Programación Concurrente
Práctico 4: Manejando la Complejidad

J. Blanco, N. Wolovick

5 de mayo de 2003

1. Demuestre la corrección de la postcondicion en el siguiente multiprograma[1].

Pre: $i = 0 \wedge j = 0$	
A: **do** $i \neq 47 \rightarrow$ $i := i + 1$ **od**	B: **do** $j \neq 47 \rightarrow$ $x.j := i;$ $j := j + 1$ **od**
Post: $\langle \forall k : 1 \leq k < 47 : x.(k-1) \leq x.k \rangle$	

2. Encuentre enteros C y D para que en el siguiente multiprograma el predicado $x \leq y + C \wedge y \leq x + D$ resulte invariante.

Pre: $x = 0 \wedge y = 0$	
A: $*[x := y;$ $x := x + 1$ $]$	B: $*[y := x;$ $y := y + 1$ $]$

3. Muestre la invariancia de $x + y \leq z$ en el siguiente multiprograma[2]. ¿Qué pasa si el multiprograma contiene varias instancias de A y B?

Pre: $x = 0 \wedge y = 0 \wedge z = 0$	
A: $*[y := y - 1;$ $x := x + 1$ $]$	B: $*[z := z + 1;$ $y := y + 1$ $]$

4. Demuestre la siguiente versión del teorema "general" de topología[3].

Pre: $x = 0 \wedge y = 0$
Comp.i: $*[\ y := y + 1;$ $\{x < y\}$ $x := x + 1$ $x := x - 1$ $\{x < y\}$ $y := y - 1$ $]$
Inv: $0 \leq x \wedge x \leq y$

[1] Utilize que $x.(-1) \leq 0$.
[2] Hint: se puede ver como el 1er lema de topología
[3] Hint: se puede demostrar utilizando 3 veces el 1er lema de topología.

5. Demuestre el siguiente multiprograma donde los ángulos $\langle\ \rangle$ representan la atomicidad y $\downarrow$ es el operador binario de mínimo.

Pre: $true$	
A: **if** $\langle x \leq y \rangle \rightarrow \langle m := x \rangle$ $\Box\ \langle y \leq x \rangle \rightarrow \langle m := y \rangle$ **fi**	B: $\langle m := x \rangle$ **if** $\langle m \leq y \rangle \rightarrow$ **skip** $\Box\ \langle y \leq m \rangle \rightarrow \langle m := y \rangle$ **fi**
Post: $m = x \downarrow y$	

6. Muestre la exclusión mutua entre $cs.A$ y $cs.B$ en el siguiente multiprograma[4], donde sólo suponemos átomica la evaluación de las guardas y las asignaciones[5].

   Piense si además de cumplir con la propiedad de seguridad, también tiene buenas propiedades de progreso como las indicadas en cualquier libro de texto de Sistemas Operativos.

   Investigue si es posible utilizar este algoritmo tipo *spin-lock* en arquitecturas x86.

Pre: $\neg x \wedge \neg y$	
A: $*$ [**do** $\neg x \rightarrow x := \neg y$ **od** ; $cs.A$; $x := false$]	B: $*$ [**do** $\neg y \rightarrow y := \neg x$ **od** ; $cs.B$; $y := false$]

[4] Deberá pensar en como expresar la noción de exclusión mutua.
[5] Es decir que el **do**...**od** puede ser interrumpido luego de haber evaluado la guarda.