DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 1: Especificación y WP

21 de abril de 2024 — Algoritmos y Estructuras de Datos (Algo2)

AlgoCrafters

Integrante	LU	Correo electrónico
Calobini, Luciano	1352/23	`lcalobini@dc.uba.ar`
Carriedo, Eliseo	392/23	`carriedo.eliseo@gmail.com`
Piaggio, Joaquin	700/22	`joacopiaggio@hotmail.com`
Albarracín, Joaquín Ignacio	1013/22	`joacoalbarracin@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

1.1. redistribucionDeLosFrutos

Calcula los recursos que obtiene cada uno de los individuos luego de que se redistribuyen los recursos del fondo monetario común en partes iguales. El fondo monetario común se compone de la suma de *recursos* iniciales aportados por todas las personas que *cooperan*. La salida es la lista de recursos que tendrá cada jugador.

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

`requiere` $\{|recursos| = |cooperan| \ \land \ |cooperan| > 0\}$

`asegura` $\{|recursos| = |res| \ \land \ (\forall i : \mathbb{Z}) \ ((0 \leq i < |recursos|) \longrightarrow_L$
$((cooperan[i] \ \land \ res[i] = parteFondo(recursos, cooperan)) \lor$
$(\neg(cooperan[i]) \ \land \ res[i] = recursos[i] + parteFondo(recursos, cooperan))))\}$

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

En este ejecicio, la idea es asegurarnos que:

Para cada individuo: si coopera, sus recursos serán su parte del fondo monetario, y si no coopera, sus recursos serán la suma de su parte del fondo monetario más sus recursos anteriores.

1.2. trayectoriaDeLosFrutosIndividualesALargoPlazo

Actualiza (In/Out) la lista de *trayectorias* de los los recursos de cada uno de los individuos. Inicialmente, cada una de las trayectorias (listas de recursos) contiene un único elemento que representa los recursos iniciales del individuo. El procedimiento agrega a las *trayectorias* los recursos que los individuos van obteniendo a medida que se van produciendo los resultados de los *eventos* en función de la lista de *pagos* que le ofrece la naturaleza (o casa de apuestas) a cada uno de los individuos, las *apuestas* (o inversiones) que realizan los individuos en cada paso temporal, y la lista de individuos que *cooperan* aportando al fondo monetario común.

`proc trayectoriaDeLosFrutosIndividualesALargoPlazo` (inout trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in cooperan: $seq\langle\mathsf{Bool}\rangle$, 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` $\{|trayectorias| > 0 \ \wedge\ |trayectorias| = |cooperan| = |apuestas| = |pagos| = |eventos|\}$
`requiere` $\{(\forall i:\mathbb{Z})(1 \le i < |apuestas| \longrightarrow_L |apuestas[i-1]| = |apuestas[i]| = |pagos[i-1]| = |pagos[i]| > 0\}$
`requiere` $\{(\forall i:\mathbb{Z})(0 \le i < |trayectoria| \longrightarrow_L |trayectoria[i]| = 1 \ \wedge\ trayectoria[i][0] > 0\}$
`requiere` $\{apuestasValidas(apuestas) \ \wedge\ pagosValidos(pagos) \ \wedge$
$eventosValidos(eventos, apuestas, pagos)\}$
`asegura` $\{|trayectorias| = |old(trayectorias)| \ \wedge\ (\forall i:\mathbb{Z})(0 \le i < |trayectorias| \longrightarrow_L$
$|trayectorias[i]| = |eventos[i]| + 1 \ \wedge\ trayectorias[i][0] = old(trayectorias)[i][0]\}$
`asegura` $\{(\forall i,t:\mathbb{Z})(0 \le i < |trayectorias| \ \wedge\ 0 < t < |trayectorias[i]|) \longrightarrow_L$
$trayectorias[i][t] = recursoTiempoT(trayectorias, cooperan, apuestas, pagos, eventos, i, t)\}$

`aux recursoTiempoT` (trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{N}\rangle\rangle$, i: $\mathbb{Z}$, t: $\mathbb{Z}$) : $\mathbb{R}$ =
if $cooperan[i]$ then $parteFondo(trayectorias, cooperan, apuestas, pagos, eventos, t)$
else $ganancia(trayectorias, apuestas, pagos, eventos, i, t) +$
$parteFondo(trayectorias, cooperan, apuestas, pagos, eventos, t)$ fi ;

`aux parteFondo` (trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{N}\rangle\rangle$, t: $\mathbb{Z}$) : $\mathbb{R}$ =
$(\sum_{i=0}^{|trayectorias|-1}$ if $cooperan[i]$ then $ganancia(trayectorias, apuestas, pagos, eventos, i, t)$ else 0 fi$)/|cooperan|$;

`aux ganancia` (trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{N}\rangle\rangle$, i: $\mathbb{Z}$, t: $\mathbb{Z}$) : $\mathbb{R}$ = $trayectorias[i][t-1] * apuestas[i][eventos[i][t-1]] * pagos[i][eventos[i][t-1]]$;

/*las apuestas deben ser fracciones de recursos y la suma de ellas debe ser el total del recurso del individuo*/
`pred apuestasValidas` (apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i,j:\mathbb{Z})(0 \le i < |apuestas| \wedge_L 0 \le j < |apuestas[i]|) \longrightarrow_L (0 \le apuestas[i][j] \le 1 \ \wedge$
$(\sum_{j=0}^{|apuestas[i]|-1} apuestas[i][j]) = 1)$
}

/*los pagos deben ser positivos*/
`pred pagosValidos` (pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i,j:\mathbb{Z})(0 \le i < |pagos| \wedge_L 0 \le j < |pagos[i]|) \longrightarrow_L (1 < pagos[i][j])$
}

/*debe corresponderle una apuesta y un pago a cada evento que ocurra*/
`pred eventosValidos` (apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$,pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$,eventos: $seq\langle seq\langle\mathbb{N}\rangle\rangle$) {
$(\forall i,j:\mathbb{Z})(0 \le i < |eventos| \wedge_L 0 \le j < |eventos[i]|) \longrightarrow_L$
$(|eventos[i]| > 0 \ \wedge\ 0 \le eventos[i][j] < |apuestas[i]| \ \wedge\ 0 \le eventos[i][j] < |pagos[i]|)$
}

En este ejercicio, la idea fue:

Determinar la longitud de trayectoria en la salida, y declarar que contiene cada uno de los indices dentro de las subsecuencias de trayectoria usando el auxiliar `recursoTiempoT`.

Dentro de `recursoTiempoT` utilizamos los predicados `parteFondo` y `ganancia` para determinar el valor del fondo monetario sobre la cantidad de individuos y del producto de los recursos, apuestas y pagos, en cada paso de tiempo t.

1.3. trayectoriaExtrañaEscalera

Esta función devuelve $True$ sii en la trayectoria de un individuo existe un único punto mayor a sus vecinos (llamado máximo local). Un elemento es máximo local si es mayor estricto que sus vecinos inmediatos.

`proc` `trayectoriaExtrañaEscalera` (in trayectoria : $\mathbb{R}$) : Bool

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

`asegura` $\{(res = \text{true}) \longleftrightarrow hayUnicoMaxLocal(trayectoria)\}$

`pred` `hayUnicoMaxLocal` (trayectoria: $seq\langle\mathbb{R}\rangle$) {

$(\exists\, i : \mathbb{Z})(0 \leq i < |trayectoria| \wedge_L esElMax(trayectoria, i) \ \wedge$
$(\forall j, k : \mathbb{Z})(0 \leq j < i < k < |trayectoria| \longrightarrow_L trayectoria[j] \leq trayectoria[j+1] \ \wedge$
$trayectoria[k-1] \geq trayectoria[k]))$

}

`pred` `esElMax` (s: $seq\langle\mathbb{R}\rangle$, i: $\mathbb{Z}$) {

$(\forall j : \mathbb{R})(0 \leq j < |s| \ \wedge \ j \neq i \longrightarrow_L s[i] > s[j]$

}

En este ejercicio, la idea fue:

Si existe un único maximo local dentro de la trayectoria de índice i, este debe ser mayor estricto que el resto de los elementos de la secuencia; esto lo aseguramos con el predicado `esElMax`. Además la secuencia debe cumplir que los elementos con indice entre el índice 0 e i, deben estar ordenados de manera creciente (no estricto), y que los elementos con índice entre i y la longitud de la secuencia (-1), deben estar ordenados de manera decreciente (no estricto).

1.4. individuoDecideSiCooperarONo

Un *individuo* actualiza su comportamiento cooperativo / no-cooperativo ($cooperan[individuo]$) en función de los *recursos* iniciales, de quienes *cooperan*, de los *pagos* que se le ofrecen a cada individuo, de las inversiones o *apuestas* de cada individuo, y del resultado los *eventos* que recibe cada individuo, eligiendo el comportamiento que maximiza sus recursos individuales luego de que ocurren todos los eventos.

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

`requiere` $\{0 \leq individuo < |cooperan|\}$
`requiere` $\{|cooperan| > 0 \ \land \ |cooperan| = |recursos| = |apuestas| = |pagos| = |eventos|\}$
`requiere` $\{|apuestas[0]| = |pagos[0]| \ \land \ |apuestas[0]| > 0 \ \land \ |eventos[0]| > 0\}$
`requiere` $\{(\forall i : \mathbb{Z})(1 \leq i < |apuestas| \longrightarrow_L |apuestas[i-1]| = |apuestas[i]|\}$
`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |recursos| \longrightarrow_L recursos[i] > 0\}$
`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |apuestas| \longrightarrow_L (\forall w : \mathbb{Z})(0 \leq w < |apuestas[i]| \longrightarrow_L$
$apuestas[i][w] \geq 0 \ \land \ pagos[i][w] \geq 1\}$
`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |eventos| \longrightarrow_L (\forall w : \mathbb{Z})(0 \leq w < |eventos[i]| \longrightarrow_L$
$0 \leq eventos[i][w] < |apuestas[i]|\}$
`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |apuestas| \longrightarrow_L (\sum_{k=0}^{|apuestas[i]|-1} apuestas[i][k] = 1))\}$

`asegura` $\{|cooperan| = |old(cooperan)| \ \land \ (\forall j : \mathbb{Z})(0 \leq j < |cooperan| \ \land \ j \neq individuo \longrightarrow_L$
$cooperan[j] = old(cooperan)[j]\}$
`asegura` $\{(\exists\ trayectoria : seq\langle seq\langle\mathbb{R}\rangle\rangle)$
$(esTrayectoriaCompleta(trayectoria, recursos, cooperan, apuestas, pagos, eventos) \ \land \ (\forall\ bools : seq\langle\mathsf{Bool}\rangle)$
$(|bools| = |cooperan| \ \land \ (\forall j : \mathbb{Z})(0 \leq j < |bools| \ \land \ j \neq individuo \longrightarrow_L$
$bools[j] = cooperan[j] \ \land \ bools[indiviuo] \neq cooperan[individuo]) \ \land \ (\forall\ otraTrayectoria : seq\langle seq\langle\mathbb{R}\rangle\rangle)$
$(esTrayectoriaCompleta(otraTrayecotoria, recursos, bools, apuestas, pagos, eventos) \longrightarrow_L$
$trayectoria[individuo][|eventos|] \geq otraTrayectoria[individuo][|eventos|]))))\}$

`pred` `esTrayectoriaCompleta` (trayectoria: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, recursos: $seq\langle\mathbb{R}\rangle$, cooperan: $seq\langle\mathbb{R}\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$|trayectoria| = |eventos| \ \land \ (\forall i : \mathbb{Z})(0 \leq i < |trayectoria| \longrightarrow_L |trayectoria[i]| = |eventos[i] + 1| \ \land$
$trayectoria[i][0] = recursos[i] \ \land \ (\forall j : \mathbb{Z})(1 \leq w < |trayectoria[i]| \longrightarrow_L$
$trayectoria[i][w] = recursoDelIndividuoEnW(trayectoria, cooperan, apuestas, pagos, eventos, i, w)$
}

`aux` `recursoDelIndividuoEnW` (trayectoria: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, i: $\mathbb{Z}$, w: $\mathbb{Z}$) : $\mathbb{R}$ =
if $cooperan[i] = \text{true}$ then $parteDelFondoEnW(trayectoria, cooperan, w)$ else
$trayectoria[i][w-1] * apuesta[i][evento[i][w-1]] * pago[i][evento[i][w-1]] +$
$parteDelFondoEnW(trayectoria, cooperan, apuestas, pagos, eventos, w)$ fi ;

`aux` `parteDelFondoEnW` (trayectoria: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, w: $\mathbb{Z}$) : $\mathbb{R}$ =
$(\sum_{i=0}^{|cooperan|-1}$ if $cooperan[i]$ then $trayectoria[i][w-1] * apuesta[i][evento[i][w-1]] * pago[i][evento[i][w-1]]$ else 0 fi)
$/\ |cooperan|$;

En este ejercicio, la idea es asegurarnos que:

Existe una *trayectoria* fabricada con recursos, **cooperan**, apuestas, pagos y eventos; que para toda *otraTrayectoria* fabricada con recursos, apuestas, pagos, eventos y cualquier otra secuencia de booleanos *bools* (que es exactamente igual a cooperan salvo en el indice de individuo, donde es diferente); tal que el último elemento de individuo en *trayectoria* es mayor igual al último elemento de individuo en *otraTrayectoria*.

Con el predicado `esTrayectoriaCompleta` determinamos todos los recursos de cada uno de los individuos en cada paso de tiempo, utilizando los recursos del paso de tiempo anterior. Además este predicado utiliza `parteDelFondoEnW` y `recursoDelIndividuoEnW` para determinar el valor del fondo monetario sobre la cantidad de individuos y del producto de los recursos, apuestas y pagos, en cada paso de tiempo w.

1.5. individuoActualizaApuesta

Un *individuo* actualiza su apuesta ($apuestas[individuo]$) en función de los *recursos* iniciales, de la lista de individuos que *cooperan*, de los *pagos* que se le ofrecen a cada individuo, de las inversiones o *apuestas* de cada individuo y del resultado los *eventos* que recibe cada individuo, eligiendo la apuesta que maximiza sus recursos individuales luego de que ocurren todos los eventos.

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

`requiere` $\{0 \leq individuo < |apuestas|\}$
`requiere` $\{|apuestas| > 0 \ \wedge \ |apuestas| = |recursos| = |cooperan| = |pagos| = |eventos|\}$
`requiere` $\{|cooperan[0]| = |recursos[0]|\}$
`requiere` $\{|apuestas[0]| = |pagos[0]| \ \wedge \ |apuestas[0]| > 0 \ \wedge \ |eventos[0]| > 0\}$
`requiere` $\{(\forall i : \mathbb{Z})(1 \leq i < |apuestas| \longrightarrow_L |apuestas[i-1]| = |apuestas[i]|\}$
`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |recursos| \longrightarrow_L recursos[i] > 0\}$
`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |apuestas| \longrightarrow_L (\forall w : \mathbb{Z})(0 \leq w < |apuestas[i]| \longrightarrow_L$
$apuestas[i][w] \geq 0 \ \wedge \ pagos[i][w] \geq 1\}$
`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |eventos| \longrightarrow_L (\forall w : \mathbb{Z})(0 \leq w < |eventos[i]| \longrightarrow_L$
$0 \leq eventos[i][w] < |apuestas[i]|\}$
`requiere` $\{(\forall i : \mathbb{Z})(0 \leq i < |apuestas| \longrightarrow_L (\sum\limits_{k=0}^{|apuestas[i]|-1} apuestas[i][k] = 1))\}$

`asegura` $\{|apuestas| = |old(apuestas)| \ \wedge \ (\forall j : \mathbb{Z})(0 \leq j < |apuestas| \ \wedge \ j \neq individuo \longrightarrow_L$
$esMismaApuesta(apuestas[j], old(apuestas[j])\}$
`asegura` $\{(\exists \ trayectoria : seq\langle seq\langle\mathbb{R}\rangle\rangle)$
$(esTrayectoriaCompleta(trayectoria, recursos, cooperan, apuestas, pagos, eventos) \wedge (\forall \ otrasApuestas : seq\langle seq\langle\mathbb{R}\rangle\rangle)$
$(|otrasApuestas| = |apuesta| \ \wedge \ (\forall j : \mathbb{Z})(0 \leq j < |otrasApuestas| \ \wedge \ j \neq individuo \longrightarrow_L$
$esMismaApuesta(otrasApuestas[j], apuestas[j]) \ \wedge$
$esDistintaApuesta(otrasApuestas[individuo], apuestas[individuo])) \ \wedge$
$(\forall \ otraTrayectoria : seq\langle seq\langle\mathbb{R}\rangle\rangle)$
$(esTrayectoriaCompleta(otraTrayecotoria, recursos, cooperan, otrasApuestas, pagos, eventos) \longrightarrow_L$
$trayectoria[individuo][|eventos|] \geq otraTrayectoria[individuo][|eventos|]))))\}$

`pred esDistintaApuesta` (apuesta1: $seq\langle\mathbb{R}\rangle$, apuesta2: $seq\langle\mathbb{R}\rangle$) {
$|apuesta1| = |apuesta2| \ \wedge \ (\exists \ i : \mathbb{Z})(0 \leq i < |apuesta1| \wedge_L apuesta1[i] \neq apuesta2[i]$
}

`pred esMismaApuesta` (apuesta1: $seq\langle\mathbb{R}\rangle$, apuesta2: $seq\langle\mathbb{R}\rangle$) {
$|apuesta1| = |apuesta2| \ \wedge \ (\forall \ i : \mathbb{Z})(0 \leq i < |apuesta1| \longrightarrow_L apuesta1[i] = apuesta2[i]$
}

`pred esTrayectoriaCompleta` (trayectoria: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, recursos: $seq\langle\mathbb{R}\rangle$, cooperan: $seq\langle\mathbb{R}\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$|trayectoria| = |eventos| \ \wedge \ (\forall i : \mathbb{Z})(0 \leq i < |trayectoria| \longrightarrow_L |trayectoria[i]| = |eventos[i] + 1| \ \wedge$
$trayectoria[i][0] = recursos[i] \ \wedge \ (\forall j : \mathbb{Z})(1 \leq w < |trayectoria[i]| \longrightarrow_L$
$trayectoria[i][w] = recursoDelIndividuoEnW(trayectoria, cooperan, apuestas, pagos, eventos, i, w)$
}

`aux recursoDelIndividuoEnW` (trayectoria: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, i: $\mathbb{Z}$, w: $\mathbb{Z}$) : $\mathbb{R}$ =
if $cooperan[i] = \text{true}$ then $parteDelFondoEnW(trayectoria, cooperan, w)$ else
$trayectoria[i][w-1] * apuesta[i][evento[i][w-1]] * pago[i][evento[i][w-1]] +$
$parteDelFondoEnW(trayectoria, cooperan, apuestas, pagos, eventos, w)$ fi ;

`aux parteDelFondoEnW` (trayectoria: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, w: $\mathbb{Z}$) : $\mathbb{R}$ =
$(\sum\limits_{i=0}^{|cooperan|-1}$ if $cooperan[i]$ then $trayectoria[i][w-1] * apuesta[i][evento[i][w-1]] * pago[i][evento[i][w-1]]$ else 0 fi$)$
$/ \ |cooperan|$;

En este ejercicio, la idea es asegurarnos que:

Existe una *trayectoria* fabricada con recursos, cooperan, **apuestas**, pagos y eventos; que para toda $otraTrayectoria$ fabricada con recursos, cooperan, pagos, eventos y cualquier $otrasApuestas$ (secuencia de secuencias de reales, exactamente igual a apuestas, excepto que cambian las apuestas de individuo); tal que el último elemento de individuo en trayectoria es mayor igual al último elemento de individuo en $otraTrayectoria$.

Con el predicado `esTrayectoriaCompleta` determinamos todos los recursos de cada uno de los individuos en cada paso de tiempo, utilizando los recursos del paso de tiempo anterior. Además este predicado utiliza `parteDelFondoEnW` y `recursoDelIndividuoEnW` para determinar el valor del fondo monetario sobre la cantidad de individuos y del producto de los recursos, apuestas y pagos, en cada paso de tiempo w.

Con el predicado `esMismaApuesta` nos aseguramos que dos secuencias de reales tengan exactamente el mismo largo y exactamente los mismos elementos, en el mismo orden.

Con el predicado `esDistintaApuesta` nos aseguramos que dos secuencias de reales tengan al menos, un elemento distinto entre si o que estén en distinto orden.

2. Demostraciones de correctitud

Dada la siguiente especificación:

`proc` `frutoDelTrabajoPuramenteIndividual` (in recurso : $\mathbb{R}$, in apuesta: $\langle$s: $\mathbb{R}$, c : $\mathbb{R}\rangle$, in pago : $\langle$s : $\mathbb{R}$, c : $\mathbb{R}\rangle$, in eventos : $seq\langle$Bool$\rangle$) : $\mathbb{R}$

  `requiere` $\{apuestas_c + apuesta_s = 1 \ \wedge \ pago_c > 0 \ \wedge \ pago_s > 0 \ \wedge \ apuesta_s > 0 \ \wedge \ recurso > 0\}$

  `asegura` $\{res = recurso(apuesta_c pago_c)^{apariciones(eventos,T)}(apuesta_s pago_s)^{apariciones(eventos,F)}\}$

Para probar que es correcto dada la siguiente implementación,
sea el programa S tal que:

```
res = recursos
i = 0
while (i < |eventos|) do
    if eventos[i] then
        res = (res * apuesta.c) * pago.c
    else
        res = (res * apuesta.s) * pago.s
    endif
    i = i + 1
endwhile
```

Entonces tenemos que demostrar la correctitud de la tripla de Hoare $\{P\}S\{Q\}$,

Para eso podemos dividir la demostración en 2, primero para

```
res = recursos
i = 0
```

A esta parte del programa la llamaremos S_a, y hay que demostrar que $\{P\}S_a\{P_c\}$ es válida

Y luego para

```
while (i < |eventos|) do
    if eventos[i] then
        res = (res * apuesta.c) * pago.c
    else
        res = (res * apuesta.s) * pago.s
    endif
    i = i + 1
endwhile
```

A esta parte del programa la llamaremos S_c, y hay que demostrar que $\{P_c\}S_c\{Q_c\}$ es válida

Dado que el programa hipotéticamente termina en Q_c, entonces $Q_c \equiv Q$
teniendo además para S_c un invariante I y una función variante fv:

$$P \equiv apuesta_c + apuesta_s = 1 \ \wedge \ pago_c > 0 \ \wedge \ pago_s > 0 \ \wedge \ apuesta_s > 0 \ \wedge \ recurso > 0$$
$$P_c \equiv i = 0 \ \wedge \ res = recurso$$
$$Q_c \equiv res = recurso * (apuesta_c * pago_c)^{apariciones(eventos,T)} * (apuesta_s * pago_s)^{apariciones(eventos,F)}$$
$$B \equiv i < |eventos|$$
$$I \equiv 0 \leq i \leq |eventos| \ \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}$$
$$fv = |eventos| - i$$

Demostración de $\{P\}S_a\{P_c\}$

Para esto, alcanza con demostrar que $P \rightarrow wp(S_a, P_c)$

Primero calculamos el $wp(S_a, P_c)$ tal que:

$$\begin{aligned} wp(S_a, P_c) &\equiv wp(res = recurso; i = 0, i = 0 \ \wedge \ res = recurso) \\ &\equiv wp(res = recurso, wp(i = 0, i = 0 \ \wedge \ res = recurso)) \end{aligned}$$

Para simplificar, calculo primero $wp(i = 0, i = 0 \ \wedge \ res = recurso)$

$$\begin{aligned} wp(i = 0, i = 0 \ \wedge \ res = recurso) &\equiv def(0) \wedge_L (0 = 0 \ \wedge \ res = recurso) \\ &\equiv True \wedge_L (True \ \wedge \ res = recurso) \\ &\equiv res = recurso \end{aligned}$$

Entonces queda tal que:

$$\begin{aligned} wp(S_a, P_c) &\equiv wp(res = recurso, res = recurso) \\ &\equiv def(recurso) \wedge_L res = res \\ &\equiv True \wedge_L True \\ &\equiv True \end{aligned}$$

Y tomando el antecedente P **como verdadero, tenemos que** $P \rightarrow wp(S_a, P_c) \equiv True \rightarrow True$ **lo cual es cierto**

Demostración de $\{P_c\}S_c\{Q_c\}$

2.1. $P_c \longrightarrow I$

$$P_c \to I \equiv (i = 0 \ \land \ res = recurso) \to (0 \leq i \leq |eventos| \land$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi})$$

Para facilitar la resolución, separamos en 2 partes:

P1: $(i = 0 \ \land \ res = recurso) \to 0 \leq i \leq |eventos|$

y

P2: $(i = 0 \ \land \ res = recurso) \to res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}$

Para P1, tomando el antecedente como verdadero:

$$\begin{aligned}(i = 0 \ \land \ res = recurso) \to 0 \leq i \leq |eventos| &\equiv 0 \leq 0 \leq |eventos| \\ &\equiv True\end{aligned}$$

Para P2, tomando el antecedente como verdadero:

$$\begin{aligned}(i = 0 \ \land \ res = recurso) \to & res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi} \\ &\equiv recurso = recurso * \prod_{j=0}^{0-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi} \\ &\equiv recurso = recurso * 1 \\ &\equiv recurso = recurso \\ &\equiv True\end{aligned}$$

Entonces tenemos que:

$$\begin{aligned}P_c \to I &\equiv (i = 0 \ \land \ res = recurso) \to (0 \leq i \leq |eventos| \land \\ &\quad res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) \\ &\equiv P1 \ \land \ P2 \\ &\equiv True \ \land \ True \\ &\equiv True\end{aligned}$$

2.2. $\{I \wedge B\}\ S_c\ \{I\}$

Dado que $\{I \wedge B\} S_c \{I\} \equiv I \wedge B \rightarrow wp(S_c, I)$, para simplificar las cuentas, primero resolveremos para $I \wedge B$

$$\begin{aligned}
I \wedge B \equiv\ & (0 \leq i \leq |eventos| \wedge \\
& res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) \wedge \\
& i < |eventos| \\
\equiv\ & (0 \leq i < |eventos| \wedge \\
& res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) \\
\equiv\ & (-1 \leq i \leq |eventos| - 1 \wedge \\
& res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi})
\end{aligned}$$

Ahora resolvemos para $wp(S_c, I)$:

$$\begin{aligned}
wp(S_c, I) &\equiv wp(\textsf{if } eventos[i] \textsf{ then } res = (res * apuesta.c) * pago.c \textsf{ else } res = (res * apuesta.s) * pago.s \textsf{ fi}; (i = i + 1), I) \\
&\equiv wp(\textsf{if } eventos[i] \textsf{ then } res = (res * apuesta.c) * pago.c \textsf{ else } res = (res * apuesta.s) * pago.s \textsf{ fi}, wp(i = i + 1, I))
\end{aligned}$$

Vemos primero $wp(i = i + 1, I)$

$$\begin{aligned}
wp(i = i + 1, I) \equiv\ & def(i + 1) \wedge_L \\
& (0 \leq i + 1 \leq |eventos| \wedge \\
& res = recurso * \prod_{j=0}^{i+1-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) \\
\equiv\ & True \wedge_L \\
& (-1 \leq i \leq |eventos| - 1 \wedge \\
& res = recurso * \prod_{j=0}^{i} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) \\
\equiv\ & (-1 \leq i \leq |eventos| - 1 \wedge \\
& res = recurso * \prod_{j=0}^{i} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) \\
\equiv\ & E
\end{aligned}$$

Entonces tenemos que:

$$wp(\textsf{if } eventos[i] \textsf{ then } res = (res * apuesta.c) * pago.c \textsf{ else } res = (res * apuesta.s) * pago.s \textsf{ fi}, wp(i = i+1, I))$$
$$\equiv wp(\textsf{if } eventos[i] \textsf{ then } res = (res * apuesta.c) * pago.c \textsf{ else } res = (res * apuesta.s) * pago.s \textsf{ fi}, E)$$

$$\equiv def(eventos[i]) \wedge_L ((eventos[i]) \ \wedge \ wp(res = (res * apuesta.c) * pago.c, E)) \vee$$
$$(\neg(eventos[i]) \ \wedge \ wp(res = (res * apuesta.s) * pago.s, E))$$

$$\equiv (def(eventos) \ \wedge \ def(i) \wedge_L 0 \leq i \leq |eventos|) \wedge_L ((eventos[i]) \ \wedge \ wp(res = (res * apuesta.c) * pago.c, E)) \vee$$
$$(\neg(eventos[i]) \ \wedge \ wp(res = (res * apuesta.s) * pago.s, E))$$

$$\equiv (True \ \wedge \ True \wedge_L 0 \leq i \leq |eventos|) \wedge_L ((eventos[i]) \ \wedge \ wp(res = (res * apuesta.c) * pago.c, E)) \vee$$
$$(\neg(eventos[i]) \ \wedge \ wp(res = (res * apuesta.s) * pago.s, E))$$

$$\equiv (0 \leq i \leq |eventos|) \wedge_L ((eventos[i]) \ \wedge \ wp(res = (res * apuesta.c) * pago.c, E)) \vee$$
$$(\neg(eventos[i]) \ \wedge \ wp(res = (res * apuesta.s) * pago.s, E))$$

Resolvemos para $eventos[i] \ \wedge \ wp(res = (res * apuesta.c) * pago.c, E)$:

$$eventos[i] \ \wedge \ wp(res = (res * apuesta.c) * pago.c, E))$$

$$\equiv eventos[i] \ \wedge \ (def((res * apuesta.c) * pago.c) \wedge_L (-1 \leq i \leq |eventos| - 1 \wedge$$
$$(res * apuesta.c) * pago.c = recurso * \prod_{j=0}^{i} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}))$$

$$\equiv eventos[i] \ \wedge \ (True \wedge_L (-1 \leq i \leq |eventos| - 1 \wedge$$
$$(res * apuesta.c) * pago.c = recurso * \prod_{j=0}^{i} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}))$$

$$\equiv eventos[i] \ \wedge \ (-1 \leq i \leq |eventos| - 1 \wedge$$
$$(res * apuesta.c) * pago.c = recurso * \prod_{j=0}^{i} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi})$$

Sabiendo que $eventos[i]$ se cumple, rearmamos la expresión tal que:

$$\equiv (-1 \leq i \leq |eventos| - 1 \wedge$$
$$res * apuesta.c * pago.c = recurso * (\prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) * apuesta.c * pago.c)$$

$$\equiv (-1 \leq i \leq |eventos| - 1 \wedge$$
$$res = \frac{recurso * (\prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) * apuesta.c * pago.c}{apuesta.c * pago.c}$$

$$\equiv (-1 \leq i \leq |eventos| - 1 \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi})$$

Resolvemos para $\neg(eventos[i]) \ \wedge \ wp(res = (res * apuesta.s) * pago.s, E)$:

$$\neg(eventos[i]) \ \wedge \ wp(res = (res * apuesta.s) * pago.s, E))$$

$$\equiv \neg(eventos[i]) \ \wedge \ (def((res * apuesta.s) * pago.s) \wedge_L (-1 \leq i \leq |eventos| - 1 \wedge$$

$$(res * apuesta.s) * pago.s = recurso * \prod_{j=0}^{i} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}))$$

$$\equiv \neg(eventos[i]) \ \wedge \ (True \wedge_L (-1 \leq i \leq |eventos| - 1 \wedge$$

$$(res * apuesta.s) * pago.s = recurso * \prod_{j=0}^{i} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}))$$

$$\equiv \neg(eventos[i]) \ \wedge \ (-1 \leq i \leq |eventos| - 1 \wedge$$

$$(res * apuesta.s) * pago.s = recurso * \prod_{j=0}^{i} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi})$$

Sabiendo que $eventos[i]$ NO se cumple, rearmamos la expresión tal que:

$$\equiv (-1 \leq i \leq |eventos| - 1 \wedge$$

$$res * apuesta.s * pago.s = recurso * (\prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) * apuesta.s * pago.s)$$

$$\equiv (-1 \leq i \leq |eventos| - 1 \wedge$$

$$res = \frac{recurso * (\prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) * apuesta.s * pago.s}{apuesta.s * pago.s}$$

$$\equiv (-1 \leq i \leq |eventos| - 1 \wedge$$

$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi})$$

Y la expresión $wp(S_c, I)$ nos quedaría tal que:

$$wp(S_c, I) \equiv (0 \leq i \leq |eventos|) \wedge_L ((-1 \leq i \leq |eventos| - 1 \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}) \vee$$
$$(-1 \leq i \leq |eventos| - 1 \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}))$$

$$wp(S_c, I) \equiv (0 \leq i \leq |eventos|) \wedge_L (-1 \leq i \leq |eventos| - 1 \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi})$$

$$wp(S_c, I) \equiv -1 \leq i \leq |eventos| - 1 \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}$$

Nos queda ver que $I \wedge B \rightarrow wp(S_c, I)$

Teniendo:
$$I \wedge B \equiv -1 \leq i \leq |eventos| - 1 \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}$$

$$wp(S_c, I) \equiv -1 \leq i \leq |eventos| - 1 \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \textsf{if } eventos[j] \textsf{ then } apuesta.c * pago.c \textsf{ else } apuesta.s * pago.s \textsf{ fi}$$

Notamos que ambas expresiones son equivalentes,
por lo que $I \wedge B \rightarrow wp(S_c, I)$ y por lo tanto, $\{I \wedge B\} S_c \{I\}$

2.3. $I \wedge \neg B \longrightarrow Q_c$

Tomando el antecedente como verdadero:

$$I \wedge \neg B \longrightarrow Q_c \equiv \neg(i < |eventos|) \wedge (0 \leq i \leq |eventos| \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}) \rightarrow$$
$$res = recurso * (apuesta_c * pago_c)^{apariciones(eventos,T)} * (apuesta_s * pago_s)^{apariciones(eventos,F)}$$

$$\equiv i \geq |eventos| \wedge (0 \leq i \leq |eventos| \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}) \rightarrow$$
$$res = recurso * (apuesta_c * pago_c)^{apariciones(eventos,T)} * (apuesta_s * pago_s)^{apariciones(eventos,F)}$$

$$\equiv i = |eventos| \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi} \rightarrow$$
$$res = recurso * (apuesta_c * pago_c)^{apariciones(eventos,T)} * (apuesta_s * pago_s)^{apariciones(eventos,F)}$$

$$\equiv res = recurso * \prod_{j=0}^{|eventos|-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi} \rightarrow$$
$$res = recurso * (apuesta_c * pago_c)^{apariciones(eventos,T)} * (apuesta_s * pago_s)^{apariciones(eventos,F)}$$

Y dado que estas 2 expresiones son equivalentes, la implicación es verdadera. Por lo tanto, $I \wedge \neg B \longrightarrow Q_c$

2.4. $\{I \ \wedge \ B \ \wedge \ fv = v_0\} \ S \ \{fv < v_0\}$

Empezamos calculando $wp(S_c, fv < v_0)$

$wp(S_c, fv < v_0)$
$\equiv wp(S_{ite}; i+1, |eventos| - i < v_0)$
$\equiv wp(S_{ite}, |eventos| - i - 1 < v_0)$
$\equiv def(eventos[i]) \wedge_L ((eventos[i] \ \wedge \ wp(S1_{ite}, eventos - i - 1 < v_0)) \ \vee \ (\neg(eventos[i]) \ \wedge \ wp(S2_{ite}, eventos - i - 1 < v_0)))$

Desarrollamos $wp(S1_{ite}, eventos - i - 1 < v_0)$ **y** $wp(S2_{ite}, eventos - i - 1 < v_0)$

$wp(S1_{ite}, eventos - i - 1)$
$\equiv wp(res = res * apuesta_c * pago_c, |eventos| - i - 1 < v_0)$
$\equiv def(res) \wedge_L Q^{res}_{res*apuesta_c*pago_c}$
$\equiv True \wedge_L |eventos| - i - 1 < v_0$
$\equiv |eventos| - i - 1 < v_0$

$wp(S2_{ite}, eventos - i - 1)$
$\equiv wp(res = res * apuesta_s * pago_s, |eventos| - i - 1 < v_0)$
$\equiv def(res) \wedge_L Q^{res}_{res*apuesta_s*pago_s}$
$\equiv True \wedge_L |eventos| - i - 1 < v_0$
$\equiv |eventos| - i - 1 < v_0$

Siguiendo con $wp(S_{ite}, eventos - i - 1 < v_0)$

$def(eventos[i]) \wedge_L ((eventos[i] \ \wedge \ wp(S1_{ite}, eventos - i - 1 < v_0)) \ \vee \ (\neg(eventos[i]) \ \wedge \ wp(S2_{ite}, eventos - i - 1 < v_0)))$
$\equiv def(eventos[i]) \wedge_L ((eventos[i] \ \wedge \ |eventos| - i - 1 < v_0) \ \vee \ (\neg(eventos[i]) \ \wedge \ |eventos| - i - 1 < v_0))$
$\equiv 0 \leq i < |eventos| \wedge_L ((eventos[i] \ \wedge \ |eventos| - i - 1 < v_0) \ \vee \ (\neg(eventos[i]) \ \wedge \ |eventos| - i - 1 < v_0))$
$\equiv 0 \leq i < |eventos| \wedge_L (eventos[i] \ \vee \ \neg(eventos[i])) \ \wedge \ (|eventos| - i - 1 < v_0)$
$\equiv 0 \leq i < |eventos| \wedge_L True \ \wedge \ (|eventos| - i - 1 < v_0)$
$\equiv 0 \leq i < |eventos| \wedge_L |eventos| - i - 1 < v_0$

Ahora desarrollamos $\{I \ \wedge \ B \ \wedge \ fv = v_0\}$

$I \ \wedge \ B \ \wedge \ fv = v_0$

$$\equiv 0 \leq i \leq |eventos| \wedge_L res = recursos * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta_c * pago_c \text{ else } apuesta_s * pago_S \text{ fi}$$
$$\wedge \ i < |eventos| \ \wedge \ |eventos| - i = v_0$$
$$\equiv 0 \leq i < |eventos| \wedge_L res = recursos * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta_c * pago_c \text{ else } apuesta_s * pago_S \text{ fi}$$
$$\wedge \ |eventos| - i = v_0$$

Con $\{I \ \wedge \ B \ \wedge \ fv = v_0\}$ **implicamos la conjunción** $(0 \leq i < |eventos|) \wedge_L (|eventos| - i - 1 < v_0)$ **por partes:**

$(I \ \wedge \ B \ \wedge \ fv = v_0) \rightarrow (0 \leq i < |eventos|)$

$$\equiv (0 \leq i < |eventos| \wedge_L res = recursos * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta_c * pago_c \text{ else } apuesta_s * pago_S \text{ fi}$$
$$\wedge \ |eventos| - i = v_0) \rightarrow (0 \leq i < |eventos|)$$

Es suficiente con:

$$(0 \leq i < |eventos|) \rightarrow (0 \leq i < |eventos|)$$

Lo cual es verdadero.

Nos queda ver que:

$$(I \ \wedge \ B \ \wedge \ fv = v_0) \rightarrow (|eventos| - i - 1 < v_0)$$
$$\equiv (0 \leq i < |eventos| \wedge_L res = recursos * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta_c * pago_c \text{ else } apuesta_s * pago_s \text{ fi}$$
$$\wedge \ |eventos| - i = v_0) \rightarrow (|eventos| - i - 1 < v_0)$$

Es suficiente con:

$$(v_0 = |eventos| - i) \rightarrow (|eventos| - i - 1 < v_0)$$

Lo cual es verdadero.

Por lo tanto $(I \ \wedge \ B \ \wedge \ fv = v_0) \rightarrow wp(S_c, fv < v_0)$
Lo cual es equivalente a $\{I \ \wedge \ B \ \wedge \ fv = v_0\} \ S_c \ \{fv < v_0\}$

2.5. $I \ \wedge \ fv \leq 0 \longrightarrow \neg B$

Tomando el antecedente como verdadero:

$$I \ \wedge \ fv \leq 0 \longrightarrow \neg B \ \equiv (0 \leq i \leq |eventos| \ \wedge$$
$$res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}) \ \wedge$$
$$|eventos| - i \leq 0 \rightarrow \neg(i < |eventos|)$$

$$\Rightarrow (0 \leq i \leq |eventos|) \ \wedge \ |eventos| - i \leq 0 \rightarrow \neg(i < |eventos|)$$
$$\equiv (0 \leq i \leq |eventos|) \ \wedge \ |eventos| - i \leq 0 \rightarrow |eventos| \leq i$$
$$\equiv (0 \leq i \leq |eventos|) \ \wedge \ |eventos| \leq i \rightarrow |eventos| \leq i$$
$$\equiv 0 \leq i = |eventos| \rightarrow |eventos| = i \rightarrow |eventos| \leq i$$
$$\equiv |eventos| = i \rightarrow |eventos| \leq i$$
$$\equiv True$$

Podemos concluir que como $\{P\}S_a\{P_c\}$ y $\{P_c\}S_c\{Q_c\}$ son válidas, y dado que $Q_c \equiv Q$, entonces la tripla de Hoare $\{P\}S\{Q\}$ es válida.