

# Problema: Coches y peatones

( SIN TENER EN CUENTA INDICACIÓN )

↙ coche que va hacia el norte.

Coche Norte

loop

secc no crítica

monitor. pasar NC)

operacions de pasar

monitor. llegar NC)

↙ entrar en el puente para ir al otro lado

↘ llegar al otro lado, esto es, salir del puente.

Coche Sur

loop

secc no crítica

monitor. pasar SC)

operacions de pasar

monitor. llegar NC)

Peaton

loop

secc no crítica

monitor. pasar P

operacions de pasar

monitor. llegar PC)

Monitor

↙ nº de coches hacia el norte en el puente (pasando)

nCN : int = 0

nCS : int = 0 ↙ nº de coches hacia el sur en el puente

nP : int = 0 ↙ nº de peatones pasando por el puente

puede pasar Sur : VC = True

puede pasar Norte : VC = True

puede pasar Peaton : VC = True

$$\varphi_{INV} = \left( \begin{array}{l} nCN \geq 0 \wedge nCS \geq 0 \wedge nP \geq 0 \wedge \\ \wedge nP > 0 \rightarrow (nCS = 0 \wedge nCN = 0) \wedge \\ \wedge nCS > 0 \rightarrow (nP = 0 \wedge nCN = 0) \wedge \\ \wedge nCN > 0 \rightarrow (nP = 0 \wedge nCS = 0) \end{array} \right)$$

- $\text{passer } N ()$   
 $\Delta \text{EWU} \&$   
 $\text{puede pasar } N. \text{ wait } (nCS == 0 \wedge nP == 0)$   
 $nCN = nCN + 1$   
 $\Delta \text{EWU} \&$  c. se conserva porque la VC hace que  
 operemos hasta que  $nCS$  y  $nP$  sean 0

- $\text{llegar } N ()$   
 $\Delta \text{EWU} \& \wedge nCN > 0 \&$   
 $nCN = nCN - 1$   
 $\text{puede pasar } S. \text{ notify } ()$   
 $\text{puede pasar } P. \text{ notify } ()$   
 $\Delta \text{EWU} \&$

- $\text{passer } S ()$   
 $\Delta \text{EWU} \&$   
 $\text{puede pasar } S. \text{ wait } (nCN == 0 \wedge nP == 0)$   
 $nCS = nCS + 1$   
 $\Delta \text{EWU} \&$

- $\text{llegar } S ()$   
 $\Delta \text{EWU} \& \wedge nCS > 0 \&$   
 $nCS = nCS - 1$   
 $\text{puede pasar } P. \text{ notify } ()$   
 $\text{puede pasar } N. \text{ notify } ()$   
 $\Delta \text{EWU} \&$

- $\text{passer } P ()$   
 $\Delta \text{EWU} \& \& \text{ puede pasar } P. \text{ wait } (nCN == 0 \wedge nCS == 0)$   
 $nP = nP + 1$   
 $\Delta \text{EWU} \&$

- $\text{llegar } P ()$   
 $\Delta \text{EWU} \& \wedge nP > 0 \&$   
 $nP = nP - 1$   
 $\text{puede pasar } S. \text{ notify } ()$   
 $\text{puede pasar } N. \text{ notify } ()$   
 $\Delta \text{EWU} \&$

- Demostración de que el puente es seguro:

Si un coche o un peatón quieren entrar, tienen que esperar antes a que el puente esté vacío.

- No hay dead locks ya que siempre que cambia el número de ocupantes del puente se notifica a todos los que pudieran estar esperando.

• Esta solución, por desgracia, tiene problemas de inicialización, pero con unos pequeños arreglos nos disponemos a solucionarlos.

- Que el invariante se conserve lo consideramos demostrado con las anotaciones en el código.



ponemos el nuevo monitor en otra hoja:

La idea del código es:

La entrada al puente se decide por turnos, que van en orden, de tal manera que después del Norte van los peatones, después el Sur, y después otra vez el Norte.

El turno cambia cuando el primer individuo del grupo que está pasando sale del puente, esto es, llega a su destino. Una vez se ha cambiado el turno se deja pasar a los que ya estaban pasando, y cuando cesan de pasar, entonces pueden empezar a pasar los del nuevo turno, y cuando el primero de ellos llegue, se volverá a cambiar el turno.



# CORRECCIONES EN EL CÓDIGO

## Monitor-keio

igual  
que antes, con  
le mismo

turno:  $int = 0$   
 $nC10: int = 0$   
 $nCS: int = 0$   
 $nP: int = 0$   
 $wCS: int = 0$   
 $wC10: int = 0$   
 $wP: int = 0$   
 $sTurnoN: VC = True$   
 $sTurnoS: VC = True$   
 $sTurnoP: VC = True$   
 $puede pasar N: VC = True$   
 $puede pasar S: VC = True$   
 $puede pasar P: VC = True$

los nombres del turno  
 serán 0 = North (no  
 hay ningún turno vigente,  
 puede entrar quien quiera  
 en el puente) ;  
 1 = s el turno de los  
 peafens := lo llamaremos  
 PED  
 2 = s el turno de los  
 coches que van al norte,  
 lo llamaremos NORTH.  
 3 = s el turno de los  
 coches que van al sur,  
 lo llamaremos SUR..

Se creó el número de coches que van hacia el sur  
 esperando a entrar al puente,  $wCN$  se le mismo  
 pero con los coches que van hacia el norte, y  
 $wP$  lo mismo con peafens

4 ~~peafens~~  $wP$  (4)

$dEUSE = 1$   $wCS \geq 0$ ,  $wCP \geq 0$ ,  $wP \geq 0$ ,  
 y  $hola$  del anterior  $EUSE$ .

paramos a definir las funciones del  
 monitor



passer SC)

q1000t

wcs = wcs + 1

if turno S. visit Cturno == SOUTH or turno == NORTH)

if turno == NULL

turno = SOUTH

no se pueden a alguien de the grupo  
alternar per si acaso  
después

prede pasar S. visit CneD == 0 or NP == 0)

wcs = wcs - 1

ncs = ncs + 1

q1000t

deleg SC)

q1000t and wcs > 0

ncs = ncs - 1

if turno == SOUTH

if wcs > 0

turno = NORTH  
S turno / 0. notifs()

elid wP > 0

turno = DEHP  
S turno P. notifs()

elid wcs == 0

turno = NULL

else  
pass

else  
pass

if turno == NORTH

prede pasar S. notifs()

elid turno == SOUTH

prede pasar S. notifs()

q1000t

7

prevP(1)  
qIndex

wP = wP + 1

if hmo == BULL or hmo == SELL

if hmo == BULL

hmo = BULL

prevP, wP, wL C n EU == 0 and CS == 0

wP = wP - 1

wP = nP + 1

return

RegeP(1)

qIndex and nP > 0

nP = nP - 1

if hmo == BULL

if ~~hmo~~ w CS > 0

hmo = SELL

if hmo == SELL

if w CS > 0

hmo = BULL

if hmo == BULL

if w CS == 0

hmo = BULL

else

prevP

else

prevP

if hmo == BULL

prevP, wL, wR(1)

if hmo == SELL

prevP, wL, wR(1)

return



- La conversión del mensaje a clave, se que sólo se puede romper al estar en WGS, WCA o WP, la cual se hace siempre después de armar, o al estar de MCS o NCN o NP, pero sta siempre se hace asumiendo que son 0.

otra manera en la que se pueden romper se es viendo los ns > 0, la cual es imposible porque antes de incrementarlos a min que los otros sean cero.

- Por los argumentos anteriores, la seguridad del prente queda garantizada

- Los hay deadlocks si que si no hay nadie en ninguna celda, no hay turnos, y nadie más eligien entre cast el turno, que sigue su curso.

Si se sucesos natural en el Turno no sta, se pasa al siguiente, y si en el siguiente tampoco hay nadie, se cancela hasta que vuelen y elegen otro cast. Si no hay más celda, el turno es NULL.

En caso de que haya gente en todas las celdas, el turno sigue su curso natural.

• Lo hay iniciación ya que:

Si un proceso quiere entrar, entonces está dormido en alguno de los puentes o está despierto;

Si está despierto y le llega pasar por el puente de su turno, ¿pasa? ¿espera? ¿que acaben los que están pasando del turno anterior o pasa, si es el primer caso está número 2 por lo tanto. finally y controlado, si se puede tomar control del puente los que hay en el puente durante el turno anterior, y no pueden entrar más.

Si está dormido en el puente de su turno, entonces, como los turnos van en orden, una vez haya pasado en proceso del grupo del orden anterior se despertará y podrá pasar.