

PRACTICA 2

JAVIER CASTELLANO SORIA

Solución sencilla para exclusión mutua:

Monitor P "número de coches sentido Norte pasando"

ncochesN : int = 0 Nwait.VC
ncochesS : int = 0 Swait.VC
nped : int = 0 Pwait.VC

INV = { ncochesN ≥ 0 ∧ ncochesS ≥ 0 ∧ nped ≥ 0
 ∧ ncochesN > 0 → ncochesS + nped = 0
 ∧ ncochesS > 0 → ncochesN + nped = 0
 ∧ nped > 0 → ncochesN + ncochesS = 0 }

wants_enter_car (direc)

```
{INV}
if direc == NORTH:
    {INV}
    Nwait.wait (ncochesS + nped == 0)
    ncochesN += 1
    ← {INV ∧ ncochesS + nped == 0}
else:
    {INV}
    Swait.wait (ncochesN + nped == 0)
    ncochesS += 1
    ← {INV ∧ ncochesN + nped == 0}
```

leaves_car (direc)

```
if direc == NORTH:
    ← {INV ∧ ncochesN > 0}
    ncochesN -= 1
    if ncochesN == 0:
        Swait.notify-all()
        Pwait.notify-all()
    {INV}
else:
    {INV}
    ncochesS -= 1
    if ncochesS == 0:
        Nwait.notify-all()
        Pwait.notify-all()
    ← {INV}
```

wants_enter_pedestrian()

{INV}

Pwait.wait($ncochesN + ncochesS == 0$)

$nped += 1 \leftarrow \{INV \wedge ncochesN + ncochesS == 0\}$

{INV}

leaves_pedestrian()

{INV $\wedge nped > 0$ }

$nped -= 1$

if $nped == 0$:

Nwait.notify-all()

Swait.notify-all()

{INV}

El invariante se cumple en la inicialización y suponiendo cierto al inicio de cada método se sigue cumpliendo también al acabar. Con esto tenemos garantizada la seguridad pues no puede darse que $ncochesN > 0 \wedge ncochesS > 0$ ó $ncochesN > 0 \wedge nped > 0$ ó $ncochesS > 0 \wedge nped > 0$, debido al invariante.

Sin embargo, hay un problema de inanición. Si por ejemplo entra un coche con dirección Norte, Sur y peatón no pueden pasar, y si ahora entra otro Norte antes de acabar el primero siguen Sur y peatón sin pasar y esta secuencia se puede continuar si siguen llegando coches Norte dejando al resto sin poder pasar.

Solución La idea para solucionar la inanición va a consistir en añadir una condición más para dar permiso para pasar (no solo pedir que no haya nadie de los otros en el puente). Para ello vamos a introducir una variable que indicará los turnos de quién puede pasar. Así, cuando alguien tiene su turno, p.e. N, y no hay en el puente ni S ni peatón podrá pasar. Mientras está N dentro del puente, podrán pasar más N's pero cuando el primero salga se cederá el turno a otro solucionando la inanición. Nótese que no necesariamente necesitas tener el turno para pasar pues si no hay nadie del resto esperando puedes pasar.

NORTH = 0
SOUTH = 1
PED = 2

Monitor P

ncoches N: int = 0

ncoches S: int = 0

nped: int = 0

turno = 0

Nwaiting: int = 0

Swaiting: int = 0

Pwaiting: int = 0

Nwait. VC

Swait. VC

Pwait. VC

"número de coches sentido N que están esperando a pasar"

INV $\equiv \{ ncochesN \geq 0 \wedge$

$ncochesS \geq 0 \wedge nped \geq 0$

$\wedge ncochesN > 0 \rightarrow ncochesS + nped = 0$

$\wedge ncochesS > 0 \rightarrow ncochesN + nped = 0$

$\wedge nped > 0 \rightarrow ncochesN + ncochesS = 0$

$\wedge 0 \leq turno \leq 2 \wedge Nwaiting \geq 0$

$\wedge Swaiting \geq 0 \wedge Pwaiting \geq 0 \}$

waiter_enter_car (direc):

if direc == NORTH:
 Nwaiting += 1

```

    } Nwait.wait( ncochesS + nped == 0 ^ (turn == NORTH
                  v Swaiting + Pwaiting == 0))
    }
    Nwaiting -= 1
    ncochesN += 1
else :

```

#ANÁLOGO

leaves_car(direc)

```

if direc == NORTH :
    ncochesN -= 1
    if Swaiting > 0 :
        turn = SOUTH
    else
        turn = PED
    if ncochesN == 0
        Swait.notify_all()
        Pwait.notify_all()
else :

```

```

    ncochesS -= 1
    if Pwaiting > 0 :
        turn = PED
    else :
        turn = NORTH

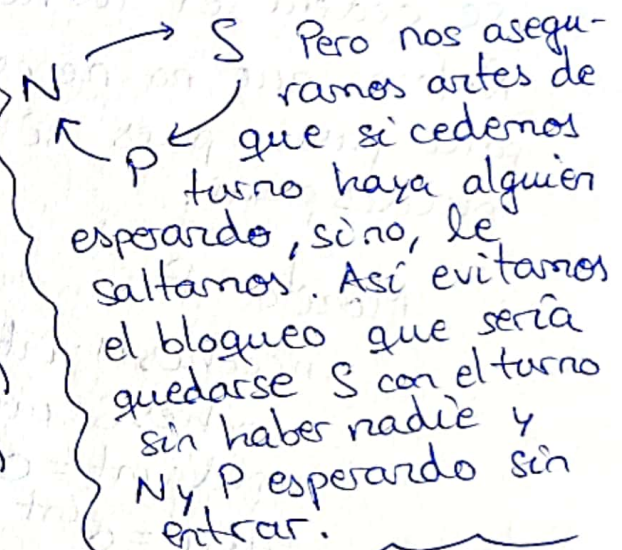
```

```

    if ncochesS == 0
        Pwait.notify_all()
        Nwait.notify_all()

```

la idea es ceder el turno según el esquema:



wards_enter_pedestrian()

Pwaiting += 1

Pwait.wait(ncochesN + ncochesS == 0 \wedge (turn == PED
 \vee Nwaiting + Swaiting == 0))

Pwaiting -= 1

nped += 1

leaves_pedestrian()

nped -= 1

if Nwaiting > 0:

turn = NORTH

else:

turn = SOUTH

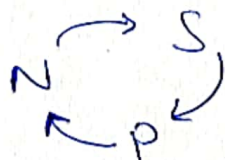
if nped == 0

Nwait.notify_all()

Swait.notify_all()

La exclusión mutua se garantiza por el invariante como en la solución anterior.

No hay problemas de inanición pues si alguien quiere entrar, p.e. N, si es su turno solo debe esperar a que salga del puente los de S o los peatones. Si no es su turno basta esperar a que le llegue según el esquema:



o incluso no hace falta si ya no hay en S o peatones esperando.

Veamos la ausencia de deadlocks. Si el sistema se bloquea llegaríamos a que no hay nadie pasando por el puente. Supongamos que turn vale SOUTH, entonces como está bloqueado no hay nadie esperando de S, si no, al tener turno y estar vacío el puente pasarían los de S y no estaría bloqueado. Además, para que esté bloqueado, debería darse que $Nwaiting > 0$ y $Pwaiting > 0$. Si alguno fuese cero entonces el otro podría pasar y no estaría bloqueado. Sin embargo, esto último no puede ocurrir por el "if" que ponemos a la hora de ceder el turno: "Si al que le toca el turno no tiene gente esperando se lo paso al siguiente del esquema anterior".

Por tanto, al invariante añadiremos: $\wedge Swaiting == 0$

$$\neg (turn == SOUTH \wedge Nwaiting > 0 \wedge Pwaiting > 0) \wedge$$

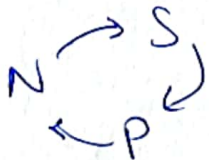
$$\neg (turn == NORTH \wedge Swaiting > 0 \wedge Pwaiting > 0) \wedge$$

$$\neg (turn == PED \wedge Nwaiting > 0 \wedge Swaiting > 0) \wedge Nwaiting = 0$$

Con esto tendríamos la ausencia de deadlock.

$\wedge Pwaiting == 0$

NOTA: El turno se va cediendo según la preferencia;



Pero también podríamos ceder a aquel en el cual están esperando más aunque podría darse inanición si se queda uno esperando con una cantidad baja y hay una gran afluencia en el resto.