```
-- Definiciones:
data AB a = Nil | Bin (AB a) a (AB a)

-- elemAB :: Eq a => a -> AB a -> Bool
{P0} elemAB _ Nil = False
{P1} elemAB x (Bin i r d) = (x == r) || (elemAB x i) || (elemAB x d)

-- mapAB :: (a -> b) -> AB a -> AB b
{M0} mapAB _ Nil = Nil
{M1} mapAB f (Bin i r d) = Bin (mapAB f i) (f r) (mapAB f d)

{CONGRUENCIA ==} x == y => (f x) == (f y)

-- tener en cuenta que:
∀ x, y ,z:: Bool. (x => y) => (x => z V y)

Usando inducción estructural sobre t basta ver que:
1). P(Nil)
2). ∀ i,d:: AB a, ∀ r::a. (P(i) && P(d)) => P(Bin i r d)
Con P(t) ∀ t :: AB a, ∀ x:: a, ∀ f:: (a -> b). (elem x t => elem (f x)
(map f t))

Caso P(Nil)
elem x Nil => elem (f x) (map f Nil)
False => elem (f x) (map f Nil) ≡ por {P0}
True ≡ por {Bool}
Queda demostrado caso P(Nil)

Caso (P(i) && P(d)) => P(Bin i r d)
-- Sea
P(i) ≡ ∀ i :: AB a, ∀ x:: a, ∀ f:: (a -> b). (elem x i => elem (f x) (map
f i))
P(d) ≡ ∀ d :: AB a, ∀ x:: a, ∀ f:: (a -> b). (elem x d => elem (f x) (map
f d))
{HI}: P(i) && P(d)

-- Qvq P(Bin i r d):
elem x (Bin i r d) => elem (f x) (map f (Bin i r d))
(x == r) || (elemAB x i) || (elemAB x d) => elem (f x) (map f (Bin i r d))
≡ por {P1}
-- Por lema de generación de bool separo en casos:
A. x == r = True
B. x == r = False

Caso A.
True || (elemAB x i) || (elemAB x d) => elem (f x) (map f (Bin i r d)) ≡
True => elem (f x) (map f (Bin i r d)) ≡ por {Bool}
True => elem (f x) (Bin (mapAB f i) (f r) (mapAB f d)) ≡ por {M1}
True => (f x) == (f r) || (elemAB (f x) (mapAB f i)) || (elemAB (f x)
(mapAB f d)) ≡ por {P1}
-- Como x == r, por {Congruencia ==} f x == f r
True => True || (elemAB (f x) (mapAB f i)) || (elemAB (f x) (mapAB f d)) ≡
```

```
por {Congruencia}
True => True ≡ por {Bool}
True ≡ por {Bool}
-- Queda demostrado caso A.

Caso B.
False || (elemAB x i) || (elemAB x d) => elem (f x) (map f (Bin i r d))
(elemAB x i) || (elemAB x d) =>  elem (f x) (map f (Bin i r d)) ≡ por
{Bool}
-- Llamo ei || ed = (elemAB x i) || (elemAB x d)
ei || ed => elemAB (f x) (Bin (mapAB f i) (f r) (mapAB f d)) ≡ por {M1}
ei || ed => (f x) == (f r) || (elemAB (f x) (map f i)) || (elemAB (f x)
(map f d)) ≡ por {P1}
{-
Por {HI}  elemAB x i => elem (f x) (map f i))
y tambien elemAB x d => elem (f x) (map f d))
 Recordando que (x => y) => [x => z V y]
Tenemos que:
elemAB x i => elem (f x) (map f i)) V [(f x == f r) || (elemAB (f x) (map f
d))]
Idem caso:
elemAB x d => elem (f x) (map f d)) V [(f x == f r) || (elemAB (f x) (map f
i))]
Por lo tanto por {Bool} vale la implicación:
ei || ed => (f x) == (f r) || (elemAB (f x) (map f i)) || (elemAB (f x)
(map f d))
-}

-- Queda demostrado Caso B.

-- Como valen caso A y B queda demostrado caso P(Bin i r d)
-- Luego como valen caso P(Nil) y (P(i) && P(d) => P(Bin i r d)), queda
demostrado P(t).
```

Deducción natural

```
                                    ________________Ax  ___________Ax
                                    Γ, P => T ⊢ P => T  Γ, P => T ⊢ P
                    ______Ax          ________________________________=>e
                    Γ,Q ⊢ Q                     Γ, P => T ⊢ T
______________Ax    _________vi1         ___________________vi2
Γ ⊢ Q v (P => T)    Γ,Q ⊢ Q v T          Γ, P => T ⊢ Q v T
_____________________________________________________________ve
                    P, Q v (P => T) ⊢ Q v T
_____________________________________________=>i
          P ⊢ (Q v (P => T)) => (Q v T)
_________________________________________=>i
       ⊢ P => (Q v (P => T)) => (Q v T)

Γ = P, Q v (P => T)
```