Práctica 2 Puente de Ambite con Monitores

Jose Ignacio Alba Rodriguez

Monitor

```
mutex ← Lock(0)

nVehiN : int = 0   } Número de vehículos
nVehiS : int = 0   } en el puente en dirección N/S
nPeat  : int = 0   } Número de peatones en el puente

wVehiN : int = 0   } Número de vehículos en
wVehiS : int = 0   } dirección N/S o peatones
wPeat  : int = 0   } esperando para acceder al puente

N_puente: VC   } VC para los que quieren
S_puente: VC   } acceder al puente
P_puente: VC

N_espera: VC   } VC para los que quieren
S_espera: VC   } pedir acceso al puente
P_espera: VC
```

```
wants_enter_pedestrian()
mutex.wait()
P_espera.wait(nPeat==0 or wVehiN+wVehiS==0)
wPeat += 1
P_puente.wait(nVehiS + nVehiN ==0)
wPeat -= 1
nPeat += 1
N_espera.notify()
S_espera.notify()
mutex.signal()
```

```
leaves_pedestrian()
mutex.wait()
nPeat -= 1
N_puente.notify()
S_puente.notify()
mutex.signal()
```

```
wants_enter_car(dirección)
mutex.wait()
if dirección == N:
    N_espera.wait(nVehiN==0 or wVehiS+wPeat==0)
    wVehiN += 1
    N_puente.wait(nVehiS + nPeat == 0)
    wVehiN -= 1
    nVehiN += 1
    S_espera.notify()
    P_espera.notify()
else:
    S_espera.wait(nVehiS==0 or wVehiS+wPeat==0)
    wVehiS += 1
    S_puente.wait(nVehiN + nPeat == 0)
    wVehiS -= 1
    nVehiS += 1
    P_espera.notify()
    N_espera.notify()
mutex.signal()
```

```
leaves_car(direccion)
mutex.wait()
if dirección == N:
    nVehiN -= 1
    S_puente.notify()
    P_puente.notify()
else:
    nVehiS -= 1
    P_puente.notify
    N_puente.notify
mutex.signal()
```

Peaton (monitor)

```
monitor.wants_enter_pedestrian()
cruzar_puente_peaton()
monitor.leaves_pedestrian()
```

Coche / Vehículo (dirección D, monitor)

```
monitor.wants_enter_car(D)
cruzar_puente_vehiculo()
monitor.leaves_car(D)
```

Invariante del Monitor y seguridad del Puente

- $0 \leq nVehiN \wedge 0 \leq nVehiS \wedge 0 \leq nPeat$ → Para cada una de las variables, estas son incrementadas en la llamada a wants_enter_. Solo se ven reducidas en las llamadas a leaves_, pero estas solo son ejecutadas después de que se complete la llamada a wants_enter_

- $0 \leq wVehiN \wedge 0 \leq wVehiS \wedge 0 \leq wPeat$ → Son siempre incrementadas en wants_enter y solo son reducidas después de haber sido incrementadas, una vez el cliente accede al puente

- $(nVehiN > 0 \rightarrow nVehiS == 0 \wedge nPeat == 0) \wedge$
  $(nVehiS > 0 \rightarrow nVehiN == 0 \wedge nPeat == 0) \wedge$
  $(nPeat > 0 \rightarrow nVehiN == 0 \wedge nVehiS == 0)$

  } Esta es la condición principal que exigimos para que el puente sea seguro para los peatones y evite accidentes. Si la aseguramos, solo hay peatones o vehículos en una dirección.

Podemos asegurarla ya que estas variables son ~~se~~ únicamente incrementadas después de esperar a la condición de la derecha de la flecha. Si la variable ha incrementado su valor, esto solo ha podido ocurrir si las otras dos variables son 0. A su vez, para que las otras variables dejen de valer 0, la primera variable tendría que haber vuelto a tomar el valor 0

Lema Los vehículos o peatones solo pueden empezar a esperar si { nadie de otro tipo está esperando / no están ocupando el puente los de su tipo

Esto se debe a que para solicitar el acceso, primero deben esperar a que se cumpla esta condición. De esta forma, aseguramos que cuando uno de un grupo comienza a cruzar, solo una cantidad limitada puede pasar antes de que se le permita el acceso a un grupo distinto. Los que no hubiesen solicitado el acceso antes de que alguien de su grupo comience a cruzar, deben esperarse a que nadie más espere o que cruce un tipo distinto.

DeadLock

- Si el puente está ocupado por un grupo, gracias al lema, solo entra una cantidad limitada, por lo que en algún momento, el puente quedará vacío
- Si el puente está vacío, pueden darse los siguientes casos: - Nadie ha entrado aún ⇒ No pueden estar bloqueados en la condición de antes de cruzar
  - nadie está esperando a tener acceso. Todos están en condiciones de empezar a esperar y por tanto, alguien terminará entrando, por lo que no hay bloqueo
  - El último en salir fue de algún grupo (digamos un peatón por ejemplo) y hay de otro grupo esperando (siguiendo el ejemplo, algún vehículo está esperando). En este caso, no pueden entrar a esperar más del primer grupo (no ~~hay más~~ puede pedir el acceso ningún peatón). Si había alguno del primer grupo esperando, como el puente está vacío, puede entrar. En caso contrario, como había otro grupo esperando, al salir, este grupo es notificado en leaves_(car / pedestrian) y por tanto, puede entrar. Al entrar, notifica a los del primer grupo que no pudieron pedir acceso y por tanto ahora pueden y entrarán más adelante.
  - El último en salir fue de un grupo y no hay de otros grupos esperando. En este caso, los del mismo grupo que estén esperando están en condiciones de entrar. Los del mismo grupo que aún no están esperando, van a poder pasar a esperar. Y los de grupos contrarios que aún no estén esperando también están en condiciones de esperar, si lo hacen, pasaríamos al caso anterior

En ningún caso se produce una situación de DeadLock, ya que en algún momento, alguien entra al puente, habilitando que los siguientes entren después de ser notificados una vez termine de cruzar.

Inanición

Sea H un peatón o vehículo en alguna dirección (H es un proceso), queremos ver que en algún momento, H cruza el puente.

- Si hay alguien del mismo tipo que H en el puente
  - si H está esperando a tener acceso → lo recibirá si no hay procesos distintos con acceso, en otro caso, recibirá acceso cuando otro de otro tipo cruce el puente. Esto terminará ocurriendo gracias al Lema y el caso ⊗ para el proceso que está esperando con acceso
  - si H ya tiene acceso, no puede estar bloqueado (si no, el proceso del puente estaría bloqueado también ya que son notificados a la vez). Por lo tanto, puede entrar
- Si hay alguien de un tipo distinto que H en el puente
  - H no tiene acceso → está en condiciones de recibirlo y pasa al siguiente caso
  - ⊗ - H tiene acceso pero está bloqueado. Por el Lema, en algún momento, dejarán de entrar más en el puente y en algún momento el puente queda vacío
- Si el puente está vacío (nadie en el puente)
  - H no tiene acceso → está en condiciones de recibirlo y ~~entra~~ empieza a esperar
  - H está esperando. Por hipótesis de justicia, como el puente está vacío, todos tienen las mismas condiciones para entrar, por lo que H terminará entrando en algún momento