Ejercicios Práctico 3

1)a) $psum.xs = \langle \forall i : 0 \leq i < \#xs : sum(xs \uparrow i) \geq 0 \rangle$ Caso base: $xs = []$

$$psum.[]$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle \forall i : 0 \leq i < \#[] : sum([] \uparrow i) \geq 0 \rangle$$

$$\equiv \{\text{Def de \#, evaluo rango, rango vacio}\}$$

$$True$$

Caso recursivo: $xs = (y : ys)$

$$HI = psum.ys = \langle \forall i : 0 \leq i < \#ys : sum(ys \uparrow i) \geq 0 \rangle$$

psum.(y:ys)

$$\equiv \{\text{Especificacion}\}$$

$$\langle \forall i : 0 \leq i < \#(y : ys) : sum((y : ys) \uparrow i) \geq 0 \rangle$$

$$\equiv \{\text{Aritmetica, particion de rango, rango unitario}\}$$

$$sum((y : ys) \uparrow 0) \geq 0 \wedge \langle \forall i : 0 \leq i < \#(y : ys) : sum((y : ys) \uparrow i) \geq 0 \rangle$$

$$\equiv \{\text{Def } \uparrow\text{, def de sum y aritmetica}\}$$

$$True \wedge \langle \forall i : 1 \leq i < \#(y : ys) : sum((y : ys) \uparrow i) \geq 0 \rangle$$

$$\equiv \{\text{Def \#, Cambio de variable f.x=x+1, aritmetica}\}$$

$$True \wedge \langle \forall i : 0 \leq i < \#ys : sum((y : ys) \uparrow i + 1) \geq 0 \rangle$$

$$\equiv \{\text{Def de } \uparrow\}$$

$$True \wedge \langle \forall i : 0 \leq i < \#ys : sum(y : (ys \uparrow i)) \geq 0 \rangle$$

$$\equiv \{\text{Def de sum}\}$$

$$True \wedge \langle \forall i : 0 \leq i < \#ys : y + sum((ys \uparrow i)) \geq 0 \rangle$$

Como no se puede aplicar la HI procedemos a realizar una generalizacion por abstraccion:

$gpsum.n.xs = \langle \forall i : 0 \leq i < \#xs : n + sum((xs \uparrow i)) \geq 0 \rangle$ Caso base: $xs = []$
gpsum.n.[]

$$\equiv \{\text{Especificacion}\}$$

$$\langle \forall i : 0 \leq i < \#[] : n + sum(([] \uparrow i)) \geq 0 \rangle$$

$$\equiv \{\text{Def \#, evaluo rango, rango vacio}\}$$

$$True$$

Caso recursivo: $xs = (y : ys)$

$$HI = gpsum.n.ys = \langle \forall i : 0 \leq i < \#ys : n + sum((ys \uparrow i)) \geq 0 \rangle \forall n$$

$$gpsum.n.(y : ys)$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \forall i : 0 \leq i < \#(y : ys) : n + sum(((y : ys) \uparrow i)) \geq 0 \rangle$$
$$\equiv \{\text{Aritmetica, particion de rango, rango unitario}\}$$
$$n + sum(((y : ys) \uparrow 0)) \geq 0 \wedge \langle \forall i : 1 \leq i < \#(y : ys) : n + sum(((y : ys) \uparrow i)) \geq 0 \rangle$$
$$\equiv \{\text{Def de } \uparrow\text{, def de sum}\}$$
$$n + 0 \geq 0 \wedge \langle \forall i : 1 \leq i < \#(y : ys) : n + sum(((y : ys) \uparrow i)) \geq 0 \rangle$$
$$\equiv \{\text{def de \#, cambio de variable f.x=x+1, aritmetica}\}$$
$$n \geq 0 \wedge \langle \forall i : 0 \leq i < \#ys : n + sum(((y : ys) \uparrow i + 1)) \geq 0 \rangle$$
$$\equiv \{\text{Def de } \uparrow\}$$
$$n \geq 0 \wedge \langle \forall i : 0 \leq i < \#ys : n + sum((y : (ys \uparrow i)) \geq 0 \rangle$$
$$\equiv \{\text{Def de sum}\}$$
$$n \geq 0 \wedge \langle \forall i : 0 \leq i < \#ys : n + y + sum(ys \uparrow i) \geq 0 \rangle$$
$$\equiv \{\text{HI}\}$$
$$n \geq 0 \wedge gpsum.(n + y).ys$$

El resultado final de la derivacion es:

```
gpsum.n.[] = True
gpsum.n.(y:ys) = n >= 0 && gpsum.(n+y).ys

psum.(y:ys) = gpsum.0.(y:ys)
```

Ahora verifiquemos que $psum.(y : ys) = gpsum.0.(y : ys)$:

$$gpsum.0.(y : ys)$$
$$\equiv \{\text{Especificacion gpsum}\}$$
$$\langle \forall i : 0 \leq i < \#(y : ys) : 0 + sum(((y : ys) \uparrow i)) \geq 0 \rangle \forall n$$
$$\equiv \{\text{Aritmetica}\}$$
$$\langle \forall i : 0 \leq i < \#(y : ys) : sum(((y : ys) \uparrow i)) \geq 0 \rangle \forall n$$
$$\equiv \{\text{Especificacion psum}\}$$
$$gpsum.(y : ys)$$

b) $sumAnt.xs = \langle \exists i : 0 \leq i < \#xs : xs \,!!\, i = sum.(xs \uparrow i) \rangle$

Caso base: $xs = []$

$$sumAnt.[]$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists i : 0 \leq i < \#[] : [] \,!!\, i = sum.([] \uparrow i) \rangle$$
$$\equiv \{\text{Def de \#, evaluo rango, rango vacio}\}$$
$$False$$

Caso recursivo: $xs = (y : ys)$

$$HI = sumAnt.ys = \langle \exists i : 0 \leq i < \#ys : ys \,!!\, i = sum.(ys \uparrow i) \rangle$$

$$sumAnt.(y : ys)$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists i : 0 \leq i < \#(y : ys) : (y : ys) \,!!\, i = sum.((y : ys) \uparrow i) \rangle$$
$$\equiv \{\text{Aritmetica, particion de rango, rango unitario}\}$$
$$(y : ys) \,!!\, 0 = sum.((y : ys) \uparrow 0) \vee \langle \exists i : 1 \leq i < \#(y : ys) : (y : ys) \,!!\, i = sum.((y : ys) \uparrow i) \rangle$$
$$\equiv \{\text{Def de !! y def de } \uparrow\}$$
$$y = sum.([]) \vee \langle \exists i : 1 \leq i < \#(y : ys) : (y : ys) \,!!\, i = sum.((y : ys) \uparrow i) \rangle$$
$$\equiv \{\text{Def de sum}\}$$
$$y = 0 \vee \langle \exists i : 1 \leq i < \#(y : ys) : (y : ys) \,!!\, i = sum.((y : ys) \uparrow i) \rangle$$
$$\equiv \{\text{Def de \#, cambio de variable f.x=x+1, aritmetica}\}$$
$$y = 0 \vee \langle \exists i : 0 \leq i < \#ys : (y : ys) \,!!\, i + 1 = sum.((y : ys) \uparrow i + 1) \rangle$$
$$\equiv \{\text{Def de !!, def de } \uparrow \text{ y Def de sum}\}$$
$$y = 0 \vee \langle \exists i : 0 \leq i < \#ys : ys \,!!\, i = y + sum.(ys \uparrow i) \rangle$$

Como no se puede aplicar la HI procedemos a realizar una generalizacion por abstraccion:

$gSumAnt.n.xs = \langle \exists i : 0 \leq i < \#xs : xs \,!!\, i = n + sum.(xs \uparrow i) \rangle$

Caso base: $xs = []$

$$gSumAnt.n.[]$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists i : 0 \leq i < \#[] : [] \,!!\, i = n + sum.([] \uparrow i) \rangle$$
$$\equiv \{\text{Def de \#, evaluo rango, rango vacio}\}$$

$$False$$

Caso recursivo: $xs = (y : ys)$

$$HI = gSumAnt.n.ys = \langle \exists i : 0 \leq i < \#ys : ys \ !! \ i = n + sum.(ys \uparrow i) \rangle \ \forall n$$

$$gSumAnt.n.(y : ys)$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists i : 0 \leq i < \#(y : ys) : (y : ys) \ !! \ i = n + sum.((y : ys) \uparrow i) \rangle \ \forall n$$
$$\equiv \{\text{Aritmetica, particion de rango, rango unitario}\}$$
$$(y : ys) \ !! \ 0 = n + sum.((y : ys) \uparrow 0) \vee \langle \exists i : 1 \leq i < \#(y : ys) : (y : ys) \ !! \ i = n + sum.((y : ys) \uparrow i) \rangle$$
$$\equiv \{\text{Def de !! y def de } \uparrow\}$$
$$y = n + sum.([]) \vee \langle \exists i : 1 \leq i < \#(y : ys) : (y : ys) \ !! \ i = n + sum.((y : ys) \uparrow i) \rangle$$
$$\equiv \{\text{Def de sum}\}$$
$$y = n \vee \langle \exists i : 1 \leq i < \#(y : ys) : (y : ys) \ !! \ i = n + sum.((y : ys) \uparrow i) \rangle$$
$$\equiv \{\text{Def de \#, cambio de variable f.x=x+1, aritmetica}\}$$
$$y = n \vee \langle \exists i : 0 \leq i < \#ys : (y : ys) \ !! \ i + 1 = n + sum.((y : ys) \uparrow i + 1) \rangle$$
$$\equiv \{\text{Def de !!, def de } \uparrow \text{ y Def de sum}\}$$
$$y = n \vee \langle \exists i : 0 \leq i < \#ys : ys \ !! \ i = y + n + sum.(ys \uparrow i) \rangle$$
$$\equiv \{\text{HI}\}$$
$$y = n \vee gSumAnt.(y + n).ys$$

Resultado de la derivacion:

```
gSumAnt.n.[] = False
gSumAnt.n.(y:ys) = y = n || gSumAnt.(y+n).ys

sumAnt.ys = gSumAnt.0.ys
```

Ahora verifiquemos que $sumAnt.ys = gSumAnt.0.ys$

$$gSumAnt.0.ys$$
$$\equiv \{\text{Especificacion gSumAnt}\}$$
$$\langle \exists i : 0 \leq i < \#ys : ys \ !! \ i = 0 + sum.(ys \uparrow i) \rangle$$
$$\equiv \{\text{Aritmetica}\}$$
$$\langle \exists i : 0 \leq i < \#ys : ys \ !! \ i = sum.(ys \uparrow i) \rangle$$

$$\equiv \{\text{Especificacion sumAnt}\}$$

$$sumAnt.ys$$

2)a) $esCuad.n = \langle \exists i : 0 \leq i \leq n : i * i = n \rangle$

Caso base: $n = 0$

$$esCuad.0$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle \exists i : 0 \leq i \leq 0 : i * i = n \rangle$$

$$\equiv \{\text{Evaluo rango, rango unitario}\}$$

$$True$$

Caso recursivo: $n = k + 1$

$$HI = esCuad.k = \langle \exists i : 0 \leq i \leq k : i * i = k \rangle$$

$$esCuad.(k + 1)$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle \exists i : 0 \leq i \leq (k + 1) : i * i = (k + 1) \rangle$$

$$\equiv \{\text{Aritmetica, particion de rango, rango unitario}\}$$

$$\equiv \{\text{Magia negra y metafisica}\}$$

$$\langle \exists i : 0 \leq i < k : i * i = k + 1 \rangle \vee (k + 1) * (k + 1) = k + 1$$

Como no se puede aplicar HI procedemos a generalizar:

$gEsCuad.n.m = \langle \exists i : 0 \leq i \leq n : i * i = (n + m) \rangle$

Caso base: $n = 0$

$$gEsCuad.0.m$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle \exists i : 0 \leq i \leq 0 : i * i = (0 + m) \rangle$$

$$\equiv \{\text{Evaluo rango, rango unitario y aritmetica}\}$$

$$0 = m$$

Caso recursivo: $n = (k + 1)$

$$HI = gEsCuad.k.m = \langle \exists i : 0 \leq i \leq k : i * i = k + m \rangle \ \forall m$$

$$gEsCuad.(k + 1).m$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle \exists i : 0 \leq i \leq (k+1) : i * i = (k+1) + m \rangle$$
$$\equiv \{\text{Aritmetica, particion de rango, rango unitario}\}$$
$$(k+1) * (k+1) = k+1+m \vee \langle \exists i : 0 \leq i \leq k : i * i = (k+1) + m \rangle$$
$$\equiv \{\text{asociatividad}\}$$
$$(k+1) * (k+1) = (k+1) + m \vee \langle \exists i : 0 \leq i \leq k : i * i = k + (m+1) \rangle$$
$$\equiv \{\text{HI con m:=m+1}\}$$
$$(k+1) * (k+1) = (k+1) + m \vee gEsCuad.k.(m+1)$$

Resultado final de la derivacion:

```
gEsCuad.0.m = (m == 0)
gEsCuad.(k+1).m = (k+1) * (k+1) = (k+1)+m || gEsCuad.k.(m+1)

esCuad.k = gEsCuad.k.0
```

Ahora verifiquemos que $esCuad.k = gEsCuad.k.0$:

$$gEsCuad.k.0$$
$$\equiv \{\text{Especificacion gEsCuad}\}$$
$$\langle \exists i : 0 \leq i \leq k : i * i = (k+0) \rangle$$
$$\equiv \{\text{Aritmetica}\}$$
$$\langle \exists i : 0 \leq i \leq k : i * i = k \rangle$$
$$\equiv \{\text{Especificacion esCuad}\}$$
$$esCuad.k$$

b) $sumanOcho.xs = \langle \exists as, bs : xs = as + +bs : sum.as = 8 \rangle$

Caso base: $xs = []$

$$sumanOcho.[]$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as, bs : [] = as + +bs : sum.as = 8 \rangle$$
$$\equiv \{\text{Igualdad de listas}\}$$
$$\langle \exists as, bs : as = [] \wedge bs = [] : sum.as = 8 \rangle$$
$$\equiv \{\text{Anidado}\}$$
$$\langle \exists as : as = [] : \langle \exists bs : bs = [] : sum.as = 8 \rangle \rangle$$
$$\equiv \{\text{Termino constante}\}$$
$$\langle \exists as : as = [] : sum.as = 8 \rangle$$

$$\equiv \{\text{Rango unitario}\}$$

$$sum.[] = 8$$

$$\equiv \{\text{Def de sum}\}$$

$$0 = 8$$

$$\equiv \{\text{Logica}\}$$

$$False$$

Caso recursivo: $xs = (y : ys)$

$$sumanOcho.(y : ys)$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle N\ as, bs : (y : ys) = as + +bs : sum.as = 8\rangle$$

$$\equiv \{\text{Neutro } \wedge\}$$

$$\langle N\ as, bs : (y : ys) = as + +bs \wedge True : sum.as = 8\rangle$$

$$\equiv \{\text{Tercero excluido}\}$$

$$\langle N\ as, bs : (y : ys) = as + +bs \wedge (as = [] \vee as \neq []) : sum.as = 8\rangle$$

$$\equiv \{\text{Distributividad}\}$$

$$\langle N\, as, bs : ((y : ys) = as{+}{+}bs{\wedge}as = []){\vee}((y : ys) = as{+}{+}bs{\wedge}as \neq []) : sum.as = 8\rangle$$

$$\equiv \{\text{Particion de rango}\}$$

$$\langle N\, as, bs : (y : ys) = as{+}{+}bs{\wedge}as = [] : sum.as = 8\rangle{+}\langle N\, as, bs : (y : ys) = as{+}{+}bs{\wedge}as \neq [] : sum.as = 8\rangle$$

$$\equiv \{\text{Cambio de variable f.as = (a:as)}\}$$

$$\langle N\, as, bs : (y : ys) = as{+}{+}bs{\wedge}as = [] : sum.as = 8\rangle{+}\langle N\, as, bs : (y : ys) = (a : as){+}{+}bs{\wedge}(a : as) \neq [] : sum.(a :$$

$$\equiv \{\text{Def de ++ y propiedad de listas}\}$$

$$\langle N\, as, bs : (y : ys) = as{+}{+}bs{\wedge}as = [] : sum.as = 8\rangle{+}\langle N\, as, bs : (y : ys) = a : (as{+}{+}bs){\wedge}True : sum.(a : as) =$$

$$\equiv \{\text{Elemento neutro conjuncion y propiedad de listas}\}$$

$$\langle N\, as, bs : (y : ys) = as{+}{+}bs{\wedge}as = [] : sum.as = 8\rangle{+}\langle N\, as, bs : (y = a){\wedge}ys = as{+}{+}bs : sum.(a : as) = 8\rangle$$

$$\equiv \{\text{Eliminacion de variable}\}$$

$$\langle N\, as, bs : (y : ys) = as{+}{+}bs{\wedge}as = [] : sum.as = 8\rangle{+}\langle N\, as, bs : ys = as{+}{+}bs : sum.(y : as) = 8\rangle$$

$$\equiv \{\text{Def de sum}\}$$

$$\langle N\, as, bs : (y : ys) = as{+}{+}bs{\wedge}as = [] : sum.as = 8\rangle{+}\langle N\, as, bs : ys = as{+}{+}bs : y{+}sum.(as) = 8\rangle$$

$$\equiv \{\text{Aritmetica}\}$$

$$\langle N\, as, bs : (y : ys) = as{+}{+}bs{\wedge}as = [] : sum.as = 8\rangle{+}\langle N\, as, bs : ys = as{+}{+}bs : sum.(as) = 8{-}y\rangle$$

Como no se puede aplicar HI procedemos a realizar una generalizacion por abstraccion: $gSumanOcho.xs.n = \langle N\ as, bs : xs = as + +bs : sum.as = n\rangle$

Caso base: $xs = []$

$$gSumanOcho.[].n$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle N\ as, bs : [] = as + +bs : sum.as = n\rangle$$
$$\equiv \{\text{Propiedad de listas}\}$$
$$\langle N\ as, bs : [] = as \wedge [] = bs : sum.as = n\rangle$$
$$\equiv \{\text{Anidado}\}$$
$$\langle N\ as : [] = as : \langle N\ bs : [] = bs : sum.as = n\rangle\rangle$$
$$\equiv \{\text{Termino constante}\}$$
$$\langle N\ as : [] = as : sum.as = n\rangle$$
$$\equiv \{\text{Rango unitario}\}$$
$$sum.[] = n$$
$$\equiv \{\text{Def de sum}\}$$
$$0 = n$$

Caso recursivo: $xs = (y : ys)$ $\$HI = gSumanOcho.ys.n = \langle N\ as, bs : ys = as + +bs : sum.as = n\rangle \forall n$

$$gSumanOcho.(y : ys).n$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle N\ as, bs : (y : ys) = as + +bs : sum.as = n\rangle$$
$$\equiv \{\text{Neutro } \wedge\}$$
$$\langle N\ as, bs : (y : ys) = as + +bs \wedge True : sum.as = n\rangle$$
$$\equiv \{\text{Tercero excluido}\}$$
$$\langle N\ as, bs : (y : ys) = as + +bs \wedge (as = [] \vee as \neq []) : sum.as = n\rangle$$
$$\equiv \{\text{Distributividad}\}$$
$$\langle N\ as, bs : ((y : ys) = as{+}{+}bs \wedge as = []) \vee ((y : ys) = as{+}{+}bs \wedge as \neq []) : sum.as = n\rangle$$
$$\equiv \{\text{Particion de rango}\}$$
$$\langle N\ as, bs : (y : ys) = as{+}{+}bs \wedge as = [] : sum.as = n\rangle + \langle N\ as, bs : (y : ys) = as{+}{+}bs \wedge as \neq [] : sum.as = n\rangle$$
$$\equiv \{\text{Cambio de variable f.as = (a:as)}\}$$
$$\langle N\ as, bs : (y : ys) = as{+}{+}bs \wedge as = [] : sum.as = n\rangle + \langle N\ as, bs : (y : ys) = (a : as){+}{+}bs \wedge (a : as) \neq [] : sum.(a$$

$$\equiv \{\text{Def de ++ y propiedad de listas}\}$$

$$\langle N\ as, bs : (y : ys) = as{+\!\!+}bs \wedge as = [\,] : sum.as = n\rangle + \langle N\ as, bs : (y : ys) = a : (as{+\!\!+}bs) \wedge True : sum.(a : as) =$$

$$\equiv \{\text{Elemento neutro conjuncion y propiedad de listas}\}$$

$$\langle N\ as, bs : (y : ys) = as{+\!\!+}bs \wedge as = [\,] : sum.as = n\rangle + \langle N\ as, bs : (y = a) \wedge ys = as{+\!\!+}bs : sum.(a : as) = n\rangle$$

$$\equiv \{\text{Eliminacion de variable}\}$$

$$\langle N\ as, bs : (y : ys) = as{+\!\!+}bs \wedge as = [\,] : sum.as = n\rangle + \langle N\ as, bs : ys = as{+\!\!+}bs : sum.(y : as) = n\rangle$$

$$\equiv \{\text{Def de sum}\}$$

$$\langle N\ as, bs : (y : ys) = as{+\!\!+}bs \wedge as = [\,] : sum.as = n\rangle + \langle N\ as, bs : ys = as{+\!\!+}bs : y + sum.(as) = n\rangle$$

$$\equiv \{\text{Aritmetica}\}$$

$$\langle N\ as, bs : (y : ys) = as{+\!\!+}bs \wedge as = [\,] : sum.as = n\rangle + \langle N\ as, bs : ys = as{+\!\!+}bs : sum.(as) = n - y\rangle$$

$$\equiv \{\text{HI}\}$$

$$\langle N\ as, bs : (y : ys) = as{+\!\!+}bs \wedge as = [\,] : sum.as = n\rangle + gSumanOcho.ys.(n - y)$$

$$\equiv \{\text{Eliminacion de variable}\}$$

$$\langle N\ as, bs : (y : ys) = [\,] + +bs : sum.[\,] = n\rangle + gSumanOcho.ys.(n - y)$$

$$\equiv \{\text{Def de ++ y de sum}\}$$

$$\langle N\ as, bs : (y : ys) = bs : 0 = n\rangle + gSumanOcho.ys.(n - y)$$

$$\equiv \{\text{Rango unitario}\}$$

$$(n = 0 \rightarrow 1 \square n \neq 0 \rightarrow 0) + gSumanOcho.ys.(n - y)$$

$$\equiv \{\text{Meto la suma dentro dle analisis por casos}\}$$

$$(n = 0 \rightarrow 1 + gSumanOcho.ys.(n - y) \square n \neq gSumanOcho.ys.(n - y) \rightarrow 0)$$

Resultado final:

```
gSumanOcho.(x:xs).n
  | n == 0 = 1 + gSumanOcho.xs.n
  | n != 0 = gSumanOcho.xs.n

sumanOcho.xs = gSumanOcho.xs.8
```

Ahora verifiquemos que $sumanOcho.xs = gSumanOcho.xs.8$:

$$gSumanOcho.xs.8$$

$$\equiv \{\text{Especificacion gSumanOcho}\}$$

$$gSumanOcho.xs.n = \langle N\ as, bs : xs = as + +bs : sum.as = 8\rangle$$

$$\equiv \{\text{Especificacion sumanOcho}\}$$

$$sumanOcho$$

3)a) La suma de todos los elementos de cada prefijo de xs es mayor o igual a 0. b) La menor suma de un segmento intermedio de xs c) La cantidad de sufijos de xs en los que el primer elemento es mayor a la suma de los demas elementos. d) El sufijo más grande que pertenece a dos listas.

Evaluacion manual: a)

$$\langle \forall as, bs : [9, -5, 1, -3] = as ++ bs : sum.as \geq 0 \rangle$$

$$\equiv \{\text{Evaluo rango}\}$$

$$\langle \forall as, bs :$$

$$(as, bs) \in ([9, -5, 1, -3], []), ([9, -5, 1], [-3]), ([9, -5], [1, -3]), ([9], [-5, 1, -3]), ([], [9, -5, 1, -3]),$$

$$: sum.as \geq 0 \rangle$$

$$\equiv \{\text{Evaluo rango en el termino}\}$$

$$sum.[9, -5, 1, -3] \geq 0 \wedge sum.[9, -5, 1] \geq 0 \wedge sum.[9, -5] \geq 0 \wedge sum.[9] \geq 0$$

$$\equiv \{\text{Def de sum}\}$$

$$2 \geq 0 \wedge 5 \geq 0 \wedge 4 \geq 0 \wedge 9 \geq 0$$

$$\equiv \{\text{Aritmetica}\}$$

$$True \wedge True \wedge True \wedge True$$

$$\equiv \{\text{Idempotencia conjuncion}\}$$

$$True$$

b) Rango:

as	bs	cs	Termino	Evaluacion
[9,-5,1,-3]	[]	[]	sum.[]	0
[9,-5,1]	[-3]	[]	sum.[-3]	-3
[9,-5]	[1,-3]	[]	sum.[1,-3]	-2
[9]	[-5,1,-3]	[]	sum.[-5,1,-3]	-7
[]	[9,-5,1,-3]	[]	sum.[9,-5,1,-3]	2
[]	[9,-5,1]	[-3]	sum.[9,-5,1]	5
[]	[9,-5]	[1,-3]	sum.[9,-5]	4
[]	[9]	[-5,1,-3]	sum.[9]	9
[]	[]	[9,-5,1,-3]	sum.[]	0
[9,-5]	[1]	[-3]	sum.[1]	-2
[9]	[-5]	[1,-3]	sum.[-5]	-5
[9]	[-5,1]	[-3]	sum.[-5,1]	-4

Teniendo en cuenta el cuantificador utilizado nos queda lo siguiente:

$min\ (0(min\ -3(min\ -2(min\ -7(min\ 2(min\ 5(min\ 4(min\ 9(min\ 0)))))))))$

$$\equiv \{\text{Evaluamos}\}$$

$$-7$$

4)

a) La lista xs es un segmento inicial de la lista ys (prefijo.xs.ys).

$$prefijo.xs.ys = \langle \exists as :: ys = xs +\!\!+ as \rangle$$

b) $seg.xs.ys = \langle \exists as, bs :: ys = as + +xs + +bs \rangle$

c) $sufijo.xs.ys = \langle \exists as :: ys = as +\!\!+ xs \rangle$

d) $segComun.xs.ys = \langle \exists as, bs, cs : ys = as + +bs + +cs \wedge bs \neq [] : seg.bs.xs \rangle$

e)

$hayMeseta.xs = \langle \exists as, bs, cs : xs = as + +bs + +cs : P.as.bs.cs \rangle$

P.as.bs.cs = “bs no es ni prefijo ni sufijo ∧ el minimo de bs es mayor a los valores de as y cs”

$P.as.bs.cs = Q.as.bs.cs \wedge R.as.bs.cs$
Q.as.bs.cs = “bs no es ni prefijo ni sufijo”

$Q.as.bs.cs = as \neq [] \wedge cs \neq []$
R.as.bs.cs = “El minimo de bs es mayor a los valores de as y cs”

$R.as.bs.cs = min.bs > max.(as + +bs)$

$\therefore P.as.bs.cs = as \neq [] \wedge cs \neq [] \wedge min.bs > max.(as + +bs)$

Resultado final:

$hayMeseta.xs = \langle \exists as, bs, cs : xs = as + +bs + +cs : P.as.bs.cs \rangle$
donde:
$P.as.bs.cs = as \neq [] \wedge cs \neq [] \wedge min.bs > max.(as + +bs)$

f) La lista xs de numeros enteros tiene la misma cantidad de elementos pares e impares (balanceada.xs).

$balanceada.xs = cantidadPares.xs == cantidadImpares.xs$

donde:

$cantidadPares.xs = \langle N\ i : 0 \leq i < \#xs : esPar.(xs\ !!\ i) \rangle$

$cantidadImpares.xs = \#xs - cantidadPares.xs$

5)

a) caso base:

b) $xs = []$

$$prefijo.[].ys$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as :: ys = [] +\!\!+ as \rangle$$
$$\equiv \{\text{Def de}\}$$
$$\langle \exists as :: ys = as \rangle$$
$$\equiv \{\text{Intercambio}\}$$
$$\langle \exists as : ys = as : True \rangle$$
$$\equiv \{\text{Rango unitario}\}$$
$$True$$

ii) $ys = []$

$$prefijo.xs.[]$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as :: [] = xs +\!\!+ as \rangle$$
$$\equiv \{\text{ Igualdad de listas}\}$$
$$\langle \exists as :: [] = xs \wedge [] = as \rangle$$
$$\equiv \{\text{Intercambio}\}$$
$$\langle \exists as : [] = as : [] = xs \rangle$$
$$\equiv \{\text{Rango unitario}\}$$
$$[] = xs$$

Caso inductivo: $xs = (z : zs), ys = (l : ls)$ $HI = prefijo.zs.ls = \langle \exists as :: ls = zs +\!\!+ as \rangle$

$$prefijo.(z : zs).(l : ls)$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as :: (l : ls) = (z : zs) +\!\!+ as \rangle$$
$$\equiv \{\text{Def de ++}\}$$
$$\langle \exists as :: (l : ls) = z : (zs + +as) \rangle$$
$$\equiv \{\text{Igualdad de listas}\}$$
$$\langle \exists as :: l = z \wedge ls = (zs + +as) \rangle$$
$$\equiv \{\text{Distributividad}\}$$
$$l = z \wedge \langle \exists as :: ls = zs + +as \rangle$$
$$\equiv \{\text{HI}\}$$

$$l =< \wedge prefijo.zs.ls$$

b) Caso base: i) $ys = []$

$$seg.xs.[]$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as, bs :: [] = as + +xs + +bs \rangle$$
$$\equiv \{\text{Igualdad de listas}\}$$
$$\langle \exists as, bs :: [] = as \wedge [] = xs \wedge [] = bs \rangle$$
$$\equiv \{\text{Anidado}\}$$
$$\langle \exists as :: \langle \exists bs :: [] = as \wedge [] = xs \wedge [] = bs \rangle \rangle$$
$$\equiv \{\text{Intercambio}\}$$
$$\langle \exists as :: \langle \exists bs : bs = [] : [] = as \wedge [] = xs \rangle \rangle$$
$$\equiv \{\text{Rango unitario}\}$$
$$\langle \exists as :: [] = as \wedge [] = xs \rangle$$
$$\equiv \{\text{Intercambio}\}$$
$$\langle \exists as : [] = as : [] = xs \rangle$$
$$\equiv \{\text{Rango unitario}\}$$
$$[] = xs$$

Caso inductivo: $ys = (l : ls)$

$$HI = seg.xs.ls = \langle \exists as, bs :: ls = as + +xs + +bs \rangle$$

$$seg.xs.(l : ls)$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as, bs :: (l : ls) = as + +xs + +bs \rangle$$
$$\equiv \{\text{Anidado}\}$$
$$\langle \exists as :: \langle \exists bs :: (l : ls) = as + +xs + +bs \rangle \rangle$$
$$\equiv \{\text{Tercero excluido}\}$$
$$\langle \exists as : as = [] \vee as \neq [] : \langle \exists bs :: (l : ls) = as + +xs + +bs \rangle \rangle$$
$$\equiv \{\text{Particion de rango}\}$$
$$\langle \exists as : as = [] : \langle \exists bs :: (l : ls) = as{+}{+}xs{+}{+}bs \rangle \rangle \vee \langle \exists as : as \neq [] : \langle \exists bs :: (l : ls) = as{+}{+}xs{+}{+}bs \rangle \rangle$$
$$\equiv \{\text{Rango unitario}\}$$
$$\langle \exists bs :: (l : ls) = []{+}{+}xs{+}{+}bs \rangle \vee \langle \exists as : as \neq [] : \langle \exists bs :: (l : ls) = as{+}{+}xs{+}{+}bs \rangle \rangle$$

$$\equiv \{\text{Def de ++}\}$$

$$\langle \exists bs :: (l : ls) = xs + +bs\rangle \vee \langle \exists as : as \neq [] : \langle \exists bs :: (l : ls) = as + +xs + +bs\rangle\rangle$$

$$\equiv \{\text{Modularizacion}\}$$

$$prefijo.xs.(l : ls) \vee \langle \exists as : as \neq [] : \langle \exists bs :: (l : ls) = as + +xs + +bs\rangle\rangle$$

$$\equiv \{\text{Cambio de variable f.as = (a:as)}\}$$

$$prefijo.xs.(l : ls) \vee \langle \exists a, as : (a : as) \neq [] : \langle \exists bs :: (l : ls) = (a : as) + +xs + +bs\rangle\rangle$$

$$\equiv \{\text{Evaluo rango}\}$$

$$prefijo.xs.(l : ls) \vee \langle \exists a, as : True : \langle \exists bs :: (l : ls) = (a : as) + +xs + +bs\rangle\rangle$$

$$\equiv \{\text{Def de ++ e igualdad de listas}\}$$

$$prefijo.xs.(l : ls) \vee \langle \exists a, as : True : \langle \exists bs :: l = a \wedge ls = as + +xs + +bs\rangle\rangle$$

$$\equiv \{\text{Anidado}\}$$

$$prefijo.xs.(l : ls) \vee \langle \exists a, as, bs :: l = a \wedge ls = as + +xs + +bs\rangle$$

$$\equiv \{\text{Intercambio}\}$$

$$prefijo.xs.(l : ls) \vee \langle \exists a, as, bs : l = a : ls = as + +xs + +bs\rangle$$

$$\equiv \{\text{Anidado}\}$$

$$prefijo.xs.(l : ls) \vee \langle \exists a : l = a : \langle \exists as, bs :: ls = as + +xs + +bs\rangle\rangle$$

$$\equiv \{\text{Rango unitario}\}$$

$$prefijo.xs.(l : ls) \vee \langle \exists as, bs :: ls = as + +xs + +bs\rangle$$

$$\equiv \{\text{HI}\}$$

$$prefijo.xs.(l : ls) \vee seg.xs.ls$$

6) $xs = "frufru"$

as	bs	Termino	Evaluacion
[frufru]	[]	[frufru]==[]	False
[frufr]	[u]	[frufr]==[u]	False
[fruf]	[ru]	[fruf]==[ru]	False
[fru]	[fru]	[fru]==[fru]	True
[fr]	[ufru]	[fr]==[ufru]	False
[f]	[rufru]	[f]==[rufru]	False

7) $semiEco.xs = \langle \exists as, bs : xs = as +\!\!+ as +\!\!+ bs : as \neq []\rangle$

Caso base: $xs = []$

$$semiEco.[]$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle \exists as, bs : [] = as +\!\!+ as +\!\!+ bs : as \neq [] \rangle$$
$$\equiv \{\text{Propiedades de listas}\}$$
$$\langle \exists as, bs : [] = as \wedge [] = bs : as \neq [] \rangle$$
$$\equiv \{\text{Eliminacion de variable}\}$$
$$\langle \exists bs : [] = bs : [] \neq [] \rangle$$
$$\equiv \{\text{Termino constante}\}$$
$$[] \neq []$$
$$\equiv \{\text{Logica}\}$$
$$False$$

Caso inductivo: $xs = (y : ys)$

$HI = semiEco.ys = \langle \exists as, bs : ys = as +\!\!+ as +\!\!+ bs : as \neq [] \rangle$

$$semiEco.(y : ys)$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as, bs : (y : ys) = as +\!\!+ as +\!\!+ bs : as \neq [] \rangle$$
$$\equiv \{\text{Intercambio}\}$$
$$\langle \exists as, bs : (y : ys) = as +\!\!+ as +\!\!+ bs \wedge as \neq [] : True \rangle$$
$$\equiv \{\text{Cambio de variable as -> (a:as)}\}$$
$$\langle \exists a, as, bs : (y : ys) = (a : as) +\!\!+ (a : as) +\!\!+ bs \wedge (a : as) \neq [] : True \rangle$$
$$\equiv \{\text{Def de ++ y propiedades de lista}\}$$
$$\langle \exists a, as, bs : y = a \wedge ys = as +\!\!+ (a : as) +\!\!+ bs \wedge (a : as) \neq [] : True \rangle$$
$$\equiv \{\text{Logica y neutro de la conjuncion}\}$$
$$\langle \exists a, as, bs : y = a \wedge ys = as +\!\!+ (a : as) +\!\!+ bs : True \rangle$$
$$\equiv \{\text{Eliminacion de variable}\}$$
$$\langle \exists as, bs : ys = as +\!\!+ (y : as) +\!\!+ bs : True \rangle$$
$$\equiv \{\text{Neutro de la conjuncion y tercero excluido}\}$$
$$\langle \exists as, bs : ys = as +\!\!+ (y : as) +\!\!+ bs \wedge (as = [] \vee as \neq []) : True \rangle$$
$$\equiv \{\text{Distributividad}\}$$
$$\langle \exists as, bs : ys = as +\!\!+ (y : as) +\!\!+ bs \wedge as = [] \vee ys = as +\!\!+ (y : as) +\!\!+ bs \wedge as \neq [] : True \rangle$$
$$\equiv \{\text{Particion de rango}\}$$
$$\langle \exists as, bs : ys = as +\!\!+ (y : as) +\!\!+ bs \wedge as = [] : True \rangle \vee \langle \exists as, bs : ys = as +\!\!+ (y : as) +\!\!+ bs \wedge as \neq [] : True \rangle$$
$$\equiv \{\text{Prop de listas}\}$$

$\langle \exists as, bs : ys = as +\!\!+ (y : as) +\!\!+ bs \wedge as = [] : True\rangle \vee \langle \exists as, bs : ys = as +\!\!+ [y] +\!\!+ as +\!\!+ bs \wedge as \neq [] : True\rangle$

Como no se puede aplilcar HI, procedemos a realizar una generalizacion por abstraccion:

$gSemiEco.xs.ys = \langle \exists as, bs : xs = as +\!\!+ ys +\!\!+ as +\!\!+ bs \wedge as \neq [] : True\rangle$

semiEco.xs = gSemiEco.xs.[]

Demostracion:

$$gSemiEco.xs.[]$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as, bs : xs = as +\!\!+ [] +\!\!+ as +\!\!+ bs \wedge as \neq [] : True\rangle$$
$$\equiv \{\text{Def de ++}\}$$
$$\langle \exists as, bs : xs = as +\!\!+ as +\!\!+ bs \wedge as \neq [] : True\rangle$$
$$\equiv \{\text{Especificacion de semiEco}\}$$
$$semiEco.xs$$

Caso base: $ys = []$

$$gSemiEco.[].ys$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as, bs : [] = as +\!\!+ ys +\!\!+ as +\!\!+ bs \wedge as \neq [] : True\rangle$$
$$\equiv \{\text{Propiedad de listas}\}$$
$$\langle \exists as, bs : [] = as \wedge [] = ys \wedge [] = as \wedge [] = bs \wedge as \neq [] : True\rangle$$
$$\equiv \{\text{Logica}\}$$
$$\langle \exists as, bs : [] = as \wedge [] = ys \wedge [] = as \wedge [] = bs \wedge False : True\rangle$$
$$\equiv \{\text{Elemento absorbente de la conjuncion}\}$$
$$\langle \exists as, bs : False : True\rangle$$
$$\equiv \{\text{Rango vacio}\}$$
$$False$$

Caso inductivo: $xs = (z : zs)$

$HI = gSemiEco.zs.ys = \langle \exists as, bs : zs = as +\!\!+ ys +\!\!+ as +\!\!+ bs \wedge as \neq [] : True\rangle$

$$gSemiEco.(z : zs).ys$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle \exists as, bs : (z : zs) = as +\!\!+ ys +\!\!+ as +\!\!+ bs \wedge as \neq [] : True\rangle$$

$\equiv$ {cambio de variable as->(a:as)}

$\equiv$ {Def de ++ repetidas veces}

$$\langle \exists a, as, bs : (z : zs) = a : (as ++ ys ++ (a : as) ++ bs) \wedge (a : as) \neq [] : True\rangle$$

$\equiv$ {Logica y neutro de la conjuncion}

$$\langle \exists a, as, bs : (z : zs) = a : (as ++ ys ++ (a : as) ++ bs) : True\rangle$$

$\equiv$ {Propiedad de listas}

$$\langle \exists a, as, bs : z = a \wedge zs = as ++ ys ++ (a : as) ++ bs : True\rangle$$

$\equiv$ {Eliminacion de variable a}

$$\langle \exists as, bs : zs = as ++ ys ++ (z : as) ++ bs : True\rangle$$

$\equiv$ {Def de ++}

$$\langle \exists as, bs : zs = as ++ ys ++ (z : as) ++ bs : True\rangle$$

$\equiv$ {Elemento neutro de la conjuncion y tercero excluido}

$$\langle \exists as, bs : zs = as ++ ys ++ (z : as) ++ bs \wedge (as = [] \vee as \neq []) : True\rangle$$

$\equiv$ {Distributividad y particion de rango}

$$\langle \exists as, bs : zs = as ++ ys ++ (z : as) ++ bs \wedge as = [] : True\rangle \vee \langle \exists as, bs : zs = as ++ ys ++ (z : as) ++ bs \wedge as \neq [] : True\rangle$$

$\equiv$ {Propiedad de listas}

$$\langle \exists as, bs : zs = as ++ ys ++ (z : as) ++ bs \wedge as = [] : True\rangle \vee \langle \exists as, bs : zs = as ++ ys ++ [z] ++ as ++ bs \wedge as \neq [] : True\rangle$$

$\equiv$ {HI}

$$\langle \exists as, bs : zs = as ++ ys ++ (z : as) ++ bs \wedge as = [] : True\rangle \vee gSemiEco.zs.(ys ++ [z])$$

$\equiv$ {Eliminacion de variable}

$$\langle \exists bs : zs = [] ++ ys ++ (z : []) ++ bs : True\rangle \vee gSemiEco.zs.(ys ++ [z])$$

$\equiv$ {Def de ++ y propiedad de listas}

$$\langle \exists bs : zs = ys ++ [z] ++ bs : True\rangle \vee gSemiEco.zs.(ys ++ [z])$$

$\equiv$ {Modularizacion}

$$prefijo.(ys ++ [z]).zs \vee gSemiEco.zs.(ys ++ [z])$$

Resultado de la derivacion:

```
gSemiEco.[].ys = False
gSemiEco.(x:xs).ys = prefijo.(ys ++ [x]).xs || gSemiEco.xs.(ys ++ [x]

semiEco.xs = semiEco.xs.[]
```

8) $sumaMin.xs = \langle Min\ as, bs, cs : xs = as ++ bs ++ cs : sum.bs\rangle$

Caso base: $xs := []$

$$sumaMin.[]$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle Min\ as, bs, cs : [] = as +\!\!+ bs +\!\!+ cs : sum.bs\rangle$$
$$\equiv \{\text{Propiedad de listas}\}$$
$$\langle Min\ as, bs, cs : [] = as \wedge [] = bs \wedge [] = cs : sum.bs\rangle$$
$$\equiv \{\text{Eliminacion de variable bs}\}$$
$$\langle Min\ as, cs : [] = as \wedge [] = cs : sum.[]\rangle$$
$$\equiv \{\text{Termino constante y def de sum}\}$$
$$0$$

Caso inductivo: $xs := (x : xs)$

$HI = sumaMin.xs = \langle Min\ as, bs, cs : xs = as +\!\!+ bs +\!\!+ cs : sum.bs\rangle$

$$sumaMin.(x : xs)$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle Min\ as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs : sum.bs\rangle$$
$$\equiv \{\text{Neutro de la conjuncin y tercero excluido}\}$$
$$\langle Min\ as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs \wedge (as = [] \vee as \neq []) : sum.bs\rangle$$
$$\equiv \{\text{Distributividad y particion de rango}\}$$
$$\langle Min\, as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs \wedge as = [] : sum.bs\rangle min \langle Min\, as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs \wedge as \neq [] : s$$
$$\equiv \{\text{Cambio de variable: as->(a:as)}\}$$
$$\langle Min\, as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs \wedge as = [] : sum.bs\rangle min \langle Min\, a, as, bs, cs : (x : xs) = (a : as) +\!\!+ bs +\!\!+ cs \wedge (a$$
$$\equiv \{\text{Logica y neutro de la conjuncion}\}$$
$$\equiv \{\text{Def de ++}\}$$
$$\langle Min\, as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs \wedge as = [] : sum.bs\rangle min \langle Min\, a, as, bs, cs : (x : xs) = a : (as +\!\!+ bs +\!\!+ cs) : su$$
$$\equiv \{\text{Propiedad de listas}\}$$
$$\langle Min\, as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs \wedge as = [] : sum.bs\rangle min \langle Min\, a, as, bs, cs : x = a \wedge xs = as +\!\!+ bs +\!\!+ cs : sum.$$
$$\equiv \{\text{Eliminacion de variable a}\}$$
$$\langle Min\, as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs \wedge as = [] : sum.bs\rangle min \langle Min\, as, bs, cs : xs = as +\!\!+ bs +\!\!+ cs : sum.bs\rangle$$
$$\equiv \{\text{HI}\}$$
$$\langle Min\ as, bs, cs : (x : xs) = as +\!\!+ bs +\!\!+ cs \wedge as = [] : sum.bs\rangle minsumaMin.xs$$
$$\equiv \{\text{Eliminacion de variable}\}$$

$$\langle Min\ bs, cs : (x : xs) = [] ++ bs ++ cs : sum.bs\rangle min sumaMin.xs$$

$$\equiv \{\text{Def de ++}\}$$

$$\langle Min\ bs, cs : (x : xs) = bs ++ cs : sum.bs\rangle min sumaMin.xs$$

$$\equiv \{\text{Modularizamos, } sumaMin2.xs = \langle Min\ bs, cs : xs = bs ++ cs : sum.bs\rangle\}$$

$$sumaMin2.xs\ min\ sumaMin.xs$$

resultado parcial:

```
sumaMin.[] = 0
sumaMin.(x:xs) = sumaMin2.xs `min` sumaMin.xs
```

Ahora hay que derivar sumaMin2: $sumaMin2.xs = \langle Min\ bs, cs : xs = bs ++ cs : sum.bs\rangle$

Caso base: $xs := []$

$$sumaMin2.[]$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle Min\ bs, cs : [] = bs ++ cs : sum.bs\rangle$$

$$\equiv \{\text{Propiedad de listas}\}$$

$$\langle Min\ bs, cs : [] = bs \wedge [] = cs : sum.bs\rangle$$

$$\equiv \{\text{Eliminacion de variable}\}$$

$$\langle Min\ cs : [] = cs : sum.[]\rangle$$

$$\equiv \{\text{Def de sum y termino constante}\}$$

$$0$$

Caso inductivo: $xs := (x : xs)$

$$sumaMin2.(x : xs)$$

$$\equiv \{\text{Especificacion}\}$$

$$\langle Min\ bs, cs : (x : xs) = bs ++ cs : sum.bs\rangle$$

$$\equiv \{\text{Elemento neutro de la conjuncion y tercero excluido}\}$$

$$\langle Min\ bs, cs : (x : xs) = bs ++ cs \wedge (bs = [] \vee bs \neq []) : sum.bs\rangle$$

$$\equiv \{\text{Distributividad y particion de rango}\}$$

$$\langle Min\ bs, cs : (x : xs) = bs++cs \wedge bs = [] : sum.bs\rangle min \langle Min\ bs, cs : (x : xs) = bs++cs \wedge bs \neq [] : sum.bs\rangle$$

$$\equiv \{\text{Llamamos X a la primer expresion cuantificada}\}$$

$$X min \langle Min\ bs, cs : (x : xs) = bs ++ cs \wedge bs \neq [] : sum.bs\rangle$$

$$\equiv \{\text{Cambio de variable: bs->(b:bs)}\}$$

$$Xmin\langle Min\ b, bs, cs : (x : xs) = (b : bs) +\!\!+\ cs \wedge (b : bs) \neq [] : sum.(b : bs)\rangle$$
$$\equiv \{\text{Propiedad de listas y elemento neutro de la conjuncion}\}$$
$$Xmin\langle Min\ b, bs, cs : (x : xs) = (b : bs) +\!\!+\ cs\wedge : sum.(b : bs)\rangle$$
$$\equiv \{\text{Def de ++ y propiedad de listas}\}$$
$$Xmin\langle Min\ b, bs, cs : x = b \wedge xs = bs +\!\!+\ cs : sum.(b : bs)\rangle$$
$$\equiv \{\text{Eliminacion de variable b}\}$$
$$Xmin\langle Min\ b, bs, cs : xs = bs +\!\!+\ cs : sum.(x : bs)\rangle$$
$$\equiv \{\text{Def de sum}\}$$
$$Xmin\langle Min\ b, bs, cs : xs = bs +\!\!+\ cs : x + sum.(bs)\rangle$$
$$\equiv \{\text{Distributividad, ya que + es distributivo con min}\}$$
$$Xmin(\langle Min\ b, bs, cs : xs = bs +\!\!+\ cs : sum.(bs)\rangle + x)$$
$$\equiv \{\text{HI}\}$$
$$X\ min\ sumaMin2.xs + x$$
$$\equiv \{\text{Cambiemos X por la expresion cuantificada original}\}$$
$$\langle Min\ bs, cs : (x : xs) = bs +\!\!+\ cs \wedge bs = [] : sum.bs\rangle min\ sumaMin2.xs + x$$
$$\equiv \{\text{Eliminacion de variable bs}\}$$
$$\langle Min\ bs, cs : (x : xs) = [] +\!\!+\ cs : sum.[]\rangle min\ sumaMin2.xs + x$$
$$\equiv \{\text{Def de sum y termino constante}\}$$
$$0\ min\ (sumaMin2 + x)$$

Resultado final:

```
sumaMin.[] = 0
sumaMin.(x:xs) = sumaMin2.xs `min` sumaMin.xs
  where
    sumaMin2.xs = 0 min (sumaMin2 + x)
```

b)

$maxLongEq.e.xs = \langle Max\ as, bs, cs : xs = as +\!\!+\ bs +\!\!+\ cs \wedge iga.e.bs : \#bs\rangle$

Caso base: $xs = []$

$$maxLongEq.e.[]$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle Max\ as, bs, cs : [] = as +\!\!+\ bs +\!\!+\ cs \wedge iga.e.bs : \#bs\rangle$$
$$\equiv \{\text{Propiedad de listas, eliminacion de variable bs y termino constante}\}$$
$$\#[]$$

$$\equiv \{\text{Def de } \#\}$$

0

Caso inductivo:$xs := (x : xs)$

$HI = maxLongEq.e.xs = \langle Max\ as, bs, cs : xs = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs : \#bs\rangle$

$$maxLongEq.e.(x : xs)$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle Max\ as, bs, cs : (x : xs) = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs : \#bs\rangle$$
$$\equiv \{\text{Elemento neutro de la conjuncion, tercero excluido}\}$$
$$\langle Max\ as, bs, cs : (x : xs) = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs \wedge (as = [] \vee as \neq []) : \#bs\rangle$$
$$\equiv \{\text{Distributividad conjuncion disyuncion}\}$$
$$\equiv \{\text{Particion de rango}\}$$
$$\langle Max\, as, bs, cs : (x : xs) = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs \wedge as = [] : \#bs\rangle max \langle Max\, as, bs, cs : (x : xs) = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga$$
$$\equiv \{\text{Llamemos X a la primer expresion cuantificada}\}$$
$$X\ max\langle Max\ as, bs, cs : (x : xs) = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs \wedge as \neq [] : \#bs\rangle$$
$$\equiv \{\text{Cambio de variable as->(a:as)}\}$$
$$X\, max\langle Max\, a, as, bs, cs : (x : xs) = (a : as) \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs \wedge (a : as) \neq [] : \#bs\rangle$$
$$\equiv \{\text{Propiedad de listas y neutro de la conjuncion}\}$$
$$X\ max\langle Max\ a, as, bs, cs : (x : xs) = (a : as) \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs : \#bs\rangle$$
$$\equiv \{\text{Def de ++ y prop de listas}\}$$
$$X\ max\langle Max\ a, as, bs, cs : x = a \wedge xs = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs : \#bs\rangle$$
$$\equiv \{\text{Eliminacion de variable a}\}$$
$$X\ max\langle Max\ as, bs, cs : xs = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs : \#bs\rangle$$
$$\equiv \{\text{HI}\}$$
$$X\ max\ maxLongEq.e.xs$$
$$\equiv \{\text{Cambiemos X por la expresion cuantificada original }\}$$
$$\langle Max\, as, bs, cs : (x : xs) = as \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs \wedge as = [] : \#bs\rangle maxmaxLongEq.e.xs$$
$$\equiv \{\text{Eliminacion de variable}\}$$
$$\langle Max\ bs, cs : (x : xs) = [] \mathbin{+\!\!+} bs \mathbin{+\!\!+} cs \wedge iga.e.bs : \#bs\rangle maxmaxLongEq.e.xs$$
$$\equiv \{\text{Def de ++}\}$$
$$\langle Max\ bs, cs : (x : xs) = bs \mathbin{+\!\!+} cs \wedge iga.e.bs : \#bs\rangle maxmaxLongEq.e.xs$$
$$\equiv \{\text{Modularizamos}\}$$
$$maxLongEq2.e.(x : xs)\ max\ maxLongEq.e.xs$$

Resultado parcial:

```
maxLongEq.e.[] = 0
maxLongEq.e.(x:xs) = maxLongEq2.e.(x:xs) max maxLongEq.e.xs
```

Ahora hay que derivar maxLongEq2:

$maxLongEq2.e.xs = \langle Max\ bs, cs : xs = bs +\!\!+\ cs \wedge iga.e.bs : \#bs \rangle$

Caso base:

$$maxLongEq2.e.xs$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle Max\ bs, cs : xs = bs +\!\!+\ cs \wedge iga.e.bs : \#bs \rangle$$

Caso inductivo

$HI = maxLongEq2.e.xs = \langle Max\ bs, cs : xs = bs +\!\!+\ cs \wedge iga.e.bs : \#bs \rangle$

$$maxLongEq2.e.xs$$
$$\equiv \{\text{Especificacion}\}$$
$$\langle Max\ bs, cs : xs = bs +\!\!+\ cs \wedge iga.e.bs : \#bs \rangle$$