DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1 - Especificación y WP

Fondo monetario común

4 de junio de 2024 — Algoritmos y Estructura de Datos 2

Grupo pip install

Integrante	LU	Correo electrónico
Indaco, Matias	710/16	indaco.mat@gmail.com
Palomino, Leonardo	418/21	lpalomino2300@gmail.com
Pórcel, Carlos	1513/21	cporcel@fi.uba.ar
Suarez, Ricardo Javier	127/20	rjavier.suarez97@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. Definición de Tipos

type $recursos = seq\langle R\rangle$

type $cooperan = seq\langle \mathsf{Bool}\rangle$

type $trayectorias = seq\langle seq\langle R\rangle\rangle$

type $eventos = seq\langle seq\langle N\rangle\rangle$

type $apuestas = seq\langle seq\langle R\rangle\rangle$

type $pagos = seq\langle seq\langle R\rangle\rangle$

2. Especificación

2.1. Ejercicio 1

`proc` `redistribucionDeLosFrutos` (in r: recursos, in c: cooperan) : $seq\langle R\rangle$
`requiere` $\{|r| > 0 \wedge todosElementosPositivos(r) \wedge |r| = |c|\}$
`asegura` $\{(\forall i : \mathbb{Z})(((0 \leq i < |res| \wedge c[i] = true) \longrightarrow_L$
$res[i] = sumaDeLosQueCooperan(r, c)/|r|$
$\wedge ((0 <= i < |res| \wedge c[i] = false) \longrightarrow_L res[i] = (sumaDeLosQueCooperan(r, c)/|r|) + r[i]))\}$

`aux` `sumaDeLosQueCooperan` (r: recursos, c: cooperan) : R $=$
$\sum_{i=0}^{|r|-1}$ if $c[i] = true$ then $r[i]$ else 0 fi ;

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

2.2. Ejercicio 2

`proc` `trayectoriaDeLosFrutosIndividualesALargoPlazo` (inout t: trayectorias, in c: cooperan, in a: apuestas, in p: pagos, in e: eventos)
`requiere` $\{t = old(t) \wedge mismaLongitud(t, c, a, p, e) \wedge elementosDentroDeSecuenciasPositivos(t) \wedge$
$elementosDentroDeSecuenciasPositivos(p) \wedge elementosDentroDeSecuenciasPositivos(a) \wedge$
$elementosDentroDeSecuenciasPositivos(e) \wedge apuestasSumanUno(a)\}$
`asegura` $\{nuevoRecursoAgregado(t, c)\}$
`pred` `nuevoRecursoAgregado` (t: trayectorias, c: cooperan) {
$(\forall i, recursoActual : \mathbb{Z})$ (
$0 \leq i < |t| \wedge 1 \leq recursoActual < |t[i]| \longrightarrow_L t[i][recursoActual] = recursoAsignado(t, a, p, i, recursoActual, c)$
)
}

`aux` `recursoAsignado` (t: trayectorias, a: apuestas , p: pagos , eventoId: $\mathbb{Z}$, recursoActual: $\mathbb{Z}$, c: cooperan) : R $=$
if $cooperan[i] = True$ then $gananciaPorEvento(t, a, p, eventoId, recursoActual)/cantDePersonas(c) +$
$fondoDividido(fondo, cantDePersonas)$ else $gananciaPorEvento(t, a, p, eventoId, recursoActual) +$
$fondoDivido(fondo, cantDePersonas)$ fi ;

`aux` `fondoDividido` (fondo: R, cantDePersonas: $\mathbb{Z}$, c: cooperan) : R $=$
$fondoComunitario(cantDePersonas)/cantDePersonas(c)$;

`aux` `fondoComunitario` (cantDePersonas: $\mathbb{Z}$, cooperan: seq$\langle$Bool$\rangle$) : R $=$
$\sum_{i=0}^{cantDePersonas}$ if $cooperan[i] = True$ then $gananciaPorEvento(t, a, p, eventoId, recursoActual)/cantDePersonas(c)$ else 0 fi ;

`aux` `cantDePersonas` (c: cooperan) : $\mathbb{Z}$ $=$ $cantDePersonas = |c|$;

aux `gananciaPorEvento` (t: trayectorias, a: apuestas, p: pagos, eventoId: $\mathbb{Z}$, recursoActual: $\mathbb{Z}$) : R $= t[i][recursoActual - 1] * a[i][eventoId] * p[i][eventoId]$
;

pred `mismaLongitud` (s1: seq⟨T⟩, s2: seq⟨T⟩, s3: seq⟨T⟩, s4: seq⟨T⟩, s5: seq⟨T⟩) {
$|s1| = |s2| \land |s2| = |s3| \land |s3| = |s4| \land |s4| = |s5|$
}

pred `elementosDentroDeSecuenciasPositivos` (s: seq⟨seq⟨T⟩⟩) {
$(\forall i : \mathbb{Z})(0 \leq i < |s| \longrightarrow_L todosElementosPositivos(s[i])$
}

pred `apuestasSumanUno` (s: seq⟨seq⟨T⟩⟩) {
$(\forall i : \mathbb{Z})(0 \leq i < |s| \longrightarrow_L sumaDeApuestasIgualAUno(s[i])$
}

Aclaracion: sumaDeApuestasIgualUno esta definido en el ejercicio 5.

2.3. Ejercicio 3

proc `trayectoriaExtrañaEscalera` (in t: trayectoria) : Bool
requiere $\{todosElementosPositivos(t)\}$
asegura $\{res = true \leftrightarrow (\exists i : \mathbb{Z})$ (
$|t| = 1 \lor (|t| = 2 \land t[0] \neq t[1]) \lor (1 \leq i < |t| - 1 \land_L ((t[i-1] < t[i] > t[i+1] \land \neg(\exists j : \mathbb{Z})$ (
$(1 \leq j < |t| - 1 \land i \neq j) \land_L t[j-1] < t[j] > t[j+1]) \land \neg(t[0] > t[1] \lor t[|t|-1] > t[|t|-2]))$
$\lor (t[0] > t[1] \land \neg(t[|t|-1] > t[|t|-2] \lor t[i+1] < t[i] > t[i+1]))$
$\lor (t[|t|-1] > t[|t|-2] \land \neg(t[0] > t[1] \lor t[i-1] < t[i] > t[i+1])))$
)
)}

2.4. Ejercicio 4

proc `individuoDecideSiCooperarONo` (in individuo: N, in r: recursos, inout c: cooperan, in a: apuestas, in p: pagos, in e: eventos)
requiere $\{mismaLongitud(r, c, a, p, e) \land todosElementosPositivos(r) \land$
$elementosDentroDeSecuenciasPositivos(p) \land elementosDentroDeSecuenciasPositivos(a) \land$
$elementosDentroDeSecuenciasPositivos(e) \land apuestasSumanUno(a) \land individuo < |r|\}$
asegura $\{c = setAt(old(c), c[individuo], individuo\}$
asegura $\{(cooperan[individuo] = true \leftrightarrow$
$recursoFinalCooperando(individuo, r, a, c, p, e) > rercusoFinalSinCooperar(individuo, r, a, old(c), p, e)))\}$

aux `recursoFinalCooperando` (individuo: N , recursos: seq⟨R⟩, apuestas: seq⟨seq⟨R⟩⟩, cooperan: seq⟨Bool⟩, pagos: seq⟨seq⟨R⟩⟩, eventos: seq⟨seq⟨N⟩⟩) : R $=$
$recusos[individuo] * \prod_{i=0}^{|eventos|-1} (fondoComunitarioPorEvento(cooperan, apuestas, pagos, eventos, i) \div |cooperan|)$;

aux `recursoFinalSinCooperar` (individuo: N,recursos: seq⟨R⟩, apuestas: seq⟨seq⟨R⟩⟩, cooperan: seq⟨Bool⟩, pagos: seq⟨seq⟨R⟩⟩, eventos: seq⟨seq⟨N⟩⟩) : R $=$
$recusos[individuo] * \prod_{i=0}^{|eventos|-1} ((apuestas[individuo][evento[individuo][i]] * pagos[individuo][evento[individuo][i]]) +$
$(fondoComunitarioPorEvento(cooperan, apuestas, pagos, eventos, i) \div |cooperan|))$;

aux `fondoComunitarioPorEvento` (cooperan: seq⟨Bool⟩, apuestas: seq⟨seq⟨R⟩⟩,pagos: seq⟨seq⟨R⟩⟩, eventos: seq⟨seq⟨N⟩⟩, numEvento: $\mathbb{Z}$) : R $=$
$\sum_{j=0}^{|cooperan|-1}$ if $cooperan[j]$ then $pagos[j][evento[j][numEvento]] * apuestas[j][evento[j][numEvento]]$ else 0 fi ;

2.5. Demostración p(n)

Sea la sucesión que representaaux recursoFinalCooperando (individuo: N , recursos: seq⟨R⟩, apuestas: seq⟨seq⟨R⟩⟩, $\mathbf{w_0} = w_0, w_1 = w_0 * w'_1, ..., w_k = w_{k-1} * w'_k,\ con\ w'_i = FondoComun_i \div CantidadDeInidividuos,\ 1 \leq i \leq k.$

Quiero probar que, p(n):$w_n = w_0 * \prod_{i=1}^{n} w'_i,\ con\ w'_i = FondoComun_i \div CantidadDeInidividuos,\ 1 \leq i \leq n, \forall n \in N_0.$
Lo pruebo mediante Principio de Inducción.
Caso Base: p(n=0):$w_0 * \prod_{i=1}^{0} w'_i = w_0 * 1 = w_0,\ Verdadero.$
Paso Inductivo: $Sea\ n \in N_0.$
Hipótesis Inductiva :
$p(n) : w_n = w_0 * \prod_{i=1}^{n} w'_i,\ con\ w'_i = FondoComun_i \div CantidadDeInidividuos,\ 1 \leq i \leq k$
Quiero ver que:
$p(n+1) : w_{n+1} = w_0 * \prod_{i=1}^{n+1} w'_i,\ con\ w_i = FondoComun_i \div CantidadDeInidividuos,\ 1 \leq i \leq k$
Por como está definida la suceción:
$w_{n+1} = w_n * w'_{n+1}$
Por Hipótesis Inductiva:
$w_{n+1} = w_n * w'_{n+1} = w_0 * \prod_{i=1}^{n} w'_i * w'_{n+1} = w_0 * \prod_{i=1}^{n+1} w'_i,\ \forall n \in N_0$

Por lo que, p(n+1): $w_{n+1} = w_0 * \prod_{i=1}^{n+1} w'_i,\ \forall n \in N_0, Verdadero.$
Finalmente, como Caso Base y Paso Inductivo Verdaderos, entonces por el Principio de Inducción, p(n) es Verdadero.

Para probar que p(n) vale cuando el individuo no coopera la demostración es la misma con la diferencia que:
$\mathrm{w}'_i = GananciaPorEvento_i + (FondoComun_i \div CantidadDeIndividuos),\ 0 < i \leq k.$

Consideraciones:
GananciaPorEvento$_i$ $: es\ la\ Ganancia\ del\ individuo\ en\ el\ Evento_i\ sin\ tomar\ en\ cuenta\ si\ coopera\ o\ no,\ con\ 1 \leq i \leq k.$
$FondoComun_i:\ es\ la\ suma\ de\ todas\ las\ GananciaPorEvento\ de\ los\ individuos\ que\ cooperan\ en\ el\ Evento_i,\ con\ 1 \leq i \leq k.$
$CantidadDeIndividuos:\ es\ el\ total\ de\ los\ individuos\ que\ participan\ en\ los\ Eventos.$
$w'_i : es\ la\ ganacia\ final\ en\ el\ Evento_i,\ con\ 1 \leq i \leq k.$
$k : es\ el\ último\ evento$

2.6. Ejercicio 5

`proc` `individuoActualizaApuesta` (in individuo: N, in r: recursos, in c: cooperan, inout a: apuestas,in p: pagos, in e: eventos)
`requiere` $\{a = old(a) \land mismaLongitud(r, c, a, p, e) \land todosElementosPositivos(r) \land$
$elementosDentroDeSecuenciasPositivos(p) \land elementosDentroDeSecuenciasPositivos(a) \land$
$elementosDentroDeSecuenciasPositivos(e) \land apuestasSumanUno(a) \land individuo < |recursos|\}$
`asegura` $\{sumaDeApuestasIgualAUno(a[individuo]) \land\ |old(a)[individuo]| = |a[individuo]|\ \land$
$a = setAt(old(a), a[individuo], individuo)\}$
`asegura` $\{(\forall s : seq\text{<}R\text{>})(sumaDeApuestasIgualAUno(s) \land |s| = |a[individuo]|\ \land\ s \neq a[individuo] \longrightarrow_L \land_L$
$recursoFinal(s) \leq recursoFinal(a[individuo])\}$

`aux` `recursoFinal` (individuo: N , recursos: seq⟨R⟩, apuestas: seq⟨seq⟨R⟩⟩, cooperan: seq⟨Bool⟩, pagos: seq⟨seq⟨R⟩⟩, eventos: seq⟨seq⟨N⟩⟩) : R = if $cooperan[individuo] = true$ then
$recursoFinalCooperando(individuo, recursos, apuestas, cooperan, pagos, eventos)$ else
$recursoFinalSinCooperar(individuo, recursos, apuestas, cooperan, pagos, eventos)$ fi ;

`pred` `sumaDeApuestasIgualAUno` (s: seq⟨R⟩) {
$\sum_{i=0}^{|s|-1} s[i] = 1$
}

3. Demostración de correctitud

Sean:
Pre $\equiv apuesta_c + apuesta_s = 1 \land pago_c > 0 \land pago_s > 0 \land apuesta_c > 0 \land apuesta_s > 0 \land recurso > 0$
$Post \equiv res = recurso \cdot (apuesta_c.pago_c)^{(eventos,T)} \cdot (apuesta_s.pago_s)^{(eventos,F)}$
$S_1 \equiv res = recursos$

$S_2 \equiv i = 0$
$P_c \equiv (res = recurso) \wedge (i = 0) \wedge (apuesta_c + apuesta_s = 1) \wedge (pago_c > 0) \wedge (pago_s > 0) \wedge (apuesta_c > 0) \wedge (apuesta_s > 0) \wedge (recurso > 0)$
$B_{ciclo} \equiv i < |eventos|$
$B_{if} \equiv (eventos[i] == true)$
$Q_c \equiv (i = |eventos| \wedge res = recurso \cdot (apuesta_c.pago_c)^{(eventos,T)} \cdot (apuesta_s.pago_s)^{(eventos,F)}$
$fv \equiv |eventos| - i$
$I \equiv (0 \leq i \leq |eventos| \wedge res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})$

Para probar que la especificación es correcta respecto de su implementación, se tiene que cumplir la tripla de Hoare: {Pre}S1;S2;While B_{ciclo} do C endwhile$\{Post\}$
donde Pre es el requiere de la especificacion, Post es el asegura y C es el ciclo de la implementación.
La Tripla de Hoare antes mencionada es equivalente a probar que las siguientes Triplas son verdaderas, y vale por monotonía:
$\{Pre\}S1; S2\{P_c\}$
$\{P_c\}$While B_{ciclo} do C endwhile$\{Q_c\}$
$\{Q_c\}Skip\{Post\}$

Para poder probar que $\{P_c\}$While B_{ciclo} do C endwhile$\{Q_c\}$ es correcto, se tienen que cumplir los siguientes criterios:

1)$P_c \longrightarrow I$
2)$\{I \wedge B_{ciclo}\}C\{I\} \equiv (I \wedge B_{ciclo}) \longrightarrow WP(C, I)$, donde C es el cuerpo del ciclo
3)$(I \wedge \neg B_{ciclo}) \longrightarrow Q_c$
4)$\{I \wedge B_{ciclo} \wedge fv = V_0\}C\{fv < V_0\} \equiv (I \wedge B_{ciclo} \wedge fv = V_0) \longrightarrow WP(C, fv < V_0)$
5)$(I \wedge fv \leq 0) \longrightarrow (\neg B_{ciclo})$

Veamos si se cumple 1) :
$P_c \equiv (res = recurso) \wedge (i = 0) \wedge (apuesta_c + apuesta_s = 1) \wedge (pago_c > 0) \wedge (pago_s > 0) \wedge (apuesta_c > 0) \wedge (apuesta_s > 0) \wedge (recurso > 0)$
$\Rightarrow (i = 0) \wedge (0 \leq i \leq |eventos|) \wedge res = recurso \wedge (res = recurso \cdot \prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi}) \Rightarrow (0 \leq i \leq |eventos| \wedge res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi}) \equiv I$
Por lo tanto vemos que se cumple $P_c \longrightarrow I$

Veamos si se cumple 2) :
Tengo que demostrar (I $\wedge B_{ciclo}) \Rightarrow WP(C, I)$, empecemos por calcular $WP(C, I)$
$WP(i := i+1; I) \equiv (0 \leq i+1 \leq |eventos|) \wedge res = recurso \cdot \prod_{k=0}^{(i+1)-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})$

$WP(S, WP(i := i + 1; I)) \equiv def(eventos[i] = true) \wedge_L (((eventos[i] = true) \wedge (WP(res := res \cdot apuesta_c \cdot pago_c; WP(i := i + 1, I))) \vee (eventos[i] = false \wedge WP(res := res \cdot apuesta_s \cdot pago_s; Wp(i := i + 1, I))))$

Ahora calculo def(eventos[i]=true) :
$def(eventos[i] = true) \equiv 0 \leq i < |eventos|$

Calculo $WP(res := res \cdot apuesta_c \cdot pago_c; WP(i := i + 1, I)) \equiv A_1$ (lo llamamos asi para denotar la primera alternativa)
$\equiv (0 \leq i+1 \leq |eventos|) \wedge res \cdot apuesta_c \cdot pago_c = recurso \cdot \prod_{k=0}^{i}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})$

Calculo $WP(res := res \cdot apuesta_s \cdot pago_s; WP(i := i + 1; I)) \equiv A_2$ (lo llamamos asi para denotar la segunda alternativa)
$\equiv (0 \leq i+1 \leq |eventos|) \wedge res \cdot apuesta_s \cdot pago_s = recurso \cdot \prod_{k=0}^{i}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})$

Calculo $(I \wedge B_{ciclo})$
$\equiv (0 \leq i < |eventos|) \wedge_L res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi}))$

De esta manera llegué a obtener WP(C,I)
$WP(C, I) \equiv 0 \leq i < |eventos| \wedge_L ((eventos[i] = true \wedge A_1) \vee (eventos[i] = false \wedge A_2))$

Ahora tengo que ver si se cumple: $(I \wedge B_{ciclo}) \longrightarrow WP(C, I)$
$(I \wedge B_{ciclo}) \equiv 0 \leq i < |eventos| \wedge_L res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi}))$
$\Rightarrow 0 \leq i < |eventos| \wedge_L$

$((eventos[i] = true \wedge 0 \leq i+1 \leq |eventos| \wedge res \cdot apuesta_c \cdot pago_c = recurso \cdot (\prod_{k=0}^{i}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi}))) \vee$
$(eventos[i] = false \wedge 0 \leq i+1 \leq |eventos| \wedge res \cdot apuesta_s \cdot pago_s = recurso \cdot (\prod_{k=0}^{i}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})))) \equiv WP(C, I)$
Por lo tanto vemos que se cumple $(I \wedge B) \longrightarrow WP(C, I)$

Veamos si se cumple 3) :
Tengo que demostrar que (I$\wedge\neg B_{ciclo}) \longrightarrow Q_c$
$(I \wedge \neg B_{ciclo}) \equiv 0 \leq i \leq |eventos| \wedge i \geq |eventos| \wedge res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})) \Rightarrow i = |eventos| \wedge res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi}))$
$\Rightarrow i = |eventos| \wedge res = recurso \cdot (\prod_{k=0}^{|eventos|-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi}))$
$\equiv i = |eventos| \wedge res = recurso \cdot (apuesta_c.pago_c)^{(eventos,T)} \cdot (apuesta_s.pago_s)^{(eventos,F)} \equiv Q_c$
Por lo tanto vemos que se cumple $(I \wedge \neg B) \longrightarrow Q_c$

Veamos si se cumple 4) :
Tengo que demostrar que {I$\wedge B_{ciclo} \wedge fv = V_0\}C\{fv < V_0\} \equiv (I \wedge B_{ciclo} \wedge fv = V_0) \longrightarrow WP(C, fv < V_0)$
Primero calculo $WP(C, fv < V_0)$, sé que C se compone de un IfThenElse y de una asignación, así que:
$WP(C, fv < V_0) \equiv WP(S, WP(S_3, fv < V_0)$

Ahora calculo $WP(S_3, fv < V_0)$:
$WP(S_3, fv < V_0) \equiv WP(i := i+1, |eventos| - i < V_0) \equiv |eventos| - i - 1 < V_0$

Ahora calculo $WP(S, |eventos| - i - 1 < V_0)$, donde S tiene alternativas ya que es un IfThenElse :
$\equiv def(eventos[i] = true) \wedge_L ((eventos[i] \wedge WP(res = res \cdot apuesta_c \cdot pago_c, |eventos| - i - 1 < V_0)) \vee$
$(eventos[i] = false \wedge WP(res = res \cdot apuesta_s \cdot pago_s, |eventos| - i - 1 < V_0)))$
$\equiv 0 \leq i < |eventos| \wedge_L ((eventos[i] = true \wedge |eventos| - i - 1 < V_0) \vee$
$(eventos[i] = false \wedge |eventos| - i - 1 < V_0))$
$\equiv 0 \leq i < |eventos| \wedge_L (|eventos| - i - 1 < V_0 \wedge (eventos[i] = true \vee eventos[i] = false))$
$\equiv 0 \leq i < |eventos| \wedge |eventos| - i - 1 < V_0$
$\equiv WP(C, fv < V_0)$

Ahora calculo $I \wedge B_{ciclo} \wedge fv = V_0$:
$\equiv 0 \leq i \leq |eventos| \wedge res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})) \wedge i < |eventos| \wedge |eventos| - i = V_0$

Ahora tengo que probar que $(I \wedge B_{ciclo} \wedge fv = V_0) \longrightarrow WP(C, fv < V_0)$
$0 \leq i \leq |eventos| \wedge res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})) \wedge i < |eventos| \wedge |eventos| - i = V_0$
$\Rightarrow 0 \leq i \leq |eventos| \wedge i < |eventos| \wedge |eventos| - i - 1 < V_0$
$\Rightarrow 0 \leq i < |eventos| \wedge |eventos| - i - 1 < V_0$
$\equiv WP(C, f_v < V_0)$
Por lo tanto vemos que se cumple $(I \wedge B_{ciclo} \wedge fv = V_0) \longrightarrow WP(C, fv < V_0)$
y esto es equivalente a probar que $\{I \wedge B_{ciclo} \wedge fv = V_0\}C\{fv < V_0\}$

Veamos si se cumple 5) :
Tengo que demostrar que (I$\wedge fv \leq 0) \longrightarrow \neg B_{ciclo}$
$I \wedge fv \leq 0 \equiv 0 \leq i \leq |eventos| \wedge res = recurso \cdot (\prod_{k=0}^{i-1}(\textsf{if } (eventos[k] = true) \textsf{ then } apuesta_c \cdot pago_c \textsf{ else } apuesta_s \cdot pago_s \textsf{ fi})) \wedge |eventos| - i \leq 0 \Rightarrow 0 \leq i \leq |eventos| \wedge |eventos| - i \leq 0 \equiv 0 \leq i \leq |eventos| \wedge i \geq |eventos|$
$\Rightarrow i = |eventos| \Rightarrow i \geq |eventos| \equiv \neg B_{ciclo}$
Por lo tanto vemos que se cumple que $(I \wedge f_v \leq 0) \longrightarrow \neg B_{ciclo}$

Finalmente como vemos que se cumplen los 5 criterios antes mencionados, podemos decir que la Tripla de Hoare {P$_c$} while B_{ciclo} do C endwhile $\{Q_c\}$ es válida.
Ahora solo nos falta probar que $\{Pre\}S1; S2\{P_c\}$ es verdadera ya que la otra Tripla pendiente era $\{Qc\}Skip\{Post\}$ pero Skip es hacer nada, y Qc es equivalente a Post asi que esa tripla es Verdadera.

Veamos si se cumple $\{Pre\}S1;S2\{P_c\} \equiv Pre \Rightarrow WP(S_1;S_2,P_c)$:

Calculemos WP($S_1;S_2,P_c$) :
$WP(S_1,(WP(S_2,P_c)) \equiv WP(S_1,Pc) \equiv P_c$
Ahora veamos si Pre $\longrightarrow P_c$
Y esto es verdadero ya que los mismos predicados de Pre se encuentran en P_c

De esta manera, vimos que las 3 triplas son verdaderas:
$\{Pre\}S1;S2\{P_c\}$
$\{P_c\}$While B_{ciclo} do C endwhile$\{Q_c\}$
$\{Q_c\}Skip\{Post\}$
Finalmente, la implementación es correcta respecto de su especificación.