

PRÁCTICA 2: MONITOR EN EL PUENTE

Sergio Monzón Garcés

MONITOR PUENTE

personas_dentro : int = 0 número de personas dentro del puente
coches_dentro : int = 0 numero de coches dentro del puente

hay_coches : VC
hay_personas : VC

preferencia = 0 indica la preferencia de dirección de paso (0 para Norte, 1 para Sur)

hay_coches_esperando : VC

hay_coches_esperando_norte : int = 0 número de coches esperando entrar por el norte
hay_coches_esperando_sur : int = 0 número de coches esperando entrar por el sur

WANTS_ENTER_CAR (direction)

EXPLICACIÓN:

Primero se diferencia entre si el coche quiere entrar por el norte o por el sur. Esto tiene la finalidad de que, si el coche quiere entrar por el norte, luego el siguiente coche que tiene preferencia para entrar es el del sur, y viceversa, y así se evita que todos los vengan de una sola dirección.

Para cada dirección norte / sur:

- Añade un coche a la cola de los coches esperando por el norte / sur
- Espera a que no haya ningún peatón en el puente
- Espera a que no haya ningún coche en el puente
- Espera a que tengan preferencia los coches del norte / sur

Una vez pueda pasar, se elimina ese coche de la cola de los coches esperando por el norte / sur

Añade en 1 al número de los coches de dentro del puente

if direction == 0:

- hay_coches_esperando_norte += 1
- hay_personas.wait_for(personas_dentro == 0)
- hay_coches.wait_for(coches_dentro == 0)
- hay_coches_esperando.wait_for(preferencia == 0)

- hay_coches_esperando_norte -= 1
- coches_dentro += 1

if direction == 1:

- hay_coches_esperando_sur += 1
- hay_personas.wait_for(personas_dentro == 0)
- hay_coches.wait_for(coches_dentro == 0)
- hay_coches_esperando.wait_for(preferencia == 1)

- hay_coches_esperando_sur -= 1
- coches_dentro += 1

LEAVES_CAR (direction)

EXPLICACIÓN:

Una vez que el coche quiere abandonar el puente, el número de coches dentro del puente pasa a ser 0, (coches dentro = $1 - 1 = 0$) y el semáforo de si hay coches dentro del puente se libera (ya que ahora no hay coches dentro del puente)

Más adelante, si la dirección de preferencia era el norte, después de haber pasado el coche proveniente del norte, ve si hay algún coche queriendo entrar por el sur. En caso afirmativo, la preferencia pasa a ser de la dirección sur. En caso negativo, siguen pasando coches del norte. (y viceversa en caso del sur)

```
coches_dentro -= 1
hay_coches.release()
```

```
if dirección == 0 and hay_coches_esperando_sur > 0:
    preferencia = 1
if dirección == 1 and hay_coches_esperando_norte > 0:
    preferencia = 0
```

WANTS_ENTER_PEDESTRIAN

EXPLICACIÓN:

Espera a que no haya ningún coche dentro. Mas tarde, añade 1 al número de personas dentro del puente

```
hay_coches.wait_for(coches_dentro == 0)
personas_dentro += 1
```

LEAVES_PEDESTRIAN

EXPLICACIÓN:

Reduce en 1 las personas de dentro del puente, y si no hay mas personas, libera el semáforo para que los coches puedan entrar.

```
personas_dentro -= 1
if personas_dentro == 0:
    hay_personas.release()
```

INVARIANTE

`(0 <= personas_dentro)`

`(0 <= coches_dentro <= 1)`

`(0 <= hay_coches_esperando_norte/sur)`

Si coches_dentro = 1, entonces personas_dentro = 0

Si personas_dentro > 0, entonces coches_dentro = 0

DEMOSTRAR QUE EL PUENTE ES SEGURO

Veamos que no hay 2 coches a la vez en el puente:

Para demostrarlo, podemos ver que cuando un coche entra al puente, el número coches_dentro pasa a ser 1. Si otro coche quiere entrar al puente, wants_enter_car primero ve si no coches_dentro con "hay_coches.wait_for(coches_dentro == 0)". Es por eso que no puede haber 2 coches a la vez dentro del puente

Veamos que no puede haber un peatón y un coche a la vez en el puente:

Si un peatón entra al puente, entonces personas_dentro pasa a ser mayor que 0, es decir, personas_dentro > 0. Si un coche quiere entrar, entonces primero wants_enter_car mira a ver si hay o no peatones dentro con el comando "hay_personas.wait_for(personas_dentro == 0)". Por lo tanto, el coche no puede entrar a menos que no haya peatones.

Si un coche entra al puente, entonces coches_dentro pasa a ser mayor que 0, es decir coches_dentro > 0. Si un peatón quiere entrar, entonces primero wants_enter_pedestrian mira a ver si hay o no coches dentro con el comando "hay_coches.wait_for(coches_dentro == 0)". Por lo tanto, el peatón no puede entrar a menos que no haya coches.

DEMOSTRAR LA AUSENCIA DE DEADLOCKS

Se puede ver la ausencia de deadlocks viendo que cuando un coche/peatón quiere entrar, hay un semáforo con un wait, y cuando ese mismo coche/peatón sale del puente, ese semáforo se libera. Por lo tanto, siempre hay continuo paso de coches y peatones, lo que provoca que ningún coche / peatón no pueda pasar.

DEMOSTRAR LA AUSENCIA DE INANICIÓN

Todos los procesos pueden suceder, ya que no hay ningún proceso que bloquee a otro. Por eso, no se da ninguna situación de inanición