilizando los invariantes dados.

<p>a) $\{Pre : true\}$ $r, q := x, 0$ $\{Inv : x = y \times q + r\}$ do $y \leq r \rightarrow r := r - y;$ $q := q + 1$ od $\{Post : x = y \times q + r \wedge r < y\}$</p>	<p>b) $\{Pre : 0 \leq X \wedge 0 \leq Y \wedge x = X \wedge y = Y\}$ $z := 1$ $\{Inv : 0 \leq y \wedge z \times x^y = X^Y\}$ do $0 < y \wedge even(y) \rightarrow \{Inv \wedge 0 < y \wedge even(y)\}$ $y, x := y \text{ div } 2, x \times x$ $\{Inv\}$ $\square 0 < y \wedge odd(y) \rightarrow \{Inv \wedge 0 < y \wedge odd(y)\}$ $y, z := y - 1, z \times x$ $\{Inv\}$ od $\{Post : z = X^Y\}$</p>
--	--

5. Demostrar que si el programa anotado de la izquierda es correcto, entonces también lo es el de la derecha. Luego, toda demostración de un *programa secuencial no determinístico* sirve para demostrar alguna de las variantes determinísticas del mismo.

$\{P\} \text{ if } B_0 \rightarrow S_0$ $\square B_1 \rightarrow S_1$ fi $\{Q\}$	$\{P\} \text{ if } B_0 \wedge \neg B_1 \rightarrow S_0$ $\square B_1 \rightarrow S_1$ fi $\{Q\}$
---	---

6. Escriba el programa más corto que en *corrección parcial* establece para $N > 0$ la postcondición que indica que el arreglo b tiene la secuencia de los cuadrados $Post : \langle \forall i : 0 \leq i < N : b.i = i^2 \rangle$.

7. La sucesión de Fibonacci está dada por las siguientes ecuaciones recursivas.

$$fib.0 = 0 \quad fib.1 = 1 \quad fib.n = fib.(n-1) + fib.(n-2), \quad 2 \leq n$$

Definiendo primero un *invariante de ciclo*, probar la fórmula de corrección parcial $\{0 \leq n\}S\{x = fib.n\}$, donde:

$S \doteq x := 0; y := 1; count := n;$
do $0 < count \rightarrow h := y;$
 $y := x + y;$
 $x := h;$
 $count := count - 1$
od