



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Descripción del tp

Subtítulo del tp

13 de septiembre de 2023

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. Problemas

1.1. Ejercicio 1

```
proc hayBallotage (in escrutinio : seq⟨ℤ⟩) : Bool
  requiere {esEscrutinioValido(escrutinio)}
  asegura {res = true ↔ (¬algunoSupera45(escrutinio) ∧ ¬algunoSupera40y10(escrutinio))}

  pred esEscrutinioValido (escrutinio : seq⟨ℤ⟩) {
    (|escrutinio| ≥ 3) ∧ (∀i : ℤ) (0 ≤ i < |escrutinio| →L
      (∀j : ℤ) ((0 ≤ j < |escrutinio| ∧ i ≠ j) →L escrutinio[i] ≠ escrutinio[j]))
  }

  aux sumaTotalDeVotos (escrutinio : seq⟨ℤ⟩) : ℤ = ∑i=0|escrutinio|-1 escrutinio[i];

  aux porcentajeVotos (escrutinio : seq⟨ℤ⟩, i : ℤ) : ℤ =  $\frac{\text{escrutinio}[i]}{\text{sumaTotalDeVotos}(\text{escrutinio})}$ ;

  pred algunoSupera45 (escrutinio : seq⟨ℤ⟩) {
    (∃i : ℤ) (0 ≤ i < (|escrutinio| - 1) ∧L porcentajeVotos(escrutinio, i) ≥ 0,45)
  }

  pred algunoSupera40y10 (escrutinio : seq⟨ℤ⟩) {
    (∃i : ℤ) (0 ≤ i < (|escrutinio| - 1) ∧L (porcentajeVotos(escrutinio, i) ≥ 0,4 ∧
      (∀j : ℤ) ((0 ≤ j < (|escrutinio| - 1) ∧ i ≠ j) →L porcentajeVotos(escrutinio, j) <
        (porcentajeVotos(escrutinio, i) - 0,1))))
  }
```

1.2. Ejercicio 2

```
proc hayFraude (in escrutinio_presidencial : seq⟨ℤ⟩, in escrutinio_senadores : seq⟨ℤ⟩, in escrutinio_diputados : seq⟨ℤ⟩) : Bool
  requiere {esEscrutinioValido(escrutinio_presidencial) ∧ esEscrutinioValido(escrutinio_senadores) ∧ esEscrutinioValido(escrutinio_diputados)}
  asegura {res = false ↔ noHayFraude(escrutinio_presidencial, escrutinio_senadores, escrutinio_diputados)}

  pred noHayFraude (escrutinio_presidencial : seq⟨ℤ⟩, escrutinio_senadores : seq⟨ℤ⟩, escrutinio_diputados : seq⟨ℤ⟩) {
    (sumaTotalDeVotos(escrutinio_presidencial) = sumaTotalDeVotos(escrutinio_senadores)) ∧
    (sumaTotalDeVotos(escrutinio_presidencial) = sumaTotalDeVotos(escrutinio_diputados))
  }
```

1.3. Ejercicio 3

```
proc obtenerSenadoresEnProvincia (in escrutinio : seq⟨ℤ⟩) : ℤ × ℤ
  requiere {esEscrutinioValido(escrutinio)}
  asegura { (∃i : ℤ) ((0 ≤ i < (|escrutinio| - 1) ∧L esListaMasVotada(escrutinio, i)) ↔ res0 = i) ∧
    (∃j : ℤ) ((0 ≤ j < (|escrutinio| - 1) ∧L esSegundaMasVotada(escrutinio, j) ↔ res1 = j) ) }

  pred esListaMasVotada (escrutinio : seq⟨ℤ⟩, i : ℤ) {
    (∀j : ℤ) ((0 ≤ j < |escrutinio| - 1 ∧ i ≠ j) →L escrutinio[i] > escrutinio[j])
  }

  pred esSegundaMasVotada (escrutinio : seq⟨ℤ⟩, i : ℤ) {
    (¬esListaMasVotada(escrutinio, i)) ∧ (∀j : ℤ) ((0 ≤ j < |escrutinio| - 1 ∧L ¬esListaMasVotada(escrutinio, j) ∧ i ≠ j) →L escrutinio[i] > escrutinio[j])
  }
```

1.4. Ejercicio 4

```
proc calcularDHontEnProvincia (in cant_bancas : ℤ, in escrutinio : seq⟨ℤ⟩) : seq⟨seq⟨ℤ⟩⟩
  requiere { (∃i : ℤ) (0 ≤ i < |escrutinio| ∧L SuperaElUmbral(escrutinio, i)) ∧ esEscrutinioValido(escrutinio) ∧ cant_bancas > 0 }
  asegura { |res| = (|escrutinio| - 1) ∧ (∀i : ℤ) (0 ≤ i < |res| - 1 →L res[i] ≤ cant_bancas ∧ res[i] = res[i + 1]) ∧
    (∀j : ℤ) (0 ≤ j < |res| →L res[i][0] = escrutinio[i]) ∧
    (∀k : ℤ) (0 ≤ k < |res| →L esFilaDHont(res[k])) }
```

```

pred esFilaDHont (fila : seq⟨ℤ⟩) {
  (∀j : ℤ) (1 ≤ j < |fila| →L fila[j] =  $\frac{fila[0]}{i+1}$ ) }

```

1.5. Ejercicio 5

```

proc obtenerDiputadosEnProvincia (in cant_bancas : ℤ, in escrutinio : seq⟨ℤ⟩, in dHont : seq⟨seq⟨ℤ⟩⟩) : seq⟨ℤ⟩
  requiere {cant_bancas > 0 ∧ esEscrutinioValido(escrutinio)
    ∧ esMatrizDHontValida(dHont, cant_bancas, escrutinio)}
  asegura {sumaDeBancas(res) = cant_bancas ∧ escañosAsignadosCorrectos(dHont, res)
    ∧ (∀i : ℤ) ((0 ≤ i < |res|) →L (res[i] > 0 ↔ superaUmbral(escrutinio)) ∧ (∀i : ℤ) ((0 ≤ i < |res|) →L res[i] > 0)
    ) }

aux sumaDeBancas (res : seq⟨ℤ⟩) : ℤ =  $\sum_{i=0}^{|res|-1} res[i]$ ;

pred escañosAsignadosCorrectos (dHont : seq⟨seq⟨ℤ⟩⟩, bancasAsignadas : seq⟨ℤ⟩) {
  (∀i : ℤ) ((0 ≤ i < |bancasAsignadas| ∧ bancasAsignadas[i] > 0) →L
  (∀j : ℤ) ((0 ≤ j < |bancasAsignadas| ∧ i ≠ j ∧ superaElUmbral(escrutinio, j)) →L
  dHont[i][bancasAsignadas[i] - 1] > dHont[j][bancasAsignadas[j]]) )
}

pred esMatrizDHontValida (dHont : seq⟨seq⟨ℤ⟩⟩, cant_bancas : ℤ, escrutinio : seq⟨ℤ⟩) {
  |dHont| = (|escrutinio| - 1) ∧ (∀i : ℤ) (0 ≤ i < |dHont| - 1 →L dHont[i] ≤ cant_bancas ∧ dHont[i] = dHont[i+1])
  ∧ (∀j : ℤ) (0 ≤ j < |dHont| →L dHont[i][0] = escrutinio[i]) ∧
  (∀k : ℤ) (0 ≤ k < |dHont| →L esFilaDHont(dHont[k])) }

```

1.6. Ejercicio 6

```

proc validarListasDiputadosEnProvincias (in cant_bancas : ℤ, in listas : seq⟨seq⟨dni : ent × genero : ℤ⟩⟩) : Bool
  requiere {cant_bancas > 0 ∧ (∀i : ℤ) (0 ≤ i < |listas| →L
  (∀j : ℤ) (0 ≤ j < |listas[i]| →L (0 ≤ listas[i][j]1 ≤ 1 ∧ 1,000,000 ≤ listas[i][j]0 ≤ 99,999,999)) ) }
  asegura {res = True ↔ alternanGenero(listas) ∧ presentanCantidadCorrecta(listas, cant_bancas)}

pred alternanGenero (listas : seq⟨seq⟨dni : ent × genero : ℤ⟩⟩) {
  (∀i : ℤ) (0 ≤ i < |listas| →L
  (∀j : ℤ) (0 ≤ j < (|listas[i]| - 1) →L listas[i][j]1 ≠ listas[i][j+1]1) )
}

pred presentanCantidadCorrecta (listas : seq⟨seq⟨dni : ent × genero : ℤ⟩⟩, cant_bancas : ℤ) {
  (∀i : ℤ) (0 ≤ i < |listas| →L |listas[i]| = cant_bancas) }

```