Demostración de una implicación mediante inducción estructural

(entre otras cosas)

Sean las siguientes definiciones:

```
data AEB a = Hoja a | Bin (AEB a) a (AEB a)

const :: a -> b -> a
{C} const = (\ x -> \ y -> x)

head :: [a] -> a
{H} head (x:xs) = x

tail :: [a] -> [a]
{T} tail (x:xs) = xs

length :: [a] -> Int
{L0} length [] = 0
{L1} length (x:xs) = 1 + length xs

null :: [a] -> Bool
{N0} null [] = True
{N1} null (x:xs) = False

altura :: AEB a -> Int
{A0} altura (Hoja _) = 1
{A1} altura (Bin i r d) = 1 + max (altura i) (altura d)

esPreRama :: Eq a => AEB a -> [a] -> Bool
{E0} esPreRama (Hoja x) = \xs -> null xs || (xs == [x]))
{E1} esPreRama (Bin i r d) = \xs -> null xs ||
        (r == head xs && (esPreRama i (tail xs) || esPreRama d (tail xs)))
```

(==):: Eq a ⇒ [a] -> [a] -> Bool (Escrita en 4 casos para facilitar la demostración)

```
{==0} [] == [] = True
{==1} [] == (_:_) = False
{==2} (_:_) == [] = False
{==3} (x:xs) == (y:ys) = (x == y) && (xs == ys)
```

Lema:

$\forall t::AEB\ a\ .\ 0 \leq altura\ t$

Asumiendo Eq a, demostrar la siguiente propiedad:

$$\forall t::AEB\ a.\forall xs::[a].(esPreRama\ t\ xs \Rightarrow length\ xs \leq altura\ t)$$

Queremos ver que ∀t::AEB a.∀xs::[a].(esPreRama t xs ⇒ length xs ≤ altura t).

P(t) = ∀xs::[a].(esPreRama t xs ⇒ length xs ≤ altura t)

Hacemos inducción estructural en t, desplegando definiciones y propiedades hasta probar que la implicación es verdadera. No copiamos el ∀xs::[a] en cada paso, sabemos que está.

Caso t = Hoja x:

esPreRama (Hoja x) xs ⇒ length xs ≤ altura (Hoja x) = {A0}
esPreRama (Hoja x) xs ⇒ length xs ≤ 1 = {E0}
(\xs -> null xs || (xs == [x])) xs ⇒ length xs ≤ 1 = {β}
null xs || (xs == [x]) ⇒ length xs ≤ 1

Por lema de generación de listas, xs es de la forma [] o y:ys (para alguna y::a e ys::[a]).
Caso xs = []:
null [] || ([] == [x]) ⇒ length [] ≤ 1 = {L0}
null [] || ([] == [x]) ⇒ 0 ≤ 1 = {Int}
null [] || ([] == [x]) ⇒ True = {Bool} True
Caso xs = y:ys:
null (y:ys) || ((y:ys) == [x]) ⇒ length (y:ys) ≤ 1 = {N1}
False || ((y:ys) == [x]) ⇒ length (y:ys) ≤ 1 = {Bool}
((y:ys) == [x]) ⇒ length (y:ys) ≤ 1 = {==3}
(y == x) && (ys == []) ⇒ length (y:ys) ≤ 1
Por lema de generación de listas, ys puede ser [] o z:zs.
Caso ys = z:zs:
(y == x) && (z:zs == []) ⇒ length (y:ys) ≤ 1 = {==2}
(y == x) && False ⇒ length (y:ys) ≤ 1 = {Bool} True
Caso ys = []:
(y == x) && ([] == []) ⇒ length (y:[]) ≤ 1 = {L1}
(y == x) && ([] == []) ⇒ 1 + length [] ≤ 1 = {L0}
(y == x) && ([] == []) ⇒ 1 + 0 ≤ 1 = {Int}
(y == x) && ([] == []) ⇒ True = {Bool} True

Con esto queda probado el caso base t = Hoja x.

Caso t = Bin i r d:
Hipótesis inductiva: P(i) ∧ P (d), es decir:
(∀xs::[a].(esPreRama i xs ⇒ length xs ≤ altura i)) ∧ (∀xs::[a].(esPreRama d xs ⇒ length xs ≤ altura d))

esPreRama (Bin i r d) xs ⇒ length xs ≤ altura (Bin i r d) = {E1}
(\xs -> null xs ||(r == head xs && (esPreRama i (tail xs) || esPreRama d (tail xs)))) xs ⇒
length xs ≤ altura (Bin i r d) = {β}
null xs ||(r == head xs && (esPreRama i (tail xs) || esPreRama d (tail xs))) ⇒ length xs ≤ altura (Bin i r d) = {A1}
null xs ||(r == head xs && (esPreRama i (tail xs) || esPreRama d (tail xs))) ⇒
length xs ≤ 1 + max (altura i) (altura d)

Por lema de generación de listas, xs es [] o y:ys.

Caso xs = []:

null [] ||(r == head [] && (esPreRama i (tail []) || esPreRama d (tail []))) ⇒
length [] ≤ 1 + max (altura i) (altura d) = {L0}

null [] ||(r == head [] && (esPreRama i (tail []) || esPreRama d (tail []))) ⇒ 0 ≤ 1 + max (altura i) (altura d)

Por lema, 0 ≤ altura i ∧ 0 ≤ altura d.

Luego, por Int: 0 ≤ max (altura i) (altura d) ≤ 1 + max (altura i) (altura d).

Luego el consecuente de la implicación es verdadero, y por Bool la implicación es verdadera.

Caso xs = y:ys:

null (y:ys) ||(r == head (y:ys) && (esPreRama i (tail (y:ys)) || esPreRama d (tail (y:ys)))) ⇒
length (y:ys) ≤ 1 + max (altura i) (altura d)

= {T} null (y:ys) ||(r == head (y:ys) && (esPreRama i ys) || esPreRama d ys)) ⇒
length (y:ys) ≤ 1 + max (altura i) (altura d)

= {N1} False ||(r == head (y:ys) && (esPreRama i ys) || esPreRama d ys)) ⇒
length (y:ys) ≤ 1 + max (altura i) (altura d)

= {Bool} (r == head (y:ys) && (esPreRama i ys) || esPreRama d ys)) ⇒
length (y:ys) ≤ 1 + max (altura i) (altura d)

Sean:

A = r == head (y:ys)

B = (esPreRama i ys) || esPreRama d ys)

C = length (y:ys) ≤ 1 + max (altura i) (altura d)

Si probamos B ⇒ C, entonces, por Bool, vale (A ∧ B) ⇒ C.

Probemos que vale B ⇒ C.

esPreRama i ys || esPreRama d ys ⇒ length (y:ys) ≤ 1 + max (altura i) (altura d) = {L1}

esPreRama i ys || esPreRama d ys ⇒ 1 + length ys ≤ 1 + max (altura i) (altura d) = {Int}

esPreRama i ys || esPreRama d ys ⇒ length ys ≤ max (altura i) (altura d)

Por lema de generación de Bool, esPreRama i xs es True o False.

Caso True:

True || esPreRama d ys ⇒ length ys ≤ max (altura i) (altura d) = {Bool}

length ys ≤ max (altura i) (altura d)

Por HI, sabemos que: esPreRama i ys ⇒ length ys ≤ altura i.

Por caso True, True ⇒ length ys ≤ altura i.

Por Bool, length ys ≤ altura i.

Por Int, altura i ≤ max (altura i) (altura d).

Luego, por Int, length ys ≤ max (altura i) (altura d), que es lo que queríamos demostrar.

Caso False:

False || esPreRama d ys ⇒ length ys ≤ max (altura i) (altura d) = {Bool}

esPreRama d ys ⇒ length ys ≤ max (altura i) (altura d)

Por HI, esPreRama d ys ⇒ length ys ≤ altura d.

Por Int, altura d ≤ max (altura i) (altura d).

Luego, por Int, esPreRama d ys ⇒ length ys ≤ max (altura i) (altura d),

que es lo que queríamos demostrar. □