DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

26 de abril de 2019 Algoritmos y Estructuras de Datos I

Grupo: El Matriarcado

Integrante	LU	Correo electrónico
Arévalo, Carla	307/14	carlii95@gmail.com
Gurruchaga, Sofía	173/18	sofigurru@gmail.com
Juarez, Nazareth		madenajr@gmail.com
González, Sheila	801/18	gonzsheilam@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

Ejercicio 1

proc `enTerritorio` (in v: *Viaje*,in r: *Dist*, out res: Bool) {
    `Pre` { $esViajeValido(v) \ \wedge \ r > 0$ }
    `Post` { $res = True \leftrightarrow (\exists \ centro : GPS)((\forall i : \mathbb{Z})(\ 0 \leq i < |v| \ \longrightarrow_L \ dist(centro, v[i]_1)$
    $\leq r))$ }
}

Ejercicio 2

proc `excesoDeVelocidad` (in v: *Viaje*, out res: Bool) {
    `Pre` { $esViajeValido(v)$ }
    `Post` { $res = True \leftrightarrow huboExcesoDeVelocidad(v)$ }
}

pred $huboExcesoDeVelocidad(v : Viaje)\{ \ (\exists i : \mathbb{Z})(\exists j : \mathbb{Z})((\ 0 \leq i < |v| \ \wedge_L \ 0 \leq j < |v|) \ \wedge \ sonParadasSiguientes(v, i, j)) \ \wedge \ velocidad(v, i, j) > 80)\}$

aux `velocidad` (v: *Viaje*, i: $\mathbb{Z}$, j:$\mathbb{Z}$) : $\mathbb{Z}$ = $dist(v[i]_1, v[j]_1)/(v[j]_0 - v[i]_0)$;

Ejercicio 3

proc `tiempoTotal` (in v: *Viaje*, out t: *Tiempo*) {
    `Pre` { $esViajeValido(v)$ }
    `Post` { $t = ultimaParada(v) - primeraParada(v)$ }
}

aux `ultimaParada` (v: *Viaje*) : *Tiempo* = $\sum_{i=0}^{|v|-1}$ if $(esElMaximoTiempo(v, i))$ then $v[i]_0$ else 0 fi ;

pred $esElMaximoTiempo(v : Viaje, i : \mathbb{Z})\{ \ \neg(\exists j : \mathbb{Z})(\ 0 \leq j < |v| \wedge_L v[j]_0 \ > v[i]_0) \ \}$

aux `primeraParada` (v: *Viaje*) : *Tiempo* = $\sum_{i=0}^{|v|-1}$ if $(esElMinimoTiempo(v, i))$ then $v[i]_0$ else 0 fi;

pred $esElMinimoTiempo(v : Viaje, i : \mathbb{Z})\{ \ \neg(\exists j : \mathbb{Z})(\ 0 \leq j < |v| \wedge_L v[j]_0 \ < v[i]_0) \ \}$

Ejercicio 4

proc `distanciaTotal` (in v: *Viaje*, out d: *Dist*) {
    `Pre` { $esViajeValido(v)$ }
    `Post` { $d = distanciaRecorrida(v)$ }
}

aux distanciaRecorrida (v: *Viaje*) : *Dist* $= \sum_{i=0}^{|v|-1}$ if $(existeParadaSiguiente(v, i))$ $then$ $dist(v[i]_1,$ gPSParadaSiguiente(v,i)) else 0 fi ;

pred $existeParadaSiguiente(v : Viaje, i : \mathbb{Z})\{$ $(\exists j : \mathbb{Z})((\ 0 \leq i < |v|\ \wedge_L\ v[i]_0 < \mathrm{v[j]}_0)\ \wedge \neg(\exists k : \mathbb{Z})(\ 0 \leq k < |v|\ \wedge_L\ v[i]_0\ < \mathrm{v[k]}_0\ < \mathrm{v[j]}_0))$ $\}$

aux gPSParadaSiguiente (v: *Viaje*,i: $\mathbb{Z}$) : *GPS* $= \sum_{k=0}^{|v|-1}$ if $(sonParadasSiguiente(v, i, k))$ $then$ $v[k]_1$ else 0 fi ;

Ejercicio 5

proc flota (in v: $seq\langle Viaje\rangle$, in t_0: *Tiempo*, in t_f: *Tiempo*, out res: $\mathbb{Z}$) {
Pre { $(\forall i : \mathbb{Z})(\ 0 \leq i < |v| \longrightarrow_L\ esViajeValido(v[i])\ \wedge\ 0\ \leq \mathrm{t}_0\ < t_f\ )$ }
Post { $res =\ \sum_{j=0}^{|v|-1}$ if $(viajoEnLaFranja(v, j))$ then 1 else 0 fi }
}

pred $viajoEnLaFranja(v : seq\langle Viaje\rangle, j : \mathbb{Z})\{$ $(\exists i : \mathbb{Z})(\ 0 \leq i < |v[j]| \wedge_L \mathrm{t}_0\ \leq\ v[j[i]_0]\ \leq \mathrm{t}_f)$
}

Ejercicio 6

proc recorridoCubierto (in v: *Viaje*,in r: *Recorrido*,in u: *Dist*, out res: $seq\langle GPS\rangle$) {
Pre { $esViajeValido(v)$ }
Post { $(\forall i : \mathbb{Z})((\ 0 \leq i < |r|\ \wedge_L\ \neg(\ puntoCubierto(r, i, v, u))\ \longrightarrow_L\ r[i] \in res\ )$}
}

pred $puntoCubierto(r : Recorrido, i : \mathbb{Z}, v : Viaje, u : Dist)\{$ $(\exists j : \mathbb{Z})(\ 0 \leq j < |v|\ \wedge_L\ dist(v[j]_1,$ r[i]) $<$ u }

Ejercicio 7

proc construirGrilla (in esq1: *GPS*, in esq2: *GPS*, in n: $\mathbb{Z}$, in m: $\mathbb{Z}$, out g: *Grilla*) {
Pre { $coordenadasValidas(esq1, esq2) \wedge (n > 0) \wedge (m > 0) \wedge conseguirLadoLatitudinal$ $(esq1, esq2, n) = conseguirLadoLongitudinal(esq1, esq2, m)$ }
Post { $grillaOK(g, esq1, esq2, n, m)$ }
}

pred $coordenadasValidas(esq1 : GPS,\ esq2 : GPS)\{$ $-90 \leq esq2_0\ < \mathrm{esq1}_0 \leq 90\ \wedge\ -180 \leq esq1_1\ < \mathrm{esq2}_1 \leq 180$ }

aux conseguirLadoLatitudinal (esq1: *GPS*, esq2: *GPS*, n: $\mathbb{Z}$) : $\mathbb{R}$ $=$ $(esq1_0$ - $\mathrm{esq2}_0)$/n ;

aux conseguirLadoLongitudinal (esq1: *GPS*, esq2: *GPS*, m: $\mathbb{Z}$) : $\mathbb{R}$ $=$ $(esq2_1$ - $\mathrm{esq1}_1)$/m ;

pred $cantidadCeldasOK(g : Grilla, esq1 : GPS, esq2 : GPS)\{$ $|g| = n * m$ }

pred $gpsOK(g : Grilla,\ esq1 : GPS,\ esq2 : GPS,\ n : \mathbb{Z},\ m : \mathbb{Z},\ indLat : \mathbb{Z},\ indLong : \mathbb{Z},\ i : \mathbb{Z})$\{ $(g[i]_{0_0}$ = esq1_0 - conseguirLadoLatitudinal(esq1, esq2, n) * indLat) $\land$ ($g[i]_{0_1}$ = esq1_1 + conseguirLadoLongitudinal(esq1, esq2, m) * indLong) $\land$ ($g[i]_{1_0}$ = esq2_0 + conseguirLadoLatitudinal(esq1, esq2, n) * indLat) $\land$ ($g[i]_{1_1}$ = esq2_1 - conseguirLadoLongitudinal(esq1, esq2, m) * indLong) \}

pred $nombreOK(g : Grilla,\ esq1 : GPS,\ esq2 : GPS,\ indLat : \mathbb{Z},\ indLong : \mathbb{Z}, i : \mathbb{Z})$\{ ($g[i]_{2_0}$ = indLat + 1) $\land$ ($g[i]_{2_1}$ = indLong + 1) \}

Ejercicio 8

proc aPalabra (in trayecto: $seq\langle GPS\rangle$, in g: $Grilla$, out res: $seq\langle Nombre\rangle$) {
Pre { $trayectoValido(trayecto) \land grillaOK(g) \land trayectoEnGrilla(trayecto, g)$ }
Post { $|res| = |trayecto| \land (\forall i : \mathbb{Z})((\ 0 \leq i < |trayecto| \land (\exists j : \mathbb{Z})(0 \leq j < |g|) \land_L$
$puntoDeTrayectoEnAlgunaCeldaDeGrilla(trayecto, g, i, j)) \longrightarrow_L res[i] = g[j]_2)$ }
}

pred $trayectoValido(trayecto : seq\langle GPS\rangle)$\{ $(\forall i : \mathbb{Z})(\ 0 \leq i < |trayecto| \longrightarrow_L (-90 \leq trayecto[i]_0 \leq 90 \land -180 \leq trayecto[i]_1 \leq 180))$ \}

pred $trayectoEnGrilla(trayecto : seq\langle GPS\rangle,\ g : Grilla)$\{ $(\forall i : \mathbb{Z})(\ 0 \leq i < |trayecto| \longrightarrow_L (\exists j : \mathbb{Z})(\ 0 \leq j < |g| \land_L puntoDeTrayectoEnAlgunaCeldaDeLaGrilla(trayecto, g, i, j))$ \}

pred $puntoDeTrayectoEnAlgunaCeldaDeLaGrilla(trayecto : seq\langle GPS\rangle,\ g : Grilla,\ i : \mathbb{Z},\ j : \mathbb{Z})$\{ $(g[j]_{1_0} \leq trayecto[i]_0 \leq g[j]_{0_0} \land g[j]_{0_1} \leq trayecto[i]_1 \leq g[j]_{1_1})$ \}

Ejercicio 9

proc cantidadDeSaltos (in g: $Grilla$, in v: $Viaje$, out res: $\mathbb{Z}$) {
Pre { $grillaOK(g) \land esViajeValido(v)$ }
Post { $res = numeroDeSaltos(g, v)$ }
}

aux numeroDeSaltos (g: $Grilla$, v: $Viaje$) : $\mathbb{Z}$ = $\sum_{i=0}^{|v|-1}$ if $(hayUnSalto(v, g, i))$ then 1 else 0 fi ;

pred $hayUnSalto(v : Viaje,\ g : Grilla,\ i : \mathbb{Z})$\{ $(\exists j : \mathbb{Z})(\ 0 \leq j < |v| \land_L sonParadasSiguientes(v, i, j) \land \neg(\ losPuntosEstanComoMuchoAUnaCeldaDeDistancia(v, g, i, j)))$ \}

pred $losPuntosEstanComoMuchoAUnaCeldaDeDistancia(v : Viaje,\ g : Grilla,\ i : \mathbb{Z},\ j : \mathbb{Z})$\{ $(\exists k : \mathbb{Z})(\exists l : \mathbb{Z})(\ 0 \leq k < |g| \land_L 0 \leq l < |g| \land_L ((\ g[k]_{10} \leq v[i]_{10} \leq g[k]_{00} \land g[k]_{01} \leq v[i]_{11} \leq g[k]_{11}\ ) \land (\ g[l]_{10} \leq v[j]_{10} \leq g[l]_{00} \land g[l]_{01} \leq v[j]_{11} \leq g[l]_{11}\ ) \land (((g[k]_{21}$ = g[l]$_{21}$ $\land\ g[k]_{22}$ = g[l]$_{22}$) $\lor$ ($g[k]_{21}$ = g[l]$_{21}$ $\land\ g[k]_{22}$ = g[l]$_{22}$ + 1 $\lor\ g[k]_{22}$ = g[l]$_{22}$ - 1)) $\lor$ ($g[k]_{22}$ = g[l]$_{22}$ $\land$ ($g[k]_{21}$ = g[l]$_{21}$ + 1 $\lor\ g[k]_{21}$ = g[l]$_{21}$ - 1)))) \}

Ejercicio 10

proc completarHuecos (inout v: $Viaje$, in faltantes: $seq\langle\mathbb{Z}\rangle$) {

Pre { $v = v0 \ \wedge \ esViajeValido(v0) \ \wedge \ faltantesValido(faltantes, v0)$ }
Post { $|v| = |v0| \ \wedge_L \ (\forall i : \mathbb{Z})(\ (0 \leq i < |v0| \ \wedge_L \ i \in faltantes \ ) \wedge \ (\exists j : \mathbb{Z})(\exists k : \mathbb{Z})((\ 0 \leq j < |v0| \ \wedge_L \ 0 \leq k < |v0|) \wedge tieneParadasQueLlenenElHueco(v, i, j, k) \ \longrightarrow_L \ v[i] = llenarHuecos(v0, faltantes, i)))$ }
}

pred $faltantesValido(faltantes : seq\langle\mathbb{Z}\rangle, \ v : Viaje)$\{ $(\forall i : \mathbb{Z})(\ i \in faltantes \longrightarrow_L \ (\ 0 \leq i < |v0| \ \wedge_L \ (\ v0[i]_0 \geq 0 \ \wedge \ v0[i]_{1_0} = 0 \ \wedge v0[i]_{1_1} = 0 \ )))$ \}

aux `llenarHuecos` (v0: *Viaje*, faltantes: $seq\langle\mathbb{Z}\rangle$, i: $\mathbb{Z}$, j: $\mathbb{Z}$, k: $\mathbb{Z}$) : <*Tiempo* x *GPS*> $=$ ($v0[i]_{1_0} =$ posLatitudinalxMRU(v,i,j,k) $\wedge \ v0[i]_{1_1} =$ posLongitudinalxMRU(v,i,j,k))) ;

pred $tieneParadasQueLlenenElHueco(v : Viaje, \ i : \mathbb{Z}, \ j : \mathbb{Z}, \ k : \mathbb{Z})$\{ $j \notin faltantes \ \wedge \ k \notin faltantes \ \wedge \ (v[j]_0 \ < \mathrm{v[i]}_0 \ < \mathrm{v[k]}_0) \ \wedge \neg(\exists l : \mathbb{Z})((\exists m : \mathbb{Z})(\ (\ 0 \leq l < |v0| \ \ \wedge_L \ \ 0 \leq m < |v0| \ \wedge \ (\ l \notin faltantes \ \wedge \ m \notin faltantes \ ) \ \wedge \ v[j]_0 \ < \mathrm{v[l]}_0 \ < \mathrm{v[i]}_0 \ < \mathrm{v[m]}_0 \ < \mathrm{v[k]}_0)))$ \}

aux `posLatitudinalxMRU` (v: *Viaje*, i: $\mathbb{Z}$, j: $\mathbb{Z}$, k: $\mathbb{Z}$) : $\mathbb{R} = (v[i]_{1_0} = \mathrm{v[j]}_{1_0} + ((\mathrm{v[k]}_{1_0} - \mathrm{v[j]}_{1_0})/(\mathrm{v[k]}_0 - \mathrm{v[j]}_0)) * (\mathrm{v[i]}_0 - \mathrm{v[j]}_0)$;

aux `posLongitudinalxMRU` (v: *Viaje*, i: $\mathbb{Z}$, j: $\mathbb{Z}$, k: $\mathbb{Z}$) : $\mathbb{R} = (v[i]_{1_1} = \mathrm{v[j]}_{1_1} + ((\mathrm{v[k]}_{1_1} - \mathrm{v[j]}_{1_1})/(\mathrm{v[k]}_0 - \mathrm{v[j]}_0)) * (\mathrm{v[i]}_0 - \mathrm{v[j]}_0)$;

Ejercicio 11

proc `histograma` (in xs: $seq\langle Viaje\rangle$, in bins: $\mathbb{Z}$, out cuentas: $seq\langle\mathbb{Z}\rangle$, out limites: $seq\langle\mathbb{R}\rangle$) {
Pre { $listaDeViajesValida(xs)$ }
Post { /* *primero detallamos para límites* */
($| \ limites \ | = bins{+}1 \wedge (\forall i : \mathbb{Z})(\ 0 \leq i < |limites| \ \longrightarrow_L \ limites[i] = minVelMax(xs) + i * anchoBin(xs, bins)) \ ) \ \wedge$
/* ahora detallamos para cuentas */
($| \ cuentas \ | = bins \ \wedge \ cuentas[|cuentas| - 1] = cantDeElemEnUltimoBin(xs, bins) \ \wedge \ (\forall p : \mathbb{Z})(\ 0 \leq p \ \leq \ |cuentas|{-}2 \ \longrightarrow_L \ cuentas[p] = cantElemEnPBin(xs, p, bins)) \ )$ }
}

pred $listaDeViajesValida(xs : seq\langle Viaje\rangle)$\{$(\forall i : \mathbb{Z})(\ 0 \leq i < |xs| \longrightarrow_L viajeValido(xs[i]))$ \}

aux `anchoBin` (xs: $seq\langle Viaje\rangle$, bins: $\mathbb{Z}$) : $\mathbb{R} = (maxVelMax(xs) - minVelMax(xs))/bins$;

aux `cantElemEnPBin` (xs: $seq\langle Viaje\rangle$, p: $\mathbb{Z}$, bins: $\mathbb{Z}$) : $\mathbb{Z} = \sum_{i=0}^{|xs|-1} \ if \ (estaEnElpBin(xs, p, i, bins)) \ then \ 1 \ else \ 0 \ fi$;

aux `cantElemEnUltimoBin` (xs: $seq\langle Viaje\rangle$, bins: $\mathbb{Z}$) : $\mathbb{Z} = \sum_{i=0}^{|xs|-1} \ if \ (estaEnUltimoBin(xs, i, bins)) \ then \ 1 \ else \ 0 \ fi$;

pred $estaEnElpBin(xs : seq\langle Viaje\rangle, \ p : \mathbb{Z}, \ i : \mathbb{Z}, \ bins : \mathbb{Z})$\{ $minVelMax(xs) + (p * anchoBin(xs, bins)) \leq velMaxDelViaje(xs, i) < minVelMax(xs) + ((p{+}1) * anchoBin(xs, bins))$ \}

pred $estaEnElUltimoBin(xs : seq\langle Viaje\rangle, \ i : \mathbb{Z}, \ bins : \mathbb{Z})$\{ $maxVelMax(xs) - anchoBin(xs,$

$bins) \leq velMaxDelViaje(xs, i) \leq maxVelMax(xs)$ }

pred $noHayOtraVelMayor(xs : seq\langle Viaje\rangle,\ i : \mathbb{Z},\ j : \mathbb{Z},\ k : \mathbb{Z})\{\ \neg(\exists l : \mathbb{Z})((\exists m : \mathbb{Z})((\ 0 \leq l < |xs[i]|\ \wedge_L\ 0 \leq m < |xs[i]|)\ \wedge_L\ sonParadasSiguientesEnViajes(xs, i, l, m)\ \wedge$ $(dist(xs[i[l]_1], \mathrm{xs[i[m]_1]}))/(\mathrm{xs[i[m]_0]} - \mathrm{xs[i[l]_0]}))\ > \mathrm{velocidadEntrejYk(xs,i,j,k)}))$ }

pred $sonParadasSiguientesEnViajes(xs : seq\langle Viaje\rangle,\ i : \mathbb{Z},\ j : \mathbb{Z},\ k : \mathbb{Z})\{\ (xs[i[j]_0] < \mathrm{xs[i[k]_0]})$ $\wedge\ \neg(\exists p : \mathbb{Z})(\ 0 \leq p < |xs[i]|\ \wedge_L\ xs[i[j]_0]\ < \mathrm{xs[i[p]_0]})\ < \mathrm{xs[i[k]_0]})$ }

Ejercicio 12

proc `limpiar` (inout r: $seq\langle Viaje\rangle$, out borrados: $seq\langle \mathbb{Z}\rangle$) {
    `Pre` { $r = r0\ \wedge\ listaDeViajesValida(r0)$}
    `Post` {   /* *primero detallamos para borrados* */
    $(\forall i : \mathbb{Z})(\ 0 \leq i < |r0|\ \wedge_L\ esViajeExtremo(r0, i)\ \longrightarrow_L\ i \in borrados)\ \wedge\ (\forall j : \mathbb{Z})(\ j \in borrados \longrightarrow_L (\ 0 \leq j < |r0|\ \wedge_L\ esViajeExtremo(r0, j))\ \wedge$
        /* ahora detallamos para r */
    $|r| = |\ r0|\ \wedge\ (\forall i : \mathbb{Z})(\ 0 \leq i < |r0|\ \wedge_L\ (\ i \in borrados\ \longrightarrow_L\ |r[i]| = 0)\ )$}
}

pred $esViajeExtremo(r0 : seq\langle Viaje\rangle,\ i : \mathbb{Z})\{\ maxVelMax(r0) - anchoBin10(r0) \leq velMaxDelViaje(r0, i) \leq maxVelMax(r0)$ }

aux `anchoBin10` (r0: $seq\langle Viaje\rangle$) : $\mathbb{R}$ = ($maxVelMax(r0) - minVelMax(r0)$)/10 **;**

2. Predicados y Auxiliares generales

———————————— Auxiliares ————————————

aux `maxVelMax` (xs: $seq\langle Viaje\rangle$) : $\mathbb{R} = \sum_{i=0}^{|xs|-1}\ if\ \ (estaVelMaxDelViajeEsLaMayorDeTodas(xs, i))\ \ then\ velMaxDelViaje(xs, i)\ \ else\ \ 0\ \ fi$ **;**

aux `minVelMax` (xs: $seq\langle Viaje\rangle$) : $\mathbb{R} = \sum_{i=0}^{|xs|-1}\ if\ estaVelMaxDelViajeEsLaMenorDeTodas(xs, i)\ then\ velMaxDelViaje(xs, i)\ else\ 0\ fi$ **;**

aux `velMaxDelViaje` (xs: $seq\langle Viaje\rangle$, i: $\mathbb{Z}$) : $\mathbb{R} = \sum_{j=0}^{|xs[i]|-1}\ if\ \ esVelocidadMaxima(xs, i, j)\ then\ velocidadEntrejYk(xs, i, j, k)\ \ else\ \ 0\ fi$ **;**

aux `velocidadEntrejYk` (xs: $seq\langle Viaje\rangle$, i: $\mathbb{Z}$,j: $\mathbb{Z}$,k: $\mathbb{Z}$) : $\mathbb{R} = (dist(xs[i[j]_1], \mathrm{xs[i[k]_1]}))/((\mathrm{xs[i[k]_0]} - \mathrm{xs[i[j]_0]})$;

———————————— Predicados ————————————-

pred $esViajeValido(v : Viaje)\{\ |v| > 1\ \wedge\ (\forall i : \mathbb{Z})(0 \leq i < |v| \longrightarrow_L tiempoValido(v) \wedge (-90 \leq v[i]_{10} \leq 90) \wedge (-180 \leq v[i]_{11} \leq 180))\}$

pred $estaVelMaxDelViajeEsLaMayorDeTodas(xs : seq\langle Viaje\rangle,\ i : \mathbb{Z})\{\ \neg(\exists q : \mathbb{Z})(\ (\ 0 \leq q <$

$|xs| \wedge_L velMaxDelViaje(xs, i) < velMaxDelViaje(xs, q))$ }

pred $estaVelMaxDelViajeEsLaMenorDeTodas(xs : seq\langle Viaje\rangle,\ i : \mathbb{Z})\{\ \neg(\exists q : \mathbb{Z})(\ 0 \leq q < |xs| \wedge_L velMaxDelViaje(xs, i) > velMaxDelViaje(xs, q))\ \}$

pred $esVelocidadMaxima(xs : seq\langle Viaje\rangle, i : \mathbb{Z}, j : \mathbb{Z})\{\ (\exists k : \mathbb{Z})(\ 0 \leq k < |xs[i]| \wedge_L sonParadasSiguientesEnViajes(xs, i, j, k) \wedge_L noHayOtraVelMayor(xs, i, j, k))\ \}$

pred $grillaOK(g : Grilla,\ esq1 : GPS,\ esq2 : GPS,\ n : \mathbb{Z},\ m : \mathbb{Z})\{\ cantidadDeCeldasOK(g, n, m) \wedge\ (\forall i : \mathbb{Z})(\ 0 \leq i < |g| \longrightarrow_L (\exists\ indLat : \mathbb{Z})(\exists\ indLong : \mathbb{Z})((0 \leq indLat < n \wedge_L 0 \leq indLong < m) \wedge\ gpsOK(g, esq1, esq2, n, m, indLat, indLong, i) \wedge nombreOK(g, esq1, esq2, n, m, indLat, indLong, i)))\ \}$

pred $sonParadasSiguientes(v : Viaje, i : \mathbb{Z}, j : \mathbb{Z})\{\ v[i]_0 < \mathrm{v[j]}_0 \wedge \neg(\exists x : \mathbb{Z})(0 \leq x < |v| \wedge_L \mathrm{v[i]}_0 < \mathrm{v[x]}_0 < \mathrm{v[j]}_0)\ \}$

pred $tiempoValido(v : Viaje)\{\ (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < |v| \wedge_L (v[i]_0 = \mathrm{v[j]}_0) \longrightarrow_L (v[i]_0 > 0 \wedge v[i]_1 = \mathrm{v[j]}_1))\ \}$

3. Decisiones tomadas

A lo largo del trabajo se presentaron ciertas disyuntivas sobre cómo definir algunos conceptos respecto a la problemática de los viajes, por lo tanto se tomaron una serie de decisiones que ayudaron a la definición de los mismos y consecuentemente a la especificación de los problemas.

1. Ante la disyuntiva de qué es viajar consideramos que un viaje consiste de al menos 2 elementos registrados en el viaje puesto que si hubiese sólo uno (o, peor aún, ninguno) no estaríamos viajando a ningún lado (¡o ni siquiera existiendo!)
2. Ante la problemática de qué implicaría que un colectivo viaje dentro de una franja horaria decidimos que lo haría si y sólo si en algún momento del recorrido se registró al menos un punto en ese intervalo de tiempo.
3. A la hora de grillar, como no quedaba claro si se pedía que las celdas fueran un cuadrilátero cualquiera o exactamente cuadradas se decidió que los lados de cada celda fueran iguales (y por lo tanto, las celdas fueran cuadradas).
4. A la hora de considerar si un trayecto estaba contenido en una grilla o no, dictaminamos que éste en su totalidad esté incluído en la misma.
5. Como lo establece la consigna, un salto en el trayecto de un colectivo se produce cuando dos puntos del viaje consecutivos temporalmente se encuentran a dos o más celdas de distancia, pero nada dice de cómo deben estar unidas esas celdas, por lo que consideramos que una celda es consecutiva a otra si estas comparten un lado (es decir, que la segunda está o bien arriba o abajo, o a la izquierda o la derecha de la primera). Es decir, de este modo se descarta como posibilidad que celdas consecutivas se encuentren en diagonal puesto que estarían a exactamente 2 celdas de distancia.
6. A la hora de completar huecos en el viaje cuando se produjera un faltante se consideró que el colectivo viajaría a velocidad constante entre los dos puntos inmediatamente anteriores y posteriores al hueco que no fueran también faltantes. Esto es fundamental en la resolución puesto que se logró calcular la posición para ese tiempo con datos faltantes considerando entonces un MRU.

7. Al confeccionar el histograma se tuvieron en cuenta como límites la máxima velocidad máxima y la mínima velocidad máxima de todos los viajes comprendidos en xs.