



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Descripción del tp

Subtítulo del tp

17 de abril de 2024

Materia de la carrera

Grupo 42

Integrante	LU	Correo electrónico
Apellido, Nombre1	001/01	email1@dominio.com
Apellido, Nombre2	002/01	email2@dominio.com
Apellido, Nombre3	003/01	email3@dominio.com
Apellido, Nombre4	004/01	email4@dominio.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. Ejercicio 1 de Especificación

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

```
pred todosPositivos (recurso : seq⟨ℝ⟩) {  
  (∀j : ℤ) (0 ≤ j < |recurso| →L recurso[j] > 0)  
}
```

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

2. Ejercicio 2 de Especificacion

```

proc trayectoriaDeLosFrutosIndividualesALargoPlazo (inout trayectorias:seq<seq<ℝ>>,
in cooperan:seq<Bool>, in apuestas:seq<seq<ℝ>>,in pagos:seq<seq<ℝ>>,in eventos:seq<seq<ℕ>>) :
  requiere {trayectorias = To ∧ mismaLongitud(pagos, apuestas, eventos)
  ∧ sonPositivos(apuestas, pagos) ∧ trayectoriaValida(trayectorias)}
  asegura {longitudValida(trayectorias, eventos) ∧
  (∀i : ℤ) (1 ≤ i ≤ |To[0]|) →L ((∀k : ℤ) (0 ≤ k < |To| →L
  trayectorias[k][i] = if coop(cooperan, k) then siCoop(apuestas, pagos, eventos, To, cooperan, i) else
  noCoop(apuestas, pagos, eventos, To, cooperan, k, i) fi))}

pred coop (cooperan:seq<Bool>,jugador:ℤ) {
  cooperan[individuo] = True
}

pred mismaLongitud (s1,s2:seq<seq<ℝ>>,s3:seq<seq<ℕ>>) {
  (∀i : ℤ) (0 ≤ i < |s1| →L |s1[i]| = |s2[i]|) ∧ (|s1| = |s2|) ∧ (|s1| = |s3|)
}

pred sonPositivos (s1,s2:seq<seq<ℝ>>) {
  (∀i : ℤ) (0 ≤ i < |s1| →L s1[i] > 0 ∧ s2[i] > 0)
}

pred trayectoriaValida (s1:seq<seq<ℝ>>) {
  (∀i : ℤ) (0 ≤ i < |s1| →L (|s1[i]| = 1 ∧ s1[i][0] > 0))
}

pred buenaApuesta (s1:seq<seq<ℝ>>) {
  (∀i : ℤ) (0 ≤ i < |s1| →L ∑j=0|s1[i]|-1 s1[i][j] = 1)
}

pred longitudValida (s1:seq<seq<ℝ>>,s2:seq<seq<ℕ>>) {
  (∀i : ℤ) (0 ≤ i < |s1| →L |s1[i]| = |s2[i]| + 1)
}

aux FMC (apuestas,pagos:seq<seq<ℝ>>,eventos:seq<seq<ℕ>>,cooperan:seq<Bool>,i:ℤ) : ℝ =
|cooperan|-1
  ∑j=0 if coop(cooperan, j) then frutos(apuestas[j], pagos[j], eventos[j], i)/|cooperan| else 0 fi ;

aux siCoop (apuestas,pagos,To:seq<seq<ℝ>>,eventos:seq<seq<ℕ>>,cooperan:seq<Bool>,i:ℤ) : ℝ =
  fondoInicial(apuestas, pagos, To, eventos, cooperan, i) * ∏k=1i FMC(apuestas, pagos, eventos, cooperan, k) ;

aux noCoop (apuestas,pagos,To:seq<seq<ℝ>>,eventos:seq<seq<ℕ>>,cooperan:seq<Bool>,k,i:ℤ) : ℝ =
  To[k][0] * ∏n=0i frutos(apuestas[k], pagos[k], eventos[k], i) + FMC(apuestas, pagos, eventos, cooperan, n) ;

aux fondoInicial (apuestas,pagos,To:seq<seq<ℝ>>,eventos:seq<seq<ℕ>>,cooperan:seq<Bool>,i:ℤ) : ℝ =
|eventos[0]|-1
  ∑j=0 if coop(cooperan, j) then To[j][0] * frutos(apuestas[j], pagos[j], eventos[j], i - i)/|To| else 0 fi ;

aux frutos (apuesta,pago:seq<ℝ>,evento:seq<ℕ>,i:ℤ) : ℝ = apuesta[evento[i]] * pago[evento[i]] ;

```

3. Ejercicio 3 de Especificacion

```
proc trayectoriaExtrañaEscalera (in trayectoria : seq⟨ℝ⟩) : Bool
  requiere {|trayectoria| > 0}
  asegura {res = True ⇔ (casosTriviales(trayectoria) ∨
    moduloMayorATres(trayectoria))}

pred casosTriviales (l:seq⟨ℝ⟩) {
  |l| = 1 ∨ (|l| = 2 ∧ l[0] ≠ l[1])
}

pred moduloMayorATres (l:seq⟨ℝ⟩) {
  (∃i : ℤ) (
    (0 ≤ i < |l|) ∧ secuenciaCreciente(subseq(l, 0, i + 1)) ∧ secuenciaDecreciente(subseq(l, i, |l|))
  )
}

pred secuenciaCreciente (l:seq⟨ℝ⟩) {
  |l| = 1 ∨
  (|l| = 2 ∧ l[0] < l[1]) ∨
  (|l| ≥ 3 ∧ l[|l| - 1] > l[|l| - 2] ∧ ((∀j : ℤ) (
    1 ≤ j < |l| →L l[j] ≤ l[j + 1]
  )))
}

pred secuenciaDecreciente (l:seq⟨ℝ⟩) {
  |l| = 1 ∨
  (|l| = 2 ∧ l[0] < l[1]) ∨
  (|l| ≥ 3 ∧ l[0] > l[1] ∧ ((∀j : ℤ) (
    0 ≤ j < |l| - 1 →L l[j] ≤ l[j + 1]
  )))
}
```

4. Ejercicio 4 de Especificacion

```

proc individuoDecideSiCoopera0No (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⟨ℕ⟩⟩) :
  requiere {mismosElementos(pagos, apuestas) ∧ todosPositivos(apuestas, pagos, eventos)}
  asegura {coop(cooperan, individuo) ⇔ gananciaNoCoop ≤ gananciaCoop}
}
pred coop (cooperan:seq⟨Bool⟩,individuo:ℕ) {
  cooperan[individuo] = True
}

aux gananciaCoop (apuestas,pagos,eventos,recursos,cooperan,individuo) : ℝ =
fondoInicial() * ∏k=1|eventos[0]−1 fondoConmigo();
aux fondoConmigo (apuestas,pagos,eventos,recursos,cooperan,individuo,k) : ℝ =
∑j=0|cooperan|−1 if coop(cooperan, j) ∨ j = individuo then frutos()/|cooperan| else 0 fi;
aux gananciaNoCoop (apuestas,pagos,eventos,recursos,cooperan,individuo) : ℝ =
recursos[individuo] * ∏k=1eventos[0]−1 frutos() + fondoSinMi();
aux fondoSinMi (apuestas,pagos,cooperan,evento,i) : ℝ =
∑j=0|cooperan|−1 if coop(cooperan, j) ∧ (j = ¬i) then frutos()/|cooperan| else 0 fi;
aux fondoInicial (apuestas,pagos,eventos,recursos,coperan,individuo) : ℝ =
∑j=0|eventos[0]−1 if coop(cooperan, j) ∨ j = individuo then recursos[j] * frutos()/|cooperan| else 0 fi;
aux frutos (apuesta,pago,evento,k) : ℝ = apuestas[evento[k]] * pagos[evento[k]];

```

5. Ejercicio 5 de Especificacion

6. Ejemplo de sección

6.1. Subsección: ambientes comunes de \LaTeX

Lo principal: las fórmulas. Se puede poner en una linea, como $x_i = x_{i-1} + x_{i-2}$, o ponerse más grande:

$$\sum_{i=0}^n i \tag{1}$$

Y se pueden citar ecuaciones con `\eqref{nombreDeEq}`: (1)

Ejemplo de itemizado:

- Item 1
- Item 2
- Item 3

Ejemplo de enumerado con menor distancia entre items:

1. Item 1
2. Item 2
3. Item 3

Podemos escribir mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto.

Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo.

Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos.

La tabla 1 es un ejemplo de cómo se hace una tabla.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Tabla 1: Ejemplo de tabla

La figura 2 es un ejemplo de cómo se agrega una imagen.



Figura 1: Ejemplo de figura



(a) Logo de LaTeX



(b) Logo de TeX

Figura 2: Ejemplo para poner dos figuras juntas. Y citarlas por separado a (a) y (b).

```

1 res := 0;
2 i := 0;
3 while (i < s.size()) do
4     res := res + s[i];
5     i := i + 1
6 endwhile

```

Código 1: Ejemplo de código (usando los estilos de la cátedra, ver las macros para más detalles)

Si se pone un label al `lstlisting`, se puede referenciar: Código 1.

6.2. Macros de la cátedra para especificar

```

proc nombre (in paramIn : N, inout paramInout : seq⟨Z⟩) : tipoRes
    requiere {expresionBooleana1}
    asegura {expresionBooleana2}
    aux auxiliar1 (parametros) : tipoRes = expresion;
    pred pred1 (parametros) {
        expresion
    }

aux auxiliarSuelto (parametros) : tipoRes = expresion;
pred predSuelto (parametros) {
    (∀variable : tipo) (algo →L expresion)
}
pred predSuelto (parametros) {
    (∃variable : tipo) (algo ∧L expresion)
}

```