

Programa con paralela practica 2 demostración

Invariante:

Sea $hcoches$, N el número de ~~coches~~ coches de dirección Norte dentro del puente, $hcoches$, S los coches con dirección Sur dentro del puente y $hpers$ el número de personas en el puente el invariante es (usando CN como $hcoches$, N ; CS y P como $hcoches$, S y $hpers$)

$$hcoches \wedge (CS \geq 0 \wedge CN \geq 0 \wedge P \geq 0 \wedge (CS \geq 0 \rightarrow CN + P = 0) \wedge$$

$$\wedge (CN \geq 0 \rightarrow CS + P = 0) \wedge (P \geq 0 \rightarrow CN + CS = 0))$$

Monitor CP

int $CS \leftarrow 0$

int $CN \leftarrow 0$

int $P \leftarrow 0$

Condition $entra.coche.S$

Condition $entra.coche.N$

Condition $entra.pers$

lock mutex

operation leaves car (dir: ^{direction: int} ~~int~~)

if $direction == 1$: (Sur)

$CS++$

if $CS == 0$:

signal($entra.coche.N$)

signal($entra.pers$)

signal($entra.coche.S$)

else: ~~ad~~

$CN++$

if $CN == 0$

signal($entra.pers$)

signal($entra.coche.S$)

signal($entra.coche.N$)

operation ~~leaves~~ leaves pedestrian $ent.pers()$

$P++$

if $P == 0$:

signal($entra.coche.S$)

signal($entra.coche.N$)

signal($entra.pers$)

operation ~~wait~~ waits enter pedestrian $ent.pers()$

$entra.pers.wait(CN + CS == 0)$

$P++$

operation ~~wait~~ waits enter car (direction: ^{int} ~~int~~)

if $direction == 1$: (Sur)

$entra.coche.S.wait(CN + P == 0)$

$CS++$

else

$entra.coche.N.wait(CS + P == 0)$

$CN++$

iniciamente
- Al invariante se cumple puesto
que empezamos con $CS + CN + P = 0$

Comprobamos las señales

Consideramos las operaciones del lockito y comprobamos que el invariante se cumple (no hay coches y peatones al mismo tiempo)

- wants_enter_car: Si la dirección es 1 (sur) entonces aumenta CS pero espera a que $CN + P = 0$ por lo que se cumple ($CS > 0 \wedge CN + P = 0$). En otro caso (Norte) aumenta CN pero espera a que $CS + P = 0$ por lo que se cumple ($CN > 0 \wedge CS + P = 0$)
- leave_car: reduce CS (si dirección es sur) y reduce CN (si dirección es norte) pero ocurre si $CS > 0$ o $CN > 0$ respectivamente por lo que se cumple tras la operación que $CS > 0 \vee CN > 0$ por lo que cumple el invariante
- wants_enter_pedestrian: aumenta P pero espera a que $CN + CS = 0$ por lo que se cumple ($P > 0 \wedge CS + CN = 0$)
- leave_pedestrian: disminuye P pero ocurre cuando $P > 0$ por lo que cumple tras la operación que $P > 0$

Deadlocks

$$\begin{aligned} P > 0 &\Rightarrow CN + CS = 0 \\ CS > 0 &\Rightarrow CN + P = 0 \\ CN > 0 &\Rightarrow CS + P = 0 \end{aligned}$$

que el invariante se mantenga implica la ausencia de deadlocks puesto que la única manera de que alguno sea > 0 sucede solo si el resto son 0 y si son todos 0 entonces se permite la entrada de peatones o coches

Comprobamos las señales

- $\text{Signal}(\text{entra-coche-N})$ en leve el car (si dirección es Sur)
 $\text{Signal}(\text{entra-pers})$ si esta en dirección Sur es que $\text{CN} + \text{P} = 0$
 $\text{Signal}(\text{entra-coche-S})$ y PS-Signal lo hace si $\text{CS} = 0$ por lo que aseguramos que se cumple el invariante
- $\text{Signal}(\text{entra-pers})$ en leve el car (si dirección es Norte)
 $\text{Signal}(\text{entra-coche-S})$ si está en dirección Norte es que $\text{CS} + \text{P} = 0$
 $\text{Signal}(\text{entra-coche-N})$ y lo Signal lo hace si $\text{CN} = 0$ por lo que aseguramos que se cumple el invariante
- $\text{Signal}(\text{entra-coche-S})$ en leve el peatón por lo que
 $\text{Signal}(\text{entra-coche-N})$ $\text{CN} + \text{CS} = 0$ y lo Signal lo hace si $\text{P} = 0$
 $\text{Signal}(\text{entra-pers})$ por lo que aseguramos que se cumple el invariante

IV Ausencia de deadlocks

$\text{P} > 0 \rightarrow \text{CN} + \text{CS} = 0$ y eventualmente $\text{P} = 0 \rightarrow \text{Signal}(\text{a todo})$
 $\text{CS} > 0 \rightarrow \text{CN} + \text{P} = 0$ y eventualmente $\text{CS} = 0 \rightarrow \text{Signal}(\text{a todo})$
 $\text{CN} > 0 \rightarrow \text{CS} + \text{P} = 0$ y eventualmente $\text{CN} = 0 \rightarrow \text{Signal}(\text{a todo})$

Si alguno de $(\text{P}, \text{CS}, \text{CN}) > 0$ implica que los otros son 0 pero según vaya saliendo el coche o el peatón - para $\text{P} = 0$ o $\text{CS} = 0$ o $\text{CN} = 0$ y entonces avisará a los demás para que puedan entrar

① transición ausencia de llamada

- si un peer está esperando en `entra-peer` es que
o bien `CN > 0` o `CS > 0`, eventualmente `CN` o `CS` serán 0 ~~pero~~
y entonces habrá un `signal(entra-peer)` ~~que~~
- si un `code_N` está esperando en `entra-code-n` es que
o bien `CS > 0` o `P > 0` según vaya saliendo eventualmente `CS` o
`P` serán 0 y entonces habrá un `signal(entra-code-n)`
- si un `code_S` está esperando en `entra-code-s` es que
o bien `CN > 0` o `P > 0` según vaya saliendo eventualmente
`CN` o `P` serán 0 y entonces habrá un `signal(entra-code-s)`