DEMOSTRACIÓN DEL INVARIANTE

Tenemos que $I = (c0 > 0 \Rightarrow p = 0 \wedge c1 = 0) \wedge (c1 > 0 \Rightarrow p = 0 \wedge c0 = 0)$

donde
- c0: coches cruzando en dirección 0
- c1: " " " " 1
- p: peatones "

De las condiciones del invariante tenemos también:

$$p > 0 \Rightarrow c0 = 0 \wedge c1 = 0$$

Cuando se inicia el invariante se cumple, ya que todos empiezan en valor 0 (no hay nadie cruzando)
Consideremos las ejecuciones del monitor (de principio a fin):

*) can_enter_c0, can_enter_c1 y can_enter_ped: son funciones que no modifican c0, c1 ni p, luego no cambian I, sigue cumpliéndose.

*) wants_enter_car: en esta función se puede incrementar c0, pero en ese caso el programa se asegura de que $c1 = p = 0$. Análogo con c1.

*) leaves_car: c0 o c1 se podrían reducir, pero esto no hace que I no se cumpla, se sigue cumpliendo.

*) wants_enter_ped: se incrementa p, pero el programa se asegura de que $c0 = c1 = 0$

*) leaves_ped: se reduce p, pero esto no hace que I no se cumpla, se sigue cumpliendo I.

Veamos que pasa en los notify:

*) leaves_car (notify): se podría desbloquear un peatón ($p>0$) o un coche en dirección contraria, pero se asegura de que esto sólo ocurre si los coches en la dirección que está son 0.

*) leaves_ped (notify): podría desbloquear un coche ($c0 \circ c1 > 0$) pero el programa se asegura de que esto ocurre sólo si $p=0$, y como un peatón ha cruzado se tiene que $c0=0$ y $c1=0$.

Luego, el invariante se cumple durante todo el programa.