Programación Concurrente
Práctico 5: Sincronización y Progreso

J. Blanco, N. Wolovick

5 de mayo de 2003

1. Para cada uno de los siguientes programas, **analice** sus propiedades de progreso[1].

   a) $\{\neg b\}\ b := true \parallel [b]; b := false\ \{\neg b\}$

   b) $\{\neg b\}\ b := true \parallel b := false; [b]\{b\}$

   c) $\{\neg b\}\ b := true; b := false \parallel [b]\{?\}$

   d) $\{\neg b \wedge \neg c\}\ b := true; [c] \parallel\ c := true; [b]\{b \wedge c\}$

   e) $\{\neg b\}\ *[b := true; b := false] \parallel [b]$

2. Demuestre progreso en los siguientes multiprogramas

a)

<table>
<tr><td colspan="2">Pre: true</td></tr>
<tr><td>A: $v := q$
; if $v = p \rightarrow$ skip fi
; $v := q$</td><td>B: $v := p$
; if $v = q \rightarrow$ skip fi
; $v := p$</td></tr>
<tr><td colspan="2">Post: ?</td></tr>
</table>

b)

<table>
<tr><td colspan="2">Pre: $x = y$</td></tr>
<tr><td>A: $*[$ if $x \neq y \rightarrow$ skip fi
; $x := x + 1$
]</td><td>B: $*[$ if $x = y \rightarrow$ skip fi
; $y := y + 1$
]</td></tr>
<tr><td colspan="2">Inv: ?</td></tr>
</table>

3. En [AO91] plantea la siguiente regla para demostrar *deadlock freedom* (individual y total) para programas paralelos con variables compartidas y sincronización.

   Considere un programa paralelo $S \equiv [S_1 \parallel \ldots \parallel S_n]$.

   a) Una tupla $(R_1, \ldots, R_n)$ de sentecias se denomina *deadlock potencial de S* si se dan las siguientes condiciones

   - Para todo $i \in \{1, \ldots, n\}$, R_i es una sentencia guardada en la componente S_i o el símbolo E que denota la sentencia vacía y representa la terminación de S_i,
   - para algún $i \in \{1, \ldots, n\}$, R_i es una sentencia guardada.

   b) Dada un esquema de prueba estandard de corrección parcial para no interferencia $\{p_i\}S_i^*\{q_i\}$, $i \in \{1, \ldots, n\}$, con cada deadlock potencial de S asociamos una tupla $(r_1, \ldots, r_n)$ de aserciones, poniendo para $i \in \{1, \ldots, n\}$:

   - $r_i \equiv pre(R_i) \wedge \neg B$ si $R_i \equiv$ **if** $B \rightarrow S$ **fi**

[1] Suponga atomicidad en todas las asignaciones y sentencias guardadas. La composición paralela tiene menor precedencia que la secuencial. La notación $[b]$ es el skip guardado **if** $b \rightarrow$ **skip fi**.

- $r_i \equiv q_i$ si $R_i \equiv E$.

  Si podemos demostrar $\neg \bigwedge_{i=1}^{n} r_i$ para <u>todas</u> las tuplas de aserciones $(r_1, \ldots, r_n)$, ninguno de los deadlocks potenciales pueden ocurrir, estableciendo *deadlock freedom*.

Utilizando esta regla demuestre ausencia de deadlock para

a) el ejercicio $(2a)$ de este práctico,

b) el algoritmo de *Concurrent Vector Writing* derivado en clase,

c) la resolución del ejercicio 1 del primer práctico.

$A:\ \ y := false$ $;\ x := true$ $;\ \mathbf{if}\ y \rightarrow \mathbf{skip\ fi}$ $;\ x := true$	$B:\ \ x := false$ $;\ y := true$ $;\ \mathbf{if}\ x \rightarrow \mathbf{skip\ fi}$ $;\ y := true$

4. Considere un multiprograma con 1 componente de tipo A y 47 componentes de tipo B.

Pre: $s = 0$	
A: $*[s := s + 1]$	B: $*[\ \mathbf{if}\ K \leq s \rightarrow \mathbf{skip\ fi}$ $;\ s := s - 1$ $]$

Derive para que valores de K se cumple el invariante del sistema $0 \leq s$.

5. Demuestre el progreso de las N componentes del multiprograma

Pre: $\langle \forall q :: x.q = 0 \rangle$
Comp.p: $*[\ \mathbf{if}\ \langle \forall q :: x.p \leq x.q \rangle \rightarrow \mathbf{skip\ fi}$ $;\ x.p := x.p + 1$ $]$

6. *"No siempre debilitar las guardas de un* ***if*** *o fortalecer las de un* ***do*** *ayuda al progreso".*

a) Muestre la terminación del siguiente programa:

Pre: *true*	
A: $\mathbf{do}\ x \vee y \rightarrow x := false\ \mathbf{od}$	B: $\mathbf{do}\ x \vee y \rightarrow y := false\ \mathbf{od}$

b) Pero la terminación no se garantiza si cambiamos a ...

Pre: *true*	
A: $\mathbf{do}\ y \rightarrow x := false\ \mathbf{od}$	B: $\mathbf{do}\ x \rightarrow y := false\ \mathbf{od}$

c) Analice la situación para esta última modificación.

Pre: *true*	
A: $\mathbf{do}\ y \rightarrow x := false\ \mathbf{od};$ $x := false$	B: $\mathbf{do}\ x \rightarrow y := false\ \mathbf{od};$ $y := false$