

Irma Alonso Sánchez

monitor

mutex.lock()

numpedpas: int=0

numcarspas: int=0

numcarmpas: int=0

no-car-s-ped.vc

no-car-n-ped.vc

no-car-s-n.vc

d → direction { d=1 → SUR
d=0 → NORTE

are-no-car-s-ped()

(numcarspas==0 ∧ numpedpas==0)

are-no-car-n-ped()

(numcarmpas==0 ∧ numpedpas==0)

are-no-car-s-n()

(numcarspas==0 ∧ numcarmpas==0)

wants-enter-car(d)

mutex.wait()

if d==1:

no-car-n-ped.wait-for(are-no-car-n-ped)

numcarspas += 1

else:

no-car-s-ped.wait-for(are-no-car-s-ped)

numcarmpas += 1

mutex.signal()

leaves-car(d)

mutex.wait()

if d==1:

numcarspas -= 1

if numcarspas==0:

no-car-s-ped.notify-all()

no-car-n-ped.notify-all()

else:

numcarmpas -= 1

if numcarmpas==0:

no-car-n-ped.notify-all()

no-car-s-ped.notify-all()

mutex.signal()

wants-enter-pedestrian()

mutex.wait()

no-car-s-n.wait-for(are-no-car-s-n)

numpedpas += 1

mutex.signal()

leaves-pedestrian()

mutex.wait()

numpedpas -= 1

if numpedpas==0:

no-car-n-ped.notify-all()

no-car-s-ped.notify-all()

mutex.signal()

El invariante es:

INV ≡ { numpedpas ≥ 0

numcarspas ≥ 0

numcarmpas ≥ 0

numpedpas > 0 → numcarspas = 0 ∧ numcarmpas = 0

numcarspas > 0 → numcarmpas = 0 ∧ numpedpas = 0

numcarmpas > 0 → numcarspas = 0 ∧ numpedpas = 0 }

Con esta versión se garantiza la seguridad ya que:

- Un peatón sólo puede entrar si se cumple que no hay coches del norte ni del sur pasando.
- Un coche del norte sólo puede entrar si no hay coches del sur ni peatones cruzando.
- Un coche del sur sólo puede pasar si no hay coches del norte ni peatones cruzando.

No hay deadlocks ya que cuando sale un peatón, coche del norte o coche del sur si quedan o elementos pasando hacemos un notify a los otros dos que puede ser que estuvieran esperando, de manera que no se quedan todos esperando en ningún momento.

Sin embargo, con esta solución si hay inanición.

Por ejemplo, si están cruzando peatones y llegan coches se ponen a esperar, pero mientras sigan llegando peatones entrarán en el puente y los coches no podrán entrar nunca (por muchos que haya esperando).

Análogamente, con los coches del sur y del norte, ocurre lo mismo.

Para evitar esto añadimos al monitor tres variables que llevan la cuenta del número de coches del sur, del norte y de peatones esperando, que serán carswait, carnwait, pedwait respectivamente.

Además, añadimos una variable turn que indica el turno de los que tienen que pasar, de manera que vale:

- 0 \Rightarrow cuando pueden pasar los coches del sur.
- 1 \Rightarrow cuando pueden pasar los coches del norte.
- 2 \Rightarrow cuando pueden pasar los peatones.
- -1 \Rightarrow cuando pueden pasar todos (turno neutro).

El turno neutro es necesario, ya que cada turno se activará cuando haya más elementos de algún grupo esperando, pero si alguno de los grupos termina de pasar y de los otros hay menos esperando que el límite que habríamos puesto k se activa el turno neutro y podrá pasar cualquiera. Si no se quedaria esperando. Además, se empieza con el turno neutro para que cuando no hay nadie, el primero que llegue pueda entrar).

Con esto se evita la inanición, ya que cuando la cant. de elementos de un grupo esperando sea mayor que k (una cant. que elegimos nosotros), se le da el turno y entrará al puente.

En este caso el invariante queda:

$INV \equiv \{ \text{numpedpas} \geq 0$
 $\text{numcarspas} \geq 0$
 $\text{numcarmpas} \geq 0$
 $\text{turn} \in \{0, 1, 2, -1\}$
 $\text{carwait} \geq 0$
 $\text{sorswait} \geq 0$
 $\text{pedwait} \geq 0$
 $\text{numpedpas} > 0 \rightarrow \text{numcarspas} = 0 \wedge \text{numcarmpas} = 0 \wedge (\text{turn} = 2 \vee \text{turn} = -1)$
 $\text{numcarspas} > 0 \rightarrow \text{numcarmpas} = 0 \wedge \text{numpedpas} = 0 \wedge (\text{turn} = 0 \vee \text{turn} = -1)$
 $\text{numcarmpas} > 0 \rightarrow \text{numcarspas} = 0 \wedge \text{numpedpas} = 0 \wedge (\text{turn} = 1 \vee \text{turn} = -1) \}$

Y lo que se añade en las funciones `leave_car` y `leave_pedestrian` es un "if" de manera que después de que haya salido el elemento que sea y antes de hacer el notify a los otros dos grupos, se comprueba si para alguno de los otros dos grupos hay más de 5 elementos esperando. Y en caso de que los haya se les da el turno. Si no hay ninguno con más de 5 esperando se asigna el turno neutro.