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 IJIENAFFOXDQXVPNKFQU

Integrante	LU	Correo electrónico
Mayo, Francisco	333/21	pancho.mayo@gmail.com
Cogliano, Tobias Gabriel	1330/23	tobiasgabrielcogliano@gmail.com
Sandoval, Francsico Dante	1212/23	franciscodtsandoval@gmail.com
Campitelli, Nicolas	640/23	campid2009@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

Comentarios:
1)No nos parecio necesario el uso de comentarios en la mayoria de los ejercicios de especificación ya que la modularización y los nombres de los predicados nos parecieron lo suficientemente correctos para que se entiendan los problemas.
2)Algunos predicados de los ejercicios 1.4 y 1.5 son los mismos que se usan en el ejercicio 1.2.

1.1. redistribucionDeLosFrutos

`proc` `redistribucionDeLosFrutos` (in recursos : $seq\langle\mathbb{R}\rangle$, in cooperan : $seq\langle\mathsf{Bool}\rangle$) : $seq\langle\mathbb{R}\rangle$
`requiere` $\{|recursos| = |cooperan| > 0 \wedge_L recursosIncicialesPositivos(recursos)\}$
`asegura` $\{|res| = |recursos| \wedge$
$(\forall i : \mathbb{Z})\ (0 \leq i < |recursos| \longrightarrow_L res[i] = cuantoRedistribuye(i, recursos, cooperan))\}$

`pred` `recursosIncialesPositivos` (recursos : $seq\langle\mathbb{R}\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |recursos| \longrightarrow_L recursos[i] > 0)$
}
`aux` `cuantoRedistribuye` (individuo : $\mathbb{Z}$, recursos : $seq\langle\mathbb{R}\rangle$, cooperan : $seq\langle\mathsf{Bool}\rangle$) : $\mathbb{R}$ =
if $cooperan[individuo] = \text{true}$ then $fondoMonetarioPorPersona(recursos, cooperan)$ else $recursos[individuo] + fondoMonetarioPorPersona(recursos, cooperan)$ fi ;
`aux` `fondoMonetarioPorPersona` (recursos : $seq\langle\mathbb{R}\rangle$, cooperan : $seq\langle\mathsf{Bool}\rangle$) : $\mathbb{R}$ =
$$\frac{\sum_{i=0}^{|recursos|-1} \text{if } cooperan[i] = \text{true then } recursos[i] \text{ else } 0 \text{ fi}}{|recursos|};$$

1.2. trayectoriasDeLosIndividuosALargoPlazo

`proc` `trayectoriaDeLosFrutosALargoPlazo` (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` $\{|trayectorias| = |eventos| = |pagos| = |apuestas| = |cooperan| > 0 \wedge_L$
$unElementoEnCadaTrayectoria(trayectorias) \wedge recursosInicialesMayoresA0(trayectorias) \wedge$
$mismaCantidadDeEventos(apuestas, pagos) \wedge hayAlgunEvento(eventos) \wedge$
$mismaCantidadDeEventosParaTodosLosIndividuos(eventos) \wedge$
$eventosEnRango(eventos, pagos) \wedge pagosPositivos(pagos) \wedge apuestaEsCorrecta(apuestas) \wedge$
$apuestasPositivas(apuestas)\}$
`asegura` $\{|trayectorias| = |old(trayectorias)| \wedge cantidadCorrectaDeRecursosAlFinal(trayectorias, eventos) \wedge$
$recursosInicialesNoCambian(trayectorias, old(trayectorias)) \wedge recursosBienDefinidos(trayectorias, cooperan,$
$apuestas, pagos, eventos)\}$

`pred` `unElementoEnCadaTrayectoria` (trayectorias : $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias| \longrightarrow_L |trayectorias[i]| = 1)$
}
`pred` `recursosInicialesMayoresA0` (trayectorias : $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias| \longrightarrow_L trayectorias[i][0] > 0)$
}
`pred` `mismaCantidadDeEventos` (apuestas : $seq\langle seq\langle\mathbb{R}\rangle\rangle$,pagos : $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |apuestas| \longrightarrow_L |apuestas[i]| = |pagos[j]|)$
}
`pred` `hayAlgunEvento` (eventos : $seq\langle seq\langle\mathbb{N}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |eventos| \longrightarrow_L |eventos[i]| > 0)$
}
`pred` `mismaCantidadDeEventosParaTodosLosIndividuos` (eventos : $seq\langle seq\langle\mathbb{N}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |eventos| \longrightarrow_L |eventos[i]| = |eventos[j]|)$
}
`pred` `eventosEnRango` (eventos : $seq\langle seq\langle\mathbb{N}\rangle\rangle$, pagos : $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i < |eventos| \wedge 0 \leq j < |eventos[0]| \longrightarrow_L (\exists k : \mathbb{Z})\ (0 \leq k < pagos[0] \wedge eventos[i][j] = k))$
}
`pred` `pagosPositivos` (pagos : $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i < |pagos| \wedge 0 \leq j < |pagos[0]| \longrightarrow_L pagos[i][j] > 0)$
}
`pred` `apuestaEsCorrecta` (apuestas : $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$$(\forall i : \mathbb{Z})\ (0 \leq i < |apuestas|]) \longrightarrow_L \sum_{j=0}^{|apuestas[0]|-1} apuestas[i][j] = 1$$

}
`pred apuestasPositivas` (apuestas : $seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i < |apuestas| \land 0 \leq j < |apuestas[0]| \longrightarrow_L apuestas[i][j] > 0)$
}
`pred cantidadCorrectaDeRecursosAlFinal` (trayectorias : $seq\langle seq\langle \mathbb{R}\rangle\rangle$, eventos : $seq\langle seq\langle \mathbb{N}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |eventos| \longrightarrow_L |eventos[i]| + 1 = |trayectorias[j]|)$
}
`pred recursosInicialesNocambian` (trayectorias : $seq\langle seq\langle \mathbb{R}\rangle\rangle$, old(trayectorias) : $seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias| \longrightarrow_L trayectorias[i][0] = old(trayectorias[i][0]))$
}
`pred recursosBienDefinidos` (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 : $seq\langle seq\langle \mathbb{N}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i < |trayectorias| \land 1 \leq j < |trayectorias[0]| \longrightarrow_L$
$trayectorias[i][j] = \text{if } cooperan[i] = \text{true} \text{ then } fondoPorpesonaConApuestaPagoYEvento(j-1, trayectorias,$
$cooperan, apuestas, pagos, eventos) \text{ else } trayectorias[i][j-1].apuestas[eventos[i][j-1]].pagos[eventos[i][j-1]] +$
$fondoPorPesonaConApuestaPagoYEvento(j-1, trayectorias, cooperan, apuestas, pagos, eventos) \text{ fi})$
}
`aux fondoPorPesonaConApuestaPagoYEvento` (dt : $\mathbb{Z}$, 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 : $seq\langle seq\langle \mathbb{N}\rangle\rangle$) : $\mathbb{R}$ =

$$\frac{\sum\limits_{i=0}^{|trayectorias|-1} \text{if } cooperan[i] = \text{true then } trayectorias[i][dt].pagos[eventos[i][dt]].apuesta[eventos[i][dt]] \text{ else } 0 \text{ fi}}{|taryectorias|};$$

1.3. trayectoriaExtrañaEscalera

Comentario: En este ejercicio hay tres predicados que identifican si hay algun máximo al comienzo, en el medio o al final de la trayectoria. Teniendo esto solo hace falta un "o excluyente" para tomar solo uno de los tres casos.

`proc trayectoriaExtrañaEscalera` (in trayectoria : $seq\langle \mathbb{R}\rangle$) : Bool
requiere $\{|trayectoria| > 0 \land_L trayectoria[0] > 0\}$
asegura $\{(maximoAlPrincipio(trayectoria) \land \neg maximoEnElMedio(trayectoria) \land \neg maximoAlFinal(trayectoria)) \lor$
$(\neg maximoAlPrincipio(trayectoria) \land maximoEnElMedio(trayectoria) \land \neg maximoAlFinal(trayectoria)) \lor$
$(\neg maximoAlPrincipio(trayectoria) \land \neg maximoEnElMedio(trayectoria) \land maximoAlFinal(trayectoria))\}$

`pred maximoAlPrincipio` (trayectoria : $seq\langle \mathbb{R}\rangle$) {
$trayecoria[0] > trayectoria[1]$
}
`pred maximoEnElMedio` (trayectoreia : $seq\langle \mathbb{R}\rangle$) {
$(\exists i : \mathbb{Z})\ (1 \leq |trayectoria| - 1 \land_L trayectoria[i] > trayectoria[i-1] \land_L trayectoria[i] > trayectoria[i+1] \land_L$
$(\forall j : \mathbb{Z})\ (1 \leq j < |trayectoria| - 1 \land j \neq i \longrightarrow_L trayectoria[j] \leq trayectoria[j-1] \land trayectoria[j] \leq trayectoria[j+1]))$
}
`pred maximoAlFinal` (trayectoria : $seq\langle \mathbb{R}\rangle$) {
$trayectoria[|trayectoria| - 1] > trayectoria[|trayectoria| - 2]$
}

1.4. individuoDecideSiCooperarONo

`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 $seq\langle seq\langle \mathbb{N}\rangle\rangle$)
requiere $\{|recursos| = |eventos| = |pagos| = |apuestas| = |cooperan| > 0 \land 0 \leq individuo < |recursos| \land$
$recursosInicialesPositivos(recursos) \land mismaCantidadDeEventosEn(apuestas, pagos) \land hayAlgunEvento(eventos) \land$
$mismaCantidadDeEventosParaTodosLosIndividuos(eventos) \land evetosEnRango(eventos, pagos) \land$
$pagosPositivos(pagos) \land apuestaEsCorrecta(apuestas) \land apuestasPositivas(apuestas)\}$
asegura $\{|cooperan| = |old(cooperan)| \land restoDeIndividuoNoCambian(cooperan, old(cooperan), individuo) \land$
$cooperan[individuo] = \text{if } esTrayectoriaQueDaMasRecursos(old(cooperan), individuo, recursos, pagos, apuestas, eventos)$
$\text{then } old(cooperan)[individuo] \text{ else } \neg old(cooperan)[individuo] \text{ fi}\}$

`pred restoDeIndividuosNoCambian` (cooperan : $seq\langle \mathsf{Bool}\rangle$, old(cooperan) : $seq\langle \mathsf{Bool}\rangle$, individuo : $\mathbb{N}$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |cooperan| \land i \neq individuo \longrightarrow_L cooperan[i] = old(cooperan)[individuo])$
}
`pred esTrayectoriaQueDaMasRecursos` (cooperan : $seq\langle \mathsf{Bool}\rangle$, individuo : $\mathbb{N}$, recursos : $seq\langle \mathbb{R}\rangle$, pagos : $seq\langle seq\langle \mathbb{R}\rangle\rangle$, apuestas : $seq\langle seq\langle \mathbb{R}\rangle\rangle$, eventos : $seq\langle seq\langle \mathbb{N}\rangle\rangle$) {
$(\exists s, t : seq\langle seq\langle \mathbb{R}\rangle\rangle)\ (esTrayectoriaValida(s, cooperan, recursos, pagos, apuestas, eventos) \land_L$
$esTrayectoriaValida(t, SetAt(cooperan, individuo, \neg old(cooperan)[individuo]), recursos, pagos, apuestas, eventos) \land$
$s[individuo][|s[0]| - 1] > t[individuo][|t[0]| - 1]))$

}

pred esTrayectoriaValida (trayectoria : $seq\langle seq\langle\mathbb{R}\rangle\rangle$,cooperan : $seq\langle\mathsf{Bool}\rangle$, recursos : $seq\langle\mathbb{R}\rangle$, pagos : $seq\langle seq\langle\mathbb{R}\rangle\rangle$, apuestas :$seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{N}\rangle\rangle$) {

$|trayectoria| = |cooperan| \land CantidadCorrectaDeRecursosAlFinal(trayectoria, eventos) \land$
$RecursosBienDefinidos(trayectoria, cooperan, apuestas, pagos, evento) \land$
$recursosInicialesCorrectos(trayectoria, recursos)$

}

pred recursosInicialesCorrectos (trayectoria : $seq\langle seq\langle\mathbb{R}\rangle\rangle$,recursos : $seq\langle\mathbb{R}\rangle$) {

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

}

1.5. IndividuoActualizaApuesta

proc individuoActualizaApuesta (in individuo: $\mathbb{N}$,in recurso : $seq\langle\mathbb{R}\rangle$,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 : $seq\langle seq\langle\mathbb{N}\rangle\rangle$)

requiere $\{|recursos| = |eventos| = |pagos| = |apuestas| = |cooperan| > 0 \land$
$0 \leq individuo < |recursos| \land recursosInicialesPositivos(recursos) \land mismaCantidadDeEventosEn(apuestas, pagos) \land$
$hayAlgunEvento(eventos) \land mismaCantidadDeEventosParaTodosLosIndividuos(eventos) \land$
$eventosEnRango(eventos, pagos) \land pagosPositivos(pagos) \land apuestaEsCorrecta(apuestas) \land apuestasPositivas(apuestas)\}$

asegura $\{|apuesta| = |old(apuesta)| \land mismaCantidadDeEventosEn(apuesta, old(apuesta)) \land$
$restoDeInidviduosNoCambian(apuesta, old(apuesta), individuo) \land$
$(\exists s : seq\langle\mathbb{R}\rangle)\ (EsApuestaValida(s, old(apuesta)) \land apuesta[individuo] = s \land$
$ApuestaMaximizaRecursos(individuo, recursos, cooperan, SetAt(old(apuestas), individuo, s), pagos, eventos)))$
}

pred restoDeIndividuosNoCambianApuesta (apuesta : $seq\langle seq\langle\mathbb{R}\rangle\rangle$, old(apuesta) : $seq\langle seq\langle\mathbb{R}\rangle\rangle$, individuo : $\mathbb{N}$) {

$(\forall i, j : \mathbb{Z})\ (0 \leq i < |apuesta| \land 0 \leq j < |apuesta[0]| \land i \neq individuo \longrightarrow_L apuesta[i][j] = old(apuesta)[i][j]))$

}

pred EsApuestaValida (apuesta : $seq\langle seq\langle\mathbb{R}\rangle\rangle$, old(apuesta) : $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {

$|apuesta| = |old(apuestas)[0]| \land apuestaSume1(apuesta) \land apuestaPositivas2(apuesta)$

}

pred ApuestaMaximizaRecursos (individuo: $\mathbb{N}$, recurso : $seq\langle\mathbb{R}\rangle$,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 : $seq\langle seq\langle\mathbb{N}\rangle\rangle$) {

$(\forall s : seq\langle\mathbb{R}\rangle)\ (EsApuestaValida(s, apuestas) \longrightarrow_L (\exists t, q : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ ($
$esTrayectoriaValida(t, cooperan, recursos, pagos, apuestas, eventos) \land$
$esTrayectoriaValida(q, cooperan, recursos, pagos, SetAt(apuestas, individuo, s), eventos) \land$
$t[individuo][t[0]-1] \geq q[individuo][q[0]-1]))$

}

pred apuestaSume1 (apuesta : $seq\langle\mathbb{R}\rangle$) {

$$\sum_{i=0}^{|recursos|-1} apuestas[i] = 1$$

}

pred apuestaPositivas2 (apuesta : $seq\langle\mathbb{R}\rangle$) {

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

}

2. Demostraciones de correctitud

Para demostrar la correctitud de un programa respecto de su especificación hay que demostrar si la Tripla de Hoare {P} S {Q} es válida. Esto ocurre si, dadas P y Q, siempre que el programa empiece en un estado que cumpla P, termina su ejecución en una cantidad finita de pasos y en el estado final se cumple Q.

En este caso, la especificación y el programa que queremos demostrar son los siguientes:

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

requiere {apuesta$_s$ + $apuesta_s = 1$ $\land$pago$_c > 0$ $\land$pago$_s > 0$ $\land$apuesta$_c > 0$ $\land$apuesta$_s > 0$ $\land recurso > 0\}$

asegura $\{res =$recurso.(apuesta$_c.pago_c)^{\#apariciones(eventos,T)}.(apuesta_s.pago_s)^{\#apariciones(eventos,F)}\}$

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

Código 1: codigo del programa

En este programa, la Tripla de Hoare es {requiere} S {asegura}. Para probar que es válida, tenemos que probar $requiere \longrightarrow_L WP(S, asegura)$
A S lo vamos a dividir en S1: res = recurso, S2: i = 0, S3: while B do...

Empezamos por S3. Para esto tenemos que ver si la tripla {Pc} while B do... {Qc} es válida.

Propongo, Pc = todo lo que contiene el requiere $\wedge res = recurso \wedge i = 0$. El Invariante es:
I = $0 \leq i \leq |eventos| \wedge$
$res = recurso.(\text{apuesta}_c.pago_c)^{\#apariciones(subseq(eventos,0,i),T)}.(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i),F)}$
Qc va a ser el mismo asegura de la especificación.

1) Queremos probar que $Pc \longrightarrow I$
requiere $\wedge i = 0 \wedge res = recurso \longrightarrow 0 \leq i \leq |eventos| \wedge$
$res = recurso.(\text{apuesta}_c.pago_c)^{\#apariciones(subseq(eventos,0,i),T)}.(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i),F)}$
Asumo que Pc = $requiere \wedge res = recurso \wedge i = 0$.
Subseq(secuencia, a, a) está definida en la teórica que da una lista vacía, por lo tanto las apariciones en una lista vacía de T o F serán 0. Es el caso que tenemos ya que si i=0, #apariciones(subseq(eventos, 0, i), T) = 0 y #apariciones(subseq(eventos, 0, i), F) = 0.
res = recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,0),T)}.(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,0),F)}$
y queda res = recurso.$(apuesta_c.pago_c)^0(apuesta_s.pago_s)^0$ que se simplifica a res = recurso, lo cual es verdadero.

2) Queremos probar la tripla $\{I \wedge B\}S\{I\}$. Para esto, tenemos que ver si $I \wedge B \longrightarrow wp(S, I)$.
wp(S, I) $\equiv wp(\text{S}_{if}$; i:=i+1, I) $\equiv wp(\text{S}_{if}$, wp(i:=i+1, I)). A wp(i:=i+1, I) lo vamos a llamar I'.
I'= def(i+1) $\wedge \text{I}^i_{i+1}$. def(i+1)$\equiv True$. Entonces I'$\equiv True \wedge \text{I}^i_{i+1}$. $\equiv \text{I}^i_{i+1}$.
$I^i_{i+1} \equiv 0 \leq i+1 \leq |eventos| \wedge$
$res = recurso.(\text{apuesta}_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}.(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)} \equiv I'$.
Ya teniendo I', tenemos que buscar la wp(S_{if}, I').
if eventos[i] then res:=res.$apuesta_c.pago_c$ else res:=res.$apuesta_s.pago_s$
wp(S_{if}, I') $\equiv$
$def(eventos[i]) \wedge ((eventos[i] \wedge wp(res := res.\text{apuesta}_c.pago_c, \text{I'})) \vee (\neg eventos[i] \wedge wp(res := res.\text{apuesta}_s.pago_s, \text{I'}))$
$def(eventos[i]) \equiv 0 \leq i < |eventos|$
A wp(res:=res.$apuesta_c.pago_c$, I') le vamos a decir IF1 y a wp(res:=res.$apuesta_s.pago_s$, I') IF2.
IF1 $\equiv def(res.apuesta_c.pago_c) \wedge \text{I'}^{res}_{res.apuesta_c.pago_c} \equiv True \wedge \text{I'}^{res}_{res.apuesta_c.pago_c} \equiv \text{I'}^{res}_{res.apuesta_c.pago_c}$
${I'_{res.apuesta_c.pago_c}}^{res} \equiv 0 \leq i+1 \leq |eventos| \wedge$
res.apuesta$_c.pago_c$ = recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}.(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)}$
IF2 $\equiv def(res.apuesta_s.pago_s) \wedge \text{I'}^{res}_{res.apuesta_s.pago_s} \equiv True \wedge \text{I'}^{res}_{res.apuesta_s.pago_s} \equiv \text{I'}^{res}_{res.apuesta_s.pago_s}$
${I'_{res.apuesta_s.pago_s}}^{res} \equiv 0 \leq i+1 \leq |eventos| \wedge$
res.apuesta$_s.pago_s$ = recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}.(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)}$
Hasta ahora, wp(S_{if}, I') $\equiv 0 \leq i < |eventos| \wedge ((eventos[i] \wedge IF1) \vee (\neg eventos[i] \wedge IF2))$ ésta es nuestra wp final, wp(S, I)

Tengo que probar que $I \wedge B \longrightarrow wp(S, I)$
Para esto, separo la conjunción y pruebo por casos.
Veo si $I \wedge B \longrightarrow 0 \leq i < |eventos|$ y si $I \wedge B \longrightarrow ((eventos[i] \wedge IF1) \vee (\neg eventos[i] \wedge IF2))$
Asumimos $I \wedge B$, como sabemos que el rango en I es $0 \leq i \leq |eventos|$ y B es i < $|eventos|$, llegamos a que $0 \leq i < |eventos|$, por lo que la primera implicación se cumple.
Ahora hay que ver si $I \wedge B \longrightarrow ((eventos[i] \wedge IF1) \vee (\neg eventos[i] \wedge IF2))$. Esto es:
I $\wedge B \longrightarrow ((eventos[i] \wedge 0 \leq i+1 \leq |eventos| \wedge$
(res.apuesta$_c.pago_c$ = recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$
$\vee(\neg eventos[i] \wedge 0 \leq i+1 \leq |eventos| \wedge$
(res.apuesta$_s.pago_s$ = recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)}))$
A $0 \leq i+1 \leq |eventos|$ lo podemos sacar afuera, ya que está en ambos lados del or, y probarlo por separado.

I $\wedge B \longrightarrow 0 \leq i+1 \leq |eventos|$. Asumiendo $I \wedge B$, sabemos $0 \leq i < |eventos|$. Si a cada parte de la desigualdad le añadimos 1, queda $1 \leq i+1 < |eventos|+1$, que es lo mismo que $0 \leq i+1 \leq |eventos|$, por lo tanto $I \wedge B \longrightarrow 0 \leq i+1 \leq |eventos|$.
Queda probar que $I \wedge B \longrightarrow$ ((eventos[i] $\wedge$
(res.apuesta$_c.pago_c =$ recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$
$\vee(\neg eventos[i] \wedge$
(res.apuesta$_s.pago_s =$ recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)}))$

Se puede probar I $\wedge B \wedge eventos[i] \longrightarrow$
(res.apuesta$_c.pago_c =$ recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$
Por estar en la rama del True, eventos[i] = True, sabemos que hay que agregar un elemento T a la secuencia eventos y que a res de $I \wedge B$ hay que multiplicarlo por $(apuesta_c.pago_c)$ entonces queda
$I \wedge B \wedge eventos[i]$ y se llega a
$(res.apuesta_c.pago_c =$ recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$
Entonces I $\wedge B \wedge eventos[i] \longrightarrow$
(res.apuesta$_c.pago_c =$ recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$
Ahora lo mismo pero con la rama del eventos[i] = false. I $\wedge B \wedge \neg eventos[i] \longrightarrow$
(res.apuesta$_s.pago_s =$ recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$
Por estar en la rama del False, $\neg eventos[i]$ = True, hay que agregar un elemento F a la secuencia eventos y que a res de $I \wedge B$ multiplicarlo por $(apuesta_s.pago_s)$ entonces queda
$I \wedge B \wedge eventos[i]$ y se llega a
$(res.apuesta_s.pago_s =$ recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$
Entonces I $\wedge B \wedge eventos[i] \longrightarrow$
(res.apuesta$_s.pago_s =$ recurso.$(apuesta_c.pago_c)^{\#apariciones(subseq(eventos,0,i+1),T)}(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i+1),F)})$
wp(S,I) $\equiv 0 \leq i < |eventos| \wedge ((eventos[i] \wedge IF1) \vee (\neg eventos[i] \wedge IF2))$
Habiendo probado ambas ramas de true y false, que $I \wedge B \longrightarrow 0 \leq i+1 \leq |eventos|$ queda demostrado $I \wedge B \longrightarrow ((eventos[i] \wedge IF1) \vee (\neg eventos[i] \wedge IF2))$. Esto, sumado a que al principio demostramos $I \wedge B \longrightarrow 0 \leq i < |eventos|$, están las dos implicaciones de la conjunción demostradas.
Entonces se cumple que $I \wedge B \longrightarrow wp(S, I)$ y la tripla $\{I \wedge B\}S\{I\}$ es válida.

3) Tenemos que probar que $I \wedge \neg B \longrightarrow Qc$.
Recuerdamos que nuestro Qc = Asegura.
Asegura $\equiv res = recurso.$(apuesta$_c.pago_c)^{\#apariciones(eventos,T)}.(apuesta_s.pago_s)^{\#apariciones(eventos,F)}$
El I $\equiv 0 \leq i \leq |eventos| \wedge$
res=recurso.(apuesta$_c.pago_c)^{\#apariciones((subseq(eventos,0,i),T)}.(apuesta_s.pago_s)^{\#apariciones((subseq(eventos,0,i),F)}$
Asumimos $I \wedge \neg B$.
$\neg B = |eventos| \leq i$.
Y por el rango del I, se sabe que $0 \leq i \leq |eventos|$. Juntando ambas, tenemos $i = |eventos|$ Entonces:
I $\wedge \neg B \equiv i = |eventos| \wedge$
$res = recurso.$(apuesta$_c.pago_c)^{\#apariciones(subseq(eventos,0,|eventos|),T)}.(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,|eventos|),F)}$
Por como definimos subseq("secuencia", "desde", "hasta") en la teórica, se que si "hasta" es del tamaño de la secuencia y "desde" es 0, subseq("secuencia", "desde", $|secuencia|$) es igual a la secuancia.
Por lo tanto, (subseq(eventos,0,$|eventos|$) = eventos.
Entonces I $\wedge \neg B \equiv i = |eventos|$res=recurso.(apuesta$_c.pago_c)^{\#apariciones(eventos,T)}.(apuesta_s.pago_s)^{\#apariciones(eventos,F)}$
La segunda parte de la conjunción es exactamente nuestra Qc, que es igual al asegura del programa.
De esta forma se probó que $I \wedge \neg B \longrightarrow Qc$.

4) THay que probar que I $\wedge B \wedge V_0 = fv \longrightarrow_L WP(S, fv < V_0)$ para que la tripla $\{I \wedge B \wedge V_0 = fv\}S\{fv < V_0\}$ sea válida.
WP(S, $fv < V_0) \equiv WP(S_{if}$; i:= i+1, $|eventos|$ - $i < V_0$)
WP(S_{if}, $|eventos| - i - 1 < V_0$)
If eventos[i] then res:= $res.apuesta_c.pago_c$ else res:= $res.apuesta_s.apuesta_s$
def(eventos[i]) $\wedge_L((eventos[i] \wedge WP(res :=$res.apuesta$_c.pago_c$,$|eventos|$-i-1 ¡$V_0$)) $\vee$
$(\neg eventos[i] \wedge WP(res :=$res.apuesta$_s.apuesta_s$,$|eventos|$-i-1 $< V_0$)))
def(eventos[i]) $= 0 \leq i < |eventos|$
$0 \leq i < |eventos| \wedge((eventos[i] \wedge def($res.apuesta$_s.apuesta_s) \wedge |eventos| - i - 1 < V_0) \vee$
$(\neg eventos[i] \wedge def($res.apuesta$_s.apuesta_s) \wedge |eventos| - i - 1 < V_0)$
def($res.apuesta_c.apuesta_c$) = True y def($res.apuesta_s.apuesta_s$) = True
Sabemos que $(P \wedge Q) \vee (\neg P \wedge Q) \longrightarrow Q$
Por esto, ((eventos[i] $\wedge def($res.apuesta$_s.apuesta_s) \wedge |eventos| - i - 1 < V_0) \vee$
$(\neg eventos[i] \wedge def($res.apuesta$_s.apuesta_s) \wedge |eventos| - i - 1 < V_0)$
Es lo mismo que $|eventos| - i - 1 < V_0$.
Para ser mas claros, tenemos que WP(S, fv $< V_0$) es $0 \leq i < |eventos| \wedge |eventos| - i - 1 < V_0$
Lo que queríamos probar era I $\wedge B \wedge V_0 = fv \longrightarrow_L WP(S, fv < V_0)$
Esto es, entonces, I $\wedge B \wedge V_0 = |eventos|$-i $\longrightarrow 0 \leq i < |eventos| \wedge |eventos| - i - 1 < V_0$

Probamos por separado. Asumimos $I \wedge B$, y sabemos que $0 \leq i \leq |eventos| \wedge i < |eventos|$
lo cual implica que $0 \leq i < |eventos|$, que es la primera parte de la conjunción.
Para la segunda parte, I $\wedge B \wedge$V$_0$ = $|eventos|$ - i $\longrightarrow |eventos|$ - i - 1 $< V_0$
Asumimos I $\wedge B \wedge$V$_0$ = $|eventos|$ - i y llego a V_0 - 1 $< V_0$.
Por lo tanto, si I $\wedge B \wedge$V$_0 = |eventos| - i \longrightarrow 0 \leq i < |eventos|$ y también I $\wedge B \wedge$V$_0$ = $|eventos|$-i $\longrightarrow |eventos|$ - i - 1 $< V_0$, llegamos a que I $\wedge B \wedge$V$_0$ = $|eventos|$-i $\longrightarrow 0 \leq i < |eventos| \wedge |eventos| - i - 1 <V_0$.
Así que la tripla $\{I \wedge BV_0\}S\{fv < V_0\}$ es válida.

5) Hay que probar $I \wedge fv \leq 0 \longrightarrow \neg B$(que es $|eventos| \leq i$).
Mi fv = $|eventos|$ - i.
I = 0$\leq i \leq |eventos| \wedge$
$res = recurso.(\text{apuesta}_c.pago_c)^{\#apariciones(subseq(eventos,0,i),T)}.(apuesta_s.pago_s)^{\#apariciones(subseq(eventos,0,i),F)}$
Asumo $I \wedge fv \leq 0$. Se por I que $0 \leq i \leq |eventos|$ y con $fv \leq 0$, que $|eventos| - i \leq 0$, es decir $|eventos| \leq i$.
Entonces, lo que sabemos es que $0 \leq i \leq |eventos| \wedge |eventos| \leq i$
Esto es lo mismo que decir que $|eventos|$ es exactamente igual a i. Por lo tanto, I $\wedge fv \leq 0 \longrightarrow \neg B$.

Habiendo demostrado los 5 puntos del teorema del invariante y de terminación de ciclos, puedo concluir que la tripla {Pc} while B do {Qc} es válida.

Ahora, es trivial que el requiere es la wp de la primera linea, res: = recurso porque no hay nada en el programa antes. $\{requiere\}res := recurso\{requiere \wedge res = recurso\}$. Se cumple. Ahora $\{requiere \wedge res = recurso\}i := 0\{Pc\}$, también es trivial porque antes de ésta linea no se habla de i, entonces la wp(i:=0, Pc) es $\{requiere \wedge res = recurso\}$. Ya probé antes que la tripla $\{Pc\}whileBdo\{Qc = Asegura\}$ es válida, entonces por monotonía, la tripla $\{requiere\}S\{Qc = Asegura\}$ es válida y el programa quedó correctamente demostrado.