



Trabajo práctico 1

Especificación y WP

21/4/2024

AED

Grupo IATOGYSWWBKAFJVCRWKR

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 if cooperan[i] then res[i] = totalARepartir(recursos, cooperan) else res[i] =
    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 {-}
  asegura {-}
```

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 {-}
  asegura {-}
```

1.5. individuoActualizaApuesta

```
proc individuoActualizaApuesta (in individuo: ℕ, in recursos: seq⟨ℝ⟩, in cooperan: seq⟨Bool⟩, inout apuestas: seq⟨seq⟨Bool⟩⟩,
in pagos: seq⟨seq⟨ℝ⟩⟩, in eventos: seq⟨seq⟨ℝ⟩⟩)
  requiere {-}
  asegura {-}
```

Auxiliares y predicados globales

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

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: ℝ, in apuesta: ⟨s : ℝ, c : ℝ⟩, in pago: ⟨s : ℝ, c : ℝ⟩, in eventos:
seq⟨Bool⟩, out res: ℝ)
```

requiere $\{apuesta_c + apuesta_s = 1 \wedge pago_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0\}$
asegura $\{res = recurso(apuesta_c pago_c)^{\#apariciones(eventos, T)} (apuesta_s pago_s)^{\#apariciones(eventos, F)}\}$

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

```

1  res := recursos
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

```