

$\{INV\} = \{ \text{pedr} \geq 0 \wedge \text{carrnorth} \geq 0 \wedge \text{carrsouth} \geq 0 \wedge \text{pedr} \geq 0 \rightarrow (\text{carrnorth} = 0 \wedge \text{carrsouth} = 0) \wedge \text{carrnorth} > 0 \rightarrow (\text{pedr} = 0 \wedge \text{carrsouth} = 0) \wedge \text{carrsouth} > 0 \rightarrow (\text{pedr} = 0 \wedge \text{carrnorth} = 0) \}$

$\leftarrow$   
wants-enter-car(direction):

{INV}

if direction == 0:

no-carsouth.wait(pedr == 0  $\wedge$  carrnorth == 0)

carrsouth = carrsouth + 1

{INV  $\wedge$  carrsouth > 0}

else:

no-carnorth.wait(pedr == 0  $\wedge$  carrsouth == 0)

carnorth = carrnorth + 1

{INV  $\wedge$  carrnorth > 0}

leave-car(direction):

{INV}

if direction == 0:

{INV  $\wedge$  carrsouth > 0}

carrsouth = carrsouth - 1

{INV}

if carrnorth == 0:

no-pedr.notify()

no-carnorth.notify()

{INV}

wants-enter-pedestrian():

{INV}

no-pedr.wait(carrnorth == 0  $\wedge$  carrsouth == 0)

pedr = pedr + 1

{INV  $\wedge$  pedr > 0}

leave-pedestrian():

{INV  $\wedge$  pedr > 0}

pedr = pedr - 1

{INV}

if pedr == 0:

no-carsouth.notify()

no-carnorth.notify()

Monitor CP

carrnorth : int = 0

carrsouth : int = 0

pedr : int = 0

no-carsouth : VC = False

no-carnorth : VC = False

no-pedr : VC = False

- El puente es seguro porque en el invariante se cumple que si un grupo está pasando por el puente, los otros grupos están fuera del puente y el invariante se cumple durante toda la ejecución.
- No se producen deadlocks porque en el invariante se evita que dos procesos actúen a la vez sobre el monitor. La inclusión se evita con el invariante y con la turnos que se asignan a los procesos para que puedan actuar.

in invariante:

$\{INV\} = \{pedn \geq 0 \wedge carsouth \geq 0 \wedge carnorth \geq 0 \wedge pedn - waiting \geq 0 \wedge$   
 $carsouth - waiting \geq 0 \wedge carnorth - waiting \geq 0 \wedge pedn \geq 0 \rightarrow (carnorth = 0 \wedge carsouth = 0) \wedge$   
 $carnorth > 0 \rightarrow (pedn = 0 \wedge carnorth = 0) \wedge carsouth > 0 \rightarrow (pedn = 0 \wedge carsouth = 0) \wedge 0 \leq turn \leq 2\}$

Monitor CP:

```
pedn : int = 0
carsouth : int = 0
carnorth : int = 0
pedn - waiting : int = 0
carnorth - waiting : int = 0
carsouth - waiting : int = 0
no_carsouth : VC = False
no_carnorth : VC = False
no_pedn : VC = False
turn : int = 0
```

wants\_enter\_car(direction):

```
{INV} ^ (turn = 2 ^ turn = 1) {
  if direction == 1:
    {INV} ^ turn = 1 {
      carsouth - waiting = carsouth - waiting + 1
      {INV} ^ turn = 1 ^ carsouth - waiting > 0 {
        no_carsouth.wait (pedn == 0 ^ carsouth == 0) ^ ((carsouth - waiting <= 5 ^ pedn - waiting <= 5) ^
        ^ turn = 1)
      }
      carsouth - waiting = carsouth - waiting - 1
    }
    {INV} ^ turn = 1 {

```

```
      carsouth = carsouth + 1
    }
    {INV} ^ turn = 1 ^ carsouth > 0 {

```

else:

(igual para carsouth)

leave\_car(direction):

```
{INV} ^ (carnorth > 0 ^ carsouth > 0) {
```

if direction == 1:

```
{INV} ^ carsouth > 0 {
```

```
carsouth = carsouth - 1
```

```
{INV}
```

if turn == 1:

if carsouth - waiting > 0:

```
turn = 2
```

```
{INV} ^ turn = 2 {
```

elif pedn - waiting > 0:

```
turn = 0
```

```
{INV} ^ turn = 0 {
```

```
turn = 1
```

```
{INV} ^ turn = 1 {
```

if carsouth == 0:

```
no_carsouth.notify()
```

```
no_pedn.notify()
```

else: (igual para carsouth)

wants\_enter\_pedestrian():

```
{INV} ^ turn = 0 {
```

```
pedn - waiting = pedn - waiting + 1
```

```
{INV} ^ pedn - waiting > 0 ^ turn = 0 {
```

```
no_pedn.wait ((carsouth == 0 ^ carnorth == 0) ^
```

```
(carsouth - waiting <= 5 ^ carnorth - waiting <= 5) ^
```

```
(turn = 0) {
```

```
pedn - waiting = pedn - waiting - 1
```

```
{INV} ^ turn = 0 {
```

```
pedn = pedn + 1
```

```
{INV} ^ pedn > 0 ^ turn = 0 {
```

leave\_pedestrian():

```
{INV} ^ pedn > 0 {
```

```
pedn = pedn - 1
```

```
{INV}
```

if turn == 0:

if carsouth - waiting > 0:

```
turn = 1
```

```
{INV} ^ turn = 1 {
```

elif carsouth - waiting > 0:

```
turn = 2
```

```
{INV} ^ turn = 2 {
```

if pedn == 0:

```
no_carsouth.notify()
```

```
no_carsouth.notify()
```

```
{INV} ^ turn > 0 {
```