

Práctica 2-PRPA

- Primero desarrollamos una solución sencilla, y después la mejoraremos para que no tenga deadlocks, inanición y que el puente sea seguro (no haya choques ni atropellos)

• Sol sencilla:

• Monitor

num-coches-norte: int = 0

num-coches-sur: int = 0

num-peatones: int = 0

norte-coches-esperando: int = 0

sur-coches-esperando: int = 0

peatones-esperando: int = 0

coches-sur: VC

coches-norte: VC

peatones: VC

turn: int = 0 # 0 → está vacío el puente, 1 → turno coches norte, 2 → turno coches sur, 3 → turno peatones

• El invariante será:

Inv \equiv $\left\{ \begin{array}{l} \text{num-coches-norte} \geq 0, \text{ num-coches-sur} \geq 0, \text{ num-peatones} \geq 0 \\ \text{norte-coches-esperando} \geq 0, \text{ sur-coches-esperando} \geq 0, \text{ peatones-esperando} \geq 0 \\ \text{num-coches-norte} > 0 \rightarrow \text{num-coches-sur} = 0 \wedge \text{num-peatones} = 0 \\ \text{num-coches-sur} > 0 \rightarrow \text{num-coches-norte} = 0 \wedge \text{num-peatones} = 0 \\ \text{num-peatones} > 0 \rightarrow \text{num-coches-norte} = 0 \wedge \text{num-coches-sur} = 0 \\ \text{turn} \in \{0, 1, 2, 3\} \end{array} \right.$

• Wants-enter-car (direction)

if direction == North:

norte-coches-esperando += 1

coches-norte.wait. ((num-coches-sur == 0 \wedge num-peatones == 0) \wedge 1turn == 1 \vee turn == 0)

norte-coches-esperando -= 1

turn = 1

num-coches-norte += 1

* Wants-enter-car (direction)

Continuación función
wants_enter_car

```
* elif direction == South:
    sur_coches_esperando += 1
    coches_sur.wait((num_coches_norte == 0 ^ num_peatones == 0) ^ (turn == 2 ^ turn == 0))
    sur_coches_esperando -= 1
    turn = 2
    num_coches_sur += 1
```

• leaves_car(direction)

```
if direction == North:
    num_coches_norte -= 1
    Inv ^ num_coches_norte > 0
```

```
if num_coches_norte == 0:
    coches_sur.notify_all()
    peatones.notify_all()
    turn = 0
```

```
elif direction == South:
    num_coches_sur -= 1
    Inv ^ num_coches_sur > 0
```

```
if num_coches_sur == 0:
    coches_norte.notify_all()
    peatones.notify_all()
    turn = 0
```

wants_enter_pedestrian()

```
peatones_esperando += 1
peatones.wait((num_coches_norte == 0 ^ num_coches_sur == 0) ^ (turn == 3 ^ turn == 0))
peatones_esperando -= 1
turn = 3
num_peatones += 1
```

leaves_pedestrian()

```
num_peatones -= 1
if num_peatones == 0:
    coches_norte.notify_all()
    coches_sur.notify_all()
    turn = 0
```


car(direction)

loop

monitor.wants_enter_car(direction)

monitor.leaves_car(direction)

pedestrian()

loop

monitor.wants_enter_pedestrian()

monitor.leaves_pedestrian()

Esta solución será segura en cuanto a circulación y no habrá deadlock, pues un coche en una dirección no podrá entrar si hay coches en otra dirección dentro, o peatones, y análogo para los peatones si hay coches en el puente. Esto se consigue gracias a los wait del want_enter_car y wants_enter_pedestrian()

Pero, esta solución tiene inanición, pues, por ejemplo, supongamos que tenemos 50 coches que quieren pasar al norte y 1 coche que quiera pasar al sur;



Supongamos además que ya ha pasado un coche hacia el norte, entonces podrán pasar el resto de tras suya, y puede pasar que no paren de llegar coches que quieran ir al norte y entonces el coche que quiera ir al sur no podrá pasar nunca.

Solución sin inanición:

Para resolver el problema de la inanición, cambiaremos la función `leave-car(direction)` y `leave-pedestrian()`. Para ello, cuando un coche entre en una dirección (o peatón), podrán pasar coches en esa dirección hasta que el primero salga del puente, y entonces cambiaremos el turno de forma rotativa (siempre que haya alguien esperando en el puente).

Cambiamos dichas funciones.

leave-car:

```
if direction == North:  $\rightarrow \neg Inv \wedge num\_coches\_norte > 0$ 
    num\_coches\_norte -= 1
    if sur\_coches\_esperando != 0:
        turn = 2
    elif peatones\_esperando != 0:
        turn = 3
    else:
        turn = 0
    if num\_coches\_norte == 0:
        coches\_sur.notify_all()
        peatones.notify_all()
```

```
elif direction == South:  $\rightarrow \neg Inv \wedge num\_coches\_sur > 0$ 
    num\_coches\_sur -= 1
    if peatones\_esperando != 0:
        turn = 3
    elif norte\_coches\_esperando != 0:
        turn = 1
    else:
        turn = 0
    if num\_coches\_sur == 0:
        coches\_norte.notify_all()
        peatones.notify_all()
```

leave-pedestrian: $\rightarrow \neg Inv \wedge num_peatones > 0$

```
num\_peatones -= 1
if norte\_coches\_esperando != 0:
    turn = 1
elif sur\_coches\_esperando != 0:
    turn = 2
else:
    turn = 0
if num\_peatones == 0:
    coches\_norte.notify_all()
    coches\_sur.notify_all()
```