Estructuras Discretas

Recursión en Naturales

Liliana Reyes

Universidad Nacional Autónoma de México
Facultad de Ciencias

7 de febrero de 2023

Definición (Nat)

El conjunto de números naturales Nat se define recursivamente como:

1. $0 \in Nat$.
2. Si $x \in Nat$ entonces $suc(x) \in Nat$.
3. Solo los anteriores son Nat

La función suc(x) se define como $x + 1$

Ejemplo

Notación decimal		Definición recursiva
0	$\rightarrow$	0
1	$\rightarrow$	$suc(0) = 0 + 1 = 1$
2	$\rightarrow$	$suc(suc(0)) = 1 + 1$
3	$\rightarrow$	$suc(suc(suc(0))) = 2 + 1$
4	$\rightarrow$	$suc(suc(suc(suc(0)))) = 3 + 1$
5	$\rightarrow$	$suc(suc(suc(suc(suc(0))))) = 4 + 1$

Operaciones con Naturales

Igualdad

Definición (Igualdad de naturales)

La función *igualesNat(x,y)* recibe dos naturales y devuelve *True* si son los mismos, en caso contrario devuelve *False*

1. **Casos bases**, que corresponde a los casos donde X y/o Y son 0.

$$\begin{aligned} igualesNat(0,0) &= True \\ igualesNat(0,suc(n)) &= False \\ igualesNat(suc(m),0) &= False \end{aligned}$$

2. **Caso recursivo**, dos naturales $s(n)$ y $s(m)$ son iguales si sus predecesores n y m son iguales.

$$igualesNat(suc(n),suc(m)) = igualesNat(n,m)$$

Operaciones con Naturales

Suma

Definición (Suma de naturales.)

La función *sumaNat(x,y)* recibe dos naturales y devuelve su suma.

- **Caso base:** se considera que cualquiera de los dos parámetros son 0.

$$\begin{aligned} sumaNat(n,0) &= n \\ sumaNat(0,n) &= n \end{aligned}$$

- **Caso recursivo:** se destruye el segundo argumento, mientras se agrega un sucesor al primer argumento, con el fin de llegar al caso base.

$$sumaNat(n, suc(m)) = sumaNat(suc(n), m)$$

Operaciones en Naturales

Duplicar

```
sumaNat 0 m = m
sumaNat n 0 = n
sumaNat n (suc(m)) = sumaNat (suc(n)) m
```

```
sumaNat (suc(suc(0))) (suc(suc(suc(0)))) =
    sumaNat (suc(suc(suc(0)))) (suc(suc(0))) =
        sumaNat (suc(suc(suc(suc(0))))) (suc(0)) =
            sumaNat (suc(suc(suc(suc(suc(0)))))) 0 =
                suc(suc(suc(suc(suc(0)))))
```

Definición (Producto de naturales.)

La función *prodNat(x,y)* recibe dos naturales y entrega el producto de ellos.

p1. **Caso base:** si multiplicamos 0 por cualquier natural es resultado es 0.

$$\begin{aligned} prodNat(n,0) &= 0 \\ prodNat(0,m) &= 0 \end{aligned}$$

p2. **Caso recursivo:** considerando a la multiplicación n*m como una suma abreviada de sumar n, m veces. Entonces, n*m tambien se puede reescribir como la suma de n con la multiplicación de n*(m-1). De estas manera se reduce el problema destruyendo el segundo argumento hasta llegar al caso base.

$$prodNat(n,suc(m)) = sumaNat(n,prodNat(n,m))$$

2*3 = 2+2+2

prodNat n 0 = 0
prodNat 0 n = 0
prodNat n suc(m) = sumaNat n (prodNat n m))

prod suc(suc(0)) suc(suc(suc(0)) =
sumaNat suc(suc(0)) (prodNat suc(suc(0)) suc(suc(0))) =
sumaNat suc(suc(0)) (suc(suc(suc(suc(0))) = ... =
suc(suc(suc(suc(suc(suc(0))))))

prodNat suc(suc(0)) suc(suc(0)) =
sumaNat suc(suc(0)) (prodNat suc(suc(0)) suc(0)) =
sumaNat suc(suc(0)) suc(suc(0)) =...= suc(suc(suc(suc(0))))

prodNat suc(suc(0)) suc(0) =
sumaNat suc(suc(0)) (prodNat suc(suc(0)) 0) = sumaNat suc(suc(0)) 0
= suc(suc(0))

Operaciones con Naturales

Duplicar

Definición (Duplica naturales)

La función *duplicaNat(x,y)* recibe un natural n y devuelve el doble de este. Ejemplo: *duplicaNat(2) = 4* Las reglas son:

- **Caso base:** El doble de 0 es 0

$$duplicaNat(0) \quad = \quad 0$$

- **Caso recursivo:** La regla sería, el doble de un natural n equivale a sumar 2 más el doble de predecesor de n.

$$duplicaNat(suc(n)) \quad = \quad sumaNat\ \ (suc(suc(0))\ \ duplicaNat(n))$$

Operaciones en Naturales

Duplicar

duplica 0 = 0
duplica suc(n) = sumaNat (suc(suc(0))) (duplica n)

duplica (suc(suc(suc(0)) = sumaNat (suc(suc(0))) (duplica suc(suc(0))
= sumaNat (suc(suc(0))) suc (suc (suc(suc(0))))
= suc (suc (suc(suc(suc(suc(0))))))

duplica suc(suc(0)) = sumaNat (suc(suc(0))) (duplica suc(0))
=sumaNat (suc(suc(0))) (suc (suc (0))
= suc (suc (suc(suc(0))))

duplica suc(0) = sumaNat (suc (suc (0)) 0 = (suc (suc (0))

Ejercicios

Define **recursivamente** las siguientes funciones:

1. *mayorNat(n,m)* verifica si el primer natural es mayor al segundo.
2. *mayorIgualNat(n,m)* verifica si el primer natural es mayor o igual al segundo.
3. *menorNat* verifica si el primer natural es menor al segundo.
4. *menorIgualNat(n,m)* verifica si el primer natural es menor o igual al segundo.
5. *impar(n)* que verifica si un número es impar.
6. *potNat(n,m)* calcula la potencia de n^m.