Puente de Ambite

Gabriela Ballesteros Gómez.

Desarrollaremos una solución sencilla que posteriormente mejoraremos para que evitar inanición, deadlocks y poder garantizar la seguridad del puente: no pasan coches en sentidos opuestos ni peatones y coches simultáneamente.

Solución inicial

Definimos, primero que nada, el invariante:

- car_bridge≥[0,0], peaton_bridge≥0
- car_queue≥[0,0], peaton_queue≥0
- car_bridge[n]>0 →car_bridge[n+1 (mod n)]=0 ^ peaton_bridge=0 (n=1,2)
- peaton_bridge>0 →car_bridge=[0,0]

```
Monitor
  {INV}
  car_bridge:[int,int]=[0,0] #número de coches en el momento en el norte y en el sur, respectivamente
  car_queue: [int,int]=[0,0] #número de coches esperando a entrar por el norte y por el sur
  peaton_bridge:   int=0     #número de peatones en el puente
  peaton_queue:    int=0     #número de peatones esperando a entrar
  northway:        vc #variable condición para los coches del norte
  southway:        vc #variable condición para los coches del sur
  peatonway:       vc #variable condición para los peatones
  sentido:         int=0 #indica el sentido de acceso del puente: 0:norte; 1:sur; 2:peatones

wants_enter_car(direction)
  car_queue[direction]+=1
  if direction=NORTH:
    northway.wait(sentido==0 ^(car_bridge[1]==0^peaton_bridge==0))
  else:
    southway.wait(sentido==1 ^(car_bridge[0]==0^peaton_bridge==0))
  car_queue[direction] -=1
  car_bridge[direction]+=1

leaves_car(direction)
  car_bridge[direction]-=1
  if direction==NORTH:
    {INV^car_bridge[0]>0}
    if car_bridge[0]==0:
      southway.notify()
      peatonway.notify()
  else:
    {INV^car_bridge[1]>0}
    if car_bridge[1]==0:
      northway.notify()
      peatonway.notify()
```

```
  {INV}

wants_enter_pedestrian()
  peaton_queue+=1
  peatonway.wait(sentido==2 ^ car_bridge==[0,0])
  peaton_bridge-=1

leaves_pedestrian()
  {INV^peaton_bridge>0}
  peaton_bridge-=1
  if peaton_bridge==0:
    northway.notify()
    southway.notify()
  {INV}

peaton()
  loop
    monitor.wants_enter_pedestrian()
    monitor.leaves_pedestrian()

car(direction)
  loop
    monitor.wants_enter_car(direction)
    monitor.leaves_car(direction)
```

Con este método garantizamos la seguridad del puente. Sin embargo, esta solución no es del todo correcta, ya que puede haber inanición si, por ejemplo están pasando coches desde el norte y hay peatones o coches en el sur esperando.

Veamos cómo obtener una solución correcta que cumpla lo requerido.

Solución sin inanición

Modificamos las funciones `leaves_pedestrian` y `leaves_car` y los wait de las variables condición añadiendo la posibilidad de llegar a puente vacío y así poder entrar directamente. Además, incluimos un nuevo valor para el sentido, con valor 100 para indicar que no hay tráfico esperando en ninguna de las posibles formas. El invariante se mantiene igual. La idea es que, cada vez que sale un coche en un sentido determinado, da paso a una de las otras opciones, que no se ponen en marcha hasta que el puente está vacío. Ninguna de estas modificaciones implica el incumplimiento del invariante.

```
wants_enter_car(direction)
  car_queue[direction]+=1
  if direction=NORTH:
    northway.wait(sentido==100 v sentido==0 ^(car_bridge[1]==0^peaton_bridge==0))
  else:
    southway.wait(sentido==100 v sentido==1 ^(car_bridge[0]==0^peaton_bridge==0))
  car_queue[direction] -=1
  car_bridge[direction]+=1

leaves_car(direction)
  car_bridge[direction]-=1
  if direction==NORTH:
```

```
    {INV^car_bridge[0]>0}
    if car_queue[1]>peaton_queue and car_queue[1]!=0:
      sentido=1
    elif car_queue[1]<=peaton_queue and peaton_queue!=0:
      sentido=2
    else:
      sentido=100

    if car_bridge[0]==0:
      southway.notify()
      peatonway.notify()
  else:
    {INV^car_bridge[1]>0}
    if car_queue[0]>peaton_queue and car_queue[0]!=0:
      sentido=0
    elif car_queue[0]<=peaton_queue and peaton_queue!=0:
      sentido=2
    else:
      sentido=100

    if car_bridge[1]==0:
      northway.notify()
      peatonway.notify()
  {INV}
wants_enter_pedestrian()
  peaton_queue+=1
  peatonway.wait(sentido==100 v sentido==2 ^ car_bridge==[0,0])
  peaton_bridge-=1

leaves_pedestrian()
  {INV^peaton_bridge>0}
  peaton_bridge-=1
  if peaton_bridge==0:
    northway.notify()
    southway.notify()
  {INV}
```

No hay inanición, ya que se cambia el sentido cuando un coche termina de pasar y se bloquea temporalmente el paso en ese sentido hasta que hayan pasado otros. Si no hay nadie en el puente ni esperando, entra el primero en llegar.

El puente es seguro, ya que garantizamos que ningún coche ni peatón puede pasar si hay alguien con otras intenciones en el puente. Por tanto, nadie será atropellado ni habrá choques frontales.

Tampoco hay deadlocks, ya que eso implicaría que más de un proceso de distinta naturaleza se encuentre en el puente, lo cual no está permitido según hemos diseñado el programa. Otra opción sería, por ejemplo, el bloqueo del sistema si no se cambia la dirección y no hay nadie en ninguna de las colas. Al llegar, por ejemplo, un peatón, no podría acceder al puente ni habría ningún proceso que a su salida le permita el paso. Esto se ha solucionado al introducir el sentido neutro y añadiendo en las variables condición que devuelvan cierto si el sentido es 100.