DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 1: Especificación y wp

”Fondo Monetario Común”

19 de mayo de 2024 — Algoritmos y Estructuras de Datos

UshuaiensesPorteños

Integrante	LU	Correo electrónico
Venezia, Joaquin	1338/23	joaquinvenezia@gmail.com
Vivanco, Esteban	682/23	vivancoesteban17@gmail.com
Briccola, Francina	1582/21	franbriccola2309@gmail.com
Barrios, Leandro	831/23	barrioslean2002@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

1. Especificación del juego

1.1. redistribucionDeLosFrutos:

proc `redistribucionDeLosFrutos` (in *recursos*: $seq\langle\mathbb{R}\rangle$, in *cooperan*: $seq\langle \text{Bool}\rangle$) : $seq\langle\mathbb{R}\rangle$

requiere $\{|recursos| > 0 \land |recursos| = |cooperan| \land (elementosPositivos(recursos))\}$

asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |recursos|) \longrightarrow_L res[i] =$ if $cooperan[i] = True$ then $recursosDivididos(recursos, cooperan$ else $recursos[i] + recursosDivididos(recursos, cooperan)$ fi$\}$

asegura $\{|res| = |recursos|\}$

aux `recursosDivididos` (in *recursos*: $seq\langle\mathbb{R}\rangle$, in *cooperan*: $seq\langle \text{Bool}\rangle$) : $\mathbb{R}$ $= ((\sum_{i=0}^{|recursos|-1}$ if $cooperan[i]$ then $recursos[i]$ else 0 fi$)/|recursos|)$;

1.2. trayectoriaDeLosFrutosIndividualesALargoPlazo:

proc `trayectoriaDeLosFrutosIndividualesALargoPlazo` (inout *trayectorias*: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in *cooperan*: $seq\langle \text{Bool}\rangle$, in *apuestas*: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in *pagos*: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in *eventos*: $seq\langle seq\langle\mathbb{N}\rangle\rangle$) : $seq\langle\mathbb{R}\rangle$

requiere $\{trayectorias = trayectorias_0\}$

requiere $\{|apuestas| = |trayectorias| = |pagos| = |cooperan| = |eventos| \land |eventos| > 0\}$

requiere $\{(\forall i : \mathbb{Z})(0 \leq i < |trayectorias_0| \longrightarrow_L trayectorias_0[i][0] > 0) \land (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < |pagos| \land 0 \leq j < |pagos[i]| \longrightarrow_L pagos[i][j] > 0) \land (\forall i : \mathbb{Z})(0 \leq i < |apuestas| \longrightarrow_L esApuesta(apuestas[i], apuestas))\}$

requiere $\{(\forall i : \mathbb{Z})(0 \leq i < |pagos| \longrightarrow_L |pagos[i]| = |apuesta[i]|)\}$

requiere $\{(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < |eventos| \land 0 \leq j < |eventos[i]| \longrightarrow_L 0 \leq eventos[i][j] \leq |pagos[i]|)\}$

asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |trayectorias| \longrightarrow_L |trayectorias[i]| = |eventos[i]| + 1)\}$

asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |trayectorias| \longrightarrow_L trayectorias[i][1] =$ if $cooperan[i] = True$ then $calcularFondoComun(apuestas, pagos, cooperan, trayectorias_0, 1)$ else $trayectorias_0[i][0] + calcularFondoComun(apuestas, pagos, cooperan, trayectorias_0, 1)$ fi$\}$

asegura $\{(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < |trayectorias| \land 2 \leq j < |trayectorias[i]|) \longrightarrow_L trayectorias[i][j] =$ if $cooperan[i] = True$ then $calcularFondoComun(apuestas, pagos, cooperan, trayectorias, j)$ else $trayectorias[i][j-1] + calcularFondoComun(apuestas, pagos, cooperan, trayectorias, j)$ fi$\}$

1.3. trayectoriaExtrañaEscalera:

proc `trayectoriaExtrañaEscalera` (in *trayectoria*: $seq\langle\mathbb{R}\rangle$) : Bool

requiere $\{|trayectoria| > 0\}$

asegura $\{res = True \longleftrightarrow (|trayectoria| = 1 \lor (|trayectoria| = 2 \land_L trayectoria[0] \neq trayectoria[1]) \lor (\exists i : \mathbb{Z})(0 \leq i < |trayectoria| \land esMaxLocal(trayectoria, i) \land (\forall j : \mathbb{Z})(0 \leq j < |trayectoria| \land j \neq i \longrightarrow_L \neg esMaxLocal(trayectoria, j)))\}$

aux `esMaxLocal` (in *trayectoria*: $seq\langle\mathbb{R}\rangle$, in i: $\mathbb{Z}$) : Bool $= (i = 0 \land trayectoria[0] > trayectoria[1]) \lor (i = |trayectoria| - 1 \land trayectoria[i] > trayectoria[i-1]) \lor (i \neq 0 \land i \neq |trayectoria| - 1 \land (trayectoria[i] > trayectoria[i+1] \land trayectoria[i] > trayectoria[i-1])$;

1.4. individuoDecideSiCooperarONo:

proc `individuoDecideSiCooperarONo` (in *individuo*: $\mathbb{N}$, in *recursos*: $seq\langle\mathbb{R}\rangle$, inout *cooperan*: $seq\langle \text{Bool}\rangle$, in *apuestas*: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in *pagos*: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in *eventos*: $seq\langle seq\langle\mathbb{N}\rangle\rangle$) :

requiere $\{0 \leq individuo < |eventos| \land (\forall i : \mathbb{Z})(0 \leq i < |apuestas| \longrightarrow_L esApuesta(apuestas[i], apuestas))\}$

requiere $\{cooperan = cooperan_0\}$

requiere $\{|pagos| = |recursos| = |enventos| = |cooperan|\}$

requiere $\{(\forall e : \mathbb{Z})(0 \leq e < |recursos| \longrightarrow_L recursos[e] > 0 \land pagos[e] > 0)\}$

requiere $\{(\forall i : \mathbb{Z})(0 \leq i < |pagos| \longrightarrow_L |pagos[i]| = |apuesta[i]|)\}$

requiere $\{(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < |eventos| \land 0 \leq j < |eventos[i]| \longrightarrow_L 0 \leq eventos[i][j] \leq |pagos[i]|)\}$

asegura $\{cooperan[individuo] = True \longleftrightarrow (\exists T : seq\langle seq\langle\mathbb{R}\rangle\rangle)(\exists s : seq\langle\mathbb{R}\rangle)(\exists l : seq\langle\mathbb{R}\rangle)$
$listaDeTrayectoriasValida(T, eventos, cooperan_0, apuestas, pagos)$
$\land\, esTrayectoria(individuo, s, T, setAt(cooperan_0, individuo, True), pagos, eventos, apuestas)$
$\land\, esTrayectoria(individuo, l, T, setAt(cooperan_0, individuo, False), pagos, eventos, apuestas)$
$\land\, s[|s|-1] \geq l[|l|-1])\}$

1.5. individuoActualizaApuesta:

`proc individuoActualizaApuesta` (in $individuo$: $\mathbb{N}$, in $recursos$: $seq\langle\mathbb{R}\rangle$, in $cooperan$: $seq\langle\text{Bool}\rangle$, inout $apuestas$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in $pagos$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in $eventos$: $seq\langle seq\langle\mathbb{N}\rangle\rangle$) :

`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |apuestas| \longrightarrow_L esApuesta(apuestas[i], apuestas)) \land 0 \leq individuo < |eventos|\}$

`requiere` $\{apuestas = apuestas_0\}$

`requiere` $\{|pagos| = |recursos| = |enventos| = |cooperan|\}$

`requiere` $\{(\forall e : \mathbb{Z})(0 \leq e < |recursos| \longrightarrow_L recursos[e] > 0 \land pagos[e] > 0)\}$

`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |pagos| \longrightarrow_L |pagos[i]| = |apuesta[i]|)\}$

`requiere` $\{(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < |eventos| \land 0 \leq j < |eventos[i]| \longrightarrow_L 0 \leq eventos[i][j] \leq |pagos[i]|)\}$

`asegura` $\{(\exists T : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ ((\exists s : seq\langle\mathbb{R}\rangle)\ ((\exists apuestaOptima : seq\langle\mathbb{R}\rangle)\ ($
$listaDeTrayectoriasValida(T, eventos, cooperan, apuestas, pagos) \land$
$esApuesta(apuestaOptima, apuestas) \land$
$esTrayectoria(individuo, s, T, cooperan, pagos, eventos, setAt(apuestas_0, individuo, apuestaOptima)) \land$
$(\forall l : seq\langle\mathbb{R}\rangle)(\forall apuestaInferior : seq\langle\mathbb{R}\rangle)(esApuesta(apuestaInferior, apuestas) \land$
$esTrayectoria(individuo, l, T, cooperan, pagos, eventos, setAt(apuestas_0, individuo, apuestaInferior)) \land$
$apuestaOptima \neq apuestaInferior \longrightarrow_L s[|s| - 1] \geq l[|l| - 1] \land$
$apuestas[individuo] = apuestaOptima)))\}$

1.6. Auxiliares:

`aux calcularFondoComun` (in $apuestas$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in $pagos$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in $cooperan$: $seq\langle\text{Bool}\rangle$, in $trayectorias$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in j: $\mathbb{Z}$) : $\mathbb{R}$ $= (\sum_{i=0}^{|cooperan|-1}$ if $cooperan[i] = True$ then $trayectorias[i][j-1] * apuestas[i][eventos[i][j-1]] * pagos[i][eventos[i][j-1]]$ else 0 fi$)/\ |eventos|$;

1.7. Predicados:

`pred esTrayectoria` ($individuo$: $\mathbb{N}$, $trayectoria$: $seq\langle\mathbb{R}\rangle$, $trayectorias$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, $cooperan$: $seq\langle\text{Bool}\rangle$, $pagos$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, $eventos$: $seq\langle seq\langle\mathbb{N}\rangle\rangle$, $apuestas$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {

$(\forall i : \mathbb{Z})(1 \leq i < |trayectoria|) \longrightarrow_L trayectoria[i] =$
if $cooperan[individuo] = True$ then $calcularFondoComun(apuestas, pagos, cooperan, trayectorias, individuo)$ else $trayectori$
$|trayectoria| = |eventos[individuo]| + 1 \land trayectoria[0] = trayectorias[individuo][0]$

}

.

`pred listaDeTrayectoriasValida` (in T: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in $cooperan$: $seq\langle\text{Bool}\rangle$, inout $apuestas$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in $pagos$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in $eventos$: $seq\langle seq\langle\mathbb{N}\rangle\rangle$) {

$|T| = |eventos| \land (\forall i : \mathbb{Z})(0 \leq i < |T|) \longrightarrow_L$
$esTrayectoria(individuo, T[i], T, cooperan, pagos, eventos, apuestas)$

}

`pred esApuesta` ($apuesta$: $seq\langle\mathbb{R}\rangle$, $apuestas$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {

$(\forall i : \mathbb{Z})(0 \leq i < |apuestas| \longrightarrow_L |apuesta| = |apuestas[i]| \land (\forall i : \mathbb{Z})(0 \leq i < |apuesta| \longrightarrow_L apuesta[i] > 0)$
$\sum_{i=0}^{|apuesta|-1} apuesta[i] = 1)$

}

(apuesta representa un elemento de apuestas, i.e. es la lista que representa cuánto apuesta un individuo a todos los eventos posibles, por lo que |apuesta| es la cantidad de eventos posibles)

`pred elementosPositivos` ($recursos$: $seq\langle\mathbb{R}\rangle$) {

$(\forall i : \mathbb{Z})(0 \leq i < |recursos| \longrightarrow_L recursos[i] > 0)$

}

2. Correctitud entre implementación y especificación:

Para demostrar que la especificación de la función **frutoDelTrabajoPuramenteIndividual** es correcta respecto de su implementación tenemos que probar que:

- (1) $\text{P}_c \longrightarrow I$
- (2) $\{\text{I} \land B\}S\{I\}$
- (3) $\{\text{I} \land \neg B\} \longrightarrow \{Q_c\}$
- (4) $\{\text{I} \land B \land v_0 = f_v\}S\{f_v < v_0\}$
- (5) $\{\text{I} \land f_v \leq 0 \longrightarrow \neg B\}$

Primero definimos la precondición, la guarda, la postcondición y el invariante:

P$_c = \{i = 0 \wedge res = recursos\}$
B = { i <|eventos|}
Q$_c = \{res = recursos(apuesta_c pago_c)^{\#aparicionesariciones(eventos,F)}(apuesta_s pago_s)^{\#aparicionesariciones(eventos,T)}\}$
I = $\{0 \leq i \leq |eventos| \wedge res =$
$recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i),T)} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i),F)}\}$
f$_v = \{|eventos| - i\}$

Vamos a usar el Teorema del Invariante, ya que estamos en presencia de un a tripla de Hoare con la forma {P} **while B do S1 else do S2 endwhile** {Q}

(1) P$_c \longrightarrow I$

$\{i = 0 \wedge res = recursos\} \longrightarrow \{0 \leq i \leq |eventos| \wedge res = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i),T)} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i),F)}\}$
Luego i=0 por lo que $0 \leq i \leq |eventos|$, entonces:
y como res = recursos, queda:
recursos = recursos * (apuesta$_c pago_c)^{\#apariciones(subseq(eventos,0,0),T)} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,0),F)}$
Finalmente
$\#aparicionesariciones(<>, T) = aparicionesariciones(<>, F) = 0$
Luego
$recursos = recursos$
(2) {I $\wedge B\}S\{I\}$

Buscamos, como en el anterior punto que la precondición implique la wp(S,Q). En este caso sería:
{I$\wedge B\} \longrightarrow wp(S, I)$

Como S = if $eventos[i] = True$ then $res := (res * apuesta_c) * pago_c$ else $res := (res * apuesta_s) * pago_s$ fi
Vamos definir los dos casos de S:
S$_t = res := (res * apuesta_c) * pago_c$
S$_f = res := (res * apuesta_s) * pago_s$

wp(S, I) $\equiv wp($if $eventos[i] = True$ then S_t else S_f fi$; i := i + 1, I) \equiv$
wp(if $eventos[i] = True$ then S_t else S_f fi,wp(i:=i+1;I)))

Empezando desde adentro con el primer wp(i := i + 1, I) nos queda:
wp(i := i +1, $\{0 \leq i \leq |eventos| \wedge res = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i),T)}$
$* (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i),F)}\}$
Usando el Axioma 2:
$\equiv 0 \leq i + 1 \leq |eventos| \wedge res = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}$
$* (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)}$

Ahora sigamos con el otro:

wp(if $eventos[i] = True$ then S_t else S_f fi, wp(i:=i+1;I))

Usando lo que resolvimos en el paso anterior y usando el Axioma 3 nos queda:

$\equiv 0 \leq i + 1 \leq |eventos| \wedge_L (eventos[i] = True \wedge wp(res := res * apuesta_c * pago_c, 0 \leq i + 1 \leq |eventos| \wedge res = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}$
$* (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)}) \vee (eventos[i] = False \wedge wp(res := res * apuesta_s * pago_s, 0 \leq i + 1 \leq |eventos| \wedge res = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}$
$* (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$

Ahora aplicamos el Axioma 3 y reemplazamos el valor de res:

$\equiv 0 \leq i + 1 \leq |eventos| \wedge_L (eventos[i] = True \wedge 0 \leq i + 1 \leq |eventos| \wedge res * apuesta_c * pago_c = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}$
$* (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)}) \vee (eventos[i] = False \wedge 0 \leq i + 1 \leq |eventos| \wedge res * apuesta_s * pago_s = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}$
$* (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$

Ahora hacemos un poco de algebra para reescribir el valor de res:
$\equiv 0 \leq i+1 \leq |eventos| \wedge_L (eventos[i] = True \wedge 0 \leq i+1 \leq |eventos| \wedge res = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i+1),True)-1} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i+1),False)}) \vee (eventos[i] = False \wedge 0 \leq i+1 \leq |eventos| \wedge res = recursos * (apuesta_c pago_c)^{\#apariciones(subseq(eventos,0,i+1),True)} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i+1),False)-1})$

Como $0 \leq i+1 \leq |eventos|$
en #apariciones(subseq(eventos,0,i+1), True) esto va a ser lo mismo que
#apariciones(subseq(eventos,0,i), True) si 0 <i <|eventos|.
Ese -1 que se ve en la cantidad de apariciones hace referencia a dividir ambos términos por apuesta$_c * pago_c$
En conclusión esto es lo mismo que hacer la cuenta con la subseq que va de 0 a i sin saber a priori si eventos[i] es True o False, es decir que el valor de res queda:
res = recursos * (apuesta$_c pago_c)^{\#apariciones(subseq(eventos,0,i),True)} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i),False)}$
Mientas que $0 \leq i+1 \leq |eventos|$
Esto es exactamente el invariante excepto por la parte de $0 \leq i \leq |eventos|$
que implica a $0 \leq i+1 \leq |eventos|$,
por lo tanto {I$\wedge B\} \longrightarrow wp(S, I)$

(3) {I $\wedge \neg B\} \longrightarrow \{Q_c\}$

Ahora queremos ver si efectivamente {I $\wedge \neg B\} \longrightarrow Q_c$:
Con el Invariante y la negacion de la guarda en la parte i seria:
$0 \leq i \leq |eventos| \wedge i \geq |eventos|$
Luego nuestro res queda:
res = recursos * (apuesta$_c pago_c)^{\#apariciones(subseq(eventos,0,i),T)} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i),F)}$
Juntando las dos informaciones nos queda que i = |eventos|
Con este dato, reemplanzando en res nos queda:
res = recursos * (apuesta$_c pago_c)^{\#apariciones(subseq(eventos,0,|eventos|),T)} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,|eventos|),F)}$
Entonces:
res = recursos * (apuesta$_c pago_c)^{\#apariciones(eventos,T)} * (apuesta_s pago_s)^{\#apariciones(eventos,F)}$
Ya que subseq(eventos, 0 , |eventos|) = eventos
Por lo tanto res nos queda:
res = recursos * (apuesta$_c pago_c)^{\#apariciones(eventos,T)} * (apuesta_s pago_s)^{\#apariciones(eventos,F)}$
Lo cual es exactamente Q$_c$, por lo tanto queda probado que$(I \wedge \neg B) \longrightarrow Q_c$
Con esto ya probamos que el ciclo es parcialmente correcto, falta probar su finalilzación:

(4) {I $\wedge B \wedge v_0 = f_v\} S \{f_v < v_0\}$

Para probar esto tenemos que probar que {I $\wedge B \wedge v_0 = f_v\} \longrightarrow wp(S, f_v < v_0)$
Con el Invariante y la guarda en la parte i seria:
$0 \leq i \leq |eventos| \wedge i < |eventos|$
Luego nuestro res queda:
res = recursos * (apuesta$_c pago_c)^{\#apariciones(subseq(eventos,0,i),T)} * (apuesta_s pago_s)^{\#apariciones(subseq(eventos,0,i),F)}$
Mientras que nuestra f$_v = |eventos| - i$
La wp(S, f$_v < v_0) = wp(i := i+1, wp(recursos[i] = True \wedge res := res * apuesta_c * pago_c \vee recursos[i] = False \wedge res := res * apuesta_s * pago_s, f_v < v_0)$
Luego como no existe variable res en la Q donde queremos usar el Axioma 1 nos queda Q.
Por lo tanto nos queda wp(i := i + 1, f$_v < v_0$)
Como f$_v = |eventos| - i$ al usar el Axioma 1 nos termina quedando que:
wp(i:= i + 1, f$_v < v_0) \equiv |eventos| - (i+1) < v_0$
$\equiv |eventos| - i - 1 < v_0$
Y como al principio vimos que f$_v = v_0$, termina quedándonos que
v$_0 = |eventos| - i$
Entonces |eventos|- i - 1 <|eventos|- i, lo cual es trivialmente cierto.
Se puede ver que luego del programa f$_0$es menor a$v_0$por 1.

(5) {I $\wedge f_v \leq 0 \longrightarrow \neg B\}$

Queremos ver que {I $\wedge f_v \leq 0 \longrightarrow \neg B\}$:
I $\wedge |eventos| - i \leq 0 \longrightarrow \neg(i < |eventos|)$
como I nos dice que $0 \leq i \leq |eventos|$
al unir las dos condiciones nos queda que i = |eventos|
ya que (eventos $\leq i) \wedge (0 \leq i \leq |eventos|) \longleftrightarrow i = |eventos|$
ahora con i = |eventos|vemos que:

i = |eventos|$\longrightarrow i \geq |eventos|$
esto es trivialmente verdadero y por eso se puede ver que no se cumple la guarda.