DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 1:

Especificación y WP

22 de abril de 2024 — Algoritmos y Estructuras de Datos

Grupo parenLosAlgoritmos

I

Integrante	LU	Correo electrónico
Ballerio, Francisco	986/23	francisco.ballerio@hotmail.com
Lopez, Gabriel	615/23	gabriellopezdu@gmail.com
Suárez, Francisco	104/23	plottier2002@gmail.com
Valesk, Benjamín	156/23	Benja.vales@gmail.com

Corrigió: Fermín

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

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

`requiere` $\{(|recursos| = |cooperan|) \land (0 \leq |recursos|)\}$

`asegura` $\{res = s \land |recursos| = |s| \land_L$ quién es s?

$(\forall i : \mathbb{Z})(0 \leq i < |s| \longrightarrow_L s[i] = coopero(recursos[i], cooperan[i]) + correspondeACadaUno(recursos, cooperan)\}$

`aux coopero` (in recursos[i] : $\mathbb{R}$, in cooperan[i] : Bool, in i : $\mathbb{Z}$) : $\mathbb{R}$ =
(if $cooperan[i] = true$ then 0 else $recursos[i]$ fi) ;

esto no incluye al fondo común

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

1.2. trayectoriaDeLosFrutosIndividualesALargoPlazo

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

esto se puede indefinir

`requiere` $\{(trayectorias = trayectorias_0) \;\land\; |recursos| = |cooperan| = |apuestas| = |pagos| = |eventos| \;\land (\forall i : \mathbb{Z})\ (0 \leq i < |pagos| \longrightarrow_L (|pagos[i]|) = apuestas[i]| \;\land\; recursos[i] > 0 \;\land$
$(\forall j : \mathbb{Z})\ (0 \leq j < |pagos| \longrightarrow_L (\forall j : \mathbb{Z})\ (0 \leq j < |pagos[i]| \longrightarrow_L (pagos[i][j] > 0 \;\land\; apuestas[i][j] > 0)) \;\land$
$sumatoriaApuestas(apuestas)))\}$

qué valores puede tener eventos?

`asegura` $\{tasaGanancia\ (pagos, apuestas, eventos) \;\land$
$longFinal(trayectorias, eventos) \;\land$
$elPrimeroSeMantiene\ (trayectorias, trayectorias_0) \;\land$ → qué largo tiene trayectorias?
$gananciaIndividual\ (trayectorias, pagos, apuestas, eventos) \;\land$
$fondoComúnDivVálido\ (trayectorias, apuestas, pagos, cooperan, eventos) \;\land$
$esTrayectoriaMod\ (trayectorias, apuestas, pagos, eventos, cooperan)\}$

`pred tasaGanancia` (pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) { qué valores toma j?
$(\forall i : \mathbb{Z})\ (0 \leq i < |eventos| \longrightarrow_L (\forall j : \mathbb{Z})\ ((\exists k : \mathbb{Z})\ (0 \leq k < |apuestas|[i] \land_L$
? $(k = eventos[i][j] \land (tasa(apuestas[i][k], pagos[i][k]) = (apuestas[i][k] * pagos[i][k])))$
}

`aux tasa` (in apuestas : $seq\langle seq\langle\mathbb{R}\rangle\rangle$,pago: $\mathbb{R}$) : $\mathbb{R}$ = $apuesta * pago$;

cuál era la idea de esto?

`pred longFinal` (trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias| \longrightarrow_L trayectorias[i] = |eventos| + 1)$
}

`pred elPrimeroSeMantiene` (trayectorias:$seq\langle seq\langle\mathbb{R}\rangle\rangle$, trayectorias$_0$: $seq\langle\mathbb{R}\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias| \longrightarrow_L (trayectoria[i][0] = trayectorias_0[i][0])$
}

`pred gananciaIndividual` (trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$(tasaGanancia(pagos,\ apuestas,\ eventos))$? $\land (\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias|) \longrightarrow_L$
$(\forall j : \mathbb{Z})\ (0 \leq j < |trayectorias[i]| \;\longrightarrow_L (\exists k : \mathbb{Z})\ (0 \leq k < |apuestas[i]| \land_L$
$(trayectorias[i][j] * tasa(apuestas[i][k], pagos[i][k])) = loGanado(trayectorias[i][j], tasa(apuestas[i][k], pagos[i][k]))))$
}

`pred fondoComúnDivVálido` (trayectorias:$seq\langle seq\langle\mathbb{R}\rangle\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle Bool\rangle$, eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$gananciaIndividual(trayectorias, pagos, apuestas, eventos) \land$
? $(\forall i : \mathbb{Z})\ (0 \leq i < |apuestas| \longrightarrow_L (\exists k : \mathbb{Z})\ (0 \leq k < |pagos[i]| \land_L (\forall j : \mathbb{Z})\ (0 \leq j < |trayectorias[i]| \longrightarrow_L$
$loGanado(trayectorias[i][j], tasa(apuestas[i][k], pagos[i][k])) \land (fondoComún(cooperan, loGanado)/|cooperan|) =$
$fondoComúnDiv(cooperan, loGanado))))$ para qué?
}

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

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

`pred esTrayectoriaMod` (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 $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, cooperan:$seq\langle$
{

$(gananciaIndividual(trayectorias, pagos, apuestas, eventos) \wedge$
$fondoComunDivValido(trayectorias, cooperan, apuestas, pagos, eventos) \wedge$
$(\forall i : \mathbb{Z})\ (0 \le i < |apuestas| \longrightarrow_L (\exists k : \mathbb{Z})\ (0 \le k < |pagos[i]| \wedge_L$
$(\forall j : \mathbb{Z})\ (0 \le j < |trayectorias[i]| \longrightarrow_L$
$loGanado(trayectorias[i][j], tasa(apuestas[i][k], pagos[i][k])) \wedge fondoComúnDividido(cooperan[i], loGanado) \wedge$
$(trayectorias[i][j] = decideGanancia(cooperan[i], fondoComúnDividido, loGanado))))$
}
↳ depende de lo anterior

`pred sumatoriaApuestas` (apuestas: $seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \le i < |apuestas| \wedge_L (\sum_{k=0}^{|apuestas[i]|-1} apuestas[i][k]) = 1)$
}

`aux decideGanancia` (in cooperan: Bool, fondoComúnDiv: , loGanado) : $\mathbb{R}$ =
if $coopera = true$ then $fondoComúnDiv$ else $(loGanado + fondCúnDiv)$ fi ;

1.3. trayectoriaExtrañaEscalera

`proc trayectoriaEscaleraExtraña` (in trayectoria: $seq\langle \mathbb{R}\rangle$) : Bool
`requiere` $\{|trayectoria| > 0\}$
`asegura` $\{res = \text{true} \leftrightarrow máximoRecursoPrimero(trayectoria) \vee máximoRecursoÚltimo(trayectoria) \vee máximoRecursoIntermedio(trayectoria)\}$

`pred máximoRecursoPrimero` (~~trayectoria~~ S: $seq\langle \mathbb{R}\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 < i < |S| - 1) \longrightarrow_L (S[i] \le S[i+1]) \wedge_L (S[0] < S[1]))$
}
`pred máximoRecursoÚltimo` (~~trayectoria~~ S: $seq\langle \mathbb{R}\rangle$) {
$(\forall j : \mathbb{Z})\ ((0 < j < |S| - 1) \longrightarrow_L (S[j-1] \le S[j]) \wedge_L (S[|S|-1] < S[|S|-2]))$
}
`pred máximoRecursoIntermedio` (~~trayectoria~~ S :$seq\langle \mathbb{R}\rangle$) {
$(\forall k : \mathbb{Z})\ (\neg(\exists l : \mathbb{Z})\ ((0 < k < |S-1|\ \cancel{\wedge} \rightarrow_L\ 0 < l < |S-1| \wedge k \ne l) \wedge_L (S[k-1] < S[k] > S[k+1]) \wedge_L$
$(S[l-1] < S[l] > S[l+1])))$
}

esto no vale para todo k

idem $(\neg(\exists l : \mathbb{Z})(..) \equiv (\forall l : \mathbb{Z})\ \neg(\dots))$

1.4. individuoDecideSiCooperarONo

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

se puede indefinir

`requiere` $\{(apuestas = apuestas_0) \wedge |recursos| = |cooperan| = |apuestas| = |pagos| = |eventos| \wedge 0 \le n < |cooperan| \wedge (\forall i : \mathbb{Z})\ (0 \le i < |pagos| \longrightarrow_L (|pagos[i]|) = |apuestas[i]|) \wedge recursos[i] > 0 \wedge (\forall j : \mathbb{Z})\ (0 \le j < |pagos| \longrightarrow_L (\forall j : \mathbb{Z})\ (0 \le j < |pagos[i]| \longrightarrow_L (pagos[i][j] > 0 \wedge apuestas[i][j] > 0)) \wedge sumatoriaApuestas(apuestas)))\}$

↳ qué valores pueden tomar individuo y eventos?

`asegura` $\{(\exists S : seq\langle seq\langle \mathbb{R}\rangle\rangle)\ (recursosDelInicio(recursos, S) \wedge$
$longFinal(S, eventos) \wedge$ ↳ qué largo tiene S?
? $(tasaGanancia(pagos, apuestas, eventos) \wedge$
$(gananciaIndividual(S, pagos, apuestas, eventos)) \wedge$
$esTrayectoriaMod(S, apuestas, pagos, eventos, cooperan) \wedge$
$(\exists A : seq\langle seq\langle \mathbb{R}\rangle\rangle)\ (recursosDelInicio(recursos, A) \wedge$
$longFinal(A, eventos) \wedge$
$(tasaGanancia(pagos, apuestas, eventos)) \wedge$
$(gananciaIndividal(A, pagos, apuestas, eventos) \wedge$
$fondoComúnVálido(A, apuestas, pagos, cooperanConContrario(cooperan, n), eventos) \wedge$
$esTrayectoriaMod(A, apuestas, pagos, eventos, cooperanConContrario(cooperan, n)) \wedge$
$(S[n][|S[n]| - 1] \ge A[n][S[n][|S[n]| - 1] \rightarrow cooperan = cooperan) \wedge$
$A[n][S[n][|S[n]| - 1] > S[n][S[n][|S[n]| - 1] \rightarrow cooperan = cooperanConContrario(cooperan, n)))\}$

`pred recursosDelInicio` (recursos: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, S: $seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \le i < |recursos| \wedge_L (Si][0] = recursos[i]))$
}

los aux solo devuelven números

~~`aux`~~ `cooperanConContrario` (C: $seq\langle Bool\rangle$, S: $\mathbb{R}$) : $seq\langle Bool\rangle$ =
if $C[n] = true$ then
$concat(concat(subseq(C, 0, n), [false]), subseq(C, n+1, |c|))$ else
$concat(concat(subseq(C, 0, n), [true]), subseq(C, n+1, |c|))$ fi ;

$(\exists\ coop : seq(B))$
$|coop| = |pagos| \wedge_L$
$coop[ind] = true \wedge$
$(\forall i : \mathbb{Z})(0 \le i < |coop| \wedge_i \ne ind \rightarrow_L coop[i] = coop[i]$

1.5. individuoActualizaApuesta

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

requiere $\{(apuestas = apuestas_0) \wedge$ y el resto de los parámetros?
$(\forall i : \mathbb{Z})\ (0 \leq i < |pagos| \longrightarrow_L (\forall j : \mathbb{Z})\ (0 \leq j < |pagos[i]| \longrightarrow_L pagos[i][j] > 0))\}$
asegura $\{soloCambiaIndividuo(individuo, apuestas, apuestas_0)\ \wedge$
$(\forall trayCom : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ (esTrayectoriaMod(trayCom, apuestas, pagos, eventos, cooperan)\ \wedge$
$recursoInicial(trayCom, recursos)\ \wedge\ longFinal(trayCom, eventos) \longrightarrow$
$(\exists trayMax : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ (esTrayectoriaMod(trayMax, apuestas, pagos, eventos, cooperan)\ \wedge$
$recursoInicial(trayMax, recursos)\ \wedge longFinal(trayMax, eventos) \wedge_L$
$(trayMax[individuo][|trayMax| - 1] \geq trayCom[individuo][|trayCom| - 1]) \longrightarrow$
$apuestas[indivduo] = apuestaMax[individuo]))\}$
no está definida

pred recursoInicial (in trayectoria : $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in recursos : $seq\langle\mathbb{R}\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectoria| \longrightarrow_L (trayectoria[i][0] = recursos[i]))$
}
pred soloCambiaIndiviuo (in apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in $apuestas_0$: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in inividuo: $\mathbb{Z}$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |apuestas| \longrightarrow ((i \neq inividuo\ \wedge\ apuestas[i] = apuestas_0 i)$
}

2. Demostraciones de correctitud

En este punto del trabajo vamos a probar que la especificación de la función frutoDelTrabajoPuramenteIndividual es correcta respecto de su implementación.
Probamos la correctitud del programa de la siguiente manera:

S1 $\equiv res = recurso$
$S2 \equiv i = 0$
$S3 \equiv$ while ($i < |eventos|$) do (if $eventos[i]$ then $res = (res{*}apuesta.c){*}pago.c$ else $res = (res{*}apuesta.s){*}pago.s$ fi) i = i + 1 endwhile
$Q \equiv res = recurso * (apuesta.c * pago.c)^{apariciones(eventos,T)} * (apuesta.s * pago.s)^{apariciones(eventos,t)}$

wp(S1, S2, S3, Q) $\equiv_{axioma3} wp(S1, wp(S2, wp(S3, Q)))$
$wp(S3, Q) \equiv_{axioma5}$ Por este axioma sabemos que no se puede hacer wp de un ciclo, pues quedamos encerrados en un bucle infinito.
Por eso usamos el teorema de la invariante para probar la correctitud del ciclo y que este termina.
Entonces decimos que si existe un predicado I que cumple con:

1 Pc $\longrightarrow$ I (Precondición del ciclo implica a la invariante)

2 $I \wedge B\{S\}I$ (Durante cualquier momento del ciclo la invariante sigue valiendo)

3 $I \wedge \neg B \longrightarrow Q_c$ (Se cumple la postcondición al salir del ciclo

4 $(I \wedge B \wedge V_0 = f_v)\{S\}(f_v < V_0)$ (fv es estrictamente decreciente)

5 $(I \wedge fv \leq 0) \longrightarrow \neg B$ (Si fv alcanza la cota inferior, la guarda (B) no se cumple)

Los puntos 1,2 y 3 demuestran la correctitud del ciclo. Mientras que los puntos 4 y 5 demuestran, mediante una funcion variante, que el ciclo termina.
Ahora definimos:
$Pc \equiv (res = recurso \wedge i = 0)$
$Qc \equiv Q \equiv res = recurso((apuesta_c * pago_c)^{\#(eventos),t)} * (apuestas_s * pago_s)^{\#(eventos,f)})$
$B \equiv (i < |\text{eventos}|)$
$I \equiv (0 \leq i \leq |eventos| \wedge$
$res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)})$
$fv \equiv |eventos| - i$

1 Pc $\longrightarrow$ I

$res = recurso \wedge i = 0 \wedge apuesta_c{+}apuesta_s = 1\ \wedge paco_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0\ \wedge recurso > 0)$
Todo esto es PC. Voy a asignarlo a **PC** para facilitar la lectura

PC$\longrightarrow res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)})$

Por $i = 0$

PC$\longrightarrow res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,0),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,0),f)})$

Como $subseq(lista, 0, 0) = subseq(\{\})$

PC$\longrightarrow res = recurso((apuesta_c * pago_c)^{\#(subseq(\{\}),t)} * (apuestas_s * pago_s)^{\#(subseq(\{\}),f)})$

PC$\longrightarrow res = recurso((apuesta_c * pago_c)^0 * (apuestas_s * pago_s)^0)$

PC$\longrightarrow res = recurso(((apuesta_c)^0 * (pago_c)^0) * ((apuestas_s)^0 * (pago_s)^0))$

Desarmo **PC** para que se vea claramente

$res = recurso \wedge i = 0 \wedge apuesta_c + apuesta_s = 1 \wedge paco_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0) \longrightarrow res = recurso((1)(1) * (1)(1))$

$res = recurso \wedge i = 0 \wedge apuesta_c + apuesta_s = 1 \wedge paco_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0) \longrightarrow res = recurso$

Luego, es cierto que $Pc \longrightarrow$ I ✓

2 $I \wedge B\ \{S\}\ I$

$(0 \leq i \leq |eventos|\ \wedge\ res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)})$
$\wedge\ i < |eventos|$

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

$(0 \leq i \leq |eventos|\ \wedge\ res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)})$

$i = 1 \longrightarrow (0 \leq 1 \leq |eventos| \wedge res = recurso((apuesta_c{*}pago_c)^{\#(subseq(eventos,0,1),t)}{*}(apuestas_s{*}pago_s)^{\#(subseq(eventos,0,1),f)})$
$i = 2 \longrightarrow (0 \leq 2 \leq |eventos| \wedge res = recurso((apuesta_c{*}pago_c)^{\#(subseq(eventos,0,2),t)}{*}(apuestas_s{*}pago_s)^{\#(subseq(eventos,0,2),f)})$
$i = 3 \longrightarrow (0 \leq 3 \leq |eventos| \wedge res = recurso((apuesta_c{*}pago_c)^{\#(subseq(eventos,0,3),t)}{*}(apuestas_s{*}pago_s)^{\#(subseq(eventos,0,3),f)})$
$i = |eventos| - 1 \longrightarrow (0 \leq |eventos| - 1 \leq |eventos|\ \wedge\ res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,|eventos-1|),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,|eventos|-1),f)})$ ✗
Queda probado que a medida que itera el ciclo, y hasta el último valor de i tal que cumple B, la invariante sigue valiendo.

deben hacer una demostración formal con WP

3 $I \wedge \neg B \longrightarrow Q_c$

$\neg B \longrightarrow \neg(i < |eventos|) \longrightarrow (i \geq |eventos|)$; entonces, usando que $\wedge$ es conmutativa:

$(i \geq |eventos|)\ \wedge\ 0 \wedge \leq i \leq |eventos|\ \wedge$
$res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \longrightarrow Q_c$

$\equiv i = |eventos|\ \wedge$
$res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)} \longrightarrow Q_c$

$\equiv res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,|eventos|),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,|eventos|),f)})\ \wedge$
$(i \geq |eventos|) \longrightarrow Q_c$

Pero la subsecuencia de eventos que va desde el 0 hasta la longitud de eventos $((subseq(eventos, 0, |eventos|)$ es, en realidad, la secuencia eventos original, entonces queda: ✓

$\equiv res = recurso((apuesta_c * pago_c)^{\#(eventos),t)} * (apuestas_s * pago_s)^{\#(eventos,f)})$
$\longrightarrow res = recurso((apuesta_c * pago_c)^{\#(eventos),t)} * (apuestas_s * pago_s)^{\#(eventos,f)})$

Así, queda probado que I $\wedge \neg B \longrightarrow Q_c$

4 $((I \wedge B) \wedge (V_0 = f_v))\{S\}(f_v < V_0)\ fv \equiv |eventos| - i$

$S \equiv$ if $elementos[i]$ then
        $res = (res * apuestas_c) * pago_c$
    else
        $res = (res * apuestas_s) * pago_s$
    endif
$i = i + 1$

Para probar este punto, hago la wp entre {S} y $(f_v < V_0)$.

$WP(S, F_v < V_0)$
$\equiv WP(\text{if } eventos[i] \text{ then } res = (res*apuestas_s)*pago_s \text{ else } res = (res*apuestas_c)*Pago_c \text{ fi}; \;\; i = i+1, (|eventos|-i) < V_0$
Por axioma 3;
$\equiv WP(\text{if } eventos[i] \text{ then } res = (res*apuestas_s)*pago_s \text{ else } res = (res*apuestas_c)*Pago_c \text{ fi}, WP(i = i+1, |eventos|-i < V_0)$

Por un lado, hago $WP(i = i+1, |eventos| - i < V_0)$

$WP(i = i+1, |eventos| - i < V_0)$ por axioma 1;
$\equiv def(i+1) \wedge_L (|eventos| - (i+1) < V_0)$
$\equiv |eventos| - i - 1 < V_0$
$\equiv |eventos| - i \leq V_0$
Ahora vuelvo a la WP original.

$WP(\text{if } eventos[i] \text{ then } res = ((res * apuestas_s) * pago_s \text{ else } res = (res * apuestas_c) * Pago_c \text{ fi}, \;\; |eventos - i \leq V_0)$
Por Axioma 4;
$\equiv def(eventos[i]) \wedge_L (eventos[i] \;\wedge\; WP((res = (rs * apuestas_c) * pago_c), |eventos| - i > V_0)$
$\vee (\neg(eventos[i] \;\wedge\; WP((res = (res * apuestas_s) * pagos_s), |eventos| - i > V_0)$

Como $WP((res = (rs*apuestas_c)*pago_c), |eventos|-i > V_0)$ no tiene nada en común entre {S} y Q, entonces la ejecución del programa (en este caso, $\text{if } eventos[i] \text{ then } res = (res * apuestas_s) * pago_s \text{ else } res = (res * apuestas_c) * Pago_c \text{ fi}$) no se relaciona con la postcondición. Es decir, se podría interpretar a {S} como skip. Lo mismo ocurre con $WP((res = (res * apuestas_s) * pagos_s), |eventos| - i > V_0)$ Así;

$\equiv 0 \leq i < |eventos| \wedge_L (eventos[i] \;\wedge\; WP(skip, |eventos| - i \leq V_0) \;\vee\; (\neg(eventos[i] \wedge WP(skip, |eventos| - i \leq V_0)$
$\equiv 0 \leq i < |eventos| \wedge_L (eventos[i] \;\wedge\; |eventos| - i \leq V_0) \;\vee\; (\neg(eventos[i] \wedge |eventos| - i \leq V_0)$
$\equiv 0 \leq i < |eventos| \wedge_L |eventos| - i \leq V_0$

Ahora, tomamos $(I \wedge B) \;\wedge\; (V_0 = f_v)$:

$(0 \leq i \leq |eventos| \wedge$
$res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \;\wedge\; i < |eventos| \;\wedge\; V_0 = |eventos| - i$

Y se puede ver que la implica, por lo que la wp entre {S} y $(f_v < V_0)$ demuestra que fv es estrictamente decreciente en el cuerpo del ciclo.

5 $(I \wedge fv \leq 0) \longrightarrow \neg B$

$(0 \leq i \leq |eventos| \wedge res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \;\wedge\; |eventos| - i \leq 0) \longrightarrow (\neg(i < |eventos|)$

$\equiv (0 \leq i \leq |eventos| \wedge res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \;\wedge\; (|eventos| \leq i)) \longrightarrow (i \leq |eventos|)$

$\equiv (res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \;\wedge\; (|eventos| = i)) \longrightarrow (i \leq |eventos|)$

Se puede ver en la última implicación es verdadera, demostrando así que al llegar fv a la cota inferior, la guarda deja de cumplirse.

Queda así demostrada la correctitud y la finitud del ciclo. Como el programa termina junto con el ciclo, queda también demostrada la correctitud la especificación del programa respecto a su implementación.