



# Trabajo Practico N°1

## Fondo Monetario C  mun

20 de abril de 2024

Algoritmos y Estructuras de Datos

### Grupo "Nominadores"

| Integrante       | LU      | Correo electr  nico       |
|------------------|---------|---------------------------|
| Caire, Nahuel A. | 1140/23 | cairenahuel@gmail.com     |
| Motta, Marino J. | 1372/23 | marijmotta@gmail.com      |
| Rabey, Nahuel A. | 1394/23 | nahuelrabeywork@gmail.com |
| Sola, Santiago   | 42/23   | santiagoms.ss@gmail.com   |



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabell  n I/Planta Baja)

Intendente G  iraldes 2610 - C1428EGA

Ciudad Aut  noma de Buenos Aires - Rep. Argentina

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

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

## Auxiliares y predicados utilizados en mas de un procedimiento.

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

### 1. Ejercicio 1:

```
proc redistribucionDeLosFrutos (in recursos :  $\text{seq}\langle\mathbb{R}\rangle$ , in cooperan:  $\text{seq}\langle\text{Bool}\rangle$ ) :  $\text{seq}\langle\mathbb{R}\rangle$ 
  requiere  $\{|recursos| = |cooperan|\}$ 
  requiere  $\{(\forall n : \mathbb{Z}) ((0 \leq n < |recursos| \wedge_L \text{recursos}[n] > 0))\}$ 
  asegura  $\{|res| = |recursos|\}$ 
  asegura  $\{(\forall n : \mathbb{Z}) (0 \leq n < |res| \longrightarrow_L \text{res}[n] = (\text{if } \text{cooperan}[n] \text{ then } \text{ufc} \text{ else } \text{recursos}[n] + \text{ufc} \text{ fi}))\}$ 

  donde  $\text{ufc} = \text{unidadFondoComun}(\text{recursos}, \text{cooperan})\}$ 
```

### 2. Ejercicio 2:

```
proc trayectoriaDeLosFrutosIndividualesALargoPlazo (inout trayectorias :  $\text{seq}\langle\text{seq}\langle\mathbb{R}\rangle\rangle$ ,
in cooperan :  $\text{seq}\langle\text{Bool}\rangle$ , in apuestas :  $\text{seq}\langle\text{seq}\langle\mathbb{R}\rangle\rangle$ , in pagos :  $\text{seq}\langle\text{seq}\langle\mathbb{R}\rangle\rangle$ , in eventos :  $\text{seq}\langle\text{seq}\langle\mathbb{N}\rangle\rangle$ )
  requiere  $\{|trayectorias| = |cooperan| = |apuestas| = |pagos| = |eventos|\}$ 
  requiere  $\{(\forall i : \mathbb{Z}) ((0 \leq i < \text{individuosLen}) \wedge_L (|\text{pagos}[i]| = |\text{apuestas}[i]|))\}$ 
  requiere  $\{(\forall i : \mathbb{Z}) ((0 \leq i < \text{individuosLen}) \wedge_L (|\text{trayectorias}[i]| = 1 \wedge_L \text{trayectorias}[i][0] > 0))\}$ 
  requiere  $\{(\forall i, k : \mathbb{Z}) ((0 \leq i < \text{individuosLen}, 0 \leq k < \text{pagosLen}) \wedge_L (|\text{pagos}[i]|, |\text{pagos}[i][k]| > 0))\}$ 
  requiere  $\{(\forall n : \mathbb{Z}) ((0 \leq n < |\text{eventos}|) \wedge_L |\text{eventos}[n]| > 0)\}$ 
  asegura  $\{(\forall i, p : \mathbb{Z}) (\text{recorreIndividuosPasos}(i, p) \wedge_L \text{if } \text{cooperan}[i] \text{ then } \text{trayectorias}[i][p] = \text{ufc}(p) \text{ else } \text{trayectorias}[i][p] = \text{ufc}(p) \text{ fi}))\}$ 

  pred  $\text{recorreIndividuosPasos}$  (in i:  $\mathbb{Z}$ , in p:  $\mathbb{Z}$ ) {
     $0 \leq i < \text{individuosLen} \wedge_L 0 \leq p < \text{pasosLen}$ 
  }

  aux  $\text{cuantoGano}$  (in i:  $\mathbb{Z}$ , in p:  $\mathbb{Z}$ ) :  $\mathbb{R} = \text{trayectorias}[i][p] * \text{apuestas}[i][\text{eventos}[i][p]] * \text{pagos}[i][\text{eventos}[i][p]];$ 

  donde  $\text{individuosLen} = \text{cooperan.len}$ 
  donde  $\text{pasosLen} = \text{eventos}[0].len$ 
  donde  $\text{pagosLen} = \text{pagos}[0][0].len$ 
```

### 3. Ejercicio 3:

```
proc trayectoriaExtrañaEscalera (in trayectoria:seq⟨ℝ⟩) : Bool
  requiere {|trayectoria| > 0}
  asegura {res = true ⇔ (∃!n : ℤ) (esMaximoLocal(trayectoria, n))}
  pred existeAnterior (in s:seq⟨ℝ⟩, in n:ℕ) {
    0 ≤ n < |s| ∧L 0 ≤ n - 1 < |s|
  }
  pred existePosterior (in s:seq⟨ℝ⟩, in n:ℕ) {
    0 ≤ n < |s| ∧L 0 ≤ n + 1 < |s|
  }
  pred esMaximoLocal (in s:seq⟨ℝ⟩, in n:ℕ) {
    (existeAnterior(s, n) ⇒ s[n - 1] < s[n]) ∧L (existePosterior(s, n) ⇒ s[n + 1] < s[n])
  }
}
```