



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

Análisis Habitacional Argentino

8 de septiembre de 2021

Algoritmos y Estructuras de Datos I

Grupo 6

Integrante	LU	Correo electrónico
Romero, Santiago	272/21	santiagooromero1234@gmail.com
Rosselot, Eduardo	924/11	earosselot@gmail.com
Magi, Julian	829/21	magijulian0@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. Encuesta válida

```
proc esEncuestaValida (in th: ephh, in ti: ephi, out result: Bool) {
  Pre {true}
  Post {result = true  $\iff$  cumpleRequisitos(th, ti)}
}

pred cumpleRequisitos (th: ephh, ti: ephi) {
  encuestaHogaresCorrecta(th)  $\wedge$  encuestaIndividuosCorrecta(ti)  $\wedge$  encuestasSonCompatibles(th, ti)
}

pred encuestaHogaresCorrecta (th: ephh) {
  esMatriz(th)  $\wedge$  noEsVacio(th)  $\wedge$  cantidadDeColumnasCorrecta(th, 12)  $\wedge$  noHayHogaresRepetidos(th)
   $\wedge$  latLongValidas(th)  $\wedge$  hogaresPosibles(th)  $\wedge$  atributosHogaresEnRango(th)
}

pred encuestaIndividuosCorrecta (ti: ephi) {
  esMatriz(ti)  $\wedge$  noEsVacio(ti)  $\wedge$  cantidadDeColumnasCorrecta(ti, 11)  $\wedge$  noHayIndividuosRepetidos(ti)
   $\wedge$  atributosIndividuosEnRango(ti)
}

pred encuestasSonCompatibles (th: ephh, ti: ephi) {
  todosHogaresTienenIndividuos(th, ti)  $\wedge$  todosIndividuosTienenHogar(th, ti)  $\wedge$  igualAñoTrimestre(th, ti)
   $\wedge$  miembrosMenorA20(th, ti)
}

pred esMatriz (t: seq<seq< $\mathbb{Z}$ >>) {
  ( $\exists x : \mathbb{Z}$ ) ( $\forall i : \mathbb{Z}$ ) (enRango(i, t)  $\longrightarrow_L$  |t[i]| = x)
}

pred noEsVacio (t: seq<seq< $\mathbb{Z}$ >>) {
  |t| > 0
}

pred cantidadDeColumnasCorrecta (t: seq<seq< $\mathbb{Z}$ >>, n:  $\mathbb{Z}$ ) {
  ( $\forall i : \mathbb{Z}$ ) (enRango(i, t)  $\longrightarrow_L$  |t[i]| = n)
}

pred todosHogaresTienenIndividuos (th: ephh, ti: ephi) {
  ( $\forall y : \text{hogar}$ ) ( $y \in th \longrightarrow_L (\exists x : \text{individuo}) (x \in ti \wedge_L x[@\text{indcodusu}] = y[@\text{hogcodusu}])$ )
}

pred todosIndividuosTienenHogar (th: ephh, ti: ephi) {
  ( $\forall x : \text{individuo}$ ) ( $x \in ti \longrightarrow_L (\exists y : \text{hogar}) (y \in th \wedge_L x[@\text{indcodusu}] = y[@\text{hogcodusu}])$ )
}

pred noHayIndividuosRepetidos (ti: ephi) {
  ( $\forall x, y : \text{individuo}$ ) ( $((x \in ti \wedge y \in ti \wedge x \neq y) \longrightarrow_L ((x[@\text{indcodusu}] \neq y[@\text{indcodusu}]) \vee (x[@\text{componente}] \neq y[@\text{componente}])))$ )
}

pred noHayHogaresRepetidos (th: ephh) {
  ( $\forall x, y : \text{hogar}$ ) ( $((x \in th \wedge y \in th \wedge x \neq y) \longrightarrow_L x[@\text{hogcodusu}] \neq y[@\text{hogcodusu}])$ )
}

pred latLongValidas (th: ephh) {
  ( $\forall x : \text{hogar}$ ) ( $x \in th \longrightarrow_L \text{latitudValida}(x[@\text{hoglatitud}]) \wedge \text{longitudValida}(x[@\text{hoglongitud}])$ )
}
```

Tomamos como válidas las latitudes y longitudes posibles en la Tierra $[-90, 90]$ y $[-180, 180]$ respectivamente). Si bien se pueden normalizar números fuera del rango, entendemos que ningún dispositivo de medición lo daría fuera del rango y es mas probable que sea un error en la toma del dato.

```
pred latitudValida (latitud:  $\mathbb{Z}$ ) {
   $-90 \leq \text{latitud} \leq 90$ 
}

pred longitudValida (longitud:  $\mathbb{Z}$ ) {
   $-180 \leq \text{longitud} \leq 180$ 
}
```

```

pred igualAñoTrimestre (th: ephh, ti: ephi) {
  igualAño(th, ti) ∧ igualTrimestre(th, ti)
}
pred igualAño (th: ephh, ti: ephi) {
  (∃ año : ℤ)((∀ x : hogar)(x ∈ th →L x[@hogaño] = año) ∧ (∀ y : individuo)(y ∈ ti →L y[@indaño] = año))
}
pred igualTrimestre (th: ephh, ti: ephi) {
  (∃ trimestre : ℤ)((∀ x : hogar)(x ∈ th →L x[@hogtrimestre] = trimestre) ∧
  (∀ y : individuo)(y ∈ ti →L y[@indtrimestre] = trimestre))
}
pred miembrosMenorA20 (th: ephh, ti: ephi) {
  (∀ x : hogar)(x ∈ th →L cantidadIndividuosEnHogar(x[@hogcodusu], ti) ≤ 20)
}
aux cantidadIndividuosEnHogar (hogcodusu: ℤ, ti: ephi) : ℤ = ∑i=0|ti|-1 if hogcodusu = ti[i][@indcodusu] then 1 else 0 fi ;
pred hogaresPosibles (th: ephh) {
  (∀ x : hogar)(x ∈ th →L x[@ii2] ≤ x[@iv2])
}
pred atributosIndividuosEnRango (ti: ephi) {
  (∀ x : individuo)(x ∈ ti →L (datoEnRango(x[@ch4], 1, 2) ∧ datoEnRango(x[@nivel_ed], 0, 1) ∧
  datoEnRango(x[@estado], -1, 1) ∧ datoEnRango(x[@cat_ocup], 0, 4) ∧ datoEnRango(x[@pp04g], 1, 10)))
}
pred atributosHogaresEnRango (th: ephh) {
  (∀ x : hogar)(x ∈ th →L (datoEnRango(x[@ii7], 1, 3) ∧ datoEnRango(x[@region], 1, 6) ∧ datoEnRango(x[@mas_500], 0, 1)
  ∧ datoEnRango(x[@iv1], 1, 5) ∧ datoEnRango(x[@ii3], 1, 2)))
}
pred datoEnRango (n: dato, y, z: ℤ) {
  y ≤ n ≤ z
}

```

1.2. Histograma habitacional

La secuencia res, devuelve el histograma, donde cada posición i representa la cantidad de casas con i+1 habitaciones. Ya que consideramos que no tiene sentido que el histograma tenga un valor para casas con 0 habitaciones.

```

proc histHabitacional (in th: ephh, in ti: ephi, in region: ℤ, out res: seq⟨ℤ⟩) {
  Pre {1 ≤ region ≤ 6 ∧ cumpleRequisitos(th, ti)}
  Post {largoCorrecto(th, region, res) ∧ esHistograma(th, region, res)}
}
pred esHistograma (th: ephh, region: ℤ, res: seq⟨ℤ⟩) {
  (∀ i : ℤ)(enRango(i, res) →L res[i] = numCasasConHabEnRegion(th, region, i + 1))
}
aux numCasasConHabEnRegion (th: ephh, region: ℤ, n: ℤ) : ℤ =
  ∑k=0|th|-1 if (th[k][@iv2] = i ∧ th[k][@region] = region ∧ th[k][@iv1] = 1) then 1 else 0 fi ;
pred largoCorrecto (th: ephh, region: ℤ, res: seq⟨ℤ⟩) {
  (∃ i : ℤ)((enRango(i, th) ∧L esRegionyCasa(th, region, i)) ∧L
  (∀ j : ℤ)((enRango(j, th) ∧L esRegionyCasa(th, region, j)) →L (th[j][@iv2] ≤ th[i][@iv2] ∧ |res| = th[i][@iv2])))
}
pred esRegionyCasa (th: ephh, region: ℤ, x: ℤ) {
  th[x][@iv1] = 1 ∧ th[x][@region] = region
}

```

1.3. La casa esta quedando chica

```

proc laCasaEstaQuedandoChica (in th: ephh, in ti: ephi, out res: seq⟨ℝ⟩) {
  Pre {cumpleRequisitos(th, ti)}
  Post {|res| = 6 ∧ (∀ i : ℤ)(enRango(i, res) →L res[i] =  $\frac{\text{sumaHogaresHac}(th, ti, i+1)}{\text{sumaHogares}(th, i+1)})$ }
}

```

aux sumaHogares (th: eph_h , region: \mathbb{Z}) : $\mathbb{Z} = \sum_{i=0}^{|th|} -1$ if $esRegionyCasa(th, region, i) \wedge \neg masDe500(th, i)$ then 1 else 0 fi ;

esRegionyCasa definido en sección 1.2.

pred masDe500 (th: eph_h , i: \mathbb{Z}) {
 $th[i][@mas_500] = 1$
}

aux sumaHogaresHac (th: eph_h , ti: eph_i , region: \mathbb{Z}) : $\mathbb{Z} = \sum_{i=0}^{|th|} -1$ if $esRegionyCasa(th, region, i) \wedge \neg masDe500(th, i) \wedge hacCritico(th, ti, i)$ then 1 else 0 fi ;

pred hacCritico (th: eph_h , ti: eph_i , i: \mathbb{Z}) {
 $\frac{numPersonasCasa(th, ti, i)}{th[i][@iv2]} > 3$
}

aux numPersonasCasa (th: eph_h , ti: eph_i , n: \mathbb{Z}) : $\mathbb{Z} = \sum_{i=0}^{|ti|} -1$ if $ti[i][@indcodusu] = th[n][@hogcodusu]$ then 1 else 0 fi ;

1.4. Crece el teletrabajo

proc creceElTeleworkingEnCiudadesGrandes (in t1h: eph_h , in t1i: eph_i , in t2h: eph_h , in t2i: eph_i , out res: Bool) {
Pre { $(cumpleRequisitos(t1h, t1i) \wedge cumpleRequisitos(t2h, t2i)) \wedge_L$
 $(añosCorrectos(t1h, t2h) \wedge mismoTrimestre(t1h, t2h))$ }
Post { $(res = true) \iff (\frac{proporcionTeleworking(t2h, t2i)}{proporcionTeleworking(t1h, t1i)} > 1)$ }
}

pred añosCorrectos (t1h: eph_h , t2h: eph_h) {
 $(\forall i, j : \mathbb{Z})((enRango(i, t1h) \wedge enRango(j, t2h)) \longrightarrow_L t1h[i][@hogaño] < t2h[j][@hogaño])$
}

pred mismoTrimestre (t1h: eph_h , t2h: eph_h) {
 $(\forall i, j : \mathbb{Z})((enRango(i, t1h) \wedge enRango(j, t2h)) \longrightarrow_L t1h[i][@hogtrimestre] = t2h[j][@hogtrimestre])$
}

aux proporcionTeleworking (th: eph_h , ti: eph_i) : $\mathbb{R} = \frac{cantIndCasaDepTeleworking(th, ti)}{cantIndCasaDep(th, ti)}$;

aux cantIndCasaDepTeleworking (th: eph_h , ti: eph_i) : $\mathbb{Z} = \sum_{k=0}^{|ti|} -1$ if $viveEnCasaDep500(th, ti, k) \wedge haceTeleworking(th, ti, k)$ then 1 else 0 fi ;

aux cantIndCasaDep (th: eph_h , ti: eph_i) : $\mathbb{Z} = \sum_{k=0}^{|ti|} -1$ if $viveEnCasaDep500(th, ti, k)$ then 1 else 0 fi ;

pred viveEnCasaDep500 (th: eph_h , ti: eph_i , k: \mathbb{Z}) {
 $(viveEnCasa(th, ti, k) \vee viveEnDep(th, ti, k)) \wedge viveEnMas500(th, ti, k)$
}

pred haceTeleworking (th: eph_h , ti: eph_i , k: \mathbb{Z}) {
 $trabajaEnCasa(ti, k) \wedge tieneHabitacionDeTrabajo(th, ti, k)$
}

pred viveEnCasa (th: eph_h , ti: eph_i , k: \mathbb{Z}) {
 $th[buscarHogar(th, ti[k][@indcodusu])][@iv1] = 1$
}

pred viveEnDep (th: eph_h , ti: eph_i , k: \mathbb{Z}) {
 $th[buscarHogar(th, ti[k][@indcodusu])][@iv1] = 2$
}

pred viveEnMas500 (th: eph_h , ti: eph_i , k: \mathbb{Z}) {
 $th[buscarHogar(th, ti[k][@indcodusu])][@mas500] = 1$
}

pred trabajaEnCasa (ti: eph_i , k: \mathbb{Z}) {
 $ti[k][@pp04g] = 6$
}

pred tieneHabitacionDeTrabajo (th: eph_h , ti: eph_i , k: \mathbb{Z}) {
 $th[buscarHogar(th, ti[k][@indcodusu])][@ii3] = 1$
}

aux buscarHogar (th: eph_h , hogcodusu: \mathbb{Z}) : $\mathbb{Z} = \sum_{k=0}^{|th|} -1$ if $th[k][@hogcodusu] = hogcodusu$ then k else 0 fi ;

1.5. Costo de Subsidio de Mejora

```
proc costoSubsidioMejora (in th:  $eph_h$ , in ti:  $eph_i$ , in monto:  $\mathbb{Z}$ , out res:  $\mathbb{Z}$ ) {  
    Pre {cumpleRequisitos(th, ti)}  
    Post {cantidadDeSubsidiados(th, ti)  $\times$  monto = res}  
}  
  
aux cantidadDeSubsidiados (th:  $eph_h$ , ti:  $eph_i$ ) :  $\mathbb{Z} = \sum_{k=0}^{|th|-1}$  if cumplenTenencia(th[k])  $\wedge$  cumplenHab(th, ti, k) then 1 else 0 fi;  
  
pred cumplenTenencia (h: hogar) {  
    h[@ii7] = 1  
}  
  
numPersonasCasa definido en 1.3  
  
pred cumplenHab (th:  $eph_h$ , ti:  $eph_i$ , k:  $\mathbb{Z}$ ) {  
    th[k][@ii2] < numPersonasCasa(th, ti, k) - 2  
}
```

2. Predicados y Auxiliares generales

```
pred enRango (n:  $\mathbb{Z}$ , l: seq( $T$ )) {  
     $0 \leq n < |l|$   
}
```

2.1. Auxiliares hogares

```
enum ItemHogar {  
    HOGCODUSU, HOGAÑO, HOGTRIMESTRE, HOGLATITUD, HOGLONGITUD, II7, REGION, MAS_500, IV1, IV2,  
    II2, II3  
}  
  
aux @hogcodusu :  $\mathbb{Z} = ord(HOGCODUSU)$ ;  
aux @hogañO :  $\mathbb{Z} = ord(HOGAÑO)$ ;  
aux @hogtrimestre :  $\mathbb{Z} = ord(HOGTRIMESTRE)$ ;  
aux @hoglatitud :  $\mathbb{Z} = ord(HOGLATITUD)$ ;  
aux @hoglongitud :  $\mathbb{Z} = ord(HOGLONGITUD)$ ;  
aux @ii7 :  $\mathbb{Z} = ord(II7)$ ;  
aux @region :  $\mathbb{Z} = ord(REGION)$ ;  
aux @mas_500 :  $\mathbb{Z} = ord(MAS_500)$ ;  
aux @iv1 :  $\mathbb{Z} = ord(IV1)$ ;  
aux @iv2 :  $\mathbb{Z} = ord(IV2)$ ;  
aux @ii2 :  $\mathbb{Z} = ord(II2)$ ;  
aux @ii3 :  $\mathbb{Z} = ord(II3)$ ;
```

2.2. Auxiliares individuos

```
enum ItemIndividuo {  
    INDCODUSU, COMPONENTE, INDAÑO, INDTRIMESTRE, CH4, CH6, NIVEL_ED, ESTADO, CAT_OCUP, P47T,  
    PP04G  
}  
  
aux @indcodusu :  $\mathbb{Z} = ord(INDCODUSU)$ ;  
aux @componente :  $\mathbb{Z} = ord(COMPONENTE)$ ;  
aux @indaño :  $\mathbb{Z} = ord(INDAÑO)$ ;  
aux @indtrimestre :  $\mathbb{Z} = ord(INDTRIMESTRE)$ ;  
aux @ch4 :  $\mathbb{Z} = ord(CH4)$ ;  
aux @ch6 :  $\mathbb{Z} = ord(CH6)$ ;  
aux @nivel_ed :  $\mathbb{Z} = ord(NIVEL_ED)$ ;  
aux @estado :  $\mathbb{Z} = ord(ESTADO)$ ;  
aux @cat_ocup :  $\mathbb{Z} = ord(CAT_OCUP)$ ;  
aux @p47T :  $\mathbb{Z} = ord(P47T)$ ;  
aux @pp04g :  $\mathbb{Z} = ord(PP04G)$ ;
```