



TP de Especificación

Análisis Habitacional Argentino

8 de Septiembre de 2021

Algoritmos y Estructuras de Datos I

Grupo 02, comisión 11

Integrante	LU	Correo electrónico
Lakowsky, Manuel	511/21	mlakowsky@gmail.com
Lenardi, Juan Manuel	56/14	juanlenardi@gmail.com
Arienti, Federico	316/21	fa.arianti@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. Problemas

1.1. proc. esEncuestaValida

```
proc esEncuestaValida (in th:  $eph_h$ , in ti:  $eph_i$ , out result: Bool) {
  Pre {true}
  Post {res = true  $\leftrightarrow$  validarEncuesta(th, ti)}
}

pred validarEncuesta (th:  $eph_h$ , ti:  $eph_i$ ) {
  /* tabla hogares */
  (esTabla(th, @largoItemHogar)  $\wedge_L$ 
  ( $\forall h : hogar$ )( $h \in th \rightarrow_L$  (
    codigoValido $_h$ (th, ti, h)  $\wedge$  añoTrimestreCongruente $_h$ (th, h)  $\wedge$  attEnRango $_h$ (h)
  )))  $\wedge$ 
  /* tabla individuos */
  (esTabla(ti, @largoItemIndividuo)  $\wedge_L$ 
  ( $\forall i : individuo$ )( $i \in ti \rightarrow_L$  (
    codigoValido $_i$ (th, ti, i)  $\wedge$  añoTrimestreCongruente $_i$ (th, i)  $\wedge$  attEnRango $_i$ (i)  $\wedge$ 
    validarComponente $_i$ (ti, i)
  )))
}

pred codigoValido $_h$  (th:  $eph_h$ , ti:  $eph_i$ , h: hogar) {
  ( $\exists i : individuo$ )( $i \in ti \wedge_L$ 
    h[@hogcodusu] = i[@indcodusu]
  )  $\wedge$ 
   $\neg(\exists h_2 : hogar)((h_2 \in th \wedge h_2 \neq h) \wedge_L$ 
    h[@hogcodusu] = h $_2$ [@hogcodusu]
  )
}

pred añoTrimestreCongruente $_h$  (th:  $eph_h$ , h: hogar) {
  h[@hogaño] = th[0][@hogaño]  $\wedge$  h[@hogtrimestre] = th[0][@hogtrimestre]
}

pred attEnRango $_h$  (h: hogar) {
   $0 \leq h[@hogcodusu] \wedge$ 
   $1810 \leq h[@hogaño] \wedge$ 
   $1 \leq h[@hogtrimestre] \leq 4 \wedge$ 
   $-90 \leq th[i][@hoglatitud] \leq 90 \wedge$ 
   $-180 \leq th[i][@hoglongitud] \leq 180 \wedge$ 
   $1 \leq h[@ii7] \leq 3 \wedge$ 
   $1 \leq h[@region] \leq 6 \wedge$ 
   $0 \leq h[@mas_500] \leq 1 \wedge$ 
   $1 \leq h[@iv1] \leq 5 \wedge$ 
   $0 < h[@ii2] \leq h[@iv2] \wedge$ 
   $1 \leq h[@ii3] \leq 2$ 
}

pred codigoValido $_i$  (th:  $eph_h$ , ti:  $eph_i$ , i: individuo) {
  ( $\exists h : hogar$ )( $h \in th \wedge_L$ 
    i[@indcodusu] = h[@hogcodusu]
  )  $\wedge$ 
   $\neg(\exists i_2 : individuo)((i_2 \in ti \wedge i_2 \neq i) \wedge_L$ 
    i[@indcodusu] = i $_2$ [@indcodusu]  $\wedge$  i[@componente] = i $_2$ [@componente]
  )
}

pred añoTrimestreCongruente $_i$  (th:  $eph_h$ , i: individuo) {
  i[@indaño] = th[0][@hogaño]  $\wedge$  i[@indtrimestre] = th[0][@hogtrimestre]
}
```

```

}

pred attEnRangoi (i: individuo) {
  0 ≤ i[@indcodusu] ∧
  1 ≤ i[@componente] ≤ 20 ∧
  1 ≤ i[@ch4] ≤ 2 ∧
  0 ≤ i[@ch6] ∧
  0 ≤ i[@nivel_ed] ≤ 1 ∧
  -1 ≤ i[@estado] ≤ 1 ∧
  0 ≤ i[@cat_ocup] ≤ 4 ∧
  -1 ≤ i[@p47t] ∧
  1 ≤ i[@ppo4g] ≤ 10
}

pred validarComponentei (ti: ephi, i: individuo) {
  i[@componente] = 1 ∨ (∃ i2 : individuo)(i2 ∈ ti ∧L i[@componente] - 1 = i2[@componente])
}

```

1.2. proc. histHabitacional

```

proc histHabitacional (in th: ephh, in ti: ephi, in region: ℤ, out res: seq⟨ℤ⟩) {
  Pre {validarEncuesta(th, ti) ∧ HayCasasEnLaRegion(th, region)}
  Post {HayMaximoDeHabitaciones(res) ∧L
    (∀ i : ℤ)(0 ≤ i < |res| →L res[i] = #casasPorNroDeHabitaciones(th, k, i + 1))
  }
}

pred HayCasasEnLaRegion (th: ephh, region: ℤ) {
  (∃ h : hogar)(h ∈ th ∧L esHogarValidoParaHist(h, region))
}

pred esHogarValidoParaHist (h: hogar, region: ℤ) {
  h[@region] = region ∧ h[@iv1] = 1
}

pred HayMaximoDeHabitaciones (res: seq⟨ℤ⟩) {
  (∃ max : ℤ)(max = |res| ∧ res[max - 1] > 0)
}

aux #casasPorNroDeHabitaciones (th: ephh, region: ℤ, habitaciones: ℤ) : ℤ =
  ∑h ∈ th (if esHogarValidoParaHist(h, region) ∧ h[@iv2] = habitaciones then 1 else 0 fi);

```

1.3. proc. laCasaEstaQuedandoChica

```

proc laCasaEstaQuedandoChica (in th: ephh, in ti: ephi, out res: seq⟨ℤ⟩) {
  Pre {validarEncuesta(th, ti)}
  Post {|res| = 6 ∧L (∀ region : ℤ)(1 ≤ region ≤ 6 →L res[region - 1] = %hacinado(th, ti, region))
}

aux %hacinado (th: ephh, ti: ephi, region: ℤ) : ℝ =
  if ΩhacinadosNoVacio(th, region) then
    ∑h ∈ th (if casaHacinada(ti, h, region) then 1 else 0 fi)
  else 0 fi;

pred ΩhacinadosNoVacio (th: ephh, region: ℤ) {

```

```

    }
    (∃h :hogar)(h ∈ th ∧L esHogarValidoParaHacinamiento(h, region))
}

pred esHogarValidoParaHacinamiento (h: hogar, region: ℤ) {
    h[@region] = region ∧ h[@mas500] = 0 ∧ h[@iv1] = 1
}

pred casaHacinada (ti: ephi, h: hogar, region: ℤ) {
    esHogarValidoParaHacinamiento(h, region) ∧ #individuosEnHogar(ti, h[@hogcodusu]) > 3 * h[@iv2]
}

```

1.4. proc. creceElTeleworkingEnCiudadesGrandes

```

proc creceElTeleworkingEnCiudadesGrandes (in t1h: ephh, in t1i: ephi, in t2h: ephh, in t2i: ephi, out res: Bool) {
    Pre {
        (validarEncuesta(t1h, t1i) ∧ validarEncuesta(t2h, t2i)) ∧L
        (t1h[0][@hogaño] = t2h[0][@hogaño] - 1 ∧ t1h[0][@hogtrimestre] = t2h[0][@hogtrimestre])
    }
    Post {res = true ⇔ %teleworking(t1h, t1i) < %teleworking(t2h, t2i)}
}

aux %teleworking (th: ephh, ti: ephi) : ℝ =
    if ΩteleworkingNoVacio(th) then
        
$$\frac{\sum_{i \in ti} (\text{if } \text{haceTeleworking}(th, i) \text{ then } 1 \text{ else } 0 \text{ fi})}{\sum_{i \in ti} (\text{if } \text{viveEnHogarValido}(th, i) \text{ then } 1 \text{ else } 0 \text{ fi})}$$

    else 0 fi;

pred ΩteleworkingNoVacio (th: ephh) {
    (∃h :hogar)(h ∈ th ∧L esHogarValidoParaTeleworking(h))
}

pred esHogarValidoParaTeleworking (h: hogar) {
    h[@mas_500] = 1 ∧ (h[@iv1] = 1 ∨ h[@iv1] = 2)
}

pred haceTeleworking (th: ephh, i: individuo) {
    viveEnHogarValido(th, i) ∧ i[@ii3] = 1 ∧ i[@ppo4g] = 6
}

pred viveEnHogarValido (th: ephh, i: individuo) {
    esHogarValidoParaTeleworking(th[indiceHogarPorCodusu(th, i[@indcodusu])])
}

```

1.5. proc. costoSubsidioMejora

```

proc costoSubsidioMejora (in th: ephi, in ti: ephi, in monto: ℤ, out res: ℤ) {
    Pre {validarEncuesta(th, ti) ∧ monto > 0}
    Post {res = monto *  $\sum_{h \in th} (\text{if } \text{esHogarValidoParaSubsidio}(ti, h) \text{ then } 1 \text{ else } 0 \text{ fi})$ }
}

pred esHogarValidoParaSubsidio (ti: ephi, h: hogar) {
    h[@ii7] = 1 ∧ h[@iv1] = 1 ∧ individuosEnHogar(ti, h[@hogcodusu]) - 2 > h[@ii2]
}

```

2. Predicados y Auxiliares generales

2.1. predicados generales

```
pred esMatriz (s: seq⟨seq⟨T⟩⟩) {  
  (∀fila : seq⟨T⟩)(fila ∈ s →L |fila| = |s[0]|)  
}
```

```
pred esTabla (m: seq⟨seq⟨T⟩⟩, columnas: ℤ) {  
  |m| > 0 ∧L (|m[0]| = columnas ∧ esMatriz(m))  
}
```

2.2. auxiliares generales

```
aux #individuosEnHogar (ti: ephi, codusuh: ℤ) : ℤ = ∑i ∈ ti (if i[@indcodusu] = codusuh then 1 else 0 fi);  
  
/* indiceHogarPorCodusu asume codusuh existe en la tabla y es único */  
aux indiceHogarPorCodusu (th: ephh, codusuh: ℤ) : ℤ = ∑h ∈ th if h[@hogcodusu] = codusuh then i else 0 fi;
```

2.3. tipos y enumerados

```
type dato = ℤ  
type individuo = seq⟨dato⟩  
type hogar = seq⟨dato⟩  
type ephi = seq⟨individuo⟩  
type ephh = seq⟨hogar⟩  
type joinHI = seq⟨hogar × individuo⟩  
  
enum ItemHogar {  
  hogcodusu, hogaño, hogtrimestre, hoglatitud, hoglongitud, ii7, region, mas_500, iv1, iv2, ii2, ii3  
}  
enum ItemIndividuo {  
  indcodusu, componente, indaño, indtrimestre, ch4, ch6, nivel_ed, cat_ocup, p47t, ppo4g  
}
```

2.4. referencias

```
aux @hogcodusu : ℤ = itemHogar.ord(hogcodusu);  
aux @hogaño : ℤ = itemHogar.ord(hogaño);  
aux @hogtrimestre : ℤ = itemHogar.ord(hogtrimestre);  
aux @hoglatitud : ℤ = itemHogar.ord(hoglatitud);  
aux @hoglongitud : ℤ = itemHogar.ord(hoglongitud);  
aux @ii7 : ℤ = itemHogar.ord(ii7);  
aux @region : ℤ = itemHogar.ord(region);  
aux @mas_500 : ℤ = itemHogar.ord(mas_500);  
aux @iv1 : ℤ = itemHogar.ord(iv1);  
aux @iv2 : ℤ = itemHogar.ord(iv2);  
aux @ii2 : ℤ = itemHogar.ord(ii2);  
aux @ii3 : ℤ = itemHogar.ord(ii3);  
  
aux @indcodusu : ℤ = itemIndividuo.ord(indcodusu);  
aux @componente : ℤ = itemIndividuo.ord(componente);  
aux @indaño : ℤ = itemIndividuo.ord(indaño);  
aux @indtrimestre : ℤ = itemIndividuo.ord(indtrimestre);  
aux @ch4 : ℤ = itemIndividuo.ord(ch4);  
aux @ch6 : ℤ = itemIndividuo.ord(ch6);  
aux @nivel_ed : ℤ = itemIndividuo.ord(nivel_ed);  
aux @cat_ocup : ℤ = itemIndividuo.ord(cat_ocup);  
aux @p47t : ℤ = itemIndividuo.ord(p47t);  
aux @ppo4g : ℤ = itemIndividuo.ord(ppo4g);
```

```
aux @largoItemHogar :  $\mathbb{Z}$  = 12;  
aux @largoitemIndividuo :  $\mathbb{Z}$  = 10;
```

3. Decisiones tomadas

3.1. Ejercicio 4.2. Igualdad $\text{res}[i] = \text{HabitacionesPorCasa}(\text{th}, \text{region}, i+1)$ en Postcondicion

Se asume que de existir al menos una casa entonces la misma posee al menos una habitacion o un ambiente usado a tal fin.
No hay casa con cero habitaciones