

## PRÁCTICA 2: TÚNEL DE KIYOTAKI

Marta Mas Villar  
Arturo Tarazona Pérez  
Pablo Muñoz Vicente

• Definamos las siguientes variables:

• `Ncars: int`

Es el número de coches cruzando el túnel hacia el norte

• `Scars: int`

Es el número de coches cruzando el túnel hacia el sur

• `Ncars-waiting: int`

Es el número de coches esperando en el sur para ir al norte

• `Scars-waiting: int`

Es el número de coches esperando en el norte para ir al sur

• `no-cars-goingN: condition`

• `no-cars-goingS: condition`

• El invariante será el siguiente:  $\{INV = \neg Ncars = 0 \vee Scars = 0\}$

• Procesos que vamos a implementar (Vamos que el invariante siempre se cumple en ellos):

`want-enter-north()`:

$\{INV\} \rightarrow$  es cierto

`Ncars-waiting += 1`

$\{INV\} \rightarrow$  sigue siendo cierto ya que la asignación anterior no afecta al invariante

`no-cars-goingS.wait(Scars=0  $\wedge$  Scars-waiting=0)`

$\{INV \wedge Scars=0\} \rightarrow$  es cierto ya que `Scars=0`

`Ncars-waiting -= 1`

$\{INV\}$

`Ncars += 1`

$\{INV \wedge Ncars > 0\} \rightarrow$  se sigue cumpliendo `Scars=0` luego el invariante es cierto

`leave-tunnel-north()`:

$\{INV\} \rightarrow$  es cierto puesto que cuando llamamos a esta función `Ncars > 0  $\rightarrow$  Scars=0`

`Ncars -= 1`

$\{INV\} \rightarrow$  es cierto puesto que `Scars=0` ya que la instrucción anterior solo modifica `Ncars`

`if Ncars=0:`

`no-cars-goingN.signal()`

$\{INV\} \rightarrow$  sigue siendo cierto ya que las instrucciones anteriores no afectan al invariante

Los procesos para el sentido sur los implementamos de forma análoga:

`want-enter-south()`:

$\{INV\}$

`Scars-waiting += 1`

$\{INV\}$

`no-cars-goingN.wait(Ncars=0  $\wedge$  Ncars-waiting=0)`

$\{INV \wedge Ncars=0\}$

`Scars-waiting -= 1`

$\{INV\}$

`Scars += 1`

$\{INV \wedge Scars > 0\}$

`leave-tunnel-south()`:

$\{INV\}$

`Scars -= 1`

$\{INV\}$

`if Scars=0:`

`no-cars-goingS.signal()`

$\{INV\}$