INV = { ncar_north ≥ 0, ncar_south ≥ 0, nped ≥ 0,

ncar_north ≥ 0 → ncar_south = 0 and nped = 0,

ncar_south ≥ 0 → ncar_north = 0 and nped = 0,

nped ≥ 0 → ncar_south = 0 and ncar_north = 0,

(a) turno = 0 and ncar_north = 0 and ncar_south = 0 and nped = 0

→ ncar_north_waiting > 0 or (ncar_south_waiting = 0 and nped_waiting = 0),

(b) turno = 1 and ncar_north = 0 and ncar_south = 0 and nped = 0

→ ncar_south_waiting > 0 or (ncar_north_waiting = 0 and nped_waiting = 0),

(c) turno = 2 and ncar_north = 0 and ncar_south = 0 and nped = 0

→ nped_waiting > 0 or (ncar_south_waiting = 0 and ncar_north_waiting = 0) }

Vamos a demostrar que lo anterior es invariante

① Todos los valores son inicializados a 0 de modo que las primeras condiciones se cumplen trivialmente

Además, turno = 0 y todas las variables son 0 así que también se cumplen las otras 3 condiciones con respecto a los turnos

② Veamos ahora que se sigue cumpliendo el invariante a lo largo de la ejecución

PARA LA ENTRADA DE COCHES AL PUENTE

DIRECCIÓN NORTE (LA DIRECCIÓN SUR Y EL PASO DE PEATONES ES ANÁLOGO)

El coche para entrar en el puente tiene que esperar a que se cumpla la condición ncar_south = 0 y nped = 0, así que se comprueba y, por tanto, al salir se cumple INV y además ncar_north > 0

Si el proceso se queda bloqueado

Se tiene que ncar_north_waiting = + 1 (que se acaba de actualizar)

Veamos si se cumplen las condiciones del invariante

No puede entrar ⇒ nped > 0 o ncar_south > 0 las últimas condiciones no fallan ya que no se dan las hipótesis

Ahora, si ncar_south = 0 y nped = 0

- Si turno = 0 (no hay bloqueo) el coche entra en el puente y se cumple el INV
- Si turno = 1 y ncar_south_waiting > 0 se cumple (b)
- Si turno = 1 y ncar_south_waiting = 0 no hay bloqueo puesto que el INV nos garantiza que nped_waiting = 0 (b)

Lo mismo ocurre si turno = 2

SALIDA DE COCHES DEL PUENTE

DIRECCIÓN NORTE (LA DIRECCIÓN SUR Y LA SALIDA DE PEATONES ES ANÁLOGA)

La función leaves_car se ejecuta tras la función wants_enter_car que ya vimos que cumple el INV al ejecutarse y además implica que ncar_north > 0

Como ncar_north > 0 $\Rightarrow$ ncar_north - 1 $\geq$ 0

Y el resto de variables permanece invariable por lo que las primeras condiciones se siguen cumpliendo

Para las condiciones (a), (b) y (c)

Se comprueba si ncar_south_waiting != 0 si lo es el turno es 1

entonces si ncar_north = 0, ncar_south = 0 y nped = 0 se verifica (b)

Si no, se comprueba si nped_waiting = 0 entonces el turno pasaría a ser 2 y se verifica (c)

En caso de que las dos condiciones anteriores no se cumplan el turno pasa a ser 0 y se verifica (a)

Por tanto, esta función mantiene el invariante al ejecutarse

Las otras funciones del programa no modifican el invariante

SEGURIDAD DEL PUENTE

Vamos a demostrar que el puente es seguro teniendo en cuenta las condiciones del invariante

Ya hemos verificado el invariante y las condiciones para que haya coches en dirección norte es que no haya ni coches en dirección sur ni peatones

$$ncars\_north > 0 \longrightarrow ncars\_south = 0 \text{ and } nped = 0$$

Esto se comprueba en la función wants_enter_car(0) y si no se bloquea

Lo mismo ocurre para los coches en dirección sur y los peatones

De este modo, en el puente nunca hay coches en ambos direcciones o coches y peatones de forma simultánea