

## PRÁCTICA 2 PRPA: Puente de Ambato

HELENA FERRERO CHAVEZ

- Invariante del monitor:  $\{ (carsouth \neq 0 \Rightarrow carsouth = 0 \wedge peds = 0) \vee (carsouth \neq 0 \Rightarrow carsouth = 0 \wedge peds = 0) \vee (peds \neq 0 \Rightarrow carsouth = 0 \wedge carsouth = 0) \} = \{ INV \}$

Monitor CP:

```
carsouth: int = 0
carsouth: int = 0
peds: int = 0
{ INV }
mutex = lock()
no_carsouth = Condition(mutex)
no_carsouth = Condition(mutex)
no_peds = Condition(mutex)
```

wants\_enter\_car():

```
mutex.acquire()
if direction == 0:
    no_carsouth.wait(carsouth == 0)
    no_peds.wait(peds == 0)
    carsouth += 1
else:
    no_carsouth.wait(carsouth == 0)
    no_peds.wait(peds == 0)
    carsouth += 1
mutex.release()
```

leave\_car():

```
mutex.acquire()
if direction == 0:
    carsouth -= 1
    if carsouth == 0:
        no_carsouth.notify()
        no_carsouth.notify()
        no_peds.notify()
else:
    carsouth -= 1
    if carsouth == 0:
        no_carsouth.notify()
        no_carsouth.notify()
        no_peds.notify()
mutex.release()
```

wants\_enter\_pedestrian():

```
mutex.acquire()
no_carsouth.wait(carsouth == 0)
no_carsouth.wait(carsouth == 0)
peds += 1
mutex.release()
```

leave\_pedestrian():

```
mutex.acquire()
peds -= 1
if peds == 0:
    no_peds.notify()
    no_carsouth.notify()
    no_carsouth.notify()
mutex.release()
```

- El puente es seguro porque, como se puede ver en el código, un coche en una dirección sólo puede entrar si no hay ningún peatón ni ningún coche en la otra dirección, y un peatón sólo puede entrar si no hay ningún coche en el puente.

- No se producen deadlocks porque usamos un semáforo lock() que evita que haya bloqueos, es decir, evita que dos procesos actúen a la vez sobre el Monitor.

- No se produce inanición porque utilizamos las notificaciones de las condiciones (condition.notify()) para evitar que haya algún proceso en espera permanente. Además, la inanición también se evita con los turnos que se asignan a los procesos para que puedan actuar.