Demostraciones

Enteros:

1. **(∀n:: N − {0}) sum1 n =**

**Funcion:**  
sum1::Int->Int  
sum1 0 = 0  
sum1 n = (3.n)-2

**Paso base ( n = 1 )**  
sum1 1 =   
def sum1|distributiva y resta  
(3.1)-2 = 2/2  
1 = 1  
**Paso inductivo: (n=n+1)**  
**Hi)** sum1 n = **Ti)** sum1 (n+1) = **Demostración:  
#1**  
sum (n+1) =  **def sum1**3.(n+1)+sum n  
**Hi) y distributiva**3n+3-2+ **distributiva**3n+1+ **todo sobre 2**odo  
cuentas  
**#2**

= **distributiva**=   
**= distributiva  
= cuentas**

**2) (∀n :: N) dosala (n + 1) = (sum2 n) + 1  
Funciones:**  
dosala::Int->Int  
dosala 0 = 1  
dosala n = 2 \* dosala (n-1);  
  
sum2::Int->Int  
sum2 0 = dosala 0  
sum2 n = dosala n + sum2 (n-1);  
  
**Paso base ( n = 0)**  
dosala 1 = sum2 0 +1  
= def dosala y sum2  
2\*dosala 0 = dosala 0 +1  
=def dosala  
2\*1 = 1+1  
2=2  
**Paso Inductivo ( n = n+1 )**Hi) dosala (n+1) = (sum2 n ) +1  
Ti) dosala ((n+1)+1) = ( sum2 (n+1) )+1  
Demostración:  
dosala(n+2) = (sum2 (n+1))+1  
**= def dosala y sum2**  
2\*(dosala(n+1)) = (dosala (n+1)+sum2 n)+1  
**=Hi) en ambos**2\*( sum2 n +1) = sum2 n +1 sum2 n +1  
= factor común  
2\*sum2n +2 = 2\*(sum2 n +1)

**LQQD**

Listas:  
Demosrar que (∀t1, t2::[a]) prefijo (unir t1 t2) t1 = True  
Funciones:  
prefijo::Eq a => [a]->[a]->Bool  
prefijo [] l = True  
prefijo (x:xs) (y:ys)  
 | x == y = prefijo xs ys   
 | otherwise = False;  
unir::[a]->[a]->[a]  
unir [] l = l  
unir (x:xs) l2 = x: (unir xs l2);  
  
Paso base ( lista vacia)  
prefijo(unir [] l2) [] = True  
= definicio unir   
prefijo l2 [] = True  
= def prefijo   
True = True  
Paso inductivo ( x:xs)  
Hi) prefijo (unir (xs) l2) (xs) = True  
Ti)prefijo (unir (x:xs) l2 ) (x:xs) = True

**Demostracion**  
prefijo (unir (x:xs) l2 ) (x:xs)  
**= definicion unir**prefijo ( x : (unir xs l2 ) (x:xs)  
**= definición prefijo**  
**(Caso == )**prefijo (unir xs l2) xs   
**= Hi)**True = True  
**(caso otherwise)**nunca entra por que se esta pasando la misma lista que se unió con l2  
por ende tendrán siempre el mismo comienzo.  
  
**LQQD**