



Trabajo práctico 1

Especificación y WP

21/4/2024

AED

SCARAMOUCHE & LOS FANDANGO

Integrante	LU	Correo electrónico
Calo, Agustín	390/23	caloagustin4@gmail.com
Seri, Rafael Nicolás	362/23	rafaelnicoseri@gmail.com
Pintos Oliveira, Sol María Marcela	428/23	solpintosoliveira@gmail.com
Páez Torrico, Santiago	713/23	santiagopaez122@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

1. Especificación

1.1. redistribucionDeLosFrutos

```
proc redistribucionDeLosFrutos (in recursos: seq⟨ℝ⟩, in cooperan: seq⟨Bool⟩) : seq⟨ℝ⟩
  requiere {|recursos| = |cooperan|}
  requiere {todosPositivos(recursos)}
  asegura {|res| = |recursos|}
  asegura {(∀i : ℤ) (0 ≤ i < |res| →L res[i] = if cooperan[i] then totalARepartir(recursos, cooperan) else recursos[i] +
    totalARepartir(recursos, cooperan) fi)}

aux totalARepartir (recursos: seq⟨ℝ⟩, cooperan: seq⟨Bool⟩) : ℝ =
  (∑i=0|recursos|-1 if cooperan[i] then recursos[i] else 0 fi)/|recursos|;
```

1.2. trayectoriaDeLosFrutosIndividualesALargoPlazo

```
proc trayectoriaDeLosFrutosIndividualesALargoPlazo (inout trayectorias: seq⟨seq⟨ℝ⟩⟩, in cooperan: seq⟨Bool⟩, in apues-
tas: seq⟨seq⟨ℝ⟩⟩, in pagos: seq⟨seq⟨ℝ⟩⟩, in eventos: seq⟨seq⟨ℝ⟩⟩)
  requiere {trayectorias = old(trayectorias)}
  requiere {|cooperan| = |pagos| = |apuestas| = |eventos| = |trayectorias|}
  requiere {(∀i : ℤ) (0 ≤ i < |pagos| →L todosPositivos(pagos[i]) ∧ todosPositivos(apuestas[i]))}
  requiere {(∀i : ℤ) (0 ≤ i < |trayectoria| →L trayectoria[i][0] > 0)}
  requiere {(∀j : ℤ) (0 ≤ j < |apuestas| →L ∑i=0|eventos[j]|-1 apuestas[j][i] = 1)}
  asegura {|trayectorias| = |old(trayectorias)|}
  asegura {(∀i : ℤ) (0 ≤ i < |old(trayectorias)| →L |trayectorias[i]| = |old(trayectorias)[i]| + |eventos[i]|)}
  asegura {(∀i : ℤ) (0 ≤ i < |old(trayectorias)| →L trayectorias[i][0] = old(trayectorias)[i][0])}
  asegura {(∀i : ℤ) (0 ≤ i < |old(trayectorias)| →L (∀j : ℤ) (0 ≤ j < |eventos[i]| →L trayectorias[i][j + 1] =
    if cooperan[i] then distribucion(aporteIndividual(trayectorias, apuestas, pagos, eventos, cooperan, i, j)) else aporte-
    Individual(trayectorias, apuestas, pagos, eventos, i, j) + distribucion(trayectorias, apuestas, pagos, eventos, cooperan,
    j) fi))}

aux distribucion (trayectorias: seq⟨seq⟨ℝ⟩⟩, apuestas: seq⟨seq⟨ℝ⟩⟩, pagos: seq⟨seq⟨ℝ⟩⟩, eventos: seq⟨seq⟨ℝ⟩⟩, cooperan:
seq⟨Bool⟩, m: ℕ) : ℝ =
  (∑k=0|cooperan|-1 if cooperan[k] then aporteIndividual(trayectorias, apuestas, pagos, eventos, k, m) else 0 fi)/|cooperan|;
```

1.3. trayectoriaExtrañaEscalera

```
proc trayectoriaExtrañaEscalera (in trayectorias: seq⟨ℝ⟩) : Bool
  requiere {|trayectoria| > 0}
  asegura {res = True ⇔ |trayectoria| = 1 ∨ (trayectoria[0] > trayectoria[1] ∧ ¬maximoLocal(trayectoria) ∧
    trayectoria[|trayectoria|-1] < trayectoria[|trayectoria|-2]) ∨ (trayectoria[|trayectoria|-1] > trayectoria[|trayectoria|-
    2] ∧ ¬maximoLocal(trayectoria) ∧ trayectoria[0] < trayectoria[1] ∨ (∃i : ℤ) (0 < i < |trayectoria|-1 ∧L (trayectoria[i] >
    trayectoria[i + 1] ∧ trayectoria[i] > trayectoria[i - 1]) ∧ (∀i : ℤ) (0 < j < |trayectoria| - 1 ∧L (trayectoria[j] >
    trayectoria[j + 1] ∧ trayectoria[j] > trayectoria[j - 1]) →L j = i)))}

pred maximoLocal (s: seq⟨ℝ⟩) {
  (∃i : ℤ) (0 < i < |s| - 1 ∧L (s[i] > s[i + 1] ∧ s[i] > s[i - 1]))
}
```

1.4. individuoDecideSiCooperarONo

```
proc individuoDecideSiCooperarONo (in individuo: ℕ, in recursos: seq⟨ℝ⟩, inout cooperan: seq⟨Bool⟩, in apuestas: seq⟨seq⟨ℝ⟩⟩,
in pagos: seq⟨seq⟨ℝ⟩⟩, in eventos: seq⟨seq⟨ℝ⟩⟩)
  requiere {cooperan = old(cooperan)}
  requiere {|cooperan| = |recursos| = |apuestas| = |pagos| = |eventos|}
  requiere {(∀i : ℤ) (0 ≤ i < |apuestas| →L todosPositivos(recursos) ∧ todosPositivos(apuestas[i]) ∧
    todosPositivos(pagos[i]))}
  requiere {0 ≤ individuo < |cooperan|}
  asegura {|cooperan| = |old(cooperan)|}
  asegura {(∀i : ℤ) (0 ≤ individuo < |cooperan| ∧ i ≠ individuo →L cooperan[i] = old(cooperan)[i])}
  asegura {(∃s, p : seq⟨seq⟨ℝ⟩⟩) (|s| = |p| = |cooperan| ∧ (∀n : ℤ) (0 ≤ n < |s| ∧L |s[n]| = |p[n]| = (|eventos| + 1) ∧
    s[n][0] = p[n][0] = recursos[n] ∧ (∀k : ℤ) (0 < k < |s[n]| →L s[n][k] = (if old(cooperan)[k] ∨ k = individuo then 0 else
    aporteIndividual(s, apuestas, pagos, eventos, n, k) fi) + distribucionCoop(s, apuestas, pagos, eventos, old(cooperan),
    k, individuo) ∧ p[n][k] = (if ¬(old(cooperan)[k]) ∨ k = individuo then aporteIndividual(p, apuestas, pagos, eventos, n, k)
```

```

else 0 fi) + distribucionNoCoop(p, apuestas, pagos, eventos, old(cooperan), k, individuo)))  $\wedge_L$  cooperan[individuo] =
true  $\leftrightarrow$  p[individuo][p[individuo] - 1]  $\leq$  s[individuo][s[individuo] - 1])}

aux distribucionCoop (trayectorias: seq(seq( $\mathbb{R}$ )), apuestas: seq(seq( $\mathbb{R}$ )), pagos: seq(seq( $\mathbb{R}$ )), eventos: seq(seq( $\mathbb{N}$ )), cooperan: seq(Bool), m:  $\mathbb{N}$ , individuo:  $\mathbb{N}$ ) :  $\mathbb{R}$  =
( $\sum_{k=0}^{|cooperan|-1}$  if cooperan[k]  $\vee$  k = individuo then aporteIndividual(trayectorias, apuestas, pagos, eventos, k, m)
else 0 fi)/|cooperan|;
aux distribucionNoCoop (trayectorias: seq(seq( $\mathbb{R}$ )), apuestas: seq(seq( $\mathbb{R}$ )), pagos: seq(seq( $\mathbb{R}$ )), eventos: seq(seq( $\mathbb{N}$ )), cooperan: seq(Bool), m:  $\mathbb{N}$ , individuo:  $\mathbb{N}$ ) :  $\mathbb{R}$  =
( $\sum_{k=0}^{|cooperan|-1}$  if cooperan[k]  $\wedge$  k  $\neq$  individuo then aporteIndividual(trayectorias, apuestas, pagos, eventos, k, m)
else 0 fi)/|cooperan|;

```

1.5. individuoActualizaApuesta

```

proc individuoActualizaApuesta (in individuo:  $\mathbb{N}$ , in recursos: seq( $\mathbb{R}$ ), in cooperan: seq(Bool), in/out apuestas: seq(seq(Bool)),
in pagos: seq(seq( $\mathbb{R}$ )), in eventos: seq(seq( $\mathbb{N}$ )))
  requiere {apuestas = old(apuestas)}
  requiere {|cooperan| = |recursos| = |apuestas| = |pagos| = |eventos|}
  requiere {0  $\leq$  individuo < |cooperan|}
  asegura {|apuestas| = |old(apuestas)|}
  asegura {( $\forall i : \mathbb{Z}$ ) ( $-1 < i < |apuestas| \rightarrow_L |apuestas[i]| = |old(apuestas)[i]|$ )}
  asegura {( $\exists p, s : seq(seq(\mathbb{R}))$ )( $\exists mejorApuesta : seq(\mathbb{R})$ )( $(\forall posibleApuesta : seq(\mathbb{R})) (ultElem(p, recursos, old(apuestas),$ 
pagos, cooperan, eventos, individuo, mejorApuesta)  $\geq$  ultElem(s, recursos, old(apuestas), pagos, cooperan, eventos,
individuo, posibleApuesta)  $\rightarrow_L apuestas[individuo] = mejorApuesta$ ))}

aux ultElem (t: seq(seq( $\mathbb{R}$ )), recursos: seq( $\mathbb{R}$ ), apuestas: seq(seq( $\mathbb{R}$ )), pagos: seq(seq( $\mathbb{R}$ )), cooperan: seq(Bool), eventos:
seq( $\mathbb{N}$ ), individuo:  $\mathbb{N}$ , posibleApuesta: seq( $\mathbb{N}$ )) :  $\mathbb{R}$  =
if trayectoriaPosible(t, recursos, apuestas, pagos, cooperan, eventos, individuo, posibleApuesta) then t[individuo][t[individuo] -
1] else -1 fi;
pred trayectoriaPosible (t: seq(seq( $\mathbb{R}$ )), recursos: seq( $\mathbb{R}$ ), apuestas: seq(seq( $\mathbb{R}$ )), pagos: seq(seq( $\mathbb{R}$ )), cooperan: seq(Bool),
eventos: seq( $\mathbb{N}$ ), individuo:  $\mathbb{N}$ , posibleApuesta: seq( $\mathbb{N}$ )) {
  |posibleApuesta| = |apuestas[individuo]|  $\wedge$  sumElem(posibleApuesta) = 1  $\wedge$  todosPositivos(posibleApuesta)  $\wedge$  |t| = |cooperan|  $\wedge$ 
( $\forall i : \mathbb{Z}$ ) ( $-1 < i < |t| \wedge_L |t[i]| = (|eventos| + 1) \wedge |t[i][0]| = |recursos[i]| \wedge (\forall j : \mathbb{Z}) (0 \leq j < |t[i]| \rightarrow_L |t[i][j+1]| =$ 
(if cooperan[i] then 0 else aporteIndDiferido(t[i], apuestas[i], pagos[i], eventos[i], i, j, individuo, posibleApuesta) fi) +
distribucionDiferida(t[i], apuestas[i], pagos[i], eventos[i], cooperan, i, j, individuo, posibleApuesta)))
}
aux aporteIndDiferido (trayectoria: seq( $\mathbb{R}$ ), apuestas: seq( $\mathbb{R}$ ), pagos: seq( $\mathbb{R}$ ), eventos: seq( $\mathbb{N}$ ), k:  $\mathbb{N}$ , m:  $\mathbb{N}$ , individuo:  $\mathbb{N}$ ,
apuestaInd: seq( $\mathbb{R}$ )) :  $\mathbb{R}$  =
if k = individuo then trayectorias[m]  $\cdot$  apuestaInd[eventos[m]]  $\cdot$  pagos[eventos[m]] else trayectorias[m]  $\cdot$  apuestas[eventos[m]]  $\cdot$ 
pagos[eventos[m]] fi;
aux distribucionDiferida (trayectoria: seq( $\mathbb{R}$ ), apuestas: seq( $\mathbb{R}$ ), pagos: seq( $\mathbb{R}$ ), eventos: seq( $\mathbb{N}$ ), cooperan: seq(Bool), k:
 $\mathbb{N}$ , m:  $\mathbb{N}$ , individuo:  $\mathbb{N}$ , apuestaInd: seq( $\mathbb{R}$ )) :  $\mathbb{R}$  =
( $\sum_{k=0}^{|cooperan|-1}$  if cooperan[k] then aporteIndDiferido(trayectorias, apuestas, pagos, eventos, k, m, individuo) else 0 fi)/|cooperan|;
aux sumElem (s: seq( $\mathbb{R}$ )) :  $\mathbb{R}$  =
 $\sum_{i=0}^{|s|-1} s[i]$ ;

```

Auxiliares y predicados globales

```

pred todosPositivos (s: seq( $\mathbb{R}$ )) {
  ( $\forall i : \mathbb{Z}$ ) (0  $\leq i < |s| \rightarrow_L s[i] > 0$ )
}
aux aporteIndividual (trayectorias: seq(seq( $\mathbb{R}$ )), apuestas: seq(seq( $\mathbb{R}$ )), pagos: seq(seq( $\mathbb{R}$ )), eventos: seq(seq( $\mathbb{N}$ )), k:  $\mathbb{N}$ ,
m:  $\mathbb{N}$ ) :  $\mathbb{R}$  = trayectorias[k][m]  $\cdot$  apuestas[k][eventos[k][m]]  $\cdot$  pagos[k][eventos[k][m]] ;

```

2. Demostraciones de correctitud

Demostrar que la siguiente especificación es correcta respecto de su implementación.

La función **frutoDelTrabajoPuramenteIndividual** calcula, para el ejemplo de apuestas al juego de cara o seca, cuánto se ganaría si se juega completamente solo. Se contempla que el evento True es cuando sale cara.

```

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(Bool), out res:  $\mathbb{R}$ )
  requiere {apuestac + apuestas = 1  $\wedge$  pagoc > 0  $\wedge$  pagos > 0  $\wedge$  apuestac > 0  $\wedge$  apuestas > 0  $\wedge$  recurso > 0}
  asegura {res = recurso(apuestac, pagoc)#apariciones(eventos, T)(apuestas, pagos)#apariciones(eventos, F)}

```

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

```

1  res := recurso
2  i := 0
3  while (i < |eventos|) do
4    if eventos[i] then
5      res := (res * apuesta.c) * pago.c
6    else
7      res := (res * apuesta.s) * pago.s
8    endif
9    i := i + 1
10 endwhile

```

Como el programa que nos dan cuenta con ciclos, tenemos que probar lo siguiente:

1. $Pre \rightarrow_L wp(\text{código previo al ciclo}, P_c)$
2. $P_c \rightarrow_L wp(\text{ciclo}, Q_c)$
3. $Q_c \rightarrow_L Post$

Para ello proponemos lo siguiente:

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

Equivalencias entre Q_c propuesto y asegura dado:

$asegura : res = recurso * (apuesta_c * pago_c)^{\#apariciones(eventos, T)} (apuesta_s * pago_s)^{\#apariciones(eventos, F)} \equiv recurso * (apuesta_c * pago_c)^{\sum_{i=0}^{|eventos|-1} \text{if } eventos[i]=T \text{ then } 1 \text{ else } 0 \text{ fi}} * (apuesta_s * pago_s)^{\sum_{i=0}^{|eventos|-1} \text{if } eventos[i]=F \text{ then } 1 \text{ else } 0 \text{ fi}}$

Por propiedad de potenciación: $x^{f(n)+f(m)} = x^{f(n)} * x^{f(m)}$ luego $x^{\sum_{i=0}^n f(i)} = \prod_{i=0}^n x^{f(i)}$

$\equiv recurso * \prod_{j=0}^{|eventos|-1} \text{if } eventos[j] = T \text{ then } (apuesta_c * pago_c) \text{ else } 1 \text{ fi} * \prod_{j=0}^{|eventos|-1} \text{if } eventos[j] = F \text{ then } (apuesta_s * pago_s) \text{ else } 1 \text{ fi}$

Si $A = \{0 \leq j < |eventos| - 1 : eventos[j] = T\}$ y $B = \{0 \leq j < |eventos| - 1 : eventos[j] = F\}$ tengo que $A \cap B = \emptyset$ y como en las productorias el predicado del else es 1 (neutro multiplicativo), vale que si las juntamos queda:

$asegura : res = recurso * \prod_{j=0}^{|eventos|-1} \text{if } eventos[j] \text{ then } apuesta_c * pago_c \text{ else } apuesta_s * pago_s \text{ fi} \blacksquare$

Demostración $Pre \rightarrow_L wp(\text{código previo al ciclo}, P_c)$

Vamos a utilizar los axiomas 1 (asignación) y 3 (composicional) vistos en la teórica.

1. $Pre \rightarrow_L wp(res := recurso; i := 0, P_c) \stackrel{Axioma\ 3}{\equiv} wp(res := recurso; wp(i := 0, P_c))$
2. $wp(i := 0, \{i = 0 \wedge res = recurso \wedge apuesta_c + apuesta_s = 1 \wedge pago_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0\}) \stackrel{Axioma\ 1}{\equiv} def(0) \wedge_L 0 = 0 \wedge res = recurso \wedge apuesta_c + apuesta_s = 1 \wedge pago_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0 \equiv res = recurso \wedge apuesta_c + apuesta_s = 1 \wedge pago_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0 \equiv E_1$
3. $wp(res := recurso, E_1) \stackrel{Axioma\ 1}{\equiv} def(recurso) \wedge_L recurso = recurso \wedge apuesta_c + apuesta_s = 1 \wedge pago_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0 \equiv apuesta_c + apuesta_s = 1 \wedge pago_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0 \equiv E_2$

Y como $Pre \equiv E_2$ tenemos que $Pre \rightarrow E_2 \blacksquare$

Demostración $P_c \rightarrow_L wp(\text{ciclo}, Q_c)$

Por Teorema del Invariante, vamos a mostrar que la tripla de Hoare:

$$\{P_c\} while... \{Q_c\}$$

es válida. Entonces tenemos siguiente:

1. $P_c \implies I$

2. $\{I \wedge B\}S\{I\}$
3. $I \wedge \neg B \implies Q_c$
4. $\{I \wedge v_0 = fv\}S\{fv < v_0\}$
5. $I \wedge fv \leq 0 \implies \neg B$

Demostración $P_c \implies I$:

$P_c \equiv \{i = 0 \wedge res = recurso \wedge apuesta_c + apuesta_s = 1 \wedge pago_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0\}$
 $I \equiv \{0 \leq i \leq |eventos| \wedge res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}\}$

Queremos probar que vale I :

- $0 \leq i \leq |eventos|$
vale, porque $i = 0$ y trivialmente sabemos que se encuentra entre 0 y la longitud de la secuencia eventos.
- $res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}$
tenemos que $i = 0$ y $res = recurso$ entonces, como $i = 0$
 $res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi} =$
 $recurso * \prod_{j=0}^{0-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi} = recurso * \prod_{j=0}^{-1}$, que es un rango vacío,
 es decir, valdrá 1 y teníamos en P_c que $res = recurso$ entonces, $recurso = recurso * 1 = recurso$

Tenemos entonces que $P_c \implies I$ ■

Demostración $\{I \wedge B\}S\{I\}$

Vamos a utilizar los axiomas 1 (asignación) y 4 (guardas) vistos en la teórica

$B \equiv \{i < |eventos|\}$

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

Vemos si $I \wedge B \implies wp(\text{if } \dots; i := i + 1, I)$

- $wp(i := i + 1, I) \stackrel{\text{Axioma 1}}{\equiv} def(i + 1) \wedge_L I_{i+1}^i \equiv 0 \leq i + 1 \leq |eventos| \wedge_L res = recurso * \prod_{j=0}^i \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi} \equiv E_3$
- $wp(\text{if } \dots, E_3) \stackrel{\text{Axioma 4}}{\equiv} def(eventos[i]) \wedge_L ((eventos[i] \wedge wp(res := (res * apuesta.c) * pago.c, E_3) \vee (\neg eventos[i] \wedge wp(res := (res * apuesta.s) * pago.s, E_3)))) \equiv$
 $((eventos[i] \wedge (res * apuesta.c) * pago.c = recurso * \prod_{j=0}^i \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}) \vee$
 $(\neg eventos[i] \wedge (res * apuesta.s) * pago.s = recurso * \prod_{j=0}^i \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi})) \equiv$
 $((eventos[i] \wedge res = \frac{1}{apuesta.c * pago.c} * recurso * \prod_{j=0}^i \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}) \vee$
 $(\neg eventos[i] \wedge res = \frac{1}{apuesta.s * pago.s} * recurso * \prod_{j=0}^i \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}))$
 Necesitamos llevar esto a algo similar al invariante, nos damos cuenta que $\frac{1}{apuesta.c * pago.c}$ y $\frac{1}{apuesta.s * pago.s}$ son equivalentes a $\frac{1}{\text{if } eventos[i] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}}$ respectivamente, entonces reescribimos los términos:
 Aplicamos $(P \wedge Q) \vee (\neg P \wedge Q) \equiv Q$
- $0 \leq i + 1 \leq |eventos| \wedge_L res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi} \equiv E_4$
 Chequeamos si $I \wedge B \implies E_4$
- $i \leq i + 1 \leq |eventos|$, el invariante afirma que $0 \leq i \leq |eventos|$ y la guarda afirma que $i < |eventos|$
- $res = recurso * \prod_{j=0}^{i-1} \text{if } eventos[j] \text{ then } apuesta.c * pago.c \text{ else } apuesta.s * pago.s \text{ fi}$, que lo afirma el invariante. Entonces $\{I \wedge B\}S\{I\}$ ■

Demostración $I \wedge \neg B \implies Q_c$

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

$\neg B \equiv \{i \geq |\text{eventos}|\}$

$Q_c \equiv \{\text{res} = \text{recurso} * \prod_{j=0}^{|\text{eventos}|-1} \text{if } \text{eventos}[j] \text{ then } \text{apuesta.c} * \text{pago.c} \text{ else } \text{apuesta.s} * \text{pago.s} \text{ fi}\}$

Queremos probar que vale Q_c :

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

- $0 \leq i \leq |\text{eventos}| \wedge i \geq |\text{eventos}|$, luego $i = |\text{eventos}|$

Y al reemplazar el valor de i en I , obtenemos:

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

Luego $I \wedge \neg B \implies Q_c$ ■

Demostración $\{I \wedge v_0 = fv\}S\{fv < v_0\}$

Vamos a utilizar los axiomas 1 (asignación), 3 (composicional) y 4(guardas) vistos en la teórica

$\{I \wedge B \wedge v_0 = |\text{eventos}| - i\}S\{|\text{eventos}| - i < v_0\}$

Vemos si $I \wedge B \wedge v_0 = |\text{eventos}| - i \implies wp(\text{if } \dots; i := i+1, |\text{eventos}| - i < v_0) \stackrel{\text{Axioma 3}}{\equiv} wp(\text{if } \dots; wp(i := i+1, |\text{eventos}| - i < v_0))$

$wp(i := i+1, |\text{eventos}| - i < v_0) \stackrel{\text{Axioma 1}}{\equiv} \text{def}(i+1) \wedge_L |\text{eventos}| - i - 1 < v_0 \equiv |\text{eventos}| - i < v_0 + 1$

$wp(\text{if } \dots, |\text{eventos}| - i < v_0 + 1) \stackrel{\text{Axioma 4}}{\equiv} \text{def}(\text{eventos}[i]) \wedge_L ((\text{eventos}[i] \wedge wp(\text{res} := \text{res} * \text{apuesta.c} * \text{pago.c}, |\text{eventos}| - i < v_0 + 1)) \vee (\neg \text{eventos}[i] \wedge wp(\text{res} := \text{res} * \text{apuesta.s} * \text{pago.s}, |\text{eventos}| - i < v_0 + 1))) \equiv ((\text{eventos}[i] \wedge |\text{eventos}| - i < v_0 + 1) \vee (\neg \text{eventos}[i] \wedge |\text{eventos}| - i < v_0 + 1))$

Aplicamos $(P \wedge Q) \vee (\neg P \wedge Q) \equiv Q$

Vemos si vale la implicación

$I \wedge i < |\text{eventos}| \wedge |\text{eventos}| - i = v_0 \implies |\text{eventos}| - i < v_0 + 1$, porque $A = B \implies A < B + 1$

Entonces $\{I \wedge v_0 = fv\}S\{fv < v_0\}$ ■

Demostración $I \wedge fv \leq 0 \implies \neg B$

Por la definición de fv tenemos:

- $I \wedge |\text{eventos}| - i \leq 0$
si sumamos i de ambos lados de la igualdad,
 $I \wedge |\text{eventos}| - i + i \leq 0 + i \equiv I \wedge |\text{eventos}| \leq i$

Como $|\text{eventos}| \leq i \equiv \neg B$ es una implicación del tipo $P \wedge Q \implies P$, tenemos $I \wedge \neg B \implies \neg B$ ■

Demostración $Q_c \longrightarrow_L \text{Post} = \text{asegura} \equiv Q_c$

Como la implementación termina con el ciclo y el Q_c equivale al asegura (como lo hemos demostrado anteriormente), por monotonía sabemos que $\text{Pre} \implies wp(\text{programa completo}, \text{Post})$, por lo tanto decimos que la especificación es correcta respecto a su implementación. ■