1. Definición de Tipos

type *Tiempo*= $\mathbb{R}$

type *Dist*= $\mathbb{R}$

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

type *Recorrido*= seq(*GPS*)

type *Viaje*= seq(*Tiempo*$\times$ *GPS*)

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

type *Grilla*= seq(*GPS*$\times$ *GPS*$\times$ *Nombre*)

2. Aclaraciones

En el punto 10 en la función auxiliar llamada "puntoRecorridoEnCelda"se considera que dado un punto GPS del Recorrido y una celda, se toma que dicho punto pertenece a la lista de Nombres de celdas, si está dentro de la misma con sus lados izquierdo y superior inclusive, en cambio los lados derecho y abajo corresponderían a otro nombre de celda. En el punto 13, nuestro enfoque sólo funciona para casos en que la velocidad mínima de algún viaje en la secuencia de los mismos (llamada xs) es 0.

3. Problemas

3.1. Ejercicio 1

```
proc viajeValido (in v : Viaje, out result : Bool) {
    Pre {|v| > 0}
    Post {result = true ↔ (tiempoValido(v) ∧ GPSValidoViaje(v))}
    pred tiempoValido (v : Viaje) {
        (∀i : ℤ) (
            (0 ≤ i < |v|) ⟶_L ((v[i])_0 ≥ 0)
        )
    }
    pred GPSValidoViaje (v : Viaje) {
        (∀i : ℤ) (
            (0 ≤ i < |v|) ⟶_L ((−90 ≤ ((v[i])_1)_0 ≤ 90) ∧ (−180 ≤ ((v[i])_1)_1 ≤ 180))
        )
    }
}
```

3.2. Ejercicio 2

```
proc recorridoValido (in v : Recorrido, out result : Bool) {
    Pre {|v| > 0}
    Post {result = true ↔ esRecorridoValido(v)}
}
```

3.3. Ejercicio 3

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

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

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

pred `puntosEnRadio` $(r : Dist,\ v : Viaje)$ {

$(\exists u : GPS)$ (

$(\forall elem : TiempoxGPS)$ (

$(elem \in v) \longrightarrow_L (dist((elem)_1, u) \leq r)$

)

)

}

}

3.4. Ejercicio 4

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

Pre $\{esViajeValido(v)\}$

Post $\{(\exists idTMin, idTMax : \mathbb{Z})$ (

$sonIdTiempoMinMax(idTMin, idTMax, v) \wedge (t = (v[idTMax])_0 - (v[idTMin])_0)$

$)\}$

}

3.5. Ejercicio 5

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

Pre $\{esViajeValido(v)\}$

Post $\{(\exists v_{ord} : Viaje)$ (

$esPermutacionOrdenada(v_{ord}, v) \wedge (d = sumarDistancias(v_{ord}))$

$)\}$

aux `SumarDistancias` $(v : Viaje) : Dist \ = \sum_{i=0}^{|v|-2} dist((v[i])_1, (v[i+1])_1)$;

}

3.6. Ejercicio 6

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

Pre $\{esViajeValido(v)\}$

Post $\{(\exists v_{ord} : Viaje)$ (

$res = \text{true} \leftrightarrow (esPermutacionOrdenada(v_{ord}, v) \wedge seSuperoVeloc(v_{ord}))$

$)\}$

pred `seSuperoVeloc` $(v : Viaje)$ {

$(\exists i : \mathbb{Z})$ (

$(0 \leq i < |v|) \wedge_L (\frac{dist((v[i])_1, (v[i+1])_1)}{(v[i+1])_0 - (v[i])_0} > 22,23\frac{m}{s})$

)

}

}

3.7. Ejercicio 7

`proc flota` (in V: $Seq < Viaje >$, in t_0, t_f : $Tiempo$, out res : $\mathbb{Z}$) {

`Pre` $\{sonViajesValidos(V) \wedge t_0 \geq 0 \wedge t_f > t_0\}$

`Post` $\{(\exists V_{TMinMax} : Seq < TiempoxTiempo >)$ (

$|V| = |V_{TMinMax}| \wedge esSecuenciaTMinMax(V_{TMinMax}, V) \wedge res = cantQueFlota(V_{TMinMax}, t_0, t_f)$

)}

`pred sonViajesValidos` $(V : seq < Viaje >)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |V|) \longrightarrow_L (esViajeValido(V[i]))$

)

}

`pred esSecuenciaTMinMax` $(V_{TMinMax} : seq < TiempoxTiempo >, V : seq < Viaje >)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |V|) \longrightarrow_L (\exists idTMin, idTMax : \mathbb{Z})$ (

$0 \leq idTMin < |V| \wedge_L 0 \leq idTMax < |V| \wedge sonIdTiempoMinMax(idTMin, idTMax, V[i]) \wedge_L (V_{TMinMax}[i])_0 = (V[i][idTMin])_0 \wedge_L (V_{TMinMax}[i])_1 = (V[i][idTMax])_0$

)

)

}

`aux cantQueFlota` $(V_{MinMax} : seq < TiempoxTiempo >, t_0, t_f : Tiempo) : \mathbb{Z} = \sum_{i=0}^{|V_{MinMax}|-1}$ if $enHorario((V_{MinMax}[i])_0, (V_{MinMax}[i])_1, t_i, t_f)$ then 1 else 0 fi ;

`aux enHorario` $(t_{min},\ t_{max},\ t_i,\ t_f : Tiempo)$: Bool $= ((t_{min} \geq t_i) \wedge (t_{max} \leq t_f))ø((t_{max} > t_i) \wedge (t_{min} < t_i))ø ((t_{min} < t_f) \wedge (t_{max} > t_f))ø((t_{min} < t_i) \wedge (t_{max} > t_f))$;

}

3.8. Ejercicio 8

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

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

`Post` $\{sinRepetidos_{GPS}(res) \wedge (\forall p : GPS)$ (

$p \in res \leftrightarrow p \in r \wedge \neg estaCubierto(p, v, u)$

)}

`pred estaCubierto` $(p : GPS,\ v : Viaje,\ u : Dist)$ {

$(\exists k : \mathbb{Z})$ (

$(0 \leq k < |v|) \wedge_L (dist(v[k], p) < u)$

)

}

pred $sinRepetidos_{GPS}$ $(s : GPS)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |s|) \longrightarrow (\forall j : \mathbb{Z})$ (

$((0 \leq j < |s|) \wedge (i \neq j)) \longrightarrow_L (s[i] \neq s[j])$

)

)

}

}

3.9. Ejercicio 9

proc construirGrilla (in $esq_1 : GPS$, in $esq_2 : GPS$, in $n : \mathbb{Z}$, in $m : \mathbb{Z}$, out $g : Grilla$) {

Pre $\{n > 0 \wedge m > 0 \wedge GPSValido(esq_1) \wedge GPSValido(esq_2) \wedge nombresValidos(g, n, m)\}$

Post $\{|g| = n * m \wedge (\exists AreaGrilla, b_{(1,1)}, h_{(1,1)} : \mathbb{R})$ (

$esAreaGrilla(AreaGrilla, esq_1, esq_2) \wedge sonBaseYalturaPrimerCelda(b_{(1,1)}, h_{(1,1)}, AreaGrilla, g) \wedge todasCeldasRectangulares(b_{(1,1)}, h_{(1,1)}, g)$

)}

pred nombresValidos (g : Grilla, n, m : $\mathbb{Z}$) {

$(\forall elem : Grilla)$ (

$elem \in g \longrightarrow_L (1 \leq ((elem)_2)_0 \leq n) \wedge 1 \leq ((elem)_2)_1 \leq m))$

)

}

pred esAreaGrilla (AreaGrilla : $\mathbb{R}$, esq_1, esq_2 : GPS) {

$AreaGrilla = ((esq_1)_0 - (esq_2)_0) * ((esq_2)_1 - (esq_1)_1)$

}

pred sonBaseYalturaPrimerCelda ($b_{(1,1)}$, $h_{(1,1)}$, AreaGrilla : $\mathbb{R}$, g : $Grilla$) {

$b_{(1,1)} = ((g)_0)_1 - ((g)_0)_0) \wedge (h_{(1,1)} = ((g)_0)_0 - ((g)_0)_1) \wedge (b_{(1,1)} * h_{(1,1)} = \frac{AreaGrilla}{|g|})$

}

pred sonBaseYalturaPrimerCelda ($b_{(1,1)}$, $h_{(1,1)}$, AreaGrilla : $\mathbb{R}$, g : $Grilla$) {

$b_{(1,1)} = ((g[0])_1)_1 - ((g[0])_0)_1 \wedge h_{(1,1)} = ((g[0])_0)_0 - ((g[0])_1)_0) \wedge (b_{(1,1)} * h_{(1,1)} = \frac{AreaGrilla}{|g|})$

}

pred todasCeldasRectangulares ($b_{(1,1)}$, $h_{(1,1)}$: $\mathbb{R}$, g : $Grilla$) {

$(\forall elem : GPSxGPSxNombre)$ (

$(elem \in g) \longrightarrow_L esCeldaRectangular(elem, g[0], b_{(1,1)}, h_{(1,1)})$

)

}

pred esCeldaRectangular (elem, primerCelda : $GPSxGPSxNombre$, $b_{(1,1)}$, $h_{(1,1)}$: $\mathbb{R}$) {

$(\exists n, m : \mathbb{Z})$ (

$(n = ((elem)_2)_0) \wedge (m = ((elem)_2)_1) \wedge (elem)_0 = ((primerCelda)_0 - (n-1) * h_{(1,1)}, (primerCelda)_1 + m * b_{(1,1)}) \wedge (elem)_1 = ((primerCelda)_0 - n * h_{(1,1)}, (primerCelda)_1 + m * b_{(1,1)})$

)

}

}

3.10. Ejercicio 10

`proc regiones` (in r : $Recorrido$, in g : $Grilla$, out res : $seq < Nombre >$) {

Pre $\{esRecorridoValido(r) \wedge esGrillaValida(g)\}$

Post $\{(|res| = |r|) \wedge (\forall i : \mathbb{Z})$ (

$(0 \leq i < |r|) \longrightarrow_L (\exists celda : GPSxGPSxNombre)$ (

$(celda \in_g g) \wedge_L (puntoRecorridoEnCelda(r[i], celda)) \wedge (res[i] = celda)$

)

)}

`pred puntoRecorridoEnCelda` ($p : GPS$, $celda : GPS \times GPS \times Nombre$) {

$((celda)_1)_0 < (p)_0 \leq ((celda)_0)_0 \wedge ((celda)_0)_1 \leq (p)_1 < ((celda)_1)_1$

}

}

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 $\{(\exists v_{ord} : Viaje)$ (

$esPermutacionOrdenada(v, v_{ord}) \wedge (\exists g_{viaje} : Grilla)$ (

$(|g_{viaje}| = |v_{ord}|) \wedge (esSecuenciaDeCeldasDelViaje(v_{ord}, g, g_{viaje}) \wedge (res = contadorDeSaltos(g_{viaje}))$

)

)

}

`pred esSecuenciaDeCeldasDelViaje` (v : $Viaje$, g, g_{viaje} : $Grilla$) {

$(\forall v' : Viaje)$ (

$(v' \in v) \longrightarrow_L ((\exists g' : GPSxGPSxNombre)$ (

$(g' \in_g g) \wedge puntoEnCelda(v', g') \wedge g' \in_g g_{viaje}$

))

)

}

`aux contadorDeSaltos` (g_{viaje} : Grilla) : $\mathbb{Z} = \sum_{i=0}^{|g_{viaje}|-2}$ if $\neg puntoSiguienteEsAledanio(g_{viaje}[i], g_{viaje}[i+1])$ then 1 else 0 fi

`aux puntoSiguienteEsAledanio` (g_i, g_{i+1} : $GPSxGPSxNombre$) : Bool $= (((g_i)_2)_0 - 1 \leq ((g_{i+1})_2)_0 \leq ((g_i)_2)_0 + 1)$
$\wedge (((g_i)_2)_1 - 1 \leq ((g_{i+1})_2)_1 \leq ((g_i)_2)_1 + 1)$;

`aux puntoEnCelda` (v': Tiempo$\times GPS$, g':GPSxGPSxNombre) : Bool $= (((g')_0)_0 \leq ((v')_1)_0 \leq ((g')_1)_0)$
$\wedge (((g')_1)_1 \leq ((v')_1)_1 \leq ((g')_0)_1)$;

}

3.12. Ejercicio 12

`proc corregirViaje` (inout v: $Viaje$, in $errores$: $seq < Tiempo >$) {

Pre $\{esViajeValido(v) \wedge |v| \geq 5 \wedge v = v_0 \wedge erroresValidos(errores, v)\}$

Post $\{(\exists v_{0ord} : Viaje)$ (

$|v_{0ord}| = |v_0| \wedge esPermutacionOrdenada(v_{0ord}, v_0) \wedge esViajeCorregido(v, v_{0ord})$

)}

`pred esViajeCorregido` (v, v_0 : $Viaje$, $errores$: $seq < Tiempo >$) {

$(\forall i : \mathbb{Z})$ (

$0 \leq i < |v| \wedge_L (v[i])_0 \in_t errores \longrightarrow_L (\exists elem_1, elem_2 : TiempoxGPS)$ (

$sonCorrectosProximos(v_0, v[i], elem_1, elem_2, errores) \wedge (\exists elem_{corregido} : TiempoxGPS)$ (

$esCorregido(elem_1, elem_2, elem_{corregido}) \wedge (v[i] = elem_{corregido})$

)

)

) $\wedge$ $(\forall i : \mathbb{Z})$ (

$0 \leq i < |v| \wedge_L (v[i])_0 \notin_t errores \longrightarrow_L v[i] = v_0[i]$

)

}

`aux esCorregido` (v_1, v_2, $v_{corregido}$: $TiempoxGPS$) : Bool $= ((v_{corregido})_1 = (v_1)_1 + (\frac{(v_2)_1 - (v_1)_1}{(v_2)_0 - (v_1)_0})) * ((v_{corregido})_0 - (v_1)_0))$;

`pred sonCorrectosProximos` (v : $Viaje$, v', v_1, v_2 : $TiempoxGPS$) {

$(\forall v'' : TiempoxGPS)$ (

$v'' \in_v v_0 \wedge \neg((v'')_0 \in_v errores) \longrightarrow_L esDeltaTmin(v', v_1, v'') \wedge esDeltaTminSiguiente(v', v_1, v_2, v'')$

)

}

`aux esDeltaTmin` (v', v_1, v'' : $TiempoxGPS$) : Bool $= |(v_1)_0 - (v')_0| < |(v'')_0 - (v'_0)|$;

`aux esDeltaTminSiguiente` (v', v_1, v_2, v'' : $TiempoxGPS$) : Bool $= (|(v_2)_0 - (v')_0| < |(v'')_0 - (v'_0)|) \wedge (v_2 \neq v_1)$;

`pred erroresValidos` (errores: $seq < Tiempo >$, v: Viaje) {

$|errores| \geq \frac{|v|}{10} \wedge (\forall i : \mathbb{Z})$ (

$0 \leq i < |errores| \longrightarrow_L errores[i] \geq 0$

)

}

}

3.13. Ejercicio 13

`proc histograma` (in xs: $seq < Viaje >$, in $bins$: $\mathbb{Z}$, out $Cuentas$: $seq < Z >$, out $Seq < R >$) {

`Pre` $\{sonViajesValido(xs) \wedge bins \geq 0\}$

`Post` $\{(\exists v_{Maxs} : seq < R >)$ (

$(|v_{Maxs}| = |xs| \wedge esSecuenciaDeVMax(V_{Maxs}, xs) \wedge (\exists idVmax, idVmin : \mathbb{Z})$ (

$(0 \leq IdVmax < |v_{Maxs}|) \wedge (0 \leq idVmin < |v_{Maxs}|) \wedge esIdVmax(idVmax, v_{Maxs}) \wedge esIdVmin(idVmin, v_{Maxs}) \wedge (\exists step : \Re)$ (

$(step = \frac{v_{Maxs}[idVmax] - v_{Maxs}[idVmin]}{bins}) \wedge (\exists s_{tachos} : Seq < \mathbb{Z} >)$ (

$(|s_{tachos}| = |bins|) \wedge esSecuenciaDeTachos(s_{tachos}, v_{Maxs}) \wedge cuentas = s_{tachos} \wedge (\exists s_{steps} : seq < \Re >)$ (

$(|s_{steps}| = bins + 1) \wedge esSecuenciaDeSteps(s_{steps}, step) \wedge limites = s_{steps}$

)

)

)

))

)}

`pred esSecuenciaDeVMax` $(v_{Maxs} : seq < R >, xs : seq < Viaje >)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |xs|) \longrightarrow_L (\exists idDmax : \mathbb{Z})$ (

$esIndiceDistMax(idDMax, xs[i]) \wedge (\exists v_{max} : \Re)$ (

$esVmaxDelViaje(xs[i], idDmax, v_{max}) \wedge v_{max} \in_R v_{Maxs}$

)

)

)

}

`pred esIndiceDistMax` $(idDmax : \mathbb{Z}, v : Viaje)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |v| - 1) \longrightarrow_L dist((v[idDmax])_1, (v[idDmax + 1])_1) \geq dist((v[i])_1, (v[i + 1])_1)$

)

}

`aux esVmaxDelViaje` $(v : Viaje, idDmax : \mathbb{Z}, v_{max} : R)$: Bool $= (v_{max} = \frac{(v[idDmax+1])_1 - v[idDmax])_1}{(v[idDmax+1])_0 - v[idDmax])_0})$;

`pred esIndiceDistMax` $(idDmax : \mathbb{Z}, v : Viaje)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |v| - 1) \longrightarrow_L dist((v[idDmax])_1, (v[idDmax + 1])_1) \geq dist((v[i])_1, (v[i + 1])_1)$

)

}

`pred esIndiceVMin` $(idVmin : \mathbb{Z}, v_{Maxs} : seq < R >)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |v_{Maxs}|) \longrightarrow_L (v_{Maxs}[idVmin] \leq v_{Maxs}[i])$

)

}

`pred esIndiceVMax` $(idVmax : \mathbb{Z}, v_{Maxs} : seq < R >)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |v_{Maxs}|) \longrightarrow_L (v_{Maxs}[idVmax] \geq v_{Maxs}[i])$

)

}

`pred esSecuenciaDeTachos` $(s_{tachos} : seq < Z >, v_{Maxs} : seq < R >, step : R)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |s_{tachos}|) \longrightarrow_L (s_{tachos}[i] = contarSiEnRango(v_{Maxs}, i, step))$

)

}

`aux contarSiEnRango` $(v_{Maxs} : seq < R >, i : Z, step : R) : \mathbb{Z} = \sum_{j=0}^{|v_{Maxs}|-1}$ if $i * step \leq v_{Maxs}[j] \leq (i + 1) * step$ then 1 else 0 fi;

`pred esSecuenciaDeSteps` $(s_{steps} : seq < R >, step : R)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |s_{steps}|) \longrightarrow_L (s_{steps}[i] = i * step)$

)

}

}

4. Auxiliares

`pred esViajeValido` $(v : Viaje)$ {

$(\forall elem : TiempoxGPS)$ (

$(elem \in v) \longrightarrow_L (((elem)_0 \geq 0) \wedge (-90 \leq ((elem)_1)_0) \leq 90) \wedge (-180 \leq ((elem)_1)_1) \leq 180)$

)

}

`pred` $\in$ $(elem : TiempoxGPS,\ v\text{: } Viaje)$ {

$(\exists i : \mathbb{Z})$ (

$(0 \leq i < |v|) \wedge_L (v[i] = elem)$

)

}

`pred sonIdTiempoMinMax` $(idTMin,\ idTMax : \mathbb{Z},\ v : Viaje)$ {

$esIdTiempoMin(idTMin, v) \wedge esIdTiempoMax(idTMax, v)$

}

`pred esIdTiempoMin` $(idTMin : \mathbb{Z},\ v : Viaje)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |v|) \longrightarrow_L ((v[idTMin])_0 \leq (v[i])_0)$

)

}

`pred esIdTiempoMax` $(idTMax : \mathbb{Z},\ v : Viaje)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |v|) \longrightarrow_L ((v[idTMax])_0 \geq (v[i])_0)$

)

}

`aux Apariciones` $(e : Seq < TiempoxGPS >,\ v : Viaje) : \mathbb{Z}\ = \sum_{i=0}^{|v|-1}$ if $v[i] = e$ then 1 else 0 fi ;

`pred esPermutacion` $(v_1,\ v_2 : Viaje)$ {

$(\forall e : Seq < TiempoxGPS >)$ (

$Apariciones(e, v_1) = Apariciones(e, v_2)$

)

}

`pred esRecorridoValido` $(v : Recorrido)$ {

$(\forall i : \mathbb{Z})$ (

$(0 \leq i < |v|) \longrightarrow_L ((-90 \leq (v[i])_0 \leq 90) \wedge (-180 \leq (v[i])_1 \leq 180))$

)

}

`pred estaOrdenada` $(v : Viaje)$ {

$(\forall i : \mathbb{Z})$ (

$(1 \leq i < |v|) \longrightarrow_L ((v[i-1])_0 < (v[i])_0)$

)

}

`pred esPermutacionOrdenada` $(v_1,\, v_2 : Viaje)$ {

$esPermutacion(v_1, v_2) \land estaOrdenada(v_1)$

}

`pred esGrillaValida` $(g : Grilla)$ {

$(\exists n_{max}, m_{max} : \mathbb{Z})$ (

$esFilaMax(n_{max}) \land esColumnaMax(m_{max}) \land |g| = (n_{max} * m_{max}) \land (\exists esq1, esq2 : GPS)$ (

$nombresValidos(g, n_{max}, m_{max}) \land GPSValido(esq_1) \land GPSValido(esq_2) \land (\exists AreaGrilla, b_{(1,1)}, h_{(1,1)} : \mathbb{R})$ (

$esAreaGrilla(AreaGrilla, esq_1, esq_2) \quad \land \quad sonBaseYalturaPrimerCelda(b_{(1,1)}, h_{(1,1)}, AreaGrilla, g) \quad \land$
$todasCeldasRectangulares(b_{(1,1)}, h_{(1,1)}, g)$

)

)

)

}

`pred esFilaMax` (g : Grilla, n_{max} : $\mathbb{Z}$) {

$(\exists elem : Grilla)$ (

$elem \in g \land ((elem)_2)_0 = n_{max} \land (\forall i : \mathbb{Z})$ (

$0 \leq i < |g| \longrightarrow_L n_{max} \geq ((g[i])_2)_0$

)

)

}

`pred esColumnaMax` (g : Grilla, m_{max} : $\mathbb{Z}$) {

$(\exists elem : Grilla)$ (

$elem \in g \land ((elem)_2)_1 = m_{max} \land (\forall i : \mathbb{Z})$ (

$0 \leq i < |g| \longrightarrow_L m_{max} \geq ((g[i])_2)_1$

)

)

}