

Trabajo practico 1

9 de mayo de 2024

Algoritmos y estructuras de datos

Grupo: LEQNFVIO

Integrante	LU	Correo electrónico
Festa, Bruno Alejandro	690/23	brunofesta2004@gmail.com
Tocto, Anthony Armando	342/22	armando.ddvv@gmail.com
Barg Oyanguren, Ciro	597/23	cirobargoyanguren@gmail.com
Zapata, Juan Francisco	1191/23	juanfzapata123@gmail.com

1. Ejercicio 1

```

proc redistribucionDeLosFrutos (in recursos:seq⟨ℝ⟩, in cooperan:seq⟨Bool⟩) : ℝ
  requiere {|cooperan| = |recursos|}
  requiere {(∀i : ℤ)(0 ≤ i < |recursos| →L recursos[i] ≥ 0)}
  asegura {(∃L : seq⟨ℝ⟩)(esListaDeRecursosAntesDelFondoComun(recursos, cooperan, L) ∧L res = L)}

  pred esDistribucionDeRecursos (cooperan : seq⟨bool⟩, recursos : seq⟨ℝ⟩, L : seq⟨ℝ⟩) {
    (∀i : ℤ)(0 ≤ i < |L| →L if cooperan[i] = true then L[i] =  $\frac{\text{fondoComun}(\text{recursos}, \text{cooperan})}{|\text{recursos}|}$ 
    else L[i] =  $\frac{\text{fondoComun}(\text{recursos}, \text{cooperan})}{|\text{recursos}|} + \text{recursos}[i]$ ) fi
  }

  aux fondoComun (In recursos : ℝ, In cooperan : Bool) : ℝ =  $\sum_{i=0}^{|\text{recursos}|-1} \text{if } (\text{cooperan}[i] = \text{true}) \text{ then } \text{recursos}[i] \text{ else } 0$  fi ;

```

2. Ejercicio 2

```

proc trayectoriaDeLosFrutosIndividualesALargoPlazo (inout trayectoria : seq⟨ℝ⟩, in cooperan:seq⟨Bool⟩,
in apuestas : seq⟨seq⟨ℝ⟩⟩, in pagos : seq⟨seq⟨ℝ⟩⟩, in eventos : seq⟨seq⟨ℝ⟩⟩) :

  requiere {|cooperan| = |trayectoria| = |eventos| = |pagos| = |apuestas|}

  requiere {(∀i : ℤ)(0 ≤ i < |apuestas| →L  $\sum_{k=0}^{|\text{apuestas}[i]|} \text{apuestas}[i][k] = 1$ )}}

  requiere {(∀i, e : ℤ)(0 ≤ i < |pagos| ∧ 0 ≤ e < |pagos[i]| →L pagos[i][e] > 0)}

  requiere {(∀i, e : ℤ)(0 ≤ i < |trayectoria| ∧ 0 ≤ e < |trayectoria[i]| →L trayectoria[i][e] > 0)}

  asegura {(∀individuo, nroDeEvento : ℤ)(0 ≤ individuo < |recursos| ∧ 0 < nroDeEvento
  < |cooperan| →L (∃Lista : seq⟨ℝ⟩)(esListaDespuesDeDistribuirElFondoComun(individuo, nroDeEvento, pagos, apuestas,
  , trayectoria, cooperan) ∧ trayectoria[individuo][nroDeEvento] = Lista[individuo]))}

  pred esListaDespuesDeDistribuirElFondoComun (individuo : ℤ, nroDeEvento : ℤ, pagos : seq⟨seq⟨ℝ⟩⟩, apuesta
  seq⟨seq⟨ℝ⟩⟩,
  listaDistribuida:seq⟨ℝ⟩, trayectoria:seq⟨seq⟨ℝ⟩⟩, cooperan seq⟨Bool⟩) {
    (∃listaDeRecursosActualizada : seq⟨ℝ⟩)(esListaDeRecursosActualizada(listaDeRecursosActualizada, individuo,
    nroDeEvento, trayectoria, pagos, apuestas) ∧L esListaDeRecursosAntesDelFondoComun(cooperan, listaDeRecursos
    , listaDistribuida)
  }

  pred esListaDeRecursosActualizada (lista : seq⟨ℝ⟩, individuo : ℤ, nroDeEvento : ℤ, trayectoria:seq⟨ℝ⟩, pagos :
  seq⟨seq⟨ℝ⟩⟩, apuesta seq⟨seq⟨ℝ⟩⟩) {
    lista[individuo] = calculoIndividual(individuo, nroDeEvento, trayectoria, pagos, apuestas)
  }

  aux calculoIndividual (individuo : ℤ, nroDeEvento : ℤ, trayectoria : seq⟨seq⟨ℝ⟩⟩, pagos : seq⟨seq⟨ℝ⟩⟩, apuestas :
  seq⟨seq⟨ℝ⟩⟩) : ℝ = trayectoria[individuo][nroDeEvento - 1] * pagos[individuo][nroDeEvento] *
  apuestas[individuo][nroDeEvento] ;

  pred esDistribucionDeRecursos (cooperan : seq⟨bool⟩, recursos : seq⟨ℝ⟩, L : seq⟨ℝ⟩) {
    (∀i : ℤ)(0 ≤ i < |L| →L if cooperan[i] = true then L[i] =  $\frac{\text{fondoComun}(\text{recursos}, \text{cooperan})}{|\text{recursos}|}$ 
    else L[i] =  $\frac{\text{fondoComun}(\text{recursos}, \text{cooperan})}{|\text{recursos}|} + \text{recursos}[i]$ ) fi
  }

  aux fondoComun (In recursos : ℝ, In cooperan : Bool) : ℝ =  $\sum_{i=0}^{|\text{recursos}|-1} \text{if } (\text{cooperan}[i] = \text{true}) \text{ then } \text{recursos}[i] \text{ else } 0$  fi ;

```

3. Ejercicio 3

```

proc trayectoriaExtrañaEscalera (in trayectoria: seq⟨ℝ⟩) : Bool
  requiere {|T| ≥ 2}
  requiere {res = false}
  asegura {Res = true ⇔ (∃j : ℤ)((0 ≤ j < |T|) ∧L ((|T| ≤ 2) ∧ Distintos(T)) ∨
    ((j = 0) ∧ primerElemEsMaximo(T) ∧ esUnicoMaximoLocal(T, j)) ∨
    ((j = |T| - 1) ∧ ultimoElemEsMaximo(T) ∧ esUnicoMaximoLocal(T, j)) ∨
    (esMaximoLocal(T, j) ∧ esUnicoMaximoLocal(T, j)))}
  pred primerElemEsMaximo (T:seq⟨ℝ⟩) {
    T[0] > T[1]
  }
  pred ultimoElemEsMaximo (T:seq⟨ℝ⟩) {
    T[|T| - 1] > T[|T| - 2]
  }
  pred esMaximoLocal (T:seq⟨ℝ⟩, j:ℤ) {
    T[j - 1] < T[j] ∧ T[j] > T[j + 1]
  }
  pred esUnicoMaximoLocal (T:seq⟨ℝ⟩, j:ℤ) {
    ¬(∃i : ℤ)((1 ≤ i < |T| - 1) ∧L (i ≠ j) ∧ esMaximoLocal(T, i))
  }
  pred distintos (T:seq⟨ℝ⟩) {
    T[0] ≠ T[1]
  }

```

4. Ejercicio 4

```

proc individuoDecideSiCooperarONo (in individuo : ℕ, in recursos : seq⟨ℝ⟩, inout cooperan : seq⟨Bool⟩, in apuestas : seq⟨ℝ⟩,
in pagos : seq⟨seq⟨ℝ⟩⟩, in eventos : seq⟨seq⟨ℝ⟩⟩)

  requiere {(∀i : ℤ)(0 ≤ i < |apuestas| →L ∑k=0|apuestas[i]| apuestas[i][k] = 1)}

  requiere {|cooperan| = |recursos| = |eventos| = |pagos| = |apuestas|}

  requiere {0 ≤ individuo < |recursos|}

  requiere {(∀i, e : ℤ)(0 ≤ i < |pagos| ∧ 0 ≤ e < |pagos[i]| →L pagos[i][e] > 0)}

  requiere {(∀i : ℤ)(0 ≤ i < |recursos| →L recursos[i] ≥ 0)}

  asegura {(∀i : ℤ)(0 ≤ i < |recursos| →L (∃listaFinalDeRecursosSiCoopera : seq⟨ℝ⟩)
    (esListaDespuesDeDistribuirElFondoComun(i, |eventos| - 1, pagos, apuestas, listaFinalDeRecursosSiCoopera,
    setAt(cooperan, individuo, true))))}

  asegura {(∀i : ℤ)(0 ≤ i < |recursos| →L (∃listaFinalDeRecursosSiNOCOopera : seq⟨ℝ⟩)
    (esListaDespuesDeDistribuirElFondoComun(i, |eventos| - 1, pagos, apuestas, listaFinalDeRecursosSiNOCOopera,
    setAt(cooperan, individuo, false))))}

  asegura {if (listaFinalDeRecursosSiCoopera[individuo] ≤ listaFinalDeRecursosSiNOCOopera[individuo])
    then (cooperan[individuo] = false) else (cooperan[individuo] = true) fi)}

  pred esListaDespuesDeDistribuirElFondoComun (individuo : ℤ, nroDeEvento : ℤ, pagos : seq⟨seq⟨ℝ⟩⟩, apuesta
  seq⟨seq⟨ℝ⟩⟩, listaDistribuida:seq⟨ℝ⟩, trayectorias:seq⟨seq⟨ℝ⟩⟩, cooperan seq⟨Bool⟩) {
    (∃listaDeRecursosActualizada : seq⟨ℝ⟩)(esListaDeRecursosActualizada(listaDeRecursosActualizada, individuo,
    nroDeEvento, trayectoria, pagos, apuestas)
    ∧L esListaDeRecursosAntesDelFondoComun(cooperan, listaDeRecursosActualizada
    , listaDistribuida)

```

}

```

pred esDistribucionDeRecursos (cooperan : seq⟨bool⟩, recursos : seq⟨ℝ⟩, L : seq⟨ℝ⟩) {
  (∀i : ℤ)(0 ≤ i < |L| →L if cooperan[i] = true then L[i] =  $\frac{\text{fondoComun}(\text{recursos}, \text{cooperan})}{|\text{recursos}|}$ 
  else L[i] =  $\frac{\text{fondoComun}(\text{recursos}, \text{cooperan})}{|\text{recursos}|} + \text{recursos}[i]$ ) fi
}

```

```

aux fondoComun (In recursos : ℝ, In cooperan : Bool) : ℝ =  $\sum_{i=0}^{|\text{recursos}|-1}$  if (cooperan[i] = true) then recursos[i] else 0 fi ;

```

```

pred esListaDeRecursosActualizada (lista : seq⟨ℝ⟩, individuo : ℤ, nroDeEvento : ℤ, trayectoria:seq⟨ℝ⟩, pagos :
seq⟨seq⟨ℝ⟩⟩, apuesta seq⟨seq⟨ℝ⟩⟩) {
  lista[individuo] = calculoIndividual(individuo, nroDeEvento, trayectoria, pagos, apuestas)
}

```

5. Ejercicio 5

```

proc individuoActualizaSuApuesta (in individuo : ℕ, in recursos : seq⟨ℝ⟩, in cooperan : seq⟨Bool⟩, inout apuestas : seq⟨ℝ⟩,
in pagos : seq⟨seq⟨ℝ⟩⟩, in eventos : seq⟨seq⟨ℝ⟩⟩)

```

```

  requiere {(∀i : ℤ)(0 ≤ i < |apuestas| →L  $\sum_{k=0}^{|\text{apuestas}[i]|} \text{apuestas}[i][k] = 1$ )}
```

```

  requiere {|cooperan| = |recursos| = |eventos| = |pagos| = |apuestas|}
```

```

  requiere {0 ≤ individuo < |recursos|}
```

```

  requiere {(∀i, e : ℤ)(0 ≤ i < |pagos| ∧ 0 ≤ e < |pagos[i]| →L pagos[i][e] > 0)}
```

```

  requiere {(∀i : ℤ)(0 ≤ i < |recursos| →L recursos[i] ≥ 0)}
```

```

  asegura {(∃listaDeRecursosQueMaximizaGananciasDeIndividuo, mejorApuesta : seq⟨ℝ⟩)
  (esApuestaValida(mejorApuesta, individuo) ∧L esListaDeApuestas(i, |eventos|,
pagos, setAt(apuestas, individuo, mejorApuesta), listaDeRecursosQueMaximizaGananciasDeIndividuo, cooperan))
  ∧L
  ¬(∃ListaDeRecursosCualquiera, apuestaCualquiera : seq⟨ℝ⟩)(esApuestaValida(apuestaCualquiera, individuo) ∧L
esMejorApuesta(apuestaCualquiera, mejorApuesta, individuo, eventos, pagos, cooperan)))}
```

```

pred esListaDeApuestas (individuo : ℤ, nroDeEvento : ℤ, pagos : seq⟨seq⟨ℝ⟩⟩, apuesta seq⟨seq⟨ℝ⟩⟩,
listaDistribuida:seq⟨ℝ⟩, trayectoria:seq⟨seq⟨ℝ⟩⟩, cooperan seq⟨Bool⟩) {

```

```

  (∀i : ℤ)(0 ≤ i < |recursos| →L
  esListaDespuesDeDistribuirElFondoComun(i, |eventos|,
pagos, setAt(apuestas, individuo, mejorApuesta), listaDeRecursosQueMaximizaGananciasDeIndividuo, cooperan)

```

}

```

pred esApuestaValida (apuestaAVerificar : seq⟨ℝ⟩) {

```

```

   $\sum_{i=0}^{|\text{apuestaAVerificar}|-1} \text{apuestaAVerificar}[i] = 1 \wedge |\text{apuestaAVerificar}| = |\text{apuestas}[\text{individuo}]|$ 

```

}

```

pred esMejorApuesta (apuesta1 : seq⟨ℝ⟩, apuesta2 : seq⟨ℝ⟩, individuo : ℕ, eventos : seq⟨seq⟨ℝ⟩⟩, pagos : seq⟨seq⟨ℝ⟩⟩,
cooperan : seq⟨Bool⟩) {

```

```

  (∃ListaDeRecursosApuesta1, ListaDeRecursosApuesta2 : seq⟨ℝ⟩)(esListaDespuesDeDistribuirElFondoComun
(individuo, |eventos|, pagos, apuesta1, ListaDeRecursosApuesta1, cooperan)
  ∧ (esListaDespuesDeDistribuirElFondoComun(individuo, |eventos|, pagos, apuesta2
, ListaDeRecursosApuesta2, cooperan) ∧
  ListaDeRecursosApuesta1[individuo] ≥ ListaDeRecursosApuesta2[individuo])

```

}

```

pred esListaDespuesDeDistribuirElFondoComun (individuo :  $\mathbb{Z}$ , nroDeEvento :  $\mathbb{Z}$ , pagos :  $seq\langle seq\langle \mathbb{R} \rangle \rangle$ , apuesta :  $seq\langle seq\langle \mathbb{R} \rangle \rangle$ ,
listaDistribuida :  $seq\langle \mathbb{R} \rangle$ , trayectoria :  $seq\langle seq\langle \mathbb{R} \rangle \rangle$ , cooperan :  $seq\langle Bool \rangle$ ) {
  ( $\exists listaDeRecursosActualizada : seq\langle \mathbb{R} \rangle$ ) ( $esListaDeRecursosActualizada(listaDeRecursosActualizada, individuo, nroDeEvento, trayectoria, pagos, apuestas) \wedge_L esListaDeRecursosAntesDelFondoComun$ 
  ( $cooperan, listaDeRecursosActualizada, listaDistribuida$ )
}

pred esListaDeRecursosActualizada (lista :  $seq\langle \mathbb{R} \rangle$ , individuo :  $\mathbb{Z}$ , nroDeEvento :  $\mathbb{Z}$ , trayectoria :  $seq\langle \mathbb{R} \rangle$ , pagos :  $seq\langle seq\langle \mathbb{R} \rangle \rangle$ , apuesta :  $seq\langle seq\langle \mathbb{R} \rangle \rangle$ ) {
  lista[individuo] = calculoIndividual(individuo, nroDeEvento, trayectoria, pagos, apuestas)
}

aux calculoIndividual (individuo :  $\mathbb{Z}$ , nroDeEvento :  $\mathbb{Z}$ , trayectoria :  $seq\langle seq\langle \mathbb{R} \rangle \rangle$ , pagos :  $seq\langle seq\langle \mathbb{R} \rangle \rangle$ , apuestas :  $seq\langle seq\langle \mathbb{R} \rangle \rangle$ ) :  $\mathbb{R}$  = trayectoria[individuo][nroDeEvento - 1] * pagos[individuo][nroDeEvento] *
apuestas[individuo][nroDeEvento];

pred esDistribucionDeRecursos (cooperan :  $seq\langle bool \rangle$ , recursos :  $seq\langle \mathbb{R} \rangle$ , L :  $seq\langle \mathbb{R} \rangle$ ) {
  ( $\forall i : \mathbb{Z}$ ) ( $0 \leq i < |L| \longrightarrow_L$  if cooperan[i] = true then  $L[i] = \frac{fondoComun(recursos, cooperan)}{|recursos|}$ 
  else  $L[i] = \frac{fondoComun(recursos, cooperan)}{|recursos|} + recursos[i]$ ) fi
}

aux fondoComun (In recursos :  $\mathbb{R}$ , In cooperan : Bool) :  $\mathbb{R}$  =  $\sum_{i=0}^{|recursos|-1}$  if (cooperan[i] = true) then recursos[i] else 0 fi;

```

6. Ejercicio 6

- $P_c \equiv res = recursos \wedge i = 0 \wedge 0 \leq recursos$
- $Q_c \equiv res = recurso(apuesta_c * pago_c)^{apariciones(eventos, T)} * (apuesta_s * pagos_s)^{apariciones(eventos, F)}$
- $I \equiv 0 \leq i \leq |eventos| \wedge res = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pagos_s) \text{ fi}$
- $B \equiv i < |eventos|$
- $fv = |eventos| - i$
- $\{\neg B \wedge I\} \longrightarrow Q_c$

$$\neg(i < |eventos|) \wedge 0 \leq i \leq |eventos| \wedge res = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pagos_s) \text{ fi} \equiv$$

$$|eventos| \leq i \wedge i \leq |eventos| \wedge res = \prod_{j=0}^{|eventos|-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pagos_s) \text{ fi} \equiv$$

$$res = recurso(apuesta_c * pago_c)^{apariciones(eventos, T)} * (apuesta_s * pagos_s)^{apariciones(eventos, F)} \equiv Q_c$$
- $P_c \longrightarrow I$

Como por P_c se tiene que $i=0$ en el invariante queda que

$$res = \prod_{j=0}^{-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pagos_s) \text{ fi} = 0$$

por otra parte como por P_c $res = recursos$ y $0 \leq recursos$ se tiene que $0 \leq res$ queda que $res = 0$ es una Tautologia
- $\{B \wedge I\} S\{I\} \equiv (B \wedge I) \longrightarrow wp(S, I)$

Donde $S \equiv \text{if } eventos[i] \text{ then}$

$S_1 \quad res := res * apuesta.c * pago.c$
 $else$
 $S_2 \quad res := res * apuesta.s * pago.s$
 $endif$
 $S_3 \quad i := i + 1$

$wp(S, I) \equiv wp(if \text{ eventos}[i] \text{ Then } res = res * apuesta.c * pago.c \text{ else } res * apuesta.s * pago.s, wp(i := i + 1, I))$
 $wp(S_3, I) \equiv def(i + 1) \wedge_L 0 \leq i + 1 \leq |eventos| \wedge res = \prod_{j=0}^{i+1-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi}$
 $wp(S, I) \equiv def(eventos[i]) \wedge_L ((eventos[i] = True \wedge wp(S_1, wp(S_3, I))) \vee (eventos[i] = False \wedge wp(S_2, wp(S_3, I))))$
 $wp(S, I) \equiv 0 \leq i < |eventos| \wedge_L ((eventos[i] = True \wedge wp(S_1, wp(S_3, I))) \vee (eventos[i] = False \wedge wp(S_2, wp(S_3, I))))$

$wp(S_1, wp(S_3, I)) \equiv def(res * apuesta.c * pago.c) \wedge_L 0 \leq i + 1 \leq |eventos| \wedge$
 $res * apuesta.c * pago.c = \prod_{j=0}^i \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi} \equiv$
 $0 \leq i + 1 \leq |eventos| \wedge$
 $res * apuesta.c * pago.c = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi} * (apuesta.c * pago.c) \equiv$
 $0 \leq i < |eventos| \wedge$
 $res = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi}$

$wp(S_2, wp(S_3, I)) \equiv def(res * apuesta.s * pago.s) \wedge_L 0 \leq i + 1 \leq |eventos| \wedge$
 $res * apuesta.s * pago.s = \prod_{j=0}^i \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi} \equiv$
 $0 \leq i + 1 \leq |eventos| \wedge$
 $res * apuesta.s * pago.s = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi} * (apuesta.s * pago.s) \equiv$
 $0 \leq i < |eventos| \wedge$
 $res = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi}$

Ahora calculemos $I \wedge B$ y veamos si implica lo de arriba:

$I \wedge B \equiv$
 $0 \leq i < |eventos| \wedge res = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi} \equiv$
 $0 \leq i < |eventos| \wedge ((eventos[i] = True \wedge res = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi})$
 $\vee (eventos[i] = False \wedge res = \prod_{j=0}^{i-1} \text{if } (eventos[j]) \text{ then } (apuesta_c * pago_c) \text{ else } (apuesta_s * pago_s) \text{ fi}))$

Entonces $(I \wedge B) \longrightarrow wp(S, I)$

- $\{I \wedge B \wedge v_0 = fv\} S \{fv < v_0\} \equiv (I \wedge B \wedge v_0 = fv) \longrightarrow wp(S, fv < v_0)$

Definimos la función variante como: $fv = |eventos| - i$

$wp(S, fv < v_0) \equiv wp(if \text{ eventos}[i] \text{ Then } res = res * apuesta.c * pago.c \text{ else } res * apuesta.s * pago.s, wp(i := i + 1, |eventos| - i < v_0))$
 $wp(S, fv < v_0) \equiv wp(if \text{ eventos}[i] \text{ Then } res = res * apuesta.c * pago.c \text{ else } res * apuesta.s * pago.s, |eventos| - (i + 1) < v_0)$

$wp(S, fv < v_0) \equiv def(eventos[i]) \wedge_L ((eventos[i] = True \wedge wp(S_1, |eventos| - i - 1 < v_0)) \vee (eventos[i] = False \wedge wp(S_2, |eventos| - i - 1 < v_0)))$

$wp(S_2, |eventos| - 1 - i < v_0) \equiv def(res * apuesta.s * pago.s) \wedge_L |eventos| - 1 - i < v_0 \equiv |eventos| - 1 - i < v_0$

$wp(S_1, |eventos| - 1 - i < v_0) \equiv def(res * apuesta.c * pago.c) \wedge_L |eventos| - 1 - i < v_0 \equiv |eventos| - 1 - i < v_0$

$wp(S, fv < v_0) \equiv 0 \leq i < |eventos| \wedge_L ((eventos[i] = True \wedge |eventos| - 1 - i < v_0) \vee (eventos[i] = False \wedge |eventos| - 1 - i < v_0))$

$wp(S, fv < v_0) \equiv 0 \leq i < |eventos| \wedge_L |eventos| - 1 - i < v_0$

Ahora calculemos $I \wedge B \wedge v_0 = fv$ y veamos si implica lo de arriba:

A partir del Invariante tenemos que $0 \leq i \leq |eventos| \longrightarrow 0 \leq i < |eventos|$

Luego usando que $v_0 = |eventos| - i$ es verdadero se puede llegar a que $v_0 - 1 < v_0$ Lo cual es verdadero.

Entonces $(I \wedge B \wedge v_0 = fv) \longrightarrow wp(S, fv < v_0)$

- $I \wedge fv \leq 0 \longrightarrow \neg B$

A partir de que $fv \leq 0 \equiv |eventos| - i \leq 0 \equiv |eventos| \leq i \equiv \neg B$