DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

Esperando al Bondi

29 de Abril de 2022 Algoritmos y Estructuras de Datos I

Grupo 6

Integrante	LU	Correo electrónico
Celie, Nicolás Sebastián	655/22	ncelie@dc.uba.ar
Fontana Walser, Florencia	1530/21	florfontana02@gmail.com
Fernandez Aragon, Agustin	998/21	f.a.agustin@gmail.com
Lloveras, Joaquín María	303/20	joaquinmllov@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. Definición de Tipos.

type Tiempo = $\mathbb{R}$

type Dist = $\mathbb{R}$

type GPS = $\mathbb{R}\times\mathbb{R}$

type Recorrido = $seq\langle GPS\rangle$

type Viaje = $seq\langle Tiempo \times GPS\rangle$

type Nombre = $\mathbb{Z}\times\mathbb{Z}$

type Grilla = $seq\langle GPS \times GPS \times Nombre\rangle$

2. Auxiliares y predicados generales.

En esta sección estarán ubicadas las funciones más comunes que utilizaremos a lo largo del trabajo.

pred esViajeValido (v: Viaje) $\{tiempoValido(v) \wedge (\forall i : \mathbb{Z})(0 \leq i < |v| \longrightarrow_L GPSValido(v[i]_1))$
$\wedge (\forall i, j : \mathbb{Z})(0 \leq i < j < |v| \longrightarrow_L v[i]_0 \neq v[j]_0)$
}

pred tiempoValido (t: Viaje) {
$(\forall i : \mathbb{Z})(0 \leq i < |t| \longrightarrow_L 0 \leq t[i]_0)$
}

pred GPSValido (g: GPS) {
$(-90 \leq g_0 \leq 90 \wedge -180 \leq g_1 \leq 180))$
}

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

pred enOrden (v, c: Viaje) {
$estaOrdenado(c) \wedge esPermutacion(v, c)$
}

pred estaOrdenado (v: Viaje) {
$(\forall i : \mathbb{Z})(0 \leq i < |v| - 1 \longrightarrow_L v[i]_0 < v[i+1]_0)$
}

pred esPermutacion (v,c: Viaje) {
$(\forall i : Tiempo \times GPS)(apariciones(c, i) = apariciones(v, i))$
}

pred esLaGrilla (g: Grilla, esq1: GPS, esq2: GPS, n,m: $\mathbb{Z}$) {
$(|g| = n \cdot m) \wedge$
$(\forall i, j : \mathbb{Z})((1 \leq i \leq n \wedge 1 \leq j \leq m) \longrightarrow_L$
$(\exists h : \mathbb{Z})((0 \leq h < |g| \wedge_L (g[h]_2 = (i, j) \wedge$
$g[h]_{0,0} = \frac{alturaGrilla(esq1,esq2)}{n} \cdot (n - i + 1) + esq2_0 \wedge$
$g[h]_{0,1} = \frac{baseGrilla(esq1,esq2)}{m} \cdot (j - 1) + esq1_1 \wedge$
$g[h]_{1,0} = \frac{alturaGrilla(esq1,esq2)}{n} \cdot (n - i) + esq2_0 \wedge$
$g[h]_{1,1} = \frac{baseGrilla(esq1,esq2)}{m} \cdot j + esq1_1)))$
}

- Comentarios: En el predicado esLaGrilla estamos especificando la latitud y la longitud de las esquinas 1 y 2 de **cada celda**. Siendo $g[h]_{0,0}$ la latitud de la esquina 1, $g[h]_{0,1}$ la longitud de la esquina 1, $g[h]_{1,0}$ la latitud de la esquina 2 y $g[h]_{1,1}$ la longuitud de la esquina 2.

aux alturaGrilla (esq1, esq2: GPS) : $\mathbb{Z}$ $= esq1_0 - esq2_0$;

aux baseGrilla (esq1, esq2: GPS) : $\mathbb{Z}$ $= esq2_1 - esq1_1$;

`pred esGrillaValida` (g: Grilla) {

$(\forall k : \mathbb{Z})(0 \leq k < |g| \longrightarrow_L (GPSValido(g[k]_0) \land GPSValido(g[k]_1))) \land$
$(\exists n, m : \mathbb{Z})(nmMaximos(n, m, g) \land (\forall i : \mathbb{Z})(0 \leq i < |g| \longrightarrow_L nombreValido(n, m, g, i)) \land$
$(\exists esq1, esq2 : GPS)((\forall j : \mathbb{Z})(0 \leq j < |g| \longrightarrow_L esq1esq2Maximas(esq1, esq2, j, g)) \land$
$esLaGrilla(g, esq1, esq2, n, m)))$

}

`pred nmMaximos` (n: Nombre, m: Nombre, g: Grilla) {

$(\forall i : \mathbb{Z})(0 \leq i < |g| \longrightarrow_L (n \geq g[i]_{2,0} \land m \geq g[i]_{2,1} \land$
$(\exists j : \mathbb{Z})(0 \leq j < |g| \land_L (n = g[j]_{2,0} \land m = g[j]_{2,1}))))$

}

`pred nombreValido` (n: Nombre, m: Nombre, g: Grilla, i: $\mathbb{Z}$) {

$1 \leq g[i]_{2,0} \leq n \land 1 \leq g[i]_{2,1} \leq m \land esNombreUnico(g, i)$

}

`pred esNombreUnico` (g: Grilla, i: $\mathbb{Z}$) {

$\neg(\exists j : \mathbb{Z})((0 \leq j < |g| \land j \neq i) \land_L g[i]_2 = g[j]_2)$

}

`pred esq1esq2Maximas` (esq1: GPS, esq2: GPS, j: $\mathbb{Z}$, g: Grilla) {

$esq1_0 \geq g[j]_{0,0} \land esq1_1 \leq g[j]_{0,1} \land (\exists k : \mathbb{Z})(0 \leq k < |g| \land_L esq1 = g[k]_0) \land$
$esq2_0 \leq g[j]_{1,0} \land esq2_1 \geq g[j]_{1,1} \land (\exists k : \mathbb{Z})(0 \leq k < |g| \land_L esq2 = g[k]_1)$

}

`aux apariciones` (v: viaje, e: Tiempo$\times GPS$) $: \mathbb{Z} = \sum_{i=0}^{|v|-1}$ if $v[i] = e$ then 1 else 0 fi ;

3. Problemas

3.1. Ejercicio 1

proc `viajeValido` (in v: Viaje, out res: Bool) {

Pre $\{True\}$

Post $\{res = \text{true} \leftrightarrow esViajeValido(v)\}$

}

3.2. Ejercicio 2

proc `recorridoValido` (in v: Recorrido, out res: Bool) {

Pre $\{True\}$

Post $\{res = \text{true} \leftrightarrow esRecorridoValido(v)\}$

}

3.3. Ejercicio 3

proc `enTerritorio` (in v: Viaje, in r: Dist, out res: Bool) {

Pre $\{esViajeValido(v) \land r \geq 0\}$

Post $\{res = \text{true} \leftrightarrow losPuntosGPSEstanEnRadio(v, r)\}$

}

pred **losPuntosGPSEstanEnRadio** (v: Viaje, r: Dist) {

$(\exists i : GPS)((\forall j : \mathbb{Z})(0 \leq j < |v| \longrightarrow_L dist(i, v[j]_1) \leq r \cdot 1000))$

}

3.4. Ejercicio 4

proc `tiempoTotal` (in v: Viaje, out t: Tiempo) {

Pre $\{esViajeValido(v)\}$

Post $\{(\exists tMax, tMin : Tiempo)(esMaximoTiempo(v, tMax) \land esMinimoTiempo(v, tMin) \land t = tMax - tMin)\}$

pred **esMaximoTiempo** (v: Viaje, t : Tiempo) {

$(\exists i : \mathbb{Z})(0 \leq i < |v| \land_L t = v[i]_0) \land_L \neg(\exists j : \mathbb{Z})(0 \leq j < |v| \land_L t < v[j]_0)$

}

pred **esMinimoTiempo** (v: Viaje, t : Tiempo) {

$(\exists i : \mathbb{Z})(0 \leq i < |v| \land_L t = v[i]_0) \land_L \neg(\exists j : \mathbb{Z})(0 \leq j < |v| \land_L t > v[j]_0)$

}

}

3.5. Ejercicio 5

proc `distanciaTotal` (in v: Viaje, out d: Dist) {

Pre $\{esViajeValido(v)\}$

Post $\{(\exists c : Viaje)(enOrden(v,c) \land d = \frac{\sum_{i=0}^{|c|-2} dist(c[i]_1, c[i+1]_1)}{1000})\}$

}

3.6. Ejercicio 6

proc `excesoDeVelocidad` (in v: Viaje, out res: Bool) {

Pre $\{esViajeValido(v)\}$

Post $\{res = True \leftrightarrow (\exists c : Viaje)(enOrden(v,c) \land superoLos80KMH(c)\}$

pred `superoLos80KMH` (in v: Viaje) {

$(\exists i : \mathbb{Z})(0 \leq i < |v| - 1 \land_L \frac{dist(v[i]_1, v[i+1]_1)}{v[i+1]_0 - v[i]_0} > 80 \cdot \frac{10}{36})$

}

}

3.7. Ejercicio 7

proc `flota` (in v: $seq\langle Viaje\rangle$, in $t_0 : Tiempo$, in $t_f : Tiempo$, out $res : \mathbb{Z}$){

Pre $\{0 \leq t_0 < t_f \land (\forall i : \mathbb{Z})(0 \leq i < |v| \longrightarrow_L viajeValido(v[i])\}$

Post $\{res = \sum_{i=0}^{|v|-1}(\text{if } estaEntreTiempos(v[i], t_0, t_f) \text{ then } 1 \text{ else } 0 \text{ fi})\}$

pred `estaEntreTiempos` (x: Viaje, $t_0 : Tiempo, t_f : Tiempo$){

$(\exists i : \mathbb{Z})(0 \leq i < |x| \land_L t_0 \leq x[i]_0 \leq t_f) \lor$
$(\exists c : Viaje)(enOrden(c,x) \land_L (c[0]_0 \leq t_0 \leq c[|c|-1]_0 \lor c[0]_0 \leq t_f \leq c[|c|-1]_0))$

}

}

3.8. Ejercicio 8

proc `recorridoNoCubierto` (in v: Viaje, in r: Recorrido, in u : Dist, out res : $seq\langle GPS\rangle$) {

Pre $\{esViajeValido(v) \land esRecorridoValido(r) \land u \geq 0\}$

Post $\{p \in res \leftrightarrow p \in r \land (\forall i : \mathbb{Z})(0 \leq i < |v| \longrightarrow_L dist(v[i]_1, p) \geq 1000 \cdot u) \land (\forall i : \mathbb{Z})(0 \leq i < |res| \longrightarrow_L apariciones(res, res[i]) = 1)\}$

aux `apariciones` (res:$seq\langle GPS\rangle$, e: GPS) : $\mathbb{Z} = \sum_{i=0}^{|res|-1} \text{if } res[i] = e \text{ then } 1 \text{ else } 0 \text{ fi}$;

}

3.9. Ejercicio 9

`proc constuirGrilla` (in esq1: GPS, in esq2: GPS, in n :$\mathbb{Z}$,in m :$\mathbb{Z}$, out g : Grilla) {

`Pre` $\{GPSValido(esq1) \land GPSValido(esq2) \land esquinasValidas(esq1, esq2) \land n > 0 \land m > 0\}$

`Post` $\{esLaGrilla(g, esq1, esq2, n, m)\}$

`pred esquinasValidas` (esq1: GPS, esq2: GPS) {

$esq1_0 > esq2_0 \land esq2_1 > esq1_1$

}

`pred esLaGrilla` (g: Grilla, esq1: GPS, esq2: GPS, n,m: $\mathbb{Z}$) {

$$
\begin{aligned}
&(|g| = n \cdot m) \land \\
&(\forall i, j : \mathbb{Z})((1 \leq i \leq n \land 1 \leq j \leq m) \longrightarrow_L \\
&(\exists h : \mathbb{Z})(0 \leq h < |g| \land_L (g[h]_2 = (i, j) \land \\
&g[h]_{0,0} = \tfrac{alturaGrilla(esq1,esq2)}{n} \cdot (n - i + 1) + esq2_0 \land \\
&g[h]_{0,1} = \tfrac{baseGrilla(esq1,esq2)}{m} \cdot (j - 1) + esq1_1 \land \\
&g[h]_{1,0} = \tfrac{alturaGrilla(esq1,esq2)}{n} \cdot (n - i) + esq2_0 \land \\
&g[h]_{1,1} = \tfrac{baseGrilla(esq1,esq2)}{m} \cdot j + esq1_1)))
\end{aligned}
$$

}

- Comentarios: En el predicado esLaGrilla estamos especificando la latitud y la longitud de las esquinas 1 y 2 de **cada celda**. Siendo $g[h]_{0,0}$ la latitud de la esquina 1, $g[h]_{0,1}$ la longitud de la esquina 1, $g[h]_{1,0}$ la latitud de la esquina 2 y $g[h]_{1,1}$ la longuitud de la esquina 2.

`aux alturaGrilla` (esq1, esq2: GPS) : $\mathbb{Z}$ $= esq1_0 - esq2_0$ **;**

`aux baseGrilla` (esq1, esq2: GPS) : $\mathbb{Z}$ $= esq2_1 - esq1_1$ **;**

- Comentarios: alturaGrilla trabaja sobre latitud y baseGrilla trabaja sobre longitud.

}

3.10. Ejercicio 10

`proc regiones` (in r: Recorrido, in g: Grilla, out res: $seq\langle Nombre\rangle$) {

`Pre` $\{esRecorridoValido(r) \land esGrillaValida(g) \land recorridoEnGrilla(r, g)\}$

`Post` $\{|res| = |r| \land (\forall i : \mathbb{Z})(0 \leq i < |r| \longrightarrow_L estaEnCelda(r[i], res[i], g)\}$

`pred recorridoEnGrilla` (r: Recorrido, g: Grilla) {

$(\forall j : \mathbb{Z})((\exists i : \mathbb{Z})((0 \leq i < |g| \land 0 \leq j < |r|) \longrightarrow_L (g[i]_{1,0} < r[j]_0 \leq g[i]_{0,0} \land g[i]_{0,1} \leq r[j]_1 < g[i]_{1,1})))$

}

`pred estaEnCelda` (p : GPS, n : Nombre, g : Grilla) {

$(\exists i : \mathbb{Z})(0 \leq i < |g| \land_L n = g[i]_2 \land_L (g[i]_{1,0} < p_0 \leq g[i]_{0,0} \land g[i]_{0,1} \leq p_1 < g[i]_{1,1}))$

}

}

- Comentarios: La razón por la cual en estaEnCelda y en recorridoEnGrilla a veces utilizamos $<$ o $\leq$ es debido a que no tenemos en consideración los bordes de abajo y de la derecha de la grilla.

3.11. Ejercicio 11

`proc cantidadDeSaltos` (in g: Grilla,in v: Viaje, out res: $\mathbb{Z}$) {

`Pre` $\{esViajeValido(v) \wedge esGrillaValida(g) \wedge viajeEnGrilla(v, g)\}$

`Post` $\{(\exists c : Viaje)(enOrden(v, c) \wedge res = \sum_{i=0}^{|c|-2} \text{if } haySalto(c, i, g) \text{ then } 1 \text{ else } 0 \text{ fi})\}$

`pred haySalto` (c: Viaje, i: $\mathbb{Z}$, g: Grilla) {

$\neg(\exists j : \mathbb{Z})(0 \leq j < |g| \wedge_L (g[j]_{1,0} < c[i]_{1,0} \leq g[j]_{0,0} \wedge g[j]_{0,1} \leq c[i]_{1,1} < g[j]_{1,1} \wedge$
$g[j]_{0,1} - longitudDeCelda(g) \leq c[i+1]_{1,1} < g[j]_{1,1} + longitudDeCelda(g) \wedge$
$g[j]_{1,0} - latitudDeCelda(g) < c[i+1]_{1,0} \leq g[j]_{0,0} + latitudDeCelda(g)))$

}

`pred viajeEnGrilla` (v:Viaje, g:Grilla) {

$(\forall i : \mathbb{Z})(0 \leq i < |v| \longrightarrow_L (\exists j : \mathbb{Z})(0 \leq j < |g| \wedge_L estaContenidoEnLaCelda(v[i], g[j])))$

}

`pred estaContenidoEnLaCelda` (v: Tiempo $\times GPS, g : GPS \times GPS \times Nombre$){

$g_{1,0} < v_{1,0} \leq g_{0,0} \wedge g_{0,1} \leq v_{1,1} < g_{1,1}$

}

- Comentarios: En los predicados haySalto, viajeEnGrilla y estaContenidoEnLaCelda sucede lo mismo que en el ejercicio anterior, y por eso es el uso de $<$ o $\leq$.

`aux latitudDeCelda` (g: Grilla) : $\mathbb{R}$ $= g[0]_{0,0} - g[0]_{1,0}$;

`aux longitudDeCelda` (g: Grilla) : $\mathbb{R}$ $= g[0]_{1,1} - g[0]_{0,1}$;

}

3.12. Ejercicio 12

`proc corregirViaje` (inout v: Viaje, in errores: $seq\langle Tiempo\rangle$) {

`Pre` $\{esViajeValido(v) \wedge |v| \geq 5 \wedge v = V_0 \wedge 0 < |errores| \leq \frac{|v|}{10} \wedge losErroresEstanEnElViaje(errores, v)\}$

`Post` $\{(\forall k : \mathbb{Z})((0 \leq k < |v| \wedge_L aparece(v[k]_0, errores)) \longrightarrow_L estaCorregidoSegunLosCasos(k, v, errores)) \wedge$
$(\forall j : \mathbb{Z})((0 \leq j < |v| \wedge_L \neg aparece(v[j]_0, errores)) \longrightarrow_L V_0[j] = v[j])\}$

`pred losErroresEstanEnElViaje` (errores: $seq\langle Tiempo\rangle$,v: Viaje) {

$(\forall i : \mathbb{Z})(0 \leq i < |errores| \longrightarrow_L (\exists j : \mathbb{Z})(0 \leq j < |v| \wedge_L v[j]_0 = errores[i]))$

}

`pred aparece` (t: Tiempo, e: $seq\langle Tiempo\rangle$) {

$(\exists i : \mathbb{Z})(0 \leq i < |e| \wedge_L e[i] = t)$

}

`pred estaCorregidoSegunLosCasos` (k:$\mathbb{Z}$, v: Viaje, errores: $seq\langle Tiempo\rangle$) {

$(tieneAnteriorYSiguienteCorrecto(k, v, errores) \longrightarrow_L estaCorregido1(k, v, errores)) \vee$
$(tieneAnteriorYNoSiguienteCorrecto(k, v, errores) \longrightarrow_L estaCorregido2(k, v, errores)) \vee$
$(tieneSiguienteYNoAnteriorCorrecto(k, v, errores) \longrightarrow_L estaCorregido3(k, v, errores))$

}

`pred tieneAnteriorYSiguienteCorrecto` (k:$\mathbb{Z}$, v: Viaje, errores: $seq\langle Tiempo\rangle$) {

$(\exists i : \mathbb{Z})(0 \leq i < |v| \wedge_L esElAnteriorCorrecto(v[i], k, v, errores)) \wedge$
$(\exists j : \mathbb{Z})(0 \leq j < |v| \wedge_L esElSiguienteCorrecto(v[j], k, v, errores))$

}

`pred estaCorregido1` (k:$\mathbb{Z}$, v: Viaje, errores: $seq\langle Tiempo\rangle$) {

$(\exists i, j : \mathbb{Z})(0 \leq i < j < |v| \wedge_L (esElAnteriorCorrecto(v[i], k, v, errores) \wedge$

$esElSiguienteCorrecto(v[j], k, v, errores) \wedge dist(v[i], v[k]) = velocidadMedia(v[i], v[j]) \cdot (v[k]_0 - v[i]_0) \wedge$
$dist(v[j], v[k]) = velocidadMedia(v[i], v[j]) \cdot (v[j]_0 - v[k]_0)))$

}

`pred tieneAnteriorYNoSiguienteCorrecto` (k:$\mathbb{Z}$, v: Viaje, errores: $seq\langle Tiempo\rangle$) {

$(\exists i : \mathbb{Z})(0 \leq i < |v| \wedge_L esElAnteriorCorrecto(v[i], k, v, errores)) \wedge$
$\neg(\exists j : \mathbb{Z})(0 \leq j < |v| \wedge_L esElSiguienteCorrecto(v[j], k, v, errores))$

}

`pred estaCorregido2` (k:$\mathbb{Z}$, v: Viaje, errores: $seq\langle Tiempo\rangle$) {

$(\exists i, j : \mathbb{Z})(0 \leq i < j < |v| \wedge_L (esElAnteriorCorrecto(v[j], k, v, errores) \wedge$
$esElAnteriorCorrecto(v[i], j, v, errores) \wedge dist(v[i], v[k]) = velocidadMedia(v[i], v[j]) \cdot (v[k]_0 - v[i]_0) \wedge$
$dist(v[j], v[k]) = velocidadMedia(v[i], v[j]) \cdot (v[j]_0 - v[k]_0)))$

}

- Comentario: En estaCorregido2 el primer llamado al predicado esElAnteriorCorrecto busca el punto anterior correcto del punto a corregir. Una vez encontrado este punto, lo usamos como parámetro en el segundo llamado a esElAnteriorCorrecto para buscar el anterior del anterior.

`pred tieneSiguienteYNoAnteriorCorrecto` (k:$\mathbb{Z}$, v: Viaje, errores: $seq\langle Tiempo\rangle$) {

$\neg(\exists i : \mathbb{Z})(0 \leq i < |v| \wedge_L esElAnteriorCorrecto(v[i], k, v, errores)) \wedge$
$(\exists j : \mathbb{Z})(0 \leq j < |v| \wedge_L esElSiguienteCorrecto(v[j], k, v, errores))$

}

`pred estaCorregido3` (k:$\mathbb{Z}$, v: Viaje, errores: $seq\langle Tiempo\rangle$) {

$(\exists i, j : \mathbb{Z})(0 \leq i < j < |v| \wedge_L (esElSiguienteCorrecto(v[i], k, v, errores) \wedge$
$esElSiguienteCorrecto(v[j], i, v, errores) \wedge dist(v[i], v[k]) = velocidadMedia(v[i], v[j]) \cdot (v[k]_0 - v[i]_0) \wedge$
$dist(v[j], v[k]) = velocidadMedia(v[i], v[j]) \cdot (v[j]_0 - v[k]_0)))$

}

- Comentario: En estaCorregido3 el primer llamado al predicado esElSiguienteCorrecto busca el punto siguiente correcto del punto a corregir. Una vez encontrado este punto, lo usamos como parámetro en el segundo llamado a esElSiguienteCorrecto para buscar el siguiente del siguiente.

`pred esElAnteriorCorrecto` (anterior: Tiempo$\times GPS, k : \mathbb{Z}, v : Viaje, errores : seq\langle Tiempo\rangle$){

$\neg(\exists i : \mathbb{Z})(0 \leq i < |v| \wedge_L (anterior_0 < v[i]_0 < v[k]_0 \wedge \neg aparece(v[i]_0, errores)))$

}

`pred esElSiguienteCorrecto` (siguiente: Tiempo$\times GPS, k : \mathbb{Z}, v : Viaje, errores : seq\langle Tiempo\rangle$){

$\neg(\exists i : \mathbb{Z})(0 \leq i < |v| \wedge_L (v[k]_0 < v[i]_0 < siguiente_0 \wedge \neg aparece(v[i]_0, errores)))$

}

`aux velocidadMedia` (punto1, punto2: Tiempo$\times GPS$) $: \mathbb{R} = \frac{dist(punto1, punto2)}{punto2_0 - punto1_0}$;

}

3.13. Ejercicio 13

`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` $\{(\forall i:\mathbb{Z})(0\leq i<|xs|\longrightarrow_L esViajeValido(xs[i]))\wedge bins>0\}$

`Post` $\{(\exists vels: seq\langle\mathbb{R}\rangle)(esListaDeVelocidadesMaximasOrdenada(vels,xs)$
$\wedge\, esElHistograma(cuentas,vels,bins)\wedge sonLosLimitesDeCadaBin(limites,vels,bins))\}$

`pred esVelocidadMaxima` (vel: $\mathbb{R}$, v: Viaje) {

$(\exists c:Viaje)(enOrden(v,c)\wedge_L(\forall i:\mathbb{Z})(0\leq i<|c|-1\longrightarrow_L vel\geq\frac{dist(c[i+1]_1,c[i]_1)}{|c[i+1]_0-c[i]_0|})\wedge$
$(\exists j:\mathbb{Z})(0\leq j<|c|-1\wedge_L\frac{dist(c[j+1]_1,c[j]_1)}{|c[j+1]_0-c[j]_0|}=vel))$

}

`pred esListaDeVelocidadesMaximasOrdenada` (vels: $seq\langle\mathbb{R}\rangle$, xs : $seq\langle Viaje\rangle$) {

$(|vels|=|xs|)\wedge(\forall i:\mathbb{Z})(0\leq i<|xs|\longrightarrow_L(esVelocidadMaxima(vels[i],xs[i])\wedge estaOrdenada(vels)))$

}

`pred estaOrdenada` (s: $seq\langle\mathbb{R}\rangle$) {

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

}

`pred esElHistograma` (cuentas: $seq\langle\mathbb{Z}\rangle$, vels: $seq\langle\mathbb{R}\rangle$,bins : $\mathbb{Z}$) {

$(\forall i:\mathbb{Z})(0\leq i<|cuentas|-1\longrightarrow_L$
$cuentas[i]=\sum_{j=0}^{|vels|-1}\textsf{if }(intervalo(vels,bins)\cdot i\leq vels[j]<intervalo(vels,bins)\cdot(i+1))\textsf{ then }1\textsf{ else }0\textsf{ fi})\wedge$
$cuentas[|cuentas|-1]=\sum_{j=0}^{|vels|-1}\textsf{if }(intervalo(vels,bins)\cdot(|cuentas|-1)\leq vels[j]\leq intervalo(vels,bins)\cdot$
$(|cuentas|))\textsf{ then }1\textsf{ else }0\textsf{ fi})$

}

`pred sonLosLimitesDeCadaBin` (limites: $seq\langle\mathbb{R}\rangle$,vels: $seq\langle\mathbb{R}\rangle$,bins : $\mathbb{Z}$) {

$|limites|=|bins|+1\wedge velMaxMinEsIgualAlPrimerLimite(vels,limites)\wedge$
$(\forall i:\mathbb{Z})(1\leq i<|limites|\longrightarrow_L limites[i]=intervalo(vels,bins)\cdot i+limites[0])$

}

`aux intervalo` (vels: $seq\langle\mathbb{R}\rangle$,bins : $\mathbb{Z}$) : $\mathbb{R}$ = $\frac{vels[|vels|-1]-vels[0]}{bins}$;

`pred velMaxMinEsIgualAlPrimerLimite` (vels: $seq\langle\mathbb{R}\rangle$, limites: $seq\langle\mathbb{R}\rangle$) {

$(\exists j:\mathbb{Z})(0\leq j<|vels|\wedge_L((\forall i:\mathbb{Z})(0\leq i<|vels|\longrightarrow_L vels[j]\leq vels[i])\wedge vels[j]=limites[0]))$

}

- Comentario: En sonLosLimitesDeCadaBin calculamos cada límite y nos aseguramos que estén ordenados de menor a mayor. En velMaxMinEsIgualAlPrimerLimite nos aseguramos que el primer límite es, en efecto, la velocidad máxima mínima.

}