Resolución parcial I

1. Demuestre que para todo natural de peano "n" se cumple la siguiente propiedad:

Succ 0 + n = Succ n

Utilice la definición de *suma* estudiada en clase como definición del signo "+".

Definiciones de la suma:

1. n+0 = 0
2. 0+m = m
3. n + Succ a = Succ (n+a)
4. a + Succ b = Succ a + b

Siguiendo las propiedades de la suma establecidas anteriormente, se puede realizar lo siguiente:

Succ 0 + n = Succ n

n + Succ 0 = Succ n

Succ (n + 0)= Succ n

Succ n = Succ n

2. Provea una definición inductiva para la propiedad "mayor que" (>) tal que:

$$a > b \begin{cases} Succ\ 0 & si\ a\ es\ mayor\ que\ b \\ 0 & de\ lo\ contrario \end{cases}$$

En otras palabras, la propiedad "mayor que" es equivalente a Succ 0 si el primer valor es mayor que el segundo o 0 de lo contrario. Puede utilizar el operador ">" en su definición de la misma manera que se utiliza "+" en la definición de suma.

Definiciones de la suma:

1. n+0 = 0
2. 0+m = m
3. n + Succ a = Succ (n+a)
4. a + Succ b = Succ a + b

Definiciones

1. a = b + n (siendo “n” un número natural)
2. a > b

1. a = b + 0 (siempre que “a” y “b” sean un numero natural y no sean igual a 0)
2. a > 0

1. b = a + n (siendo “n” un número natural)
2. a < b

1. b = a + n (siendo “n” un numero natural y que no sea 0)
2. Succ (a) = Succ (b)
3. Succ (a) > Succ (b)

1. a = b + n
2. Succ (a) = Succ (b)
3. Succ (a) < Succ b

Entonces podemos definir que “a > b“ siempre que “a=b + n” y si “b = a + n” entonces “a < b”. Al igual para los números de peano “Succ a > Succ b” siempre que “Succ a = Succ (b+ n), si n pertenece a los números naturales y no es 0” y si “Succ b = Succ (a + n), Succ a < Succ b siempre que n pertenezca a los números naturales y no sea 0”.

4. Provea una definición de las propiedades "esPar" e "esImpar" tal que:

$$esPar\ n \begin{cases} Succ\ 0 & si\ n\ es\ un\ numero\ par \\ 0 & de\ lo\ contrario \end{cases}$$

$$esImpar \begin{cases} Succ\ 0 & Si\ n\ es\ impar \\ 0 & de\ lo\ contrario \end{cases}$$

Se sugiere que, para implementar estas propiedades, utilice la propiedad inversa en la definición. En otras palabras, un numero "n" es par cuando cierto otro número es impar y vice versa.

Definición:

- Un número “n” es par cuando “n – 1” es impar y vece versa.
- Par n = Succ 0 + n (siendo n un numero impar)

- Impar n = Succ 0 + n (siendo n un numero par)
- Succ (Succ 0)) * n (da como resultado un numero par)
- Succ (Succ 0)) * n + Succ 0 (da como resultado un numero impar

5. Utilice el lenguaje de programación Haskell para definir la propiedad "predecesor". Esta propiedad debe aceptar un numero de peano y producir el predecesor de este. En el caso de cero, utilizar cero como su predecesor.

```
{-# LANGUAGE NoImplicitPrelude #-}
    module Main where

    import Prelude (Show, undefined, appendFile, (-), Foldable (sum) )

    data Natural = O | Succ Natural deriving Show

--De numeros naturales a numeros de peano
-- Anat 6 = Succ......
    anat 0 = O
    anat n = Succ (anat (n - 1))

--Predecesor
    pred (Succ O) = O
    pred (Succ a) = a

main = undefine
```