



Trabajo práctico 1: Especificación y WP

14 de abril de 2024

Algoritmos y Estructuras de Datos

Grupo indeterminado

Integrante	LU	Correo electrónico
Labastié, Gaspar	660/23	gaspilabastie@gmail.com
Rugo, Julian	1414/23	julianrugo22@gmail.com
Torres, Emiliano	80/23	emilianomtorres1@gmail.com
Vanotti, Franco	464/23	fvanotti15@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>

0. Aclaraciones generales

- Los índices de las listas recursos, cooperan, trayectorias, apuestas, pagos, eventos representa el identificador de los individuos.
- recursos: $\text{seq}\langle\mathbb{R}\rangle$ es la lista con el recurso de cada individuo.
- cooperan: $\text{seq}\langle\text{Bool}\rangle$ es la lista que indica T rue si el individuo en dicha posición coopera.
- trayectorias: $\text{seq}\langle\text{seq}\langle\mathbb{R}\rangle\rangle$ indica para cada individuo, en cada paso de tiempos, cuántos recursos (\mathbb{R}) cuenta.
- eventos: $\text{seq}\langle\text{seq}\langle\mathbb{N}\rangle\rangle$ indica para cada individuo, en cada paso temporal, qué evento le ha tocado.
- apuestas: $\text{seq}\langle\text{seq}\langle\mathbb{R}\rangle\rangle$ indica para cada individuo, para cada evento posible (numerados desde 0), cuánto apostará.
- pagos: $\text{seq}\langle\text{seq}\langle\mathbb{R}\rangle\rangle$ indica para cada individuo, para cada evento, cuánto se pagará. Notar que a diferencia del ejemplo, estamos resolviendo un caso más general donde el pago de cada evento puede diferir por individuo.
- Las personas que no *cooperan* no aportan nada al fondo monetario común.
- Los *recursos* iniciales son positivos.
- Todos los *pagos* son positivos.
- Las *apuestas* de los individuos representan la proporción de los recursos que los individuos invierten a cada una de los eventos posibles. Notar nuevamente que a diferencia del ejemplo, en este caso más general, podríamos tener apuestas distintas para cada evento por cada individuo.
- Cada individuo apuesta siempre el mismo porcentaje por cada evento posible (es decir, el mismo número en cada paso temporal). Por ejemplo, si tenemos dos eventos; cara y ceca y apuesta 0, 4 por cara y 0,6 por seca, en cada paso temporal apostará esas proporciones.
- Se considera al número 0 como parte de \mathbb{N} .

1. Especificación

1. **redistribucionDeLosFrutos**: Calcula los recursos que obtiene cada uno de los individuos luego de que se redistribuyen los recursos del fondo monetario común en partes iguales. El fondo monetario común se compone de la suma de *recursos* iniciales aportados por todas las personas que *cooperan*. La salida es la lista de recursos que tendrá cada jugador.

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

pred nuevosRecursosCooperan (recursos : seq⟨ℝ⟩, cooperan : seq⟨Bool⟩, res : seq⟨ℝ⟩) {
  (∀i : ℤ) (0 ≤ i < |recursos| ∧ cooperan[i] = true →L res[i] = distribuciónFondoComún(recursos, cooperan))
}

pred nuevosRecursosNoCooperan (recursos : seq⟨ℝ⟩, cooperan : seq⟨Bool⟩, res : seq⟨ℝ⟩) {
  (∀i : ℤ) (0 ≤ i < |recursos| ∧ cooperan[i] = false →L res[i] = distribuciónFondoComún(recursos, cooperan) +
    recursos[i])
}

aux distribuciónFondoComún (recursos : seq⟨ℝ⟩, cooperan : seq⟨Bool⟩) : ℝ =
  ∑i=0|recursos|-1 (if cooperan[i] = true then  $\frac{\text{recursos}[i]}{|recursos|}$  else 0 fi);
```

2. **trayectoriaDeLosFrutosIndividualesALargoPlazo**: Actualiza (In/Out) la lista de *trayectorias* de los recursos de cada uno de los individuos. Inicialmente, cada una de las trayectorias (listas de recursos) contiene un único elemento que representa los recursos iniciales del individuo. El procedimiento agrega a las *trayectorias* los recursos que los individuos van obteniendo a medida que se van produciendo los resultados de los *eventos* en función de la lista de *pagos* que le ofrece la naturaleza (o casa de apuestas) a cada uno de los individuos, las *apuestas* (o inversiones) que realizan los individuos en cada paso temporal, y la lista de individuos que *cooperan* aportando al fondo monetario común.

```

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| = |cooperan| = |apuestas| = |pagos| = |eventos|  $\wedge_L$ 
  ( $\forall i : \mathbb{Z})(0 \leq i < |trayectorias| \rightarrow_L (|trayectorias[i]| = 1 \wedge trayectorias[i][0] > 0)) \wedge |pagos| > 0 \wedge_L$ 
  ( $\forall k, l : \mathbb{Z})(0 \leq k, l < |apuestas| \rightarrow_L |apuestas[k]| = |apuestas[l]|) \wedge$ 
  ( $\forall i : \mathbb{Z})(0 \leq i < |apuestas| \rightarrow_L \text{sumarApuestasIndividuo}(apuestas[i]) = 1) \wedge_L$ 
  ( $\forall j : \mathbb{Z})(0 \leq j < |apuestas[i]| \rightarrow_L 0 \leq apuestas[i][j] \leq 1)) \wedge$ 
  ( $\forall k, l : \mathbb{Z})(0 \leq k, l < |pagos| \rightarrow_L |pagos[k]| = |pagos[l]|) \wedge$ 
  ( $\forall i : \mathbb{Z})(0 \leq i < |pagos| \rightarrow_L (\forall j : \mathbb{Z})(0 \leq j < |pagos[i]| \rightarrow_L 0 \leq pagos[i][j])) \wedge$ 
  ( $\forall i : \mathbb{Z})(0 \leq i < |pagos| \rightarrow_L (\forall j : \mathbb{Z})(0 \leq j < |apuestas| \rightarrow_L |pagos[i]| = |apuestas[j]|)) \wedge$ 
  ( $\forall i : \mathbb{Z})(0 \leq i < |eventos| \rightarrow_L |eventos[i]| > 0) \wedge$ 
  ( $\forall i : \mathbb{Z})(0 \leq i < |eventos| \rightarrow_L |eventos[0]| = |eventos[i]|) \wedge$ 
  ( $\forall i : \mathbb{Z})(0 \leq i < |eventos| \rightarrow_L (\forall j : \mathbb{Z})(0 \leq j < |eventos[i]| \rightarrow_L 0 \leq eventos[i][j] < |pagos[i]|))$  }

  asegura { |trayectorias| = |eventos|  $\wedge_L$  ( $\forall i : \mathbb{Z})(0 \leq i < |trayectorias| \rightarrow_L |trayectorias[i]| = |eventos[i]| + 1) \wedge$ 
  ( $\forall i : \mathbb{Z})(0 \leq i < |trayectorias| \rightarrow_L (\forall j : \mathbb{Z})(0 \leq j < |trayectorias[i]| + 1 \rightarrow_L))$  }

pred trayectoriasCooperan (trayectorias, cooperan, apuestas, pagos, eventos) {
  cooperan[i] = true  $\rightarrow_L$  trayectorias[i][j] = distribuciónFondoComúnTrayectoria(trayectorias, cooperan, apuestas,
pagos, eventos)
}

pred trayectoriasNoCooperan (trayectorias, cooperan, apuestas, pagos, eventos) {
  cooperan[i] = false  $\rightarrow_L$  trayectorias[i][j] = distribuciónFondoComúnTrayectoria(trayectorias, cooperan, apuestas,
pagos, eventos) + trayectoria[i][|trayectorias| - 1] * gananciaIndividuo(apuestas[i], pagos[i],
eventos[i][|trayectorias| - 1])
}

aux sumarApuestasIndividuo (apuestasIndividuo : seq<R>) : R =
 $\sum_{n=0}^{|apuestasIndividuo|-1} apuestasIndividuo[n]$ ;

aux gananciaIndividuo (apuestaIndividuo: seq<R>, pagosIndividuo: seq<R>, resultadoEventoIndividuo: R) : R =
apuestaIndividuo[resultadoEventoIndividuo] * pagosIndividuo[resultadoEventoIndividuo];

aux distribuciónFondoComúnTrayectoria (trayectorias: seq<seq<R>>, cooperan : seq<Bool>, apuestas: seq<seq<R>>,
pagos: seq<seq<R>>, eventos: seq<seq<N>>) : R =
 $\frac{\sum_{i=0}^{|trayectorias|-1} \text{if } cooperan[i]=\text{true then } trayectorias[i][|trayectorias|-1] * gananciaIndividuo(apuestas[i], pagos[i], eventos[i][|trayectorias|-1]) \text{ else } 0 \text{ fi}}{|trayectorias|}$ ;

```

3. **trayectoriaExtrañaEscalera** Esta función devuelve *True* e sii en la trayectoria de un individuo existe un único punto mayor a sus vecinos (llamado máximo local). Un elemento es máximo local si es mayor estricto que sus vecinos inmediatos.

```

proc trayectoriaExtrañaEscalera (in trayectoria: seq⟨ℝ⟩) : Bool
  requiere { |trayectoria| > 0 ∧ (∀i : ℤ) (0 ≤ i < |trayectoria| →L trayectoria[i] ≥ 0) }
  asegura { res ↔ true → |trayectoria| = 1  ∨L
    ((∀i : ℤ)((∃m : ℤ)((0 ≤ i, m ≤ |trayectoria| ∧ i ≠ m) →L trayectoria[m] > trayectoria[i]) ∧L
    (∀n : ℤ)(0 ≤ n ≤ m →L trayectoria[i] > trayectoria[i - 1]) ∧
    (∀n : ℤ)(m ≤ n < |trayectoria| - 1 →L trayectoria[i] > trayectoria[i + 1]))) }

```