



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

Análisis Habitacional Argentino

1 de Octubre de 2021

Algoritmos y Estructuras de Datos I

Grupo 2

Integrante	LU	Correo electrónico
Pizan, Alessandro	240/21	apizan@dc.uba.ar
Móline, Clara	412/21	cmoline@dc.uba.ar
Solari, Braulio	575/21	brauliosolari@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. Predicados y Auxiliares generales

```
enum ItemIndividuo {INDCODUSU, INDAni0, INDTRIMESTRE, COMPONENTE, CH4, CH6,
NIVEL_ED, ESTADO, CAT_OCUP, p47T, PP04G}
enum ItemHogar {HOGCODUSU, HOGAni0, HOGTRIMESTRE, HOGLATITUD,
HONGLONGITUD, II7, REGION, MAS_500, IV1, IV2, II2, II3}
```

```
pred rango (s: seq(T), i: Z) {
0 ≤ i < |s|
}
```

TI *

```
aux @INDCODUSU : Z = 0;
aux @INDAni0 : Z = 1;
aux @INDTRIMESTRE : Z = 2;
aux @COMPONENTE : Z = 3;
aux @CH4 : Z = 4;
aux @CH6 : Z = 5;
aux @NIVEL_ED : Z = 6;
aux @ESTADO : Z = 7;
aux @CATO_CUP : Z = 8;
aux @p47t : Z = 9;
aux @PP04G : Z = 10;
```

TH *

```
aux @HOGCODUSU : Z = 0;
aux @HOGAni0 : Z = 1;
aux @HOGTRIMESTRE : Z = 2;
aux @HOGLATITUD : Z = 3;
aux @HONGLONGITUD : Z = 4;
aux @II7 : Z = 5;
aux @REGION : Z = 6;
aux @MAS_500 : Z = 7;
aux @IV1 : Z = 8;
aux @IV2 : Z = 9;
aux @II2 : Z = 10;
aux @II3 : Z = 11;
```

2. Problemas

2.1. Problema 1

```
proc esEncuestaValida (in th: ephh, in ti: ephi, out result: Bool) {
  Pre {True}
  Post {result = true ⇔ encuestaValida(th, ti)}
}

pred encuestaValida (th: ephh, ti: ephi) {
(|th| > 0) ∧ (|ti| > 0) ∧ (mismaLongitud(th)) ∧ (mismaLongitud(ti)) ∧ (columnasYVariables(th, ti))
∧ (individuosYHogares(th, ti)) ∧ (sinHogaresRepetidos(th))
∧ (sinIndividuosRepetidos(ti)) ∧ (mismoAnioYTrimestre(th, ti))
∧ (mismoAnioYTrimestre(ti, th)) ∧ (menosDe20(ti)) ∧ (habitacionesParaDormir(th))
∧ (hogaresEnRango(th)) ∧ (individuosEnRango(ti))
}

pred columnasYVariables (th: ephh, ti: ephi) {
(∀i : Z)(rango(th, i) →L (|th[i]| = 12) ∧ (∀j : Z)(rango(ti, j) →L (|ti[j]| = 11))
}

pred individuosYHogares (th: ephh, ti: ephi) {
(∀i : Z)(rango(th, i) →L (∃j : Z)(rango(ti, j) →L (th[i][@HOGCODUSU] = ti[j][@INDCODUSU]))
∧ (∀j : Z)(rango(th, j) →L (∃i : Z)(rango(ti, i) →L (ti[i][@INDCODUSU] = th[j][@HOGCODUSU]))
}

pred sinHogaresRepetidos (th: ephh) {
(∀i : Z)(rango(th, i) →L ¬(∃j : Z)(rango(th, j) ∧L
```

2

aux gentePorCasa (th: eph_h, ti: eph_i, i:Z) : Z = $\sum_{k=0}^{|ti|-1}$ if (ti[k][@INDCODUSU] = i) then 1 else 0 fi ;

2.4. Problema 4

```

proc creceElTeleworkingEnCiudadesGrandes (in t1h: ephh, in t1i: ephi, in t2h: ephh, in t2i: ephi, out res:Bool) {
  Pre { encuestaValida(t1h, t1i) ∧ encuestaValida(t2h, t2i) ∧L ((t2h[0][@HOGAniO] > t1h[0][@HOGAniO])
    ∧ (t2h[0][@HOGTRIMESTRE] = t1h[0][@HOGTRIMESTRE])) }
  Post { res = (trabajadoresEnCasa(t2h, t2i) > trabajadoresEnCasa(t1h, t1i)) }
}

pred trabajaEnCasa (th: ephh, ti: ephi, k:Z) {
  (ti[k][@PP04G] = 6) ∧ (∃ i : Z)(rango(th, i) ∧L (th[i][@HOGCODUSU] = ti[k][@INDCODUSU])
  ∧ (th[i][@IV1] = 1 ∨ th[i][@IV1] = 2) ∧ (th[i][@II3] = 1) ∧ (ti[k][@II7] = 1))
}

aux trabajadoresEnCasa (th: ephh, ti: ephi) : Z =  $\sum_{k=0}^{|ti|-1}$  if trabajaEnCasa(th, ti, k) then 1 else 0 fi ;

```

2.5. Problema 5

```

proc costoSubsidioMejora (in th: ephh, in ti: ephi, in monto: Z, out res:Z) {
  Pre { encuestaValida(th, ti) }
  Post { res = casasASubsidiar(th, ti) * monto }
}

pred esCasaASubsidiar (th: ephh, ti: ephi, i:Z) {
  (th[i][@II7] = 1) ∧ (th[i][@IV2] < (gentePorCasa(th, ti, th[i][@HOGCODUSU]) - 2))
}

aux casasASubsidiar (th: ephh, ti: ephi) : Z =  $\sum_{k=0}^{|th|-1}$  if esCasaASubsidiar(th, ti, k) then 1 else 0 fi ;

```

3. Parte 2

3.1. Problema 6

```

proc generarJoin (in th: ephh, in ti: ephi, out junta: joinHI) {
  Pre { encuestaValida(th, ti) }
  Post { juntaVieneDeEncuesta(th, ti, junta) ∧L individuosEnTuplas(th, ti, junta) ∧ casasEnTuplas(th, ti, junta) }
}

pred individuosEnTuplas (th: ephh, ti: ephi, junta: joinHI) {
  (∀ i : Z)(∃ k : Z)(rango(ti, i) →L ((junta[k]0[@HOGCODUSU] = ti[i][@INDCODUSU]) ∧ (junta[k]1 = ti[i])))
}

pred casasEnTuplas (th: ephh, ti: ephi, junta: joinHI) {
  (∀ i : Z)(junta[i]0 = th[casaDelIndividuoI(th, ti, junta[i]1[@INDCODUSU])])
}

pred juntaVieneDeEncuesta (th: ephh, ti: ephi, junta: joinHI) {
  (∀ i : Z)(rango(junta, i) →L ((∃ k : Z)(∃ j : Z)(rango(th, k) ∧L (rango(ti, j)
  ∧L (junta[i]0 = th[k] ∧ junta[i]1 = ti[j]))
}

aux casaDelIndividuoI (th: ephh, ti: ephi, i:Z) : Z =  $\sum_{k=0}^{|th|-1}$  if (th[k][@HOGCODUSU] = i) then k else 0 fi ;

```

3.2. Problema 7

```

proc ordenarRegionYTipo (inout th: ephh, inout ti: ephi) {
  Pre { encuestaValida(th, ti) ∧ th = th0 ∧ ti = ti0 }
  Post { casasOrdenadas(th) ∧ individuosOrdenados(th, ti) ∧ conservanValoresOriginales(th, ti, th0, ti0) }
}

pred casasOrdenadas (th: ephh) { (∀ i : Z)(0 ≤ i < |th| - 1 →L ((th[i][@REGION] ≤ th[i+1][@REGION])
  ∧ ((th[i][@REGION] = th[i+1][@REGION]) → (th[i][@HOGCODUSU] < th[i+1][@HOGCODUSU])))))
}

pred individuosOrdenados (th: ephh, ti: ephi) { (∀ i : Z)(0 ≤ i < |ti| - 1
  →L (casaDelIndividuoI(th, ti, ti[i][@INDCODUSU]) ≤ casaDelIndividuoI(th, ti, ti[i+1][@INDCODUSU])
  ∧ (ti[i][@INDCODUSU] = ti[i+1][@INDCODUSU] → ti[i][@COMPONENTE] < ti[i+1][@COMPONENTE])))
}

```

```

    pred conservanValoresOriginales (th: ephh, ti: ephi, th0: ephh, ti0: ephi) {
    (|th| = |th0|) ∧ (∀i : ℤ)(rango(th, i) →L ((∃j : ℤ)(rango(th0, j) ∧L th[i] = th0[j])))
    (|ti| = |ti0|) ∧ (∀k : ℤ)(rango(ti, k) →L ((∃h : ℤ)(rango(ti0, h) ∧L ti[k] = ti0[h])))}

```

3.3. Problema 8

```

proc muestraHomogenea (in th: ephh, in ti: ephi, out res: seq⟨hogar⟩) {
    Pre {encuestaValida(th, ti)}
    Post {( |res| ≥ 3 →L (casasDeIgualDiferencia(th, ti, res) ∧ resDeMaximaLongitud(th, ti, res) ∧ resOrdenada(ti, res)
    ∧ soloTieneCasasDeTH(th, res))) ∧ ( |res| < 3 →L (¬(∃k : seq⟨hogar⟩)(casasDeIgualDiferencia(th, ti, k)
    ∧ resDeMaximaLongitud(th, ti, k) ∧ resOrdenada(ti, k) ∧ soloTieneCasasDeTH(th, k) ∧ |k| ≥ 3) ∧ (|res| = 0)))}
}

aux ingresosPorCasa (codigo: ℤ, ti: ephi) : ℤ =
    ∑k=0|ti|-1 if ti[i][@INDCODUSU] = codigo then ti[i][@p47T] else 0 fi
;

    pred casasDeIgualDiferencia (th: ephh, ti: ephi, res: seq⟨hogar⟩) {
    (∀i : ℤ)(0 ≤ i < |res| - 2 →L
    (|ingresosPorCasa(res[i][@HOGCODUSU], ti) - ingresosPorCasa(res[i + 1][@HOGCODUSU], ti)| =
    |ingresosPorCasa(res[i + 1][@HOGCODUSU], ti) - ingresosPorCasa(res[i + 2][@HOGCODUSU], ti)|)
    )
    }

    pred resDeMaximaLongitud (th: ephh, ti: ephi, res: seq⟨hogar⟩) {
    (casasDeIgualDiferencia(th, ti, res) ∧ soloTieneCasasDeTH(th, res)) ∧
    ¬(∃k : seq⟨hogar⟩)((casasDeIgualDiferencia(th, ti, k) ∧ soloTieneCasasDeTH(th, k) ∧ |k| > |res|)
    )
    }

    pred soloTieneCasasDeTH (th: ephh res: seq⟨hogar⟩) {
    (∀i : ℤ)(rango(res, i) →L (∃k : ℤ)(rango(th, k) ∧L (res[i] = th[k])))
    }

    pred resOrdenada (ti: ephi, res: seq⟨hogar⟩) {
    (∀i : ℤ)(rango(res, i) →L
    (ingresosPorCasa(res[i][@HOGCODUSU], ti) < ingresosPorCasa(res[i + 1][@HOGCODUSU], ti)))
    }

```

3.4. Problema 9

```

proc corregirRegion (inout th: ephh, in ti: ephi) {
    Pre {encuestaValida(th, ti) ∧ th = th0}
    Post {cambiarRegion(th)}
}

    pred cambiarRegion (th: ephh, th0: ephh) {
    (∀i : ℤ)(rango(th, i) →L ((th0[i][@REGION] = 1) → th[i][@REGION] = 5
    ∧ (th0[i][@REGION] ≠ 1 → th[i][@REGION] = th0[i][@REGION])
    ∧ ((th0[i][@HOGCODUSU] ≠ 1) → th[i][@HOGCODUSU] = th0[i][@HOGCODUSU])
    ∧ ((th0[i][@HOGAniO] ≠ 1) → th[i][@HOGAniO] = th0[i][@HOGAniO])
    ∧ ((th0[i][@HOGTRIMESTRE] ≠ 1) → th[i][@HOGTRIMESTRE] = th0[i][@HOGTRIMESTRE])
    ∧ ((th0[i][@HOGLATITUD] ≠ 1) → th[i][@HOGLATITUD] = th0[i][@HOGLATITUD])
    ∧ ((th0[i][@HOGLONGITUD] ≠ 1) → th[i][@HOGLONGITUD] = th0[i][@HOGLONGITUD])
    ∧ ((th0[i][@II7] ≠ 1) → th[i][@II7] = th0[i][@II7])
    ∧ ((th0[i][@MAS500] ≠ 1) → th[i][@MAS500] = th0[i][@MAS500])
    ∧ ((th0[i][@IV1] ≠ 1) → th[i][@IV1] = th0[i][@IV1])
    ∧ ((th0[i][@IV2] ≠ 1) → th[i][@IV2] = th0[i][@IV2])
    ∧ ((th0[i][@II2] ≠ 1) → th[i][@II2] = th0[i][@II2])
    ∧ ((th0[i][@II3] ≠ 1) → th[i][@II3] = th0[i][@II3])))}

```

3.5. Problema 10

```

proc histogramaDeAnillosConcentricos (in th: ephh, in centro: ℤ × ℤ, in distancias : seq⟨ℤ⟩, out result : seq⟨ℤ⟩){
    Pre {distanciasValidas(distancias)}
    Post {( |result| = |distancias| ∧L hogaresEnAnillos(th, centro, distancias, result) }
}

```

```

    pred hogaresEnAnillos (th: ephh, centro:  $\mathbb{Z} \times \mathbb{Z}$ , distancias : seq( $\mathbb{Z}$ ), result : seq( $\mathbb{Z}$ )) {
    (∀i :  $\mathbb{Z}$ )(rango(result, i) →L (result[i] = casasEnAnilloI(th, centro, distancias, result)))
    aux casasEnAnilloI (th: ephh, centro:  $\mathbb{Z} \times \mathbb{Z}$ , distancias : seq( $\mathbb{Z}$ ), i :  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =
    ∑k=0|th|-1 if estaEnAnilloI(th, centro, distancias, k) then 1 else 0 fi
    ;
    pred estaEnAnilloI (th: ephh, centro:  $\mathbb{Z} \times \mathbb{Z}$ , distancias : seq( $\mathbb{Z}$ ), i :  $\mathbb{Z}$ ) {
    ((i > 0) →L ((distanciaAlCentro(th, centro, i) ≤ distancias[i]) ∧ (distanciaAlCentro(th, centro, i) > distancias[i-1]))) ∧
    ((i = 0) → (distanciaAlCentro(th, centro, i) ≤ distancias[i]))
    }
    pred distanciasValidas (distancias: seq( $\mathbb{Z}$ )) {
    (|distancias| > 0) ∧ (∀i :  $\mathbb{Z}$ )(0 ≤ i < |distancias| - 1 →L ((distancias[i] < distancias[i+1]) ∧ (distancias[i] > 0)))
    }
    aux distanciaAlCentro (th: ephh, centro:  $\mathbb{Z} \times \mathbb{Z}$ , i :  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =
    √((th[i][@HOGLATITUD] - centro[0])2 + (th[i][@HOGLONGITUD] - centro[1])2);

```

3.6. Problema 11

```

proc quitarIndividuos (th: ephh, ti: ephi, ∈ busqueda : seq((ItemIndividuo, dato)), result : (ephh, ephi)) {
    Pre {encuestaValida(th, ti) ∧ busquedaValida(busqueda) ∧ th = th0 ∧ ti = ti0}
    Post {guardarElementos(th0, ti0, busqueda, result) ∧ sacarElementos(th, ti, busqueda, th0, ti0)}
}

pred busquedaValida (busqueda: seq((ItemIndividuo, dato))) {
(∀i :  $\mathbb{Z}$ )(rango(busqueda, i) →L
(((busqueda[i]0 = INDCODUSU) →L (busqueda[i]1 > 0)) ∧
((busqueda[i]0 = INDAniO) →L (busqueda[i]1 ≤ 2021)) ∧
((busqueda[i]0 = INDTRIMESTRE) →L (1 ≤ busqueda[i]1 ≤ 4)) ∧
((busqueda[i]0 = COMPONENTE) →L (busqueda[i]1 > 0)) ∧
((busqueda[i]0 = CH4) →L (busqueda[i]1 = 1 ∨ busqueda[i]1 = 2)) ∧
((busqueda[i]0 = CH6) →L (busqueda[i]1 > 0)) ∧
((busqueda[i]0 = NIVELED) →L (busqueda[i]1 = 0 ∨ busqueda[i]1 = 1)) ∧
((busqueda[i]0 = ESTADO) →L (-1 ≤ busqueda[i]1 ≤ 1)) ∧
((busqueda[i]0 = CATOCUP) →L (0 ≤ busqueda[i]1 ≤ 4)) ∧
((busqueda[i]0 = p47T) →L (busqueda[i]1 ≥ -1)) ∧
((busqueda[i]0 = pp04G) →L (0 ≤ busqueda[i]1 ≤ 4))
∧ ¬(∃k :  $\mathbb{Z}$ )(i ≠ k ∧ busqueda[k]0 = busqueda[i]0))
}

pred cumpleConBusqueda (ti: ephi, busqueda: seq((ItemIndividuo, dato)), i:  $\mathbb{Z}$ ) {
(∀k :  $\mathbb{Z}$ )(rango(busqueda, k) →L ((busqueda[k]0 = INDCODUSU) →L (ti[i][@INDCODUSU] = busqueda[k]1) ∧
(busqueda[k]0 = INDAniO) →L (ti[i][@INDAniO] = busqueda[k]1) ∧
(busqueda[k]0 = INDTRIMESTRE) →L (ti[i][@INDTRIMESTRE] = busqueda[k]1) ∧
(busqueda[k]0 = COMPONENTE) →L (ti[i][@COMPONENTE] = busqueda[k]1) ∧
(busqueda[k]0 = CH4) →L (ti[i][@CH4] = busqueda[k]1) ∧
(busqueda[k]0 = CH6) →L (ti[i][@CH6] = busqueda[k]1) ∧
(busqueda[k]0 = NIVELED) →L (ti[i][@NIVELED] = busqueda[k]1) ∧
(busqueda[k]0 = CATOCUP) →L (ti[i][@CATOCUP] = busqueda[k]1) ∧
(busqueda[k]0 = p47T) →L (ti[i][@p47T] = busqueda[k]1) ∧
(busqueda[k]0 = pp04G) →L (ti[i][@PP04G] = busqueda[k]1))
}

pred guardarElementos (th0: ephh, ti0: ephi, busqueda: seq((ItemIndividuo, dato), result : (ephh, ephi))) {
(∀i :  $\mathbb{Z}$ )(rango(ti0, i) →L (cumpleConBusqueda(ti0, busqueda, i) → ((∃k :  $\mathbb{Z}$ )(∃h :  $\mathbb{Z}$ )(rango(result0, h) ∧L rango(result1, k) ∧L
result1[k] = ti0[i] ∧ result0[h] = th0[casaDelIndividuoI(th0, ti0, i)])))
}

pred sacarElementos (th: ephh, ti: ephi, busqueda: seq((ItemIndividuo, dato), th0 : ephh, ti0: ephi)) {
(∀i :  $\mathbb{Z}$ )(rango(ti0, i) (¬(∃k :  $\mathbb{Z}$ ))(¬(∃h :  $\mathbb{Z}$ ))((rango(ti, k) ∧ rango(th, h)) →L (cumpleConBusqueda(ti0, busqueda, i) →
(ti[k] = ti0[i] ∧ th[h] = th0[casaDelIndividuoI(th0, ti0, i)])))
}

```