Ejercicio Propuesto 2 SCD

José Alberto Hoces Castro

24 de octubre 2022

Monitor Semaforo;

  var na, np, nv : int

  c : cond;

procedure P;

begin

$\{n_p \leq n_a, n_p \leq n_v,$
$np \geq \min(n_a, n_v)\}$

na := na + 1;

$\{n_p \leq n_a - 1, n_p \leq n_v,$
$n_p \geq \min(n_a - 1, n_v)\}$

if (na > nv)

$\{n_p \leq n_a - 1, n_p \leq n_v,$
$n_p \geq \min(n_a, n_v)\}$

$\Downarrow$

$\{n_p \leq n_a, n_p \leq n_v,$
$n_p \geq \min(n_a, n_v)\}$

then c.wait();

$\{n_p < n_a, n_p = n_v - 1\}$

$\Downarrow$ Cond. sincronización

$\{n_p \leq n_a - 1, n_p \leq n_v - 1,$
$n_p \geq \min(n_a - 1, n_v - 1)\}$

np := np + 1;

$\{n_p \leq n_a, n_p \leq n_v,$
$n_p \geq \min(n_a, n_v)\}$

end;

procedure V;

begin

$\{n_p \leq n_a, n_p \leq n_v,$
$np \geq \min(n_a, n_v)\}$

nv := nv + 1;

$\{n_p \leq n_a, n_p \leq n_v - 1,$
$np \geq \min(n_a, n_v - 1)\}$

if (na > np)

Cond. sincronización $\rightarrow \{n_p < n_a, n_p = n_v - 1\}$

then c.signal;

$\{n_p \leq n_a, n_p \leq n_v,$
$n_p \geq \min(n_a, n_v)\}$

end;

begin

$\{true\}$

na := 0;

nv := 0;

np := 0;

$\{n_p \leq n_a, n_p \leq n_v,$
$n_p \geq \min(n_a, n_v)\}$

end;

1 Justificación

El invariante del monitor IM se compone de las tres condiciones que aparecen en el enunciado del ejercicio, pues se verifican en el código de inicialización del monitor: $n_p \leq n_a, n_p \leq n_v$ y $n_p \geq min(n_a, n_v)$. Se verifica cada vez que termina cualquiera de los procedimientos del monitor.

Comenzamos analizando el procedimiento V. Al principio este se verifica el IM. Una vez ejecutada la sentencia $n_v = n_v+1$, usamos el axioma de asignación sustituyendo n_v por $n_v - 1$ y así obtenemos la postcondición $n_p \leq n_a, n_p \leq n_v - 1$ y $n_p \geq min(n_a, n_v - 1)$. En caso de verificarse la condición del if, es decir, $n_a > n_p$, por lo que de $n_a \geq min(n_a, n_v - 1)$ y de $n_p \leq n_v - 1$ se deduce que $n_p = n_v - 1$. Los asertos $n_a > n_p$ y $n_p = n_v - 1$ formarán la condición de sincronización. Aplicando el axioma del $c.signal()$, obtenemos la postcondición, que coincide con el IM.
Pasamos al procedimiento P, donde tras la primera sentencia $n_a = n_a + 1$, aplicando el axioma de asignación sustituyendo n_a por $n_a - 1$ se obtiene la postcondición de la imagen anterior.

A continuación, en caso de cumplirse la condición del if, tendríamos que $n_a - 1 \geq n_v$, por lo que $n_p \geq min(n_a - 1, n_v)$, de donde se deduce que también es cierto $n_p \geq min(n_a, n_v)$ (esto se debe a que $n_p \geq n_a - 1$ implica que $n_p \geq n_a$. Por ello, se verifica el invariante del monitor. Volveremos para seguir ejecutando el procedimiento cuando se dé la condición de sincronización. Una vez ejecutado $c.signal()$ se aplica el axioma de $c.wait()$ en el procedimiento P, obteniéndose así la condición de sincronización después del $c.wait()$, que es $n_p \leq n_a - 1$, $n_p \leq n_v - 1$ y $n_p \geq min(n_a - 1, n_v - 1)$. La última sentencia del procedimiento P incrementa en una unidad la variable n_p, volviendo a ser cierto el invariante del monitor.

En cuanto a la precondición y postcondición del código de inicialización del monitor, solo se ha usado el axioma de inicialización donde se pone como precondición $true$ y como postcondición el invariante del monitor.

Como último comentario para terminar de demostrar la corrección del monitor, las condiciones iniciales permiten que siempre se cumplan las condiciones de los bloques if que hay en los procedimientos P y Q. Sin ellas, el IM podría no verificarse al terminar la ejecución de los procedimientos y resultar en un programa incorrecto.