TP de Especificación

17 de mayo 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

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

Ejercicio 2

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

pred $huboExcesoDeVelocidad(v : Viaje)$\{ $(\exists i : \mathbb{Z})(\exists j : \mathbb{Z})((\ 0 \leq i < |v| \ \wedge \ 0 \leq j < |v|) \ \wedge_L sonParadasSiguientes(v, i, j)) \ \wedge_L \ velocidad(v[i]_1, v[j]_1, v[i]_0, v[j]_0) \ > 80)$ \}

$\texttt{aux velocidad}$ (posicion1: *GPS*, posicion2:*GPS*, tiempo1: *Tiempo*, tiempo2:*Tiempo*) : $\mathbb{Z} = dist(posicion1, posicion2)/(tiempo2 - tiempo1)$;

Ejercicio 3

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

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

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

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

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

Ejercicio 4

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

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

aux `gPSParadaSiguiente` (v: *Viaje*,i: $\mathbb{Z}$) : *GPS* $= \sum_{k=0}^{|v|-1}$ $if\ (sonParadasSiguiente(v, v[i]_0$,v[k]$_0$)) then v[k]$_1$ else 0 fi ;

Ejercicio 5

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

pred $viajoEnLaFranja(v : Viaje$, t$_0$: *Tiempo*, t_f : *Tiempo*) { $(\exists i : \mathbb{Z})(0 \leq i < |v| \wedge_L$ t$_0$ $\leq v[i]_0 \leq$ t$_f$) }

Ejercicio 6

proc `recorridoCubierto` (in v: *Viaje*,in r: *Recorrido*,in u: *Dist*, out res: *seq⟨GPS⟩*) {
Pre { $esViajeValido(v) \wedge esRecorridoValido(r) \wedge u > 0$}
Post { $(\forall i : \mathbb{Z})((0 \leq i < |r| \wedge_L \neg(puntoCubierto(r[i], v, u)) \longrightarrow_L r[i] \in res) \wedge_L (r[i] \in res \longrightarrow_L \neg(puntoCubierto(r[i], v, u))) \wedge_L |res| = cantidadDePuntosNoCubiertos(r, v, u)$}
}

pred $esRecorridoValido(r : Recorrido)\{ (\forall i : \mathbb{Z})(0 \leq i < |r| \longrightarrow_L gpsValido(r[i])\}$

aux `cantidadDePuntosNoCubiertos` (r: *Recorrido*,v: *Viaje*, u: *Dist*) : $\mathbb{Z}$ $= \sum_{i=0}^{|r|-1}$ if $(\neg(puntoCubierto(r[i], v, u))\ then\ 1\ else\ 0\ fi$ **;**

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

Ejercicio 7

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

pred $celdasCuadradas(esq1 : GPS, esq2 : GPS, n : \mathbb{Z}, m : \mathbb{Z})\{ conseguirLadoLatitudinal(esq1, esq2, n) = conseguirLadoLongitudinal(esq1, esq2, m)\}$

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

$esq1_1 \ < \mathrm{esq2}_1 \leq 180$ }

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

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

pred $cantidadCeldasOK(g : Grilla, n : \mathbb{Z}, m : \mathbb{Z})\{\ |g| = n * m\ \}$

pred $gpsOK(g : Grilla,\ esq1 : GPS,\ esq2 : GPS,\ n : \mathbb{Z},\ m : \mathbb{Z},\ indLat : \mathbb{Z},\ indLong : \mathbb{Z},\ i : \mathbb{Z})\{\ esquinaSuperiorIzq(g, esq1, esq2, n, m, indLat, inLong, i) \wedge esquinaInferiorDer(g, esq1, esq2, n, m, indLat, inLong, i)\}$

pred $esquinaSuperiorIzq(g : Grilla,\ esq1 : GPS,\ esq2 : GPS,\ n : \mathbb{Z},\ m : \mathbb{Z},\ indLat : \mathbb{Z},\ indLong : \mathbb{Z},\ i : \mathbb{Z})\{$ $(g[i]_{0_0} = \mathrm{esq1}_0$ - conseguirLadoLatitudinal(esq1, esq2, n) * indLat) $\wedge$ ($g[i]_{0_1} = \mathrm{esq1}_1 +$ conseguirLadoLongitudinal(esq1, esq2, m) * indLong) }

pred $esquinaInferiorDer(g : Grilla,\ esq1 : GPS,\ esq2 : GPS,\ n : \mathbb{Z},\ m : \mathbb{Z},\ indLat : \mathbb{Z},\ indLong : \mathbb{Z},\ i : \mathbb{Z})\{$ ($g[i]_{1_0} = \mathrm{esq2}_0 +$ conseguirLadoLatitudinal(esq1, esq2, n) * indLat) $\wedge$ ($g[i]_{1_1} = \mathrm{esq2}_1$ - 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) $\wedge$ ($g[i]_{2_1}$ = indLong + 1) }

Ejercicio 8

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

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

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

pred $puntoDeTrayectoEnAlgunaCeldaDeLaGrilla(puntotrayecto : GPS,\ g : Grilla,\ j : \mathbb{Z})\{\ (g[j]_{1_0} \leq puntotrayecto_0 \leq g[j]_{0_0}\ \wedge g[j]_{0_1} \leq puntotrayecto_1 \leq g[j]_{1_1})\ \}$

Ejercicio 9

proc `cantidadDeSaltos` (**in** g: *Grilla*, **in** v: *Viaje*, **out** res: $\mathbb{Z}$) {
    `Pre` { $grillaValida(g)\ \wedge\ esViajeValido(v)\ \wedge\ viajeEnGrilla(v, g)$ }
    `Post` { $res = numeroDeSaltos(g, v)$ }
}

pred $grillaValida(g : Grilla)\{\ (\exists n : \mathbb{Z})(\exists m : \mathbb{Z})(\exists esq1 : GPS)(\exists esq2 : GPS)((n > 0 \wedge m > 0 \wedge$

$gpsValido(esq1) \ \wedge \ gpsValido(esq2)) \wedge_L grillaOK(g, esq1, esq2, n, m)\}$

pred $viajeEnGrilla(v : Viaje,\ g : Grilla)\{\ (\forall i : \mathbb{Z})(\ 0 \leq i < |v| \longrightarrow_L \ trayectoEnGrilla(v[i]_1$,g)
}

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

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

pred $losPuntosEstanComoMuchoAUnaCeldaDeDistancia(parada1 : <Tiempo\,x\,GPS>, parada2 : <Tiempo\,x\,GPS>,\ g : Grilla)\{\ (\exists k : \mathbb{Z})(\exists l : \mathbb{Z})(\ 0 \leq k < |g| \wedge 0 \leq l < |g| \wedge_L parada1EstaEnLaKCelda (parada1, g[k]) \wedge_L parada2EstaEnLaLCelda(parada2, g[l]) \wedge_L (lasCeldasLyKSonLaMisma(g[k], g[l]) \vee \ lasCeldasCompartenUnBorde(g[k], g[l]))\ \}$

pred $parada1EstaEnLaKCelda(parada1 : <Tiempo\,x\,GPS>,\ celdaK : <GPS\,x\,GPS\,x\,Nombre>)\{ celdaK_{10} \leq parada1_{10} \leq celdaK_{00} \ \wedge celdaK_{01} \leq parada1_{11} \leq celdaK_{11} \ \ \}$

pred $parada2EstaEnLaLCelda(parada2 : <Tiempo\,x\,GPS>,\ celdaL : <GPS\,x\,GPS\,x\,Nombre>)\{ celdaL_{10} \leq parada2_{10} \leq celdaL_{00} \ \wedge celdaL_{01} \leq parada2_{11} \leq celdaL_{11} \ \ \}$

pred $lasCeldasLyKSonLaMisma(celdaK : <GPS\,x\,GPS\,x\,Nombre>,\ celdaL : <GPS\,x\,GPS\,x\,Nombre>)\{celdaK_{20} = \mathrm{celdaL}_{20} \ \wedge celdaK_{21} = \mathrm{celdaL}_{21} \ \ \}$

pred $lasCeldasCompartenUnBorde(celdaK : <GPS\,x\,GPS\,x\,Nombre>,\ celdaL : <GPS\,x\,GPS\,x\,Nombre>)\{\ ((celdaK_{20} = \mathrm{celdaL}_{20}) \ \wedge ((celdaK_{21} = \mathrm{celdaL}_{21} + 1) \ \vee (celdaK_{21} = \mathrm{celdaL}_{21}$ - 1 $))) \ \vee ((\ celdaK_{21} = \mathrm{celdaL}_{21}) \ \wedge ((celdaK_{20} = \mathrm{celdaL}_{20} + 1) \ \vee (celdaK_{20} = \mathrm{celdaL}_{20}$ - 1))) }

Ejercicio 10

`proc` `completarHuecos` (inout v: *Viaje*, in faltantes: $seq\langle\mathbb{Z}\rangle$) {
`Pre` $\{\ v = v0 \wedge esViajeValido(v0) \wedge faltantesValido(faltantes, v0) \wedge 0 \notin faltantes \wedge |v0| - 1 \notin faltantes\}$
`Post` $\{\ |v| = |v0| \ \wedge \ (\forall x : \mathbb{Z})(0 \leq x < |v0| \ \wedge \ x \notin faltantes \longrightarrow_L v[x] = v0[x]\ ) \ \wedge \ (\forall i : \mathbb{Z})(\ (0 \leq i < |v0| \wedge i \in faltantes\ ) \wedge (\exists j : \mathbb{Z})(\exists k : \mathbb{Z})((\ 0 \leq j < |v0| \wedge 0 \leq k < |v0|) \wedge_L tieneParadasQueLlenenElHueco(v, i, j, k) \ \longrightarrow_L \ v[i] = llenarHuecos(v0, faltantes, i, j, k)))\ \}$
}

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 $GPS> =$ $(v0[i]_0$,(posLatitudinalxMRU(v,i,j,k),posLongitudinalxMRU(v,i,j,k))) ;

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

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

aux `posLongitudinalxMRU` (v: *Viaje*, i: $\mathbb{Z}$, j: $\mathbb{Z}$, k: $\mathbb{Z}$) : $\mathbb{R}$ = $v[j]_{1_1}$ + velocidad(v[k]$_{1_1}$, v[j]$_{1_1}$, v[k]$_0$, v[j]$_0$) * (v[i]$_0$ - 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` $\{ \ listaDeViajesValida(xs) \ \wedge \ 1 \leq bins\}$
  `Post` $\{ \ | \ limites \ | = bins + 1 \ \wedge \ limitesCorrectos(limites, bins, xs) \ \wedge \ | \ cuentas \ | = bins \ \wedge \ cuentasCorrectas(cuentas, bins, xs)\}$
}

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

pred $limitesCorrectos(limites : seq\langle\mathbb{R}\rangle, \ bins : \mathbb{Z}, \ xs : seq\langle Viaje\rangle)\{ \ (\forall i : \mathbb{Z})(\ 0 \leq i < |limites| \ \longrightarrow_L \ limites[i] = minVelMaxSinRepetidos(xs) + i * anchoBin(xs, bins)) \ )\}$

pred $cuentasCorrectas(\ cuentas : seq\langle\mathbb{Z}\rangle, \ bins : \mathbb{Z}, \ xs : seq\langle Viaje\rangle)\{cuentas[|cuentas| - 1] = cantDeElemEnUltimoBin(xs, bins) \ \wedge \ (\forall p : \mathbb{Z})(\ 0 \leq p \ \leq \ |cuentas| - 2 \ \longrightarrow_L \ cuentas[p] = cantElemEnPBin(xs, p, bins))\}$

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

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

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

pred $noHayOtraVelMayor(xs : seq\langle Viaje\rangle, \ i : \mathbb{Z}, \ j : \mathbb{Z}, \ k : \mathbb{Z})\{ \ \neg(\exists l : \mathbb{Z})((\exists m : \mathbb{Z})((\ 0 \leq l < |xs[i]| \ \wedge \ 0 \leq m < |xs[i]|) \ \wedge_L \ sonParadasSiguientes(xs[i], xs[i][l]_0$,xs[i][m]$_0$) $\wedge_L$ $velocidad(xs[i][l]_1$,xs[i][m]$_1$,xs[i][l]$_0$,xs[i][m]$_0$) > velocidad(xs[i][j]$_1$,xs[i][k]$_1$,xs[i][j]$_0$,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` $\{(\forall i : \mathbb{Z})((\ 0 \leq i < |r0| \ \wedge_L \ esViajeExtremo(r0, i)) \ \longrightarrow \ i \in borrados) \ \wedge \ (\forall j :$

$\mathbb{Z})(\ j \in borrados \longrightarrow (\ 0 \leq j < |r0| \ \wedge_L \ esViajeExtremo(r0, j)) \ \wedge \ |borrados| = cantidadDeViajesExtremos(r0) \ \wedge \ |r| = |r0| \ \wedge \ (\forall x : \mathbb{Z})((\ 0 \leq x < |r0| \ \wedge \neg(esViajeExtremo(r0, x)) \longrightarrow_L r[x] = r0[x]\ ) \ \wedge \ (\forall i : \mathbb{Z})((\ 0 \leq i < |r0| \ \wedge \ i \in borrados\ ) \longrightarrow |r[i]| = 0)\ )\}$

}

pred $cantidadDeViajesExtremos(r0 : seq\langle Viaje\rangle)\{ \sum_{i=0}^{|r0|-1} \ if \ (esViajeExtremo(r0, i)) \ then \ 1 \ else \ 0 \ fi \ \}$

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

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

2. Predicados y Auxiliares generales

———————————————— Auxiliares ————————————————

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

aux `minVelMaxConRepetidos` (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: $\mathbb{Z}$) : $\mathbb{R} = \sum_{j=0}^{|xs[i]|-1} \ if \ esVelocidadMaxima(xs, i, j) \ then \ velocidad(xs[i][j]_1$,xs[i][k]$_1$,xs[i][j]$_0$,xs[i][k]$_0$) else 0 fi ;

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

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

aux `maxVelMaxSinRepetidos` (xs: $seq\langle Viaje\rangle$) : $\mathbb{R} = maxVelMaxConRepetidos(xs)/contadorDeVelMaxs(xs)$;

aux `minVelMinSinRepetidos` (xs: $seq\langle Viaje\rangle$) : $\mathbb{R} = maxVelMinConRepetidos(xs)/contadorDeVelMins(xs)$;

———————————————— Predicados ————————————————-

pred $esViajeValido(v : Viaje)\{\ |v| > 1 \ \wedge \ tiempoValido(v) \ \wedge \ (\forall i : \mathbb{Z})(0 \leq i < |v| \longrightarrow_L gpsValido(v[i])\}$

pred $gpsValido(posicion : GPS)\{(-90 \leq posicion_0 \leq 90) \wedge (-180 \leq posicion_1 \leq 180)\}$

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

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

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

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

pred $sonParadasSiguientes(v : Viaje,\ tiempoparada1 : Tiempo,\ tiempoparada2 : Tiempo)\{\ tiempoparada1 < tiempoparada2\ \wedge\ \neg(\exists x : \mathbb{Z})(0 \leq x < |v|\ \wedge_L\ \mathrm{tiempoparada1} < \mathrm{v[x]}_0 < \mathrm{tiempoparada2})\ \}$

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

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.