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 4

Integrante	LU	Correo electrónico
Clemente Alier, Bárbara Micaela	181/21	bclemente@dc.uba.ar
Arratia Guillen, Valeria Lucia	467/21	varratia@dc.uba.ar
Genega, Yesica Giselle	1263/21	yesica.genega@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. EJERCICIO 1

proc esEncuestaVálida (in $th : eph_h$, in $ti : eph_i$ out $result$: Bool) {

Pre $\{True\}$

Post $\{result = true \leftrightarrow laEncuestaEsValida(th, ti)\}$

}

pred esMatriz (s: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$(\forall i : \mathbb{Z})(0 \leq i < filas(s) \longrightarrow_L |s[i]| > 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < filas(s) \longrightarrow_L |s[i]| = |s[j]|))$

}

pred secuenciasNoVacias ($th : eph_h$, $ti : eph_i$)) {

$|th| \neq 0 \wedge |ti| \neq 0$

}

pred columnasIgualAVariables ($th : eph_h$, $ti : eph_i$) {

$(\forall h, i : \mathbb{Z})(rango(th, h) \wedge rango(ti, i) \longrightarrow_L |th[h]| = 12 \wedge |ti[i]| = 11)$

Está bien, pero resultaría más claro separar esto en dos "para todos", uno para h y otro para i.

}

pred asociados ($th : eph_h$, $ti : eph_i$) {

$hogaresAsociadosIndividuos(th, ti) \wedge individuosAsociadosHogares(th, ti)$

}

pred hogaresAsociadosIndividuos (s : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, t : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$(\forall h : \mathbb{Z})(rango(s, h) \longrightarrow_L ((\exists i : \mathbb{Z})(rango(t, i) \wedge_L s[h][0] = t[i][0]))$

}

pred individuosAsociadosHogares (s:$seq\langle seq\langle \mathbb{Z}\rangle\rangle$ t:$seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$(\forall i : \mathbb{Z})(rango(t, i) \longrightarrow_L ((\exists h : \mathbb{Z})(rango(s, h) \wedge_L t[i][0] = s[h][0]))$

}

pred NoSeRepitenHogares ($th : eph_h$) {

$(\forall h1, h2 : \mathbb{Z})(rango(th, h1) \wedge rango(th, h2) \wedge h1 \neq h2 \longrightarrow_L th[h1][0] \neq th[h2][0])$

}

pred NoSeRepitenIndividuos ($ti : eph_i$) {

$(\forall i1, i2 : \mathbb{Z})(rango(ti, i1) \wedge rango(ti, i2) \wedge i1 \neq i2 \longrightarrow_L$
(if $ti[i1][0] = ti[h2][0]$ then $ti[i1][@componente] \neq ti[h2][@componente]$ else $ti[i1] \neq ti[h2]$ fi))

Recuerden que el IfThenElseFi es una función auxiliar que escoge entre dos valores de un tipo de dato.
En este caso, lo estan usando para elegir entre dos fórmulas, lo cual es incorrecto. Una posible solución sería: (B ^ P) v (¬B ^ Q); donde B sería la guarda del "if", y P y Q las fórmulas que se quiere elegir en cada caso.

}

pred mismoAñoTrimestre ($th : eph_h$, $ti : eph_i$) {

$(\forall j, i : \mathbb{Z})((rango(th, j) \wedge rango(ti, j)) \longrightarrow_L$
$((th[j][@hogaño] = ti[i][@indaño]) \wedge (th[j][@hogtrimestre] = ti[i][@indtrimestre]))$

}

pred miembrosMenorIgualA20 ($ti : eph_i$) {

$(\forall i : \mathbb{Z})(rango(ti, i) \longrightarrow_L t[i][@componente] \leq 20)$

}

pred masHabitacionesQueDormitorios (s : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$(\forall h : \mathbb{Z})(rango(s, h) \longrightarrow_L s[h][@iv2] \geq s[h][@ii2])$

}

pred atributosEnRangoHogar ($th : eph_h$) {

$(\forall h : \mathbb{Z})(rango(th, h) \longrightarrow_L (1 \leq th[h][@hogcodusu]) \wedge (0 \leq th[h][@hogtrimestre] \leq 4) \wedge (1 \leq th[h][@ii7] \leq 3) \wedge (1 \leq th[h][@region] \leq 6) \wedge$
$(0 \leq th[h][@mas500] \leq 1) \wedge (1 \leq th[h][@iv1] \leq 5) \wedge (0 \leq th[h][@iv2]) \wedge (0 \leq th[h][@ii2]) \wedge (1 \leq th[h][@ii3] \leq 2))$

}

pred atributosEnRangoIndividuos ($ti : eph_i$) {

Si separan los chequeos sobres los atributos tal que quede uno por renglón, mejoraría significativamente la legibilidad de estos predicados. Se los comento únicamente acá, pero hay otros predicados que se podrían ver favorecidos si ponen "Enters" en algunos lados.

$(\forall i : \mathbb{Z})(rango(ti, i) \longrightarrow_L ((1 \leq ti[i][@indcodusu]) \wedge (1 \leq ti[i][@componente]) \wedge (0 \leq ti[i][@indtrimestre] \leq 4) \wedge$
$(1 \leq ti[i][@ch4] \leq 2) \wedge (1 \leq ti[i][@ch6]) \wedge (0 \leq ti[i][@niveled] \leq 1) \wedge (-1 \leq ti[i][@estado] \leq 1) \wedge$

$(0 \leq ti[i][@catocup] \leq 4) \land (-1 \leq ti[i][@p47t]) \land (1 \leq ti[i][@pp04g)] \leq 10))$

}

`pred atributosEnRango` $(th : eph_h,\ ti : eph_i)$ {

$atributosEnRangoHogar(th) \land atributosEnRangoIndividuos(ti)$

}

1.2. EJERCICIO 2

`proc histHabitacional` (in $th : eph_h$, in $ti : eph_i$, in $region : \mathbb{Z}$, out $res : seq\langle\mathbb{Z}\rangle$) {

`Pre` $\{laEncuestaEsVa1lida(th, ti)\}$

Poco declarativo; falta modularizar.

`Post` $\{(\exists imax : dato)(maximaCantDeHabitaciones(th, imax, r) \land imax + 1 = |res|) \land (\forall i : dato)(0 \leq i < |res| \longrightarrow_L res[i] = cantCasasConiHabitacionesEnRegion(th, i, region)))\}$

}

`aux cantCasasConiHabitacionesEnRegion` $(th : eph_h, i : dato, r : dato) : \mathbb{Z} = \sum_{h \in th}$
if $h[@region] = r \land h[@iv1] = 1 \land h[@iv2] = i$ then 1 else 0 fi ;

`pred maximaCantDeHabitaciones` $(th : eph_h, imax : dato, r : dato)$ {

$perteneceALaTabla(th, imax) \land_L (\forall$ h : hogar$)(h \in th \land h[@region] = r \land h[@iv1] = 1 \longrightarrow_L (\forall$d: dato$)(d = h[@iv2] \longrightarrow_L d \leq imax))$

Este "para todo" no hace falta; agrega ruido.
Pueden decir directamente que h[@iv2] <= imax.

}

`pred perteneceALaTabla` $(th : eph_h, imax : dato)$ {

$(\exists$ h : hogar$)(h \in th \land i = h[@iv2])$

¿Quién es i?

}

1.3. EJERCICIO 3

`proc laCasaEstaQuedandoChica` (in $th : eph_h$, in $ti : eph_i$, out $res : seq\langle\mathbb{R}\rangle$) {

`Pre` $\{laEncuestaEsValida(th, ti)\}$

`Post` $\{|res| = 6 \land (\forall i : \mathbb{Z})(0 \leq i < |res| - 1 \longrightarrow_L proporcionDeHogaresEnCadaRegion(th, ti, res, i)\}$

}

`pred proporcionDeHogaresEnCadaRegion` $(th : eph_h,\ ti : eph_i,\ res : seq\langle\mathbb{R}\rangle,$ i : $\mathbb{Z})$ {

$(cantHogaresConCondicionesEnRegion(th, i + 1) = 0 \land res[i] = 0) \lor$
$(cantHogaresConCondicionesEnRegion(th, i + 1) \neq 0 \land_L res[i] =$
$(cantHacinamientosCriticosEnLaRegion(th, ti, i + 1) \div cantHogaresConCondicionesEnRegion(th, i + 1)))$

}

`aux cantHacinamientosCriticosEnLaRegion` $(th : eph_h, ti : eph_i, r : dato) : \mathbb{Z} = \sum_{h \in th}$
if $esHacinamientoCritico(ti, h, r)$ then 1 else 0 fi ;

`aux cantHogaresConCondicionesEnRegion` $(th : eph_h, r : dato) : \mathbb{Z} = \sum_{h \in th}$ if $cumpleCondiciones(h, r)$ then 1 else 0 fi ;

`pred esHacinamientoCritico` $(ti : eph_i, h : hogar, r : dato)$ {

$cumpleCondiciones(h, r) \land (masTresPersonasPorCuarto(ti, h) \lor h[@iv2] = 0)$

El nombre de este predicado es poco declarativo. Podrías aplicar a un sinfín de situaciones.

} `pred cumpleCondiciones` $(h : hogar, r : dato)$ {

$h[@iv1] = 1 \land h[@mas500] = 0 \land h[@region] = r$

}

pred masTresPersonasPorCuarto $(ti : eph_i, h : hogar)$ {

$h[@iv2] > 0 \longrightarrow_L (cantidadIndividuosEnHogar(ti, h) \div h[@iv2]) > 3$

}

aux cantidadIndividuosEnHogar $(ti : eph_i, h : hogar) : \mathbb{Z} = \sum_{i \in ti}$ if $i[@indcodusu] = h[@hogcodusu]$ then 1 else 0 fi ;

1.4. EJERCICIO 4

proc creceElTeleworkingEnCiudadesGrandes (in $t1h : eph_h$, in $t1i : eph_i$, in $t2h : eph_h$, in $t2i : eph_i$, out res : Bool) {

Pre $\{laEncuestaEsValida(t1h, t1i) \land laEncuestaEsValida(t2h, t2i) \land añosDistintosTrimIguales(t1h, t1i, t2h, t2i)\}$

Post $\{res = true \iff (proporcionDeTeleworking(t1h, t1i) < proporcionDeTeleworking(t2h, t2i)\}$

}

pred añosDistintosTrimestresIguales $(t1h : eph_h, t2h : eph_h)$ {

$t1h[0][@hogaño] < t2h[0][@hogaño] \land t1h[0][@hogtrimestre] = t2h[0][@hogtrimestre]$

}

aux proporcionDeTeleworking $(th : eph_h, ti : eph_i) : \mathbb{Z} =$ if $indTotalesEnCondiciones(th, ti) \neq 0$ then $indQueTrabajanEnSuHogar(th, ti) \div indTotalesEnCondiciones(th, ti)$ else 0 fi ;

aux indQueTrabajanEnSuHogar $(th : eph_h, ti : eph_i) : \mathbb{Z} = \sum_{h \in th}$(if $hogaresEnCondicionesYTeleworking(h)$ then $\sum_{i \in ti}$ (if $i[0] = h[0] \land i[@pp04g] = 6$ then 1 else 0 fi) else 0 fi ;

aux indTotalesEnCondiciones $(th : eph_h, ti : eph_i) : \mathbb{Z} = \sum_{h \in th}$(if $hogarCumpleCondiciones(h)$ then $\sum_{i \in ti}$ (if $i[0] = h[0] \land i[@estado] = 1$ then 1 else 0 fi) else 0 fi ;

pred hogaresEnCondicionesYTeleworking $(h : hogar)$ {

$hogarCumpleCondiciones(h) \land h[@ii3] = 1$

}

Nombre poco declarativo.

pred hogarCumpleCondiciones $(h : hogar)$ {

$h[@mas500] = 1 \land_L (h[@iv1] = 1 \lor h[@iv1] = 2)$

}

1.5. EJERCICIO 5

proc costoSubsidioMejora ((in $th : eph_h$, in $ti : eph_i$, in monto : $\mathbb{Z}$, out $res : \mathbb{Z}$)) {

Pre $\{laEncuestaEsValida(th, ti) \land monto \geq 0\}$

monto > 0

Post $\{res = cantDeHogaresEnCondiciones(th, ti) \times monto\}$

}

aux cantDeHogaresEnCondiciones $(th : eph_h,\ ti : eph_i) : \mathbb{Z} = \sum_{h \in th}$
if $(habitacionesEnCondicionesEnHogar(h, ti) \wedge esCasaDeTenenciaPropia(h, ti))$ then 1 else 0 fi ;

pred habitacionesEnCondicionesEnHogar $(h : hogar, ti : eph_i)$ {

$h[@ii2] < (cantidadIndividuosEnHogar(h, ti) - 2)$

}

aux cantidadIndividuosEnHogar $(h : hogar, ti : eph_i) : \mathbb{Z} = \sum_{i \in ti}$ if $i[@indcodusu] = h[@hogcodusu]$ then 1 else 0 fi ;

pred esCasaDeTenenciaPropia $(h : hogar)$ {

$h[@iv1] = 1 \wedge h[@ii7] = 1)$

}

1.6. EJERCICIO 6

proc generarJoin (in $th : eph_h$, in $ti : eph_i$, out junta: joinHI) {

Pre $\{laEncuestaEsValida(th, ti)\}$

Post $\{incluyeHogarEIndividuoEnJunta(th, ti, junta) \wedge_L valoresCondusuIguales(junta)\}$

}

pred incluyeHogarEIndividuoEnJunta $(th : eph_h, ti : eph_i, junta : joinHI)$ {

$(\forall h, i : seq\langle\mathbb{Z}\rangle)(h \in th \wedge i \in ti \iff (\exists hxi : hogar \times individuo)(hxi \in junta \wedge_L (hxi_0 = h \wedge hxi_1 = i)))$

}

Estan diciendo que el join debe tener todas las combinaciones hogar-individuo en th y ti, sin importar el CODUSU.

pred valoresCodusuIguales $(junta : joinHI)$ {

$(\forall hxi : hogar \times individuo)(hxi \in junta \longrightarrow_L hxi_0[@hogcodusu] = hxi_1[@indcodusu])$

}

1.7. EJERCICIO 7

proc ordenarRegionYTipo (inout $th : eph_h$, inout $ti : eph_i$) {

Pre $\{laEncuestaEsValida(th, ti)\}$

Post $\{ordenPorRegionyHogcodusu(th) \wedge ordenPorIndCodusuyComponente(th, ti)\}$

}

Falta pedir que th y ti sean permutaciones de sus valores originales.

pred ordenPorRegionyHogcodusu $(th : eph_h)$ {

$ordenPorRegion(th) \wedge_L ordenPorHogcodusuEnRegion(th)$

}

pred ordenPorRegion $(th : eph_h)$ {

$(\forall h : \mathbb{Z})(0 \leq h < |th| - 1 \longrightarrow_L th[h][@region] \leq th[h + 1][@region])$

}

Este predicado resulta un poco confuso. ¿Por qué necesitan a h2? Bastaría con hablar de h1 y h1+1. Quedaría algo como:

"Para todo h en rango talque h y h+1 estan en la misma region, entonces el HOGCODUSU de h debe ser menor al de h+1."

pred ordenPorHogCodusuEnRegion $(th : eph_h)$ {

$(\forall h : \mathbb{Z})(0 \leq h1, h2 < |th| \wedge h1 < h2 \longrightarrow_L (th[h1][@region] = th[h1+1][@region] \wedge th[h1][@hogcodusu] < th[h1+1][@hogcodusu]) \vee (th[h1][@region] < th[h1+1][@region] \wedge th[h1+1][@region] = th[h2][@region] \wedge th[h1+1][@hogcodusu] \leq th[h2][@hogcodusu])$

}

pred ordenPorIndCodusuyComponente $(th : eph_h, ti : eph_i)$ {

$ordenPorIndCodusu(th, ti) \wedge ordenPorComponenteEnHogar(ti))$

}

pred ordenPorIndCodusu $(th : eph_h, ti : eph_i)$ {

$(\forall h1, h2, i1, i2 : \mathbb{Z})(rango(th, h1) \wedge rango(th, h2) \wedge rango(ti, i1) \wedge rango(ti, i1) \wedge_L (th[h1][@hogcodusu] = ti[i1][@indcodusu] \wedge th[h2][@hogcodusu] = ti[i2][@indcodusu]) \wedge h1 < h2 \longrightarrow_L i1 < i2)$

}

pred ordenPorComponenteEnHogar $(ti : eph_i)$ {

$(\forall i : \mathbb{Z})(0 \leq i < |ti| - 1 \longrightarrow_L ((ti[i][@indcodusu] = ti[i+1][@indcodusu] \longrightarrow ti[i][@componente] < ti[i+1][@componente]) \wedge ((ti[i][@indcodusu] \neq ti[i+1][@indcodusu]) \longrightarrow ti[i][@componente] \geq ti[i+1][@componente]))$

}

Incorrecto. El "para todo" debería decir algo como:

"para todo i en rango tal que i e i+1 tiene el mismo INDCODUSU, entonces el componente de i debe ser menor al de i+1,"

1.8. EJERCICIO 8

Falta pedir:
- que todos los hogares de la muestra estén en th,
- que no haya otra muestra homogenea más grande que res.

Les recomiendo que armen un predicado tipo "esMuestraHomogenea".

proc muestraHomogenea (in $th : eph_h$, in $ti : eph_i$, out $res : seq\langle hogar \rangle$) {

Pre $\{laEncuestaEsValida(th, ti)\}$

¿Qué indefinición están salvando con este " y luego"?

Post $\{(|res| \geq 3 \wedge_L (ordendeingresoscreciente(th, ti, res) \wedge diferenciaDeMontosIguales(th, ti, res))) \vee (\neg diferenciaDeMontosIguales(ti, th, res) \wedge |res| = 0))\}$

El resultado debería ser una secuencia vacía únicamente cuando no existe una muestra con al menos 3 elementos.

}

pred ordenDeIngresosCreciente $(th : eph_h, ti : eph_i, res : seq\langle hogar \rangle)$ {

$(\forall r : \mathbb{Z})(0 \leq r < |res| - 1 \longrightarrow_L ingresoTotalDeHogarH(ti, th, r) \leq ingresoTotalDeHogarH(ti, th, r+1)$

}

pred diferenciaDeMontosIguales $(ti : eph_i, th : eph_h, res : seq\langle hogar \rangle)$ {

$(\forall r : \mathbb{Z})(0 \leq r < |res| - 2 \longrightarrow_L (ingresoTotalDelHogarH(ti, th, r+1) - ingresoTotalDelHogarH(ti, th, r)) = (ingresoTotalDelHogarH(ti, th, r+2) - ingresoTotalDelHogar(ti, th, r+1)$

}

aux ingresoTotalDelHogarH $(ti : eph_i, th : eph_h, h : \mathbb{Z}) : \mathbb{Z} = \sum_{i \in ti}$
if $i[@indcodusu] = th[h][@hogcodusu] \wedge i[@p47t] > -1$ then $i[@p47t]$ else 0 fi ;

1.9. EJERCICIO 9

proc corregirRegion (inout $th : eph_h$, in $ti : eph_i$) {

Pre $\{laEncuestaEsValida(th, ti) \land th = TH_0\}$

Post $\{|th| = |TH_0| \land_L (\forall h : \mathbb{Z})(rango(TH_0, h) \longrightarrow_L bsasAPampeana(TH_0, th, h))\}$

}

pred bsasAPampeana $(TH_0 : eph_h, th : eph_h, h : hogar)$ {

$(TH_0[h][@region] = 1 \land th[h] = setAt(TH_0[h], @region, 5)) \lor (TH_0[h][@region] \neq 1 \land th[h] = TH_0[h])$

}

1.10. EJERCICIO 10

proc histogramaDeAnillosConcentricos (in $th : eph_h$, in centro : $\mathbb{Z}$x $\mathbb{Z}$, in distancias : $seq\langle\mathbb{Z}\rangle$, out result : $seq\langle\mathbb{Z}\rangle$) {

Pre $\{|distancias| \neq 0 \land_L listaDeDistCrecienteYNoNula(distancias)\}$ ← Falta pedir que th sea una tabla de hogares válida.

Post $\{|distancias| = |result| \land_L result[0] = cantHogaresEnPrimerAnillo(th, centro, distancia[0]) \land_L$
$(\forall i : \mathbb{Z})(0 < i < |result| \longrightarrow_L result[i] = cantHogaresEnAnillo(th, centro, distancias[i-1], distancias[i]))\}$

}

pred listaDistCrecienteYNoNula (distancias : $seq\langle\mathbb{Z}\rangle$) {

$(\forall i : \mathbb{Z})(0 \leq i < |distancias| - 1 \longrightarrow_L distancias[i] \neq 0 \land distancias[i] < distancias[i+1]$

}

aux cantHogaresEnPrimerAnillo $(th : eph_h, centro : \mathbb{Z}$x $\mathbb{Z}, d0 : \mathbb{Z}) : \mathbb{Z} = \sum_{h \in th}$
if $hogarEnPrimerAnillo(h, centro, d0)$ then 1 else 0 fi;

pred hogarEnPrimerAnillo $(h : hogar, centro : \mathbb{Z}$x $\mathbb{Z}, d0\ \mathbb{Z})$ {

$0 \leq distanciaEuclidianaAlCentro(h, centro) \leq d0$

}

aux cantHogaresEnAnilloDeD1AD2 $(th : eph_h, centro : \mathbb{Z}$x $\mathbb{Z}, d1 : \mathbb{Z}, d2 : \mathbb{Z}) : \mathbb{Z} = \sum_{h \in th}$
if $hogarEnAnillo(h, centro, d1, d2)$ then 1 else 0 fi;

pred hogarEnAnillo $(h : hogar, centro : \mathbb{Z}$x $\mathbb{Z}, d1 : \mathbb{Z}, d2\ \mathbb{Z})$ {

$d1 < distanciaEuclidianaAlCentro(h, centro) \leq d2])$

}

aux distanciaEuclidianaAlCentro $(h : hogar, centro : \mathbb{Z}$x $\mathbb{Z}) : \mathbb{R} =$
$\sqrt{(h[@hoglatitud] - centro_0)^2 + (h[@hoglongitud] - centro_1)^2}$;

1.11. EJERCICIO 11

proc quitarIndividuos (inout $th : eph_h$, inout $ti : eph_i$, in busqueda : $seq\langle ItemIndividuo, dato\rangle$, out result : (eph_h, eph_i))
{

Pre $\{laEncuestaEsValida(th, ti) \land laBusquedaEsValida(busqueda, ti) \land th = TH_0 \land ti = TI_0\}$

Post $\{laEncuestaEsValida(th, ti) \land laEncuestaEsValida(result_0, result_1) \land$
$(|TI_0| = |ti| + |result_1| \land_L (\forall i : individuo)(i \in |TI_0| \longrightarrow_L moverIndividuosAResult(ti, result, busqueda, i))) \land$

$(|TH_0| = |th| + |result_0| \wedge_L (\forall h : hogar)(h \in |TH_0| \longrightarrow_L moverHogaresAResult(th, result, h)))\}$

}

pred laBusquedaEsValida ($busqueda$:$seq\langle ItemIndividuo$,ti : eph$_i$ dato$\rangle$) {

$itemNoSeRepite(busqueda) \wedge datosEnRangos(busqueda) \wedge (\forall i : individuo)(i \in ti \longrightarrow_L incluyeValorDeTI(i, busqueda))$

Esto último no se menciona en el enunciado y por lo tanto, no deberían pedirlo.

}

pred itemNoSeRepite ($busqueda : seq\langle ItemIndividuo, dato\rangle$) {

$(\forall i, j : \mathbb{Z})(0 \leq i, j < |busqueda| \wedge i \neq j \longrightarrow_L busqueda[i]_0 \neq busqueda[j]_0)$

}

En vez de indexar ItemIndividuo, es preferible que mencionen directamente las constantes.
Por ejemplo:
en vez de ItemIndividuo(0),
poner INDCODUSU.

pred datosEnRangos ($busqueda : seq\langle ItemIndividuo, dato\rangle$) {

$(\forall b : \mathbb{Z})(0 \leq b < |busqueda| \longrightarrow_L ((busqueda[b]_0 = ItemIndividuo(0) \wedge busqueda[b]_1 \geq 1) \vee (busqueda[b]_0 = ItemIndividuo(1) \wedge busqueda[b]_1 \geq 1) \vee (busqueda[b]_0 = ItemIndividuo(3) \wedge 1 \leq busqueda[b]_1 \leq 4) \vee (busqueda[b]_0 = ItemIndividuo(4) \wedge 1 \leq busqueda[b]_1 \leq 2) \vee (busqueda[b]_0 = ItemIndividuo(5) \wedge busqueda[b]_1 \geq 1) \vee (busqueda[b]_0 = ItemIndividuo(6) \wedge 0 \leq busqueda[b]_1 \leq 1) \vee (busqueda[b]_0 = ItemIndividuo(7) \wedge -1 \leq busqueda[b]_1 \leq 1) \vee (busqueda[b]_0 = ItemIndividuo(8) \wedge 0 \leq busqueda[b]_1 \leq 4) \vee (busqueda[b]_0 = ItemIndividuo(9) \wedge busqueda[b]_1 \geq -1) \vee (busqueda[i]_0 = ItemIndividuo(10) \wedge 1 \leq busqueda[b]_1 \leq 10))$

}

Este para todo está pidiendo algo distinto a lo que supongo que realmente querían decir.
Dice algo así como: "Para todo individuo en result, únicamente su hogar debería estar en result". Esto nunca va se va a cumplir si hay dos o más individuos en result que viven en casas distintas. Cada uno va a pedir que únicamente su hogar esté en result, así excluyendo los hogares de los demás.

Más arriba dije "lo que supongo que realmente querían decir", porque supuse que su intención era decir algo como:

"Si hay un individuo que habita h y se encuentra en result, entonces h también debe estar en result y no en th; si no, h debería estar en th y no en result."

Fijense si les sale definir este predicado de manera similar a moverIndividuosAResult.

pred incluyeValorDeTI ($i : individuo, busqueda : seq\langle ItemIndividuo, dato\rangle$) {

$(\forall valor : dato)(valor \in individuo \iff (\exists ixd : itemIndividuo \times dato)(ixd \in busq$

}

pred moverHogaresAResult ($th : eph_h$, result : (eph_h, eph_i), $h : hogar$) {

$(\forall i : individuo)(i \in result_1 \longrightarrow_L ((i[@indcodusu] = h[@hogcodusu] \wedge h \notin th \wedge h \in result_0) \vee (i[@indcodusu] \neq h[@hogcodusu] \wedge h \in th \wedge h \notin result_0))$

Recuerden que ustedes no estan moviendo los individuos/hogares, si no que estan verificando que hayan sido movidos correctamente.

}

pred moverIndividuosAResult ($ti : eph_i$, result : (eph_h, eph_i), busqueda : $seq\langle ItemIndividuo, dato\rangle$, i:individuo) {

$(cantCondicionesQueCumpleIndividuo(busqueda, i) = |busqueda| \wedge i \in result_1 \wedge i \notin ti) \vee$
$(cantCondicionesQueCumpleIndividuo(busqueda, i) \neq |busqueda| \wedge i \notin result_1 \wedge i \in ti))$

Armen un predicado auxiliar para esta igualdad.

}

aux cantCondicionesQueCumpleIndividuo (busqueda : $seq\langle ItemIndividuo, dato\rangle$, i : individuo) : $\mathbb{Z} = \sum_{ixd=0}^{|busqueda|-1}$(if $i[ord(busqueda[ixd]_0)] = busqueda[ixd]_1$ then 1 else 0 fi) ;

2. Predicados y Auxiliares generales

aux filas (s: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) : $\mathbb{Z} = |s|$;

pred rango (s: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, i:$\mathbb{Z}$) {

$0 \leq i < filas(s)$

}

pred laEncuestaEsValida ($th : eph_h$, $ti : eph_i$) {

$esMatriz(th) \wedge esMatriz(ti) \wedge secuenciasNoVacias(th, ti) \wedge columnasIgualAVariables(th, ti) \wedge$
$asociados(th, ti) \wedge NoSeRepitenHogares(th) \wedge NoSeRepitenIndividuos(ti) \wedge$
$mismoAñoTrimestre(th, ti) \wedge miembrosMenorIgualA20(th) \wedge masHabitacionesQueDormitorios(th)$
$\wedge atributosEnRango(th, ti)$

}

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 @mas500 : $\mathbb{Z} = ord(MAS500)$;

aux @iv1 : $\mathbb{Z} = ord[IV1]$;

aux @iv2 : $\mathbb{Z} = ord[IV2]$;

aux @ii2 : $\mathbb{Z} = ord(II2)$;

aux @ii3 : $\mathbb{Z} = ord(II3)$;

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 @niveled : $\mathbb{Z} = ord(NIVELED)$;

aux @estado : $\mathbb{Z} = ord(ESTADO)$;

aux @catocup : $\mathbb{Z} = ord(CATOCUP)$;

aux @p47t : $\mathbb{Z} = ord(P47T)$;

aux @pp04g : $\mathbb{Z} = ord(PP04G)$;

aux filas (s: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) : $\mathbb{Z} = |s|$;

2.0.1. Enumerados

enum ItemIndividuo{
INDCODUSU, COMPONENTE, INDAÑO, INDTRIMESTRE, CH4, CH6, NIVELED, ESTADO, CATOCUP, P47T, P04G
}

enum ItemHogar{

HOGCODUSU, HOGAÑO, HOGTRIMESTRE, HOGLATITUD, HOGLONGITUD, II7, REGION, MAS500, IV1, IV2, II2, II3
}

2.1. TIPOS

type $dato = \mathbb{Z}$

type $individuo = seq\langle dato \rangle$

type $hogar = seq\langle dato \rangle$

type $eph_i = seq\langle individuo \rangle$

type $eph_h = seq\langle hogar \rangle$

3. Decisiones tomadas

Tomamos la decisión de asumir que en todo ejercicio donde se presente una proporción, ya sea sobre cantidad de hogares o individuos, el denominador no se anule, pues trabajamos con el caso en que hay al menos un individuo/hogar que cumple las condiciones pedidas.