

• Tunnel:

$\{ INV : n_S \geq 0 \wedge n_N \geq 0 \wedge \neg (S \wedge N) \}$

n_S = número de coches esperando en dir norte

n_N = número de coches esperando en dir sur

S = en el túnel hay coches en dir sur

N = en el túnel hay coches en dir norte

“Suponemos dir = N, para el caso dir = S es análogo”

waits-enter (dir) (dir = NORTH)

$\{ INV \wedge True \}$

if direction == NORTH

self.para-n.wait-for ($n_N \geq n_S$ and $\neg S$ in-tunnel)

else:

self.para-s.wait-for ($n_S \geq n_N$ and $\neg N$ in-tunnel)

$\{ INV \wedge n_N \geq n_S \geq 0 \wedge \neg S \}$

waiting-reduce (direction) ($n_N := n_N - 1$)

$\{ INV \wedge \neg S \}$

in-tunnel.append (direction) (N)

$\{ INV \wedge \neg S \wedge N \} \Rightarrow \{ INV \}$

“ Hacemos lo mismo con leaves_tunnel, suponemos que en el tunnel solo hay de dirección NORTH que podemos suponer porque $INV \Rightarrow \neg(S \wedge N)$ ”

leaves_tunnel(dir) (dir = NORTH)

{ INV \wedge True {

in_tunnel_remove(dir) $\neg N$

para_n_notify_all()

para_s_notify_all()

{ INV $\wedge \neg N$ } \Rightarrow { INV }