Fondo Monetario Común

Trabajo práctico 1: Especificación y WP

31 de mayo de 2024 | Algoritmos y Estructuras de Datos

Grupo somtirogla

Integrante	LU	Correo electrónico
Fainsod, Gastón	4/20	`gaston.fainsod@gmail.com`
Berkowsky, Sasha Nicolas	1158/23	`snberkowsky@gmail.com`
Gvirtz, Bruno	1173/23	`bgvirtz18@gmail.com`
Poutays, Manuel	1256/23	`manuelpoutays@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. Introducción

La teoría de juegos es un área de las matemáticas aplicadas que utiliza modelos para estudiar interacciones entre distintos factores dentro de una estructura, haciendo alusión a participantes en juegos de azar. Esta estructura pueden tener comportamientos estocásticos, con el fin de obtener estrategias óptimas, el estudio de dichos modelos puede ser de interes a la hora de elegir las interacciones a realizar. Esta area de la matemática es de interes para comprender comportaminetos por ejemplo en economía, de manera similar, en este trabajo se busca estudiar los comportamientos óptimos en un juego de apuestas en el que los individiduos se ven obligados a apostar en cada paso temporal todos sus recursos. Para un jugador que comienza con un capital inicial w_0, el cual distribuye dicho capital entre n eventos en distintas proporciones $\overline{b} = (b_1, b_2, ..., b_n)$, con distintos pagos por evento $\overline{Q} = (Q_1, Q_2, ..., Q_n)$, el capital resultante al salir el evento $j \in [1, n]$ sera:

$$w_1 = w_0 b_j Q_j \tag{1}$$

Si se realizan k eventos consecutivos, se puede generalizar el capital final como:

$$w_k = w_0 \prod_{i=1}^{k-1} P_i \tag{2}$$

siendo P_i el producto entre el pago y la proporción destinada en el paso i. Otro tema a considerar es cuando hay más de un participante en el juego, los cuales pueden elegir entre contribuir o no a un fondo común. El hecho de participar de un fondo común se entiende como distribuir las ganancias posteriores al evento entre todas las personas que jugaron (entre los que participan y no lo hacen en el fondo común). Por lo cual si se tiene T participantes del fondo común y N participantes en el juego, la distribución del fondo (G) será

$$G = \frac{1}{N} \sum_{i=1}^{T} {w'}_{k_i} \tag{3}$$

donde ${w'}_{k_i}$ es la ganancia que hubiese tenido en principio el participante i en el paso k. Al distribuir las ganancias, se puede redefinir los recursos resultantes (w_{k_i}) obtenidos en Ecuación 2 de la siguiente manera:

$$w_{k_i} = \begin{cases} G & \text{sii el participante i participa del fondo común} \\ G + {w'}_{k_i} & \text{sii el participante i no participa del fondo común} \end{cases} \tag{4}$$

Con el fin de obtener los parametros óptimos entre los participantes, en este trabajo se especificaran distintas funciones necesarias para llevar a cabo la solución del problema.

2. Especificación

2.1. redistribucionDeLosFrutos

Este procedimiento recibe los recursos resultantes para cada individuo y los redistribuye entre todos los participantes por si participan o no del fondo común según lo planteado en Ecuación 3 y Ecuación 4,

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

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

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

requiere $\{esListaDeRecursos(recursos)\}$

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

asegura $\{(\forall i : \mathbb{Z})\ ((0 \leq i < |recursos| \wedge_L cooperan[i] = true) \longrightarrow_L res[i] = valorCoopera(recursos, cooperan))\}$

asegura $\{(\forall i : \mathbb{Z})$ (

$(0 \leq i < |recursos|\ \wedge_L\ cooperan[i] = false) \longrightarrow_L res[i] = valorCoopera(recursos, cooperan) + recursos[i]$

)}

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

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

$(\forall i : \mathbb{Z})$ (

$0 \leq i < |recursos| \longrightarrow_L recursos[i] > 0$

)

}

2.2. trayectoriaDeLosFrutosIndividualesALargoPlazo

Este procedimiento busca obtener los recursos de cada individuo para cada paso de tiempo. Para esto se utiliza el procedimiento redistribucionDeLosFrutos para obtener los valores paso a paso.

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 $\{trayectorias = Trayectorias_0\}$

requiere $\{todosIgualesA(|trayectoria|, \langle|eventos|, |pagos|, |apuestas|, |cooperan|\rangle)\}$

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

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

requiere $\{esMatrizDeApuestas(apuestas)\}$

requiere $\{esMatrizDePagos(pagos)\}$

requiere $\{(\forall i : \mathbb{Z})$ (

$0 \leq i < |apuestas| \longrightarrow_L todosIgualesA(|apuestas[0]|, \langle|pagos[i]|, |apuestas[i]|, |eventos[i]|\rangle)$

)}

asegura $\{(\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias| \longrightarrow_L trayectorias[i][0] = trayectorias_0[i][0])\}$

asegura $\{|trayectorias| = |Trayectorias_0|\}$

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

asegura $\{(\forall i : \mathbb{Z})\ ((0 \leq i < (|eventos[0]|)) \longrightarrow_L actualizarRec(trayectorias, cooperan, apuestas, pagos, eventos, i))\}$

pred actualizarRec (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: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, ganadores: $seq\langle seq\langle \mathbb{N}\rangle\rangle$, indice: $\mathbb{Z}$) {

$(\forall j : \mathbb{Z})\ ((0 \leq j < |trayectorias| \wedge_L$

$((cooperan[j] = true\ \wedge_L$
$trayectorias[j][indice + 1] =$
$valorCoopera(trayectorias, cooperan, apuestas, pagos, ganadores, indice))$

$\vee_L$

$(cooperan[j] = false\ \wedge_L$
$trayectorias[j][indice + 1] =$
$valorCoopera(trayectorias, cooperan, apuestas, pagos, ganadores, indice) +$
$recursosObtenidos(trayectoria[j][indice], apuestas[j], pagos[j], ganadores[j][indice]))))$

}

aux valorCoopera (recursos : $seq\langle seq\langle \mathbb{R}\rangle\rangle$, cooperan : $seq\langle \mathsf{Bool}\rangle$, apuestas : $seq\langle seq\langle \mathbb{R}\rangle\rangle$, pagos : $seq\langle seq\langle \mathbb{R}\rangle\rangle$, ganadores : $seq\langle seq\langle \mathbb{N}\rangle\rangle$, indice : $\mathbb{Z}$) : $\mathbb{R}$ $= \frac{1}{|recursos|} \sum_{i=0}^{|recursos|-1}$ if $cooperan[i] = True$ then

recursosObtenidos(recursos[i][indice],apuestas[i],pagos[i],ganadores[i][indice])] else 0 fi ;

aux recursosObtenidos (recurso : $\mathbb{R}$,apuestas : $seq\langle \mathbb{R}\rangle$, pagos : $seq\langle \mathbb{R}\rangle$, ganador: $\mathbb{N}$) : $\mathbb{R}$ $= recurso * apuesta[ganador] * pago[ganador]$;

pred todosIgualesA (elemento: $\mathbb{N}$, valores: $seq\langle \mathbb{N}\rangle$) {

$(\forall i : \mathbb{Z})\ (valores[i] = elemento)$

}

pred esMatrizDeApuestas (apuestas : $seq\langle seq\langle \mathbb{R}\rangle\rangle$) {

$(\forall i : \mathbb{Z})$ (

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

$0 \leq k < |apuestas[0]| \longrightarrow_L 0 \leq apuestas[i][k] \leq 1)$

)

)

}

pred esMatrizDePagos (pagos : $seq\langle seq\langle \mathbb{R}\rangle\rangle$) {

$(\forall i : \mathbb{Z})$ (

$0 \leq i < |pagos| \longrightarrow_L (\forall k : \mathbb{Z})$ (

$0 \leq k < |pagos[0]| \longrightarrow_L 0 < pagos[i][k]$

)

)

}

2.3. trayectoriaExtrañaEscalera

Este procedimiento busca registrar si en la trayectoria de un individuo hay un único máximo local.

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

`requiere` $\{|trayectoria| > 1\}$

`requiere` $\{esListaDeRecursos(trayectoria)\}$

`asegura` $\{res \leftrightarrow 1 = \sum_{i=1}^{|trayectoria|-2}$ if $(trayectoria[i] > trayectoria[i-1] \wedge_L trayectoria[i] > trayectoria[i+1])$ then 1 else 0 fi $+\ sumaExtremos(trayectoria)\}$

`aux sumaExtremos` (trayectoria : $seq\langle\mathbb{R}\rangle$) : $\mathbb{N}$ = (if $trayectoria[0] > trayectoria[1]$ then 1 else 0 fi) $+$ if $trayectoria[|trayectoria|-1] > trayectoria[|trayectoria|-2]$ then 1 else 0 fi) ;

2.4. individuoDecideSiCooperarONo

Este procedimiento compara las ganancias que obtiene un individuo con su elección de cooperar o no en el fondo común con el caso contrario al elegido. Luego cambia o no de elección quedandose con la que genere mayor ganancia individual.

`proc individuoDecideSiCooperarONo` (in individuo: $\mathbb{N}$,inout cooperan : $seq\langle$Bool$\rangle$, in recursos : $seq\langle\mathbb{R}\rangle$, 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` $\{cooperan = cooperan_0\}$

`requiere` $\{0 \leq individuo < |cooperan|\}$

`requiere` $\{esListaDeRecursos(recursos)\}$

`requiere` $\{esMatrizDeApuestas(apuestas)\}$

`requiere` $\{esMatrizDePagos(pagos)\}$

`requiere` $\{todosIgualesA(|cooperan|, \langle|eventos|, |pagos|, |apuestas|, |recursos|\rangle)\}$

`requiere` $\{(\forall i : \mathbb{Z})\ (0 \leq i < |apuestas| \longrightarrow_L todosIgualesA(|apuestas[0]|, \langle|pagos[i]|, |apuestas[i]|\rangle))\}$

`requiere` $\{(\forall i : \mathbb{Z})\ (0 \leq i < |eventos| \longrightarrow_L todosIgualesA(|eventos[0]|, \langle|eventos[i]|\rangle))\}$

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

`asegura` $\{(\exists cooperanAlternativo : seq\langle$Bool$\rangle)$ (

$(esCooperanAlternativo(cooperanAlternativo, cooperan, individuo))$

$\wedge_L$

$((\exists trayectoria_0 : seq\langle seq\langle\mathbb{R}\rangle\rangle)$ (

$|trayectoria_0| = |eventos| \wedge_L primerColumna(trayectoria_0, recursos) \wedge_L (\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias_0| \longrightarrow_L |trayectorias_0[i]| = |eventos[0]|+1) \wedge_L (\forall j : \mathbb{Z})\ ((0 \leq j < |eventos[0]|) \implies actualizarRec(trayectoria_0, cooperan_0,$

$\wedge_L$

$(\exists trayectoria_1 : seq\langle seq\langle\mathbb{R}\rangle\rangle)$ (

$|trayectoria_1| = |eventos| \wedge_L (\forall j : \mathbb{Z})\ (0 \leq j < |trayectorias_1| \longrightarrow_L |trayectorias_1[j]| = |eventos[0]|+1) \wedge_L primerColumna(trayectoria_1, recursos) \wedge_L (\forall k : \mathbb{Z})\ (0 \leq k < |eventos| \implies actualizarRec(trayectoria_1, cooperanAlternativo, apuestas, pagos, eventos, k))$

$\wedge_L$

$((trayectoria_0[individuo][|eventos|] > trayectoria_1[individuo][|eventos|] \wedge_L cooperan = cooperan_0)$
$\vee_L$
$(trayectoria_0[individuo][|eventos|] > trayectoria_1[individuo][|eventos|] \wedge_L cooperan = cooperanAlternativo))$

)

)

)}

`pred primerColumna` (lista: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, columna: $\mathbb{R}$) {

$(\forall j : \mathbb{Z})\ (0 \leq j < |lista| \longrightarrow_L lista[j][0] = columna[0])$

}

`pred esCooperanAlternativo` (cooperanAlternativo: $seq\langle\mathsf{Bool}\rangle$, cooperanOriginal: $seq\langle\mathsf{Bool}\rangle$, ind: $\mathbb{Z}$) {

$|cooperanOriginal| = |cooperanAlternativo| \wedge_L$

$(\forall i : \mathbb{Z})\ (0 \leq i < |cooperan| \longrightarrow_L$
$((i = ind\ \wedge_L cooperanAlternativo[i] \neq cooperanOriginal[i])$
$\vee_L$
$(i \neq ind\ \wedge_L cooperanAlternativo[i] = cooperanOriginal[i])))$

}

2.5. individuoActualizaApuesta

Este procedimiento actualiza la distribución de apuestas a la que mayor cantidad de ganancias genere.

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

`requiere` $\{apuestas = Apuestas_0\}$

`requiere` $\{esListaDeRecursos(recursos)\}$

`requiere` $\{esMatrizDeApuestas(apuestas)\}$

`requiere` $\{esMatrizDePagos(pagos)\}$

`requiere` $\{0 \leq individuo < |cooperan|\}$

`requiere` $\{todosIgualesA(|cooperan|, \langle|eventos|, |pagos|, |apuestas|, |recursos|\rangle)\}$

`requiere` $\{(\forall i : \mathbb{Z})\ (0 \leq i < |apuestas| \longrightarrow_L todosIgualesA(|apuestas[0]|, \langle|pagos[i]|, |apuestas[i]|\rangle))\}$

`requiere` $\{(\forall i : \mathbb{Z})\ (0 \leq i < |eventos| \longrightarrow_L todosIgualesA(|eventos[0]|, \langle|eventos[i]|\rangle))\}$

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

`requiere` $\{(\forall i : \mathbb{Z})\ ($

$0 \leq i < |apuestas[0]| \longrightarrow_L todosIgualesA(|apuestas[0]|, \langle|pagos[i]|, |apuestas[i]|\rangle)$

$)\}$

`asegura` $\{esMatrizDeApuestas(apuestas)\}$

`asegura` $\{(\forall i : \mathbb{Z})\ ((0 \leq i < |apuestas| \wedge_L i \neq individuo) \longrightarrow_L apuestas[i] = Apuestas_0[i])\}$

`asegura` $\{(\exists trayectoria_0 : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ ($

$(|trayectoria_0| = |apuestas|) \wedge_L (\forall j : \mathbb{Z})\ (0 \leq j < |trayectoria_0| \longrightarrow_L (|eventos[0]| + 1 = trayectoria_0[j] \wedge_L$
$primerColumna(trayectoria_0, recursos) \wedge_L actualizarRec(trayectoria_0, cooperan, apuestas, pagos, eventos, j))$

$\wedge_L$

$((\forall apuestasAlternativa : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ (|apuestasAlternativa| = |apuestas| \wedge_L$
$esMatrizDeApuestas(apuestasAlternativa)$

$\wedge_L$
$(\forall i : \mathbb{Z})\ ((0 \leq i < |apuestasAlternativa| \wedge_L i \neq individuo) \longrightarrow_L apuestasAlternativa[i] = apuestas[i])$

$\wedge_L$

$(\exists trayectoria_1 : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ ($

$(|trayectoria_1| = |apuestas|) \wedge_L (\forall k : \mathbb{Z})\ (0 \leq k < |trayectoria_1| \longrightarrow_L (|eventos[0]| = trayectoria_1[k] \wedge_L$
$primerColumna(trayectoria_1, recursos) \wedge_L$
$actualizarRec(trayectoria_1, cooperan, apuestasAlternativa, pagos, eventos, k))$
$\wedge_L trayectorias_1[individuo][|evento[0]| - 1] \leq trayectorias_0[individuo][|evento[0]| - 1]$

$))$

$)\}$

3. Demostraciones de Correctitud

Se quiere demostrar la correctitud del programa de la Figura 1.

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 $eventos$: $seq\langle \mathsf{Bool}\rangle$, out res: $\mathbb{R}$)

requiere $\{apuesta_c + apuesta_s = 1 \land pago_c > 0 \land pago_s > 0 \land apuesta_c > 0 \land apuesta_s > 0 \land recurso > 0\}$

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

Donde $\#\mathtt{apariciones}(eventos, T)$ es el auxiliar utilizado en la téorica, y $\#(eventos, T)$ es su abreviación.

```
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
```

Figura 1: Especificación a tratar con su implementación correspondiente

Para demostrar que esta especificación es correcta respecto de su implementación vamos a definir la implementación como el programa S y fragmentarlo de la siguiente manera:

- $S_1 \equiv \{$

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

$\}$

- $S_2 \equiv \{$

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

- $S_3 \equiv \{$

```
while (i < |eventos|) do
   S2;
endwhile
```

$\}$

El programa será correcto si y solo si el predicado $Pre \longrightarrow_L wp(S, Post)$ es verdadero, con Pre y Post como la precoindición y postcondición de la especificación. Para corroborar esto primero demostraremos lo siguiente:

1. $Pre \longrightarrow_L wp(S_1, P_c)$
2. $P_c \longrightarrow_L wp(S_3, Post)$

Donde $P_c \equiv \{i = 0 \land res = recursos\}$

Habiendo probado lo listado anteriormente, podremos decir que como valen los puntos 1 y 2, por monotonía, el predicado $Pre \longrightarrow_L wp(S, Post)$ es verdadero y por consecuente que el programa es correcto

3.1. $Pre \longrightarrow_L wp(S_1, P_c)$

Para demostrar esta implicación debemos asumir que la precondición de la especificación es verdadera y a partir de ahí llegar a $wp(S_1, P_c)$ por lo cual primero debemos obtener $wp(S_1, P_c)$

$wp(S_1, p_c) \equiv wp(res = recursos; i = 0, \{i = 0 \land res = recursos\})$

Por el axioma 1 sabemos que esto es equivalente a

$wp(res = recursos, wp(i = 0, \{i = 0 \land res = recursos\}))$
Entonces vamos a obtener $wp(i = 0, \{i = 0 \land res = recursos\})$

$wp(i = 0, \{i = 0 \land res = recursos\}) \equiv \{res = recursos\} \implies$

$wp(res = recursos, wp(i = 0, \{i = 0 \land res = recursos\})) \equiv wp(res = recursos, \{res = recursos\}) \equiv true$

Como $wp(S_1, P_c) \equiv true$ puedo confirmar que $Pre \longrightarrow_L wp(S_1, P_c)$ es verdadero.

3.2. $P_c \longrightarrow_L wp(S_3, Post)$

El programa S_3 al ser un ciclo, para poder demostrar esta implicación debemos utilizar el teorema de corrección de ciclo el cual marca que la tripla de Hoare

$\{P_c\}$

```
while (B) do
    S2;
endwhile
```

$\{Q_c\}$

$\equiv P_c \longrightarrow_L wp(S_3, Post)$

Es verdadera si, dado un predicado I, se cumplen las siguientes condiciones:

1. $P_c \implies I$
2. $\{I \land B\} S_2 \{I\}$
3. $I \land \neg B \implies Q_c$
4. $\{I \land B \land v_0 = fv\} S_2 \{fv < v_0\}$
5. $I \land fv \leq 0 \implies \neg B$

Siendo

- $I = \{0 \leq i \leq |eventos| \land_L$ $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_s * pago_s)^{apariciones(subseq(eventos,0,i),F)}\}$
- $B = \{i < |eventos|\}$
- $Q_c = Post$
- $fv = |eventos| - i$

A partir del invariante propuesto I se quieren probar las condiciones del teorema:

1. $P_c \implies I$

   $\{i = 0 \wedge res = recursos\} \implies \{0 \leq i \leq |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),F)}\}$

   utilizando que i=0 y que la función subseq(eventos,0,i=0) = $\langle\rangle$

   $\{i = 0 \wedge res = recursos\} \implies \{0 \leq i \leq |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(\langle\rangle,T)} * (apuestas_c * pago_c)^{apariciones(\langle\rangle,F)}\}$

   utilizando que $apariciones(\langle\rangle, T) = apariciones(\langle\rangle, F) = 0$

   $\{i = 0 \wedge res = recursos\} \implies \{0 \leq i \leq |eventos| \wedge_L res = recursos * (apuestas_c * pago_c)^0 * (apuestas_c * pago_c)^0 = recursos\}$

   lo cual es cierto dado que si $i = 0 \implies 0 \leq 0 \leq |eventos|$ y res=recursos $\implies$ res=recursos

2. $\{I \wedge B\} S_2 \{I\}$

   $B = \{i < |eventos|\}$
   $I = \{0 \leq i \leq |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),F)}\}$
   $I \wedge B = \{0 \leq i < |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),F)}$

   para probar esto, se utiliza el teorema que dice que $\{P\}S\{Q\}$ es valida sii $\{P\} \implies wp(S, P)$, lo cual en este problema sería:

   $\{I \wedge B\} \implies wp(S, I)$

   $\{i < |eventos| \wedge 0 \leq i \leq |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),F)}\}$
   $\implies wp(S, I)$

   $\{0 \leq i < |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),F)}\}$
   $\implies wp(S, I)$

   Definimos res como oldres, por lo tanto queremos ver que Wp(S,I) cumple :

   $Wp(S, I) = Wp(\textsf{if } ev[i] = True \textsf{ then } res = (oldres * A_c)P_c \textsf{ else } res = (oldres * A_s)P_s \textsf{ fi}; i := i + 1, I)$

   $Wp(S, I) = Wp(\textsf{if } ev[i] = True \textsf{ then } res = (oldres * A_c)P_c \textsf{ else } res = (oldres * A_s)P_s \textsf{ fi}, I^i_{i+1}) =$

   $def(ev[i]) \wedge_L ((ev[i] = True \wedge wp(res = (oldres * A_c)P_c, I^i_{i+1})) \vee (\neg ev[i] = True \wedge wp(res = (oldres * A_s)P_s, I^i_{i+1})))$

   Utilizando $def(ev[i]) = 0 \leq i < |eventos|$ se simplifica $(0 \leq i < |eventos| \wedge 0 \leq i + 1 \leq |eventos|) = 1 \leq i + 1 \leq |eventos|$ y wp(S,I) queda:

   $Wp(S, I) = 0 \leq i < |eventos| \wedge ((ev[i] = True \wedge wp(res = (oldres * A_c)P_c, I^i_{i+1})) \vee (\neg ev[i] = True \wedge wp(res = (oldres * A_s)P_s, I^i_{i+1})))$

   $Wp(S, I) = 0 \leq i < |eventos| \wedge ((ev[i] = True \wedge (I^i_{i+1})^{res}_{(oldres*A_c)*P_c})) \vee (\neg ev[i] = True \wedge {I^i_{i+1}}^{res}_{(oldres*A_s)*P_s}))$

$Wp(S,I) = 0 \leq i < |eventos| \wedge ((ev[i] = True \wedge_L 0 \leq i+1 \leq |eventos| \wedge_L$
$(oldres * A_c)P_c = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i+1),T)} *$
$(apuestas_s * pago_s)^{apariciones(subseq(eventos,0,i+1),F)}) \vee (\neg ev[i] = True \wedge_L 0 \leq i+1 \leq |eventos| \wedge_L$
$(oldres * A_s)P_s = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i+1),T)} *$
$(apuestas_s * pago_s)^{apariciones(subseq(eventos,0,i+1),F)}))$

Para simplificar veamos que:

$recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i+1),T)} * (apuestas_s * pago_s)^{apariciones(subseq(eventos,0,i+1),F)}$
$= recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_c * pago_c)^{apariciones(subseq(eventos,i,i+1),T)} *$
$(apuestas_s * pago_s)^{apariciones(subseq(eventos,0,i),F)} * (apuestas_s * pago_s)^{apariciones(subseq(eventos,i,i+1),F)} =$
$oldres * (apuestas_c * pago_c)^{apariciones(subseq(eventos,i,i+1),T)} * (apuestas_s * pago_s)^{apariciones(subseq(eventos,i,i+1),F)}$

por lo tanto:

$Wp(S,I) = 0 \leq i < |eventos| \wedge ((ev[i] = True \wedge$
$(oldres * A_c)P_c = oldres * (apuestas_c * pago_c)^{apariciones(subseq(eventos,i,i+1),T)} *$
$(apuestas_s * pago_s)^{apariciones(subseq(eventos,i,i+1),F)}) \vee$
$(\neg ev[i] = True \wedge (oldres * A_s)P_s = oldres * (apuestas_c * pago_c)^{apariciones(subseq(eventos,i,i+1),T)} *$
$(apuestas_s * pago_s)^{apariciones(subseq(eventos,i,i+1),F)}$

Lo cual nos dice que en caso de que ev[i]=True sea cierto entonces apariciones(subseq(eventos,i,i+1),T)=1 y se cumpliría $(oldres * A_c)P_c = oldres * (apuestas_c * pago_c)^1 * (apuestas_s * pago_s)^0$, en caso contrario se cumpliría $(oldres * A_s)P_s = oldres * (apuestas_c * pago_c)^0 * (apuestas_s * pago_s)^1$

3. $I \wedge \neg B \implies Q_c$

   $I = \{0 \leq i \leq |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_s * pago_s)^{apariciones(subseq(eventos,0,i),F)}\},$
   $\neg B = \{i \geq |eventos|\}$
   $Q_c = \{res = recursos * (apuesta.c * pago.c)^{\#apariciones(eventos,True)} * (apuesta.s * pago.s)^{\#apariciones(eventos,False)}\}$

   Utilizando que $I \wedge \neg B \implies 0 \leq i \leq |eventos| \wedge i \geq |eventos| \implies i = |eventos|$
   $i = |eventos| \implies subseq(eventos, 0, i) = eventos$

   podemos reducir $I \wedge \neg B$ a

   $I \wedge \neg B = \{i = |eventos| \wedge res = recursos(apuestas_c * pago_c)^{apariciones(eventos,T)}$
   $*(apuestas_s * pago_s)^{apariciones(eventos,F)} \equiv Q_c\}$

   lo cual sería equivalente a

   $i = |eventos| \wedge Q_c \implies Q_c$

   que es lo que queriamos probar.

4. $\{I \wedge B \wedge v_0 = f_v\}S_2\{fv < v_0\}$

   $fv = |eventos| - i$
   $v_o = f_v$ $I = \{0 \leq i \leq |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_s * pago_s)^{apariciones(subseq(eventos,0,i),F)}\}$
   $B = \{i < |eventos|\}$
   $(I \wedge B \wedge v_o = f_v) = \{0 \leq i < |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),F)} \wedge$
   $f_v = |eventos| - i \wedge v_o = f_v$

   Esto es equivalente a demostrar que $\{I \wedge B \wedge v_0 = fv\} \implies wp(S, f_v < v_o)$ veamos como queda la wp :

   $wp(if....endif; i := i+1, |eventos| - i < v_o)$
   $wp(if....endif, wp(i := i+1, |eventos| - i < v_o))$
   $wp(if...endif, (|eventos| - i)^{i}{}_{i+1} < v_o)$

   $(eventos[i] = True \wedge wp(res = (res * apuesta.c) * pago.c, |eventos| - (i+1) < v_o)) \vee (eventos[i] = False \wedge wp(res = (res * apuesta.s) * pago.s, |eventos| - (i+1) < v_o))$

   $(eventos[i] = True \wedge (|eventos| - (i+1))^{res}{}_{(res*apuesta.c)*pago.c} < v_o) \vee$
   $(eventos[i] = False \wedge (|eventos| - (i+1))^{res}{}_{(res*apuesta.s)*pago.s} < v_o)$

   $(eventos[i] = True \wedge (|eventos| - (i+1)) < v_o) \vee (eventos[i] = False \wedge (|eventos| - (i+1)) < v_o)$

   $(eventos[i] = True \vee eventos[i] = False) \wedge (|eventos| - (i+1)) < v_o$

   $(|eventos| - (i+1)) < v_o$

   Como $f_v = v_0$ equivale a —eventos— - i, reemplazamos v_0 con esa expresión:

   $(|eventos| - (i+1)) < |eventos| - i$
   $-(i+1) < -i$
   $i+1 > i$
   Lo cual es verdadero. Por lo tanto, demostramos que:
   $\{I \wedge B \wedge v_0 = fv\} \implies wp(S, f_v < v_o)$

5. $I \wedge fv \leq 0 \implies \neg B$

   $fv = |eventos| - i \leq 0 \implies |eventos| \leq i$
   $I = \{0 \leq i \leq |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),F)}\}$
   $\neg B = \{i \geq |eventos|\}$
   Utilizando $|eventos| \leq i \wedge 0 \leq i \leq |eventos| \implies \{i = |eventos|\}$ se obtiene :

   $\{i = |eventos| \wedge_L$
   $res = recursos(apuestas_c * pago_c)^{apariciones(subseq(eventos,0,i),T)} * (apuestas_s * pago_s)^{apariciones(subseq(eventos,0,i),F)}\}$
   $\implies \{i \geq |eventos|\}$

   Lo cual es cierto dado que $\{i = |eventos|\} \implies \{i \geq |eventos|\}$

3.3. Conclusión

Como dijimos previamente, habiendo demostrado que valen los puntos 1 y 2 el predicado $Pre \longrightarrow_L wp(S, Post)$ es verdadero por monotonía y por lo tanto programa es correcto

4. Anexo

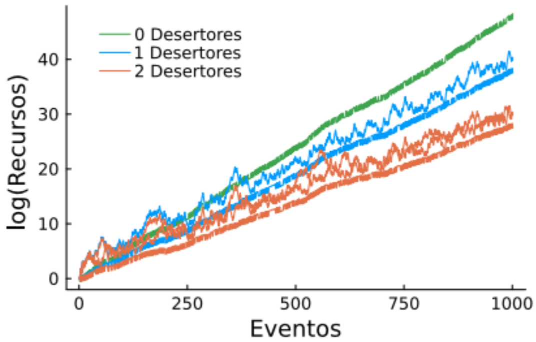

Figura 2: Recursos resultantes a lo largo de los eventos para 100 participantes, en verde se ve el caso donde todos los participantes aportan al fondo comun, en azul y rojo los casos con 1 y 2 desertores .

En la Figura 2 se ven los resultados obtenidos en un trabajo, donde se obtiene que al aumentar la cantidad de desertores, a pesar de que estos obtienen mejores resultados que el grupo participante del fondo común, los recursos para desertores y contribuyentes son inferiores a largo plazo que si todos hubiesen contribuido al fondo común. En caso de implementar un algoritmo como el tratado en este trabajo se buscaría obtener resultados consistentes con estos.