PRÁCTICA 2: PUENTE DE AMBITE Alejandro Millán Arribas

Desarrolla en papel la solución.

MONITOR-PUENTE

```
n_north: int=0    # Número de coches (norte o sur) y peatones
n_south: int=0      que están cruzando el puente
n_ped: int=0

waiting_N: int=0  # Número de coches (norte o sur) y peatones
waiting_S: int=0    que esperan para cruzar el puente.
waiting_P: int=0

pass_N: VC   # Variables condición que permiten o no el
pass_S: VC     paso de coches (norte o sur) y peatones.
pass_P: VC

wants_enter_car_N():
    waiting_N += 1
    pass_N.wait(n_north==0 ∧ n_south==0 ∧ n_ped==0)
    waiting_N -= 1
    n_north += 1
leaves_car_N():
    n_north -= 1
    if n_north == 0:
        pass_P.notify()
        pass_S.notify()
        if waiting_S==0 ∧ waiting_P==0:
            pass_N.notify()

wants_enter_car_S():
    waiting_S += 1
    pass_S.wait(n_north==0 ∧ n_south==0 ∧ n_ped==0)
    waiting_S -= 1
    n_south += 1
leaves_car_S():
    n_south -= 1
    if n_south==0:
        pass_P.notify()
        pass_N.notify()
        if waiting_N==0 ∧ waiting_P==0:
            pass_S.notify()
```

```
wants_enter_pedestrian():
    waiting_P += 1
    pass_P.wait(n_north==0 ∧ n_south==0)
    waiting_P -= 1
    n_ped += 1
leaves_pedestrian():
    n_ped -= 1
    if n_ped == 0:
        pass_N.notify()
        pass_S.notify()
```

COCHES

```
car_N():
    loop:
        monitor.wants_enter_car_N()
        operaciones_cruzar
        monitor.leaves_car_N()
car_S():
    loop:
        monitor.wants_enter_car_S()
        operaciones_cruzar
        monitor.leaves_car_S()
```

PEATONES

```
pedestrian()
    loop:
        monitor.wants_enter_pedestrian()
        operaciones_cruzar
        monitor.leaves_pedestrian()
```

Escribe el invariante del monitor.

$$\begin{aligned} INV \equiv\ & n\_north \geq 0 \wedge n\_south \geq 0 \wedge n\_ped \geq 0 \wedge \\ & waiting\_N \geq 0 \wedge waiting\_S \geq 0 \wedge waiting\_P \geq 0 \wedge \\ & n\_north > 0 \rightarrow n\_south = 0 \wedge n\_ped = 0 \wedge \\ & n\_south > 0 \rightarrow n\_north = 0 \wedge n\_ped = 0 \wedge \\ & n\_ped > 0 \rightarrow n\_north = 0 \wedge n\_south = 0 \end{aligned}$$

Demuestra que el puente es seguro.

En cuanto a los coches, estos solo pueden cruzar el puente cuando no haya ni ningún otro coche (en cualquier sentido) ni ningún peatón. Luego nunca se pueden cruzar dos coches en distintas direcciones.

Además, los peatones solo pueden entrar cuando no hay ningún coche en el puente. Es decir, los peatones se cruzan con más peatones pero nunca con un coche.

Por tanto, el puente es seguro.

Demuestra la ausencia de Deadlocks.

El deadlock surgiría si, al menos, dos procesos se impidieran mutuamente cruzar el puente. Esto no ocurre ya que lo único que impide a un proceso cruzar el puente es que ya haya otro proceso cruzando. Tarde o temprano el puente quedará libre en algún instante y podrán cruzar todos los procesos.

En el caso de los peatones es aún más claro. Lo que les impide cruzar es que haya algún coche cruzando, pero si hay peatones pueden cruzar sin problema.

Podría haber deadlocks si hubiéramos colocado las variables waiting en la variable condición y exigido que fuesen igual a 0 para poder cruzar. En el momento en que un proceso de cada tipo ya ninguno de ellos podría entrar y habría deadlock. Por ello estas variables no están en la variable condición.

Demuestra la ausencia de inanición.

La inanición se produciría si hubiese un proceso que quisiese cruzar el puente pero, por algún motivo, nunca pudiese. Imaginemos que un proceso cualquiera quiere entrar:

•) Si el puente está vacío cruzará sin problema.

•) Si el puente está ocupado en ese momento la variable condición bloqueará dicho proceso. Es en este caso donde podríamos tener problemas.

La forma en que evitamos esto es haciendo que un proceso cuando cruza el puente ~~y~~ y este queda vacío notifica a algún proceso de los otros dos tipos para que puedan entrar. Podría haber problemas con los de su mismo tipo, pero mediante las variables waiting hacemos que si no hay nadie esperando notifique también a los de su tipo. Luego, por la hipótesis de justicia, eventualmente todo proceso que haya sido bloqueado será notificado para poder cruzar el puente.