Lautaro Bachmann 44.390.167

Coloquio AYED

Considerá la función queHace :: Int → Int → [Int] → Int especificada como
queHace.b.a.xs = ⟨ Max p, q : 0 ≤ p ≤ q < #xs ∧ ⟨ ∀i : p ≤ i < q : b < xs!i < a ⟩ : q − p ⟩

a) describí con tus palabras lo que hace la función
b) transformá esa especificación en una que use segmentos
c) derivá queHace (podés usar cualquiera de las dos especificaciones).

a) La función calcula el largo del mayor segmento de xs (sin tener en cuenta nunca el ultimo elemento) que cumpla que todos los elementos de dicho segmento esten dentro del intervalo abierto (b, a).

b) Como la especificacion nunca toma en cuenta al ultimo elemento de la lista, para transformar la especificacion en una que use segmentos podemos hacer que el segmento final de xs sea no vacio. De tal modo que nos quedaria la siguiente especificación:

$$queHace.b.a.xs = \langle Max\ bs, cs, ds : xs = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge \langle \forall i : 0 \leq i < \#cs : b < cs!i < a \rangle : \#cs \rangle$$

c) Antes de proceder a derivar queHace derivemos la especificacion $estaEnRango.b.a.xs = \langle \forall i : 0 \leq i < \#xs : b < xs!i < a \rangle$, con el fin de reemplazarla en queHace y así facilitar la escritura en el desarrollo de la derivacion.

Derivacion estaEnRango.b.a.xs:

Caso base: $xs := [\,]$

$estaEnRango.b.a.[\,]$

$\equiv$ { Especificacion }

$\langle \forall i : 0 \leq i < \#[\,] : b < [\,]!i < a \rangle$

$\equiv$ { Def. cardinal }

$\langle \forall i : 0 \leq i < 0 : b < [\,]!i < a \rangle$

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

$\langle \forall i : False : b < [\,]!i < a \rangle$

$\equiv \{$ Rango vacio $\}$

$True$

Caso Inductivo: $xs := (x \triangleright xs)$

$estaEnRango.b.a.(x \triangleright xs)$

$\equiv \{$ Especificación $\}$

$\langle \forall i : 0 \leq i < \#(x \triangleright xs) : b < (x \triangleright xs)!i < a \rangle$

$\equiv \{$ Def de cardinal $\}$

$\langle \forall i : 0 \leq i < 1 + \#xs : b < (x \triangleright xs)!i < a \rangle$

$\equiv \{$ Aritmética $\}$

$\langle \forall i : i = 0 \vee 1 \leq i < 1 + \#xs : b < (x \triangleright xs)!i < a \rangle$

$\equiv \{$ Partición de rango $\}$

$\langle \forall i : i = 0 : b < (x \triangleright xs)!i < a \rangle \wedge \langle \forall i : 1 \leq i < 1 + \#xs : b < (x \triangleright xs)!i < a \rangle$

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

$b < (x \triangleright xs)!0 < a \wedge \langle \forall i : 1 \leq i < 1 + \#xs : b < (x \triangleright xs)!i < a \rangle$

$\equiv \{$ Def $!$ $\}$

$b < x < a \wedge \langle \forall i : 1 \leq i < 1 + \#xs : b < (x \triangleright xs)!i < a \rangle$

$\equiv \{$ Cambio de variable $i \leftarrow i+1$ $\}$

$b < x < a \wedge \langle \forall i : 1 \leq i+1 < 1 + \#xs : b < (x \triangleright xs)!i+1 < a \rangle$

$\equiv \{$ Aritmetica $\}$

$b < x < a \wedge \langle \forall i : 0 \leq i < \#xs : b < (x \triangleright xs)!i+1 < a \rangle$

$= \{$ Aritmetica y Def $!$ $\}$

$b < x \wedge x < a \wedge \langle \forall i : 0 \leq i < \#xs : b < xs!i < a \rangle$

$= \{$ HI $\}$

$b < x \wedge x < a \wedge estaEnRango.b.a.xs$

Resultado final estaEnRango:

$estaEnRango.b.a.[\,] = True$

$estaEnRango.b.a.(x \triangleright xs) = b < x \wedge x < a \wedge estaEnRango.b.a.xs$

Teniendo en cuenta esto, la especificacion de queHace quedaria de la siguiente forma:

$$queHace.b.a.xs = \langle Max\ bs, cs, ds : xs = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$$

Derivacion queHace.b.a.xs

Caso base: $xs := [\,]$

$queHace.b.a.[\,]$

$\equiv$ { Especificación }

$\langle Max\ bs, cs, ds : \underline{[\,] = bs ++ cs ++ ds} \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Propiedad de listas }

$\langle Max\ bs, cs, ds : [\,] = bs \wedge [\,] = cs \wedge \underline{[\,] = ds \wedge ds \neq [\,]} \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Principio de no contradiccion }

$\langle Max\ bs, cs, ds : [\,] = bs \wedge [\,] = cs \wedge \underline{False} \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Elemento absorbente conjuncion }

$\underline{\langle Max\ bs, cs, ds : False : \#cs \rangle}$

$\equiv$ { Rango vacio }

$-\infty$

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

HI: $queHace.b.a.xs = \langle Max\ bs, cs, ds : xs = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

queHace.b.a.$(x \triangleright xs)$

$\equiv$ { Especificación }

$\langle Max\ bs, cs, ds : (x \triangleright xs) = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

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

$\langle Max\ bs, cs, ds : (bs = [\,] \vee bs \neq [\,]) \wedge (x \triangleright xs) = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Distributividad conjuncion disyuncion }

$\langle Max\ bs, cs, ds : (bs = [\,] \wedge (x \triangleright xs) = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs) \vee$
$(bs \neq [\,] \wedge (x \triangleright xs) = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs) : \#cs \rangle$

$\equiv$ { Particion de rango }

$\langle Max\ bs, cs, ds : bs = [\,] \wedge (x \triangleright xs) = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle \ max$
$\langle Max\ bs, cs, ds : bs \neq [\,] \wedge (x \triangleright xs) = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Llamemos E a la primer expresion }

$E \ max \ \langle Max\ bs, cs, ds : bs \neq [\,] \wedge (x \triangleright xs) = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Cambio de variable $bs \leftarrow k \triangleright bs$ }

$E \ max \ \langle Max\ k, bs, cs, ds : k \triangleright bs \neq [\,] \wedge (x \triangleright xs) = (k \triangleright bs) ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Logica y elemento neutro conjuncion }

$E \ max \ \langle Max\ k, bs, cs, ds : (x \triangleright xs) = (k \triangleright bs) ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Concatenacion x2 }

$E \ max \ \langle Max\ k, bs, cs, ds : (x \triangleright xs) = k \triangleright (bs ++ cs ++ ds) \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Propiedad de listas }

$E \ max \ \langle Max\ k, bs, cs, ds : x = k \wedge xs = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Eliminación de variable k }

$E \ max \ \langle Max\ bs, cs, ds : xs = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { HI }

$E \ max$ queHace.b.a.xs

$\equiv$ { Reemplazo E }

$\langle Max\ bs, cs, ds : bs = [\,] \wedge (x \triangleright xs) = bs ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle \ max$
queHace.b.a.xs

$\equiv$ { Eliminacion de variable bs }

$\langle Max\ cs, ds : (x \triangleright xs) = [\,] ++ cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$ max ~~vedHace~~ queHace.b.a.xs

$\equiv$ { Def concatenación }

$\langle Max\ cs, ds : (x \triangleright xs) = cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$ max queHace.b.a.xs

$\equiv$ { Modularizamos queHace2.b.a.xs }

queHace2.b.a.(x ▷ xs) max queHace.b.a.xs

Derivación queHace2.b.a.xs

queHace2.b.a.xs $= \langle Max\ cs, ds : xs = cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

Caso base: $xs := [\,]$

queHace2.b.a.[ ]

$\equiv$ { Especificacion }

$\langle Max\ cs, ds : [\,] = cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Propiedad de listas }

$\langle Max\ cs, ds : [\,] = cs \wedge [\,] = ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Principio de no contradicion }

$\langle Max\ cs, ds : [\,] = cs \wedge False \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Elemento absorbente conjunción }

$\langle Max\ cs, ds : False : \#cs \rangle$

$\equiv$ { Rango vacío }

$-\infty$

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

HI: queHace2.b.a.xs $= \langle Max\ cs, ds : xs = cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

queHace2.b.a.(x ▷ xs)

$\equiv$ { Especificación }

$\langle Max\ cs, ds : (x \triangleright xs) = cs ++ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Neutro conjuncion y tercero excluido }

$\langle \mathrm{Max}\ cs, ds : (cs = [\,] \vee cs \neq [\,]) \wedge (x \triangleright xs) = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Distributividad conjuncion disyuncion }

$\langle \mathrm{Max}\ cs, ds : (cs = [\,] \wedge (x \triangleright xs) = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs) \vee$
$(cs \neq [\,] \wedge (x \triangleright xs) = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs) : \#cs \rangle$

$\equiv$ { Particion de rango }

$\langle \mathrm{Max}\ cs, ds : cs = [\,] \wedge (x \triangleright xs) = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle \max$
$\langle \mathrm{Max}\ cs, ds : cs \neq [\,] \wedge (x \triangleright xs) = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Llamemos E a la segunda expresión }

$\langle \mathrm{Max}\ cs, ds : cs = [\,] \wedge (x \triangleright xs) = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle \max E$

$\equiv$ { Eliminacion de variable cs }

$\langle \mathrm{Max}\ ds : (x \triangleright xs) = [\,] +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.[\,] : \#[\,] \rangle \max E$

$\equiv$ { Def. cardinal }

$\langle \mathrm{Max}\ ds : (x \triangleright xs = [\,] +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.[\,] : 0 \rangle \max E$

$\equiv$ { Termino constante }

$0 \max E$

$\equiv$ { Reemplazamos E }

$0 \max \langle \mathrm{Max}\ cs, ds : cs \neq [\,] \wedge (x \triangleright xs) = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.cs : \#cs \rangle$

$\equiv$ { Cambio de variable $cs \leftarrow (c \triangleright cs)$ }

$0 \max \langle \mathrm{Max}\ c, cs, ds : (c \triangleright cs) \neq [\,] \wedge (x \triangleright xs) = (c \triangleright cs) +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.(c \triangleright cs) : \#(c \triangleright cs) \rangle$

$\equiv$ { Lógica y elemento neutro conjuncion }

$0 \max \langle \mathrm{Max}\ c, cs, ds : (x \triangleright xs) = (c \triangleright cs) +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.(c \triangleright cs) : \#(c \triangleright cs) \rangle$

$\equiv$ { Def. concatenación }

$0 \max \langle \mathrm{Max}\ c, cs, ds : (x \triangleright xs) = c \triangleright (cs +\!\!+ ds) \wedge ds \neq [\,] \wedge estaEnRango.b.a.(c \triangleright cs) : \#(c \triangleright cs) \rangle$

$\equiv$ { Propiedad de listas }

$0 \max \langle \mathrm{Max}\ c, cs, ds : x = c \wedge xs = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.(c \triangleright cs) : \#(c \triangleright cs) \rangle$

$\equiv$ { Eliminacion de variable c }

$0 \max \langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \wedge ds \neq [\,] \wedge estaEnRango.b.a.(x \triangleright cs) : \#(x \triangleright cs) \rangle$

$\equiv$ { Def. recursiva estaEnRango }

$0 \max \langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \wedge ds \neq [\,] \wedge b < x \wedge x < a \wedge estaEnRango.b.a.cs : \#(x \triangleright cs) \rangle$

$\downarrow$

$\equiv \{$ Def. cardinal $\}$

$0 \max \langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \land ds \neq [\,] \land b < x \land x < a \land estaEnRango.b.a.cs : 1 + \#cs \rangle$

Analisis por casos:

Caso 1: $b < x \land x < a \equiv True$

$0 \max \langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \land ds \neq [\,] \land \underline{b < x \land x < a} \land estaEnRango.b.a.cs : 1 + \#cs \rangle$

$\equiv \{$ Por hipotesis $\}$

$0 \max \langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \land ds \neq [\,] \land \underline{True} \land estaEnRango.b.a.cs : 1 + \#cs \rangle$

$\equiv \{$ Elemento neutro conjunción $\}$

$0 \max \langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \land ds \neq [\,] \land estaEnRango.b.a.cs : \underline{1 + \#cs} \rangle$

$\equiv \{$ Distributividad Max $\}$

$0 \max (\underline{\langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \land ds \neq [\,] \land estaEnRango.b.a.cs : \#cs \rangle} + 1)$

$\equiv \{$ HI $\}$

$0 \max (queHace2.b.a.xs + 1)$

Caso 2: $b < x \land x < a \equiv False$

$0 \max \langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \land ds \neq [\,] \land \underline{b < x \land x < a} \land estaEnRango.b.a.cs : 1 + \#cs \rangle$

$\equiv \{$ Por hipotesis $\}$

$0 \max \langle \mathrm{Max}\ cs, ds : xs = cs +\!\!+ ds \land ds \neq [\,] \land \underline{False} \land estaEnRango.b.a.cs : 1 + \#cs \rangle$

$\equiv \{$ Elemento absorbente conjunción $\}$

$0 \max \underline{\langle \mathrm{Max}\ cs, ds : False : 1 + \#cs \rangle}$

$\equiv \{$ Rango vacio $\}$

$0 \max \underline{-\infty}$

$\equiv \{$ Elemento neutro max $\}$

0

Resultado Final:

$queHace :: int \to int \to [int] \to int$
$queHace.b.a.[\,] = -\infty$
$queHace.b.a.(x \triangleright xs) = queHace2.b.a.(x \triangleright xs) \max queHace.b.a.xs$
donde
$queHace2.b.a.[\,] = -\infty$
$queHace2.b.a.(x \triangleright xs) = (b < x \land x < a \to 0 \max (queHace2.b.a.xs + 1)$
$\square \neg(b < x \land x < a) \to 0$
$)$