FUNDAMENTOS DE COMPUTACIÓN
Repartido: Los números Naturales - Parte 3

5. La matemática de N

Como ya vimos, los números naturales constituyen una secuencia infinita de objetos, que empieza en el 0 y luego cada objeto se obtiene aplicando el constructor S al anterior:
0, S 0, S(S 0), S(S(S 0)), . . .

La pregunta es ahora: ¿Cómo demostramos afirmaciones de la forma (∀ **n :: N) P(n)**?

El método ya por todos conocido se denomina **inducción matemática.**
El mismo consiste en recorrer a todos los casos particulares, específicamente a **deducir**, a partir de las demostraciones **de la validez de una propiedad en cada natural**, la conclusión de **que ella vale para todos los números naturales**.
En otras palabras, debemos recorrer la secuencia completa de los naturales verificando la validez de la propiedad en cuestión en cada caso.

Pero, si la secuencia de los naturales es infinita... De qué manera puede realizarse una demostración por inducción si ella consiste en recorrer una secuencia infinita?
La respuesta es simple: lo que debe darse es **un dispositivo o procedimiento finito que muestre que todos los naturales tienen la propiedad en cuestión**.

La situación es la misma que afrontamos al comienzo de los repartidos precedentes: allí debíamos generar un número infinito de objetos por medio de un dispositivo finito (que llamaríamos la declaración del tipo de esos objetos, o el esquema de recursión estructural para definir funciones).

En el caso de una demostración de que todos los naturales satisfacen una propiedad P, debemos convencernos de la validez de un número infinito de proposiciones, o sea, de que se cumplen P(0), P(S 0), P(S(S 0)),... lo cual deberá necesariamente tener lugar por medio del empleo de algún dispositivo finito.

6. La inducción primitiva

Un método para demostrar una propiedad para todos los números naturales consiste en elegir un cubrimiento de la secuencia de números por medio de casos base y operaciones de modo que sea posible demostrar que cada caso base tiene la propiedad así como que ésta es propagada por cada operación considerada.

Ahora bien, la forma más simple de conseguir el cubrimiento de la secuencia de los números naturales es apegarse a su definición, o sea, considerar 0 como único caso base y la función sucesor S como única propagadora.
Esto da lugar al principio estudiado ya en Secundaria, que aquí llamaremos de **inducción primitiva**:

Para demostrar (∀ n :: N) P(n) es suficiente demostrar:

1. **<u>Caso cero</u>: P(0) se cumple.**
2. **<u>Caso S</u>: (∀ x :: N) P(S x) se cumple, asumiendo que se cumple P(x).**

La justificación de este método la daremos de la siguiente manera:
Debemos mostrar que si se cumplen los dos componentes (**1 y 2**) del método, entonces todos los naturales cumplen **P**.
Ahora bien, **0** cumple **P** debido a que vale **P(0)** por **(1)**.
Luego **S 0** también cumple **P** debido a que por **(2) S** propaga la propiedad.
Pero entonces también **S(S 0)** la cumplirá, y luego **S(S(S 0))**, **S(S(S(S 0)))**, y así sucesivamente **todos los números que se obtengan por aplicar S a uno ya obtenido previamente cumplirán P**.
Pero estos son todos los naturales, puesto que la definición del tipo **N** establece que los objetos de este son precisamente los generados por los constructores **0** y **S**.
Esto concluye la justificación deseada.

Es muy importante comprender en detalle el enunciado del caso **S**, el cual llamaremos **paso inductivo**:
Específicamente, debemos demostrar que **para cualquier objeto x** de nuestra secuencia de naturales se cumple que, si éste cumple la propiedad P, entonces su sucesor la cumplirá también. O sea, debemos demostrar que **(∀ x :: N) P(S x) se cumple, asumiendo que se cumple P(x)**, lo cual podemos escribir del siguiente modo: **(∀ x :: N) P(x) ⇒ P(S x)**.
Es muy importante entender que para demostrar este paso, **no podemos asumir que tenemos la hipótesis (∀ x :: N) P(x)**, ya que esto sería equivalente a cometer la trampa de suponer que se cumple lo que se quiere demostrar!.
En cambio, para demostrar que (∀ x :: N) P(x) ⇒ P(S x) **deberemos considerar un natural arbitrario x que cumpla la propiedad P, y probar que S x también la cumple**.
Como es costumbre en matemática, lo que se asume es llamado **hipótesis** y lo que debe demostrarse se llama **tesis**, por lo que en el caso **S**, se debe elegir **un x :: N arbitrario**, y luego **P(x)** se llamará **hipótesis inductiva** (abreviado **HI**) y **P(S x)** se llamará **tesis inductiva** (abreviado **TI**).

Dicho esto, las demostraciones de afirmaciones de la forma (∀ x :: N) P(x) por inducción primitiva, serán de la forma:

Caso n = 0: P(0)
Dem. .....
.....
Caso n = S x, con x :: N cualquiera.
HI) P(x)
TI) P(S x)
Dem.....
.....

7. Ejemplos

Empecemos a demostrar algunas propiedades sencillas de las funciones definidas en el repartido anterior.
Recordemos para ello las siguientes definiciones:

par :: N → Bool
par = λn→ case n of { 0 → True ; S x → not (par x) }

doble :: N → N
doble = λn→ case n of { 0 → 0 ; S x → S (S (doble x)) }

La primera propiedad que demostraremos vincula a estas dos funciones, y dice que el doble de todo natural es par:

$\text{Lema}_{\text{pardoble}}$: (∀ n :: N) par (doble n) = True.

Del mismo modo que con las demostraciones de propiedades de booleanos, es muy importante distinguir cuál es la propiedad **P** a la cual nos estamos refiriendo, ya que en ella deberemos poder sustituir la variable genérica **n** por **0**, **x** o **S x** en distintos lugares de la demostración.
La propiedad **P** a demostrar en este caso es **par (doble ●) = True.**

La demostración será entonces:

Demostración : Por inducción (primitiva) en n :: N.
Caso n = 0: par (doble 0) $=^?$ True *{o sea, P(0)}*
Dem. par (doble 0)
= (def. doble, β, case)
par 0
= (def. par, β, case)
True.
Caso n = S x con x :: N cualquiera
HI) par (doble x) = True *{o sea, P(x)}*
TI) par (doble (S x)) $=^?$ True *{o sea, P(S x)}*
Dem. par (doble (S x))
= (def. doble, β, case)
par (S (S (doble x)))
= (def. par, β, case)
not (par (S (doble x)))
= (def. par, β, case)
not (not (par (doble x)))
= (HI)
not (not True)
= (Lema_{not})
True. □

Otra propiedad de la función doble que podemos demostrar, vincula a la suma de naturales definida como:

(+) :: N → N → N
(+) = λm n→ case m of { 0 → n ; S x → S (x + n) }

La propiedad P a demostrar es: **doble ● = ● + ●.**

Enunciamos y demostramos entonces el siguiente lema:

Lema$_{doble+}$: (∀ n :: N) doble n = n + n
Dem. Por inducción en n :: N
Caso n = 0: doble 0 =$^{?}$ 0 + 0 *{P(0)}*
Dem.

doble 0	0 + 0
= (def. doble, β, case)	= (def. +, βx2, case)
0	0

Ambas expresiones son iguales por reducir a la misma expresión.

Caso n = S x con x :: N cualquiera
HI) doble x = x + x *{P(x)}*
TI) doble (S x) =$^{?}$ S x + S x *{P(S x)}*
Dem.

doble (S x)	S x + S x
= (def. doble, β, case)	= (def. +, βx2, case)
S (S (doble x))	S (x + S x)
= (HI)	= (*)
S (S (x + x))	S (S (x + x))

Ambas expresiones son iguales por reducir a la misma expresión. □

Observemos que el paso (*) no puede demostrarse directamente haciendo cuentas (def, β, case), ya que el valor de x + S x depende del valor de x.
Esa igualdad es un caso particular de un resultado más general:

(∀ m :: N)(∀ n :: N) m + S n = S (m + n)

Antes de empezar con la demostración, debemos decidir en cuál de los dos naturales (m o n) haremos la inducción, ya que una decisión mal tomada puede hacer que la demostración no salga tan directamente.
Si observamos la definición de la suma, vemos que se hace por casos en el primer argumento, por lo que convendrá hacer inducción en m en este caso.
Una vez decidido esto, la propiedad a demostrar es: **(∀ n :: N) ● + S n = S (● + n).**

Lema$_{+S}$: (∀ m :: N)(∀ n :: N) m + S n = S (m + n)
Dem. Por inducción en m :: N
Caso m = 0: (∀ n :: N) 0 + S n =$^{?}$ S (0 + n) *{P(0)}*
Dem. Sea n :: N cualquiera.

0 + S n	S (0 + n)
= (def. +, βx2, case)	= (def. +, βx2, case)
S n	S n

Ambas expresiones son iguales por reducir a la misma expresión

Caso m = S x con x :: N cualquiera
HI) (∀ n :: N) x + S n = S (x + n) *{P(x)}*
TI) (∀ n :: N) S x + S n =$^{?}$ S (S x + n) *{P(S x)}*
Dem. Sea n :: N cualquiera.

S x + S n	S (S x + n)
= (def. +, βx2, case)	= (def. +, βx2, case)
S (x + S n)	S (S (x + n))
= (HI)	
S (S (x + n))	

Ambas expresiones son iguales por reducir a la misma expresión. □

Observemos que en los casos de la demostración, no fue necesario hacer inducción en n, ya que ambas (sub)demostraciones pudieron hacerse para un n genérico.

El lema recién demostrado dice que el sucesor a izquierda de una suma puede "sacarse" para afuera (si hay un sucesor a la derecha de una suma se lo puede sacar por definición de la suma para el caso S).

Podemos demostrar un resultado similar para el cero: **(∀ n :: N) n + 0 = n**
Otra vez que esta igualdad no puede demostrarse directamente haciendo cuentas, ya que el valor de n + 0 depende del valor de n. La otra igualdad (0 + n = n) sí puede demostrarse directamente por definición de +, βx2 y case.

Lema$_{+0}$: (∀ n :: N) n + 0 = n
Dem. Por inducción en n :: N
Caso n = 0: 0 + 0 =$^{?}$ 0
Dem. 0 + 0
= (def. +, βx2, case)
0
Caso n = S x con x :: N cualquiera
HI) x + 0 = x
TI) S x + 0 =$^{?}$ S x
Dem. S x + 0
= (def. +, βx2, case)
S (x + 0)
= (HI)
S x □

Con estos dos lemas, podemos demostrar fácilmente la conmutatividad de la suma, o sea que se cumple m + n = n + m para todos m y n naturales.

En este caso, podemos hacer inducción en cualquiera de los dos argumentos, ya que en cada lado de la igualdad aparece uno de ellos como primer argumento de la suma.
Queda como ejercicio para el lector definir cuál es la propiedad P que se está demostrando.
Haremos entonces inducción en **m**:

$Lema_{+conm}$: ($\forall$ m :: N)($\forall$ n :: N) m + n = n + m.
Dem. Por inducción en **m :: N**
Caso m = 0: **($\forall$ n :: N) 0 + n =$^{?}$ n + 0**
Dem. Sea n :: N cualquiera.

0 + n	| n + 0
= (def. +, βx2, case)	| = ($Lema_{+0}$)
n	| n

Ambas expresiones son iguales por reducir a la misma expresión

Caso m = S x con x :: N cualquiera
HI) ($\forall$ n :: N) x + n = n + x
TI) ($\forall$ n :: N) S x + n =$^{?}$ n + S x
Dem. Sea n :: N cualquiera.

S x + n	| n + S x
= (def. +, βx2, case)	| = ($Lema_{+S}$)
S (x + n)	| S (n + x)
= (HI)	
S (n + x)	

Ambas expresiones son iguales por reducir a la misma expresión. □

Como ejercicio, también puede demostrarse la asociatividad de la suma de naturales, que se demuestra sencillamente haciendo inducción en m:

$Lema_{+asoc}$: ($\forall$ m :: N)($\forall$ n :: N)($\forall$ q :: N) m + (n + q) = (m +n) + q

Ejercicio: haciendo uso de la siguiente definición del producto y los lemas de la suma demostrados arriba, demostrar las siguientes propiedades:

(*) :: N → N → N
(*) = λm n→ case m of { 0 → 0 ; S x → n + (x * n) }

1. **$Lema_{*0}$: ($\forall$ n :: N) m * 0 = 0**
2. **$Lema_{*S}$($\forall$ m :: N)($\forall$ n :: N) m * S n = m + m * n**
3. **$Lema_{*conm}$($\forall$ m :: N)($\forall$ n :: N) m * n = n * m**
4. **$Lema_{*dist}$($\forall$ m :: N)($\forall$ n :: N)($\forall$ q :: N) (m + n) * q = (m * q) + (n * q)**
5. **$Lema_{*asoc}$($\forall$ m :: N)($\forall$ n :: N)($\forall$ q :: N) (m * n) * q = m * (n * q)**

8. La igualdad

Finalmente,recordemos la definición de la igualdad de naturales:

(==) :: N → N → Bool
(==) = λm n→ case m of { 0 → case n of { 0 → True ; S y → False } ;
S x → case n of { 0 → False ; S y → x == y } }

Podemos demostrar que esta función, pensada como una relación binaria, es reflexiva, o sea:

$Lema_{==refl}$: (∀ n :: N) n == n = True

Dejamos la demostración como ejercicio.

También podemos demostrar que **==** es conmutativa: **(∀ m :: N)(∀ n :: N) m == n = n == m.** Para demostrar este lema, podemos hacer inducción en uno de los dos naturales, por ejemplo m, quedando la propiedad P a demostrar como **(∀ n :: N) ● == n = n == ●**.

Antes de empezar la demostración, analicemos lo que deberemos demostrar.

- En el caso 0, tendremos que demostrar que **(∀ n :: N) 0 == n = n == 0**.
Observemos que para demostrar esta igualdad será necesario hacer inducción en n, ya que no podemos saber cuánto valen las dos expresiones de ambos lados del igual, a menos que sepamos el valor de n.

- Y si analizamos el caso S, rápidamente podemos darnos cuenta de que pasará lo mismo...

Entonces, la demostración deberá proceder del siguiente modo:

$Lema_{==conm}$(∀ m :: N)(∀ n :: N) m == n = n == m
Dem. Por inducción en **m :: N**
Caso m = 0: **(∀ n :: N) 0 == n =$^{?}$ n == 0**
Dem. Por inducción en **n :: N**
Caso n = 0: **0 == 0 =$^{?}$ 0 == 0**
Se cumple por reflexividad del =.
Caso n = S y con y :: N cualquiera
HI) 0 == y = y == 0
TI) 0 == S y =$^{?}$ S y == 0
Dem.$^{(**)}$

0 == S y	S y == 0
= (def. ==, βx2, casex2)	= (def. ==, βx2, casex2)
False	False

Ambas expresiones son iguales por reducir a la misma expresión.

Caso m = S x con x :: N cualquiera
HI) (∀ n :: N) x == n = n == x
TI) (∀ n :: N) S x == n ==$^{?}$ n == S x

Dem. Por inducción en **n :: N**

Caso n = 0: **S x == 0 =$^{?}$ 0 == S x**

Se demostró en (**) arriba, por lo que se cumple por simetría del =.

Caso n = S x

y :: N cualquiera

HI2) S x == y = y == S x

TI2) S x == S y =$^{?}$ S y == S x

Dem

S x == S y	| S y == S x
= (def. ==, βx2, casex2)	| = (def. ==, βx2, casex2)
x == y	| y == x
= (HI para n = y)$^{(***)}$	
y == x	

Ambas expresiones son iguales por reducir a la misma expresión. □

Observar que al haber una inducción (en n) “adentro” del caso S de la inducción principal, tenemos dos hipótesis inductivas:

- **HI) (∀ n :: N) x == n = n == x**
- **HI2) S x == y = y == S x**

La hipótesis que usamos en (***) es la hipótesis de afuera (que llamamos HI), que es la que nos sirve, ya que al estar cuantificada universalmente puede instanciarse para el y que necesitamos.

Recalcamos la importancia de escribir correctamente la propiedad P en las hipótesis y tesis que aparecen en las demostraciones por inducción. En este caso en particular, la propiedad P de la inducción principal era **(∀ n :: N) ● == n = n == ●**. Ésta tenía un **∀ n**, sin el cual nos hubiera sido imposible aplicar la HI para concluir y == x a partir de x == y!

9. El menor o igual

La relación ≤ en la matemática se define como **m ≤ n = (∃ k :: N) m + k = n.**

Además, ya vimos una definición para esta relación como una función en Haskell:

```
(<=) :: N → N → Bool
(<=) = \m n → case m of { 0 → True ;
                          S x → case n of { 0 → False ; S y → x <= y } }
```

Nos interesa demostrar que nuestra función Haskell refleja la definición matemática del ≤. Para hacerlo, podemos enunciar esta propiedad de la siguiente manera:

(∀ m:: N)(∀ n:: N) m <= n = True ⇔ (∃ k:: N) m+k=n

Para escribir la demostración de esta propiedad, observemos primero que estamos demostrando un “si y sólo sí”, o sea una afirmación de la forma **A ⇔ B**.

A ⇔ B se define en lógica como **A ⇒ B y B ⇒ A,** por lo que para demostrar **A ⇔ B** podemos:

- demostrar **ambas implicaciones**, o

- verificar que **ambas afirmaciones tienen el mismo valor de verdad**.

En efecto, el ⇔ se llama también **equivalencia lógica**, y A ⇔ B es verdadero cuando **A y B son ambas verdadera o ambas falsas**, porque:

- Si A es verdadera, B también debe serlo porque debe cumplirse A ⇒ B.

- Si A es falsa, B no puede ser verdadera, porque en ese caso no se cumpliría B ⇒ A, o sea, B debe ser falsa.

⇔ es además una **relación de equivalencia**, o sea, es reflexiva, simétrica y transitiva. En particular, la transitividad de ⇔ del nos permite hacer “cadenas” de afirmaciones equivalentes, y razonar ecuacionalmente con las mismas.

También observemos que la parte derecha del ⇔ es de la forma **(∃ k:: N)...**
Para demostrar un existe, basta con exhibir un objeto (en este caso, un natural) y probar que ese objeto cumple con la propiedad correspondiente.

Ahora sí, demostramos el lema de corrección de la función **(<=)** con respecto a la noción matemática de ≤:

Lema$_{Corr<=}$: **(∀ m:: N)(∀ n:: N) m <= n = True ⇔ (∃ k:: N) m + k = n**
Dem. Por inducción en m:: N
Caso m = 0: **(∀ n:: N) 0 <= n = True ⇔ (∃ k:: N) 0 + k = n**

Dem.

- Observemos que el lado izquierdo del ⇔ se cumple (por def. de <=, βx2 y case).
- Por otro lado, el lado derecho también se cumple, ya que existe un k:: N (en este caso, el mismo n) que cumple la igualdad 0 + k = n (por def. de +, βx2 y case).

Luego, como ambas afirmaciones a los lados del ⇔ son verdaderas, el mismo se cumple.

Caso m = S x con x :: N cualquiera

HI) $(\forall$ **n:: N) x <= n = True** $\Leftrightarrow$ **(** $\exists$ **k:: N) x + k = n**

TI) $(\forall$ **n:: N) S x <= n = True** $\Leftrightarrow^{?}$ **(** $\exists$ **k:: N) S x + k = n**

Dem. Por inducción en n:: N

Caso n = 0: S x <= 0 = True $\Leftrightarrow^{?}$ **(** $\exists$ **k:: N) S x + k = 0**

Dem.

- Observemos que el lado izquierdo del $\Leftrightarrow$ NO se cumple, ya que por def. de <=, βx2 y case x2 tenemos que S x <= 0 = False, y como True y False son constructores distintos, no pueden ser iguales.
- Por otro lado, S x + k = S (x + k) (por def. de +, βx2 y case), entonces, si existiera k que cumple con la parte derecha del $\Leftrightarrow$, tendríamos un natural (x + k en este caso) tal que su sucesor es igual a 0. Pero recordemos que S es un constructor del tipo N, y por lo tanto genera siempre objetos nuevos en el tipo, por lo que nunca puede cumplirse que 0 sea igual a S de un natural. Luego, no puede existir k tal que S x + k = 0, y por lo tanto la parte derecha del $\Leftrightarrow$ no se cumple tampoco.

Entonces, cómo ambas afirmaciones a los lados del $\Leftrightarrow$ son falsas, el mismo se cumple.

Caso n = S y con y :: N cualquiera

HI2) S x <= y = True $\Leftrightarrow$ **(** $\exists$ **k:: N) S x + k = y**

TI2) S x <= S y = True $\Leftrightarrow^{?}$ **(** $\exists$ **k:: N) S x + k = S y**

Dem.

S x <= S y = True

$\Leftrightarrow$ (def. de <=, βx2, casex2)

x <= y = True

$\Leftrightarrow$ (HI de afuera para n = y)

$(\exists$ k:: N) x + k = y

$\Leftrightarrow$ (aplicamos S de ambos lados de la igualdad)

$(\exists$ k:: N) S(x + k) = S y

$\Leftrightarrow$ (def. de +, βx2, casex)

$(\exists$ k:: N) S x + k = S y.

Acabamos de demostrar que la afirmación **S x <= S y = True** es equivalente a la afirmación **(** $\exists$ **k:: N) S x + k = S y**, con lo que este caso queda demostrado.

Propiedades de ≤

Todas las propiedades de la relación matemática ≤ pueden demostrarse usando su definición: **m ≤ n = (** $\exists$ **k:: N) m + k = n**.

Por ejemplo,

Lema$_{0\leq}$: ($\forall$ **n :: N) 0 ≤ n**

Dem. Sea n :: N.

Tomando k = n, tenemos que 0 + n = n, por def. de +, βx2 y case.

Lema$_{\leq refl}$: ($\forall$ n :: N) n $\leq$ n

Dem. Sea n :: N.

Tomando k = 0, tenemos que n + 0 = n, por Lema$_{+0}$.

Lema$_{\leq S}$: ($\forall$ n :: N) n $\leq$ S n

Dem. Sea n :: N.

Tomando k = S 0, tenemos que n + S 0 = S n, por Lema$_{+S}$ y Lema$_{+0}$.

Lema$_{\leq SS}$: ($\forall$ m :: N)($\forall$ n :: N) m $\leq$ n $\Rightarrow$ S m $\leq$ S n

Dem. Sean m, n :: N.

m $\leq$ n $\Rightarrow$ ($\exists$ k:: N) m + k = n (1).

Tomando el mismo k, tenemos que:

S m + k
= (def. de +, βx2 y case)
S (m + k)
= (1)
S n

Lema$_{\leq +}$: ($\forall$ m :: N)($\forall$ n :: N) m $\leq$ m + n

Dem. Sean m, n :: N.

Tomando k = n, tenemos que m + n = m + n, por reflexividad del =.

Lema$_{\leq ++}$: ($\forall$ m_1 :: N)($\forall$ m_2 :: N)($\forall$ n_1 :: N)($\forall$ n_2 :: N) $m_1 \leq n_1$ y $m_2 \leq n_2$ $\Rightarrow$ $m_1 + m_2 \leq n_1 + n_2$

Dem. Sean m_1, m_2, n_1, n_2 :: N.

$m_1 \leq n_1$ $\Rightarrow$ ($\exists$ k_1:: N) $m_1 + k_1 = n_1$ (1)

$m_2 \leq n_2$ $\Rightarrow$ ($\exists$ k_2:: N) $m_2 + k_2 = n_2$ (2)

Tomando $k = k_1 + k_2$, tenemos que:

$(m_1 + m_2) + (k_1 + k_2)$
= (asociatividad y conmutatividad de +)
$(m_1 + k_1) + (m_2 + k_2)$
= (1 y 2)
$n_1 + n_2$

Lema$_{\leq trans}$: ($\forall$ m :: N)($\forall$ n :: N)($\forall$ q :: N) m $\leq$ n y n $\leq$ q $\Rightarrow$ m $\leq$ q

Dem. Sean m, n, q :: N.

m $\leq$ n $\Rightarrow$ ($\exists$ k_1:: N) $m + k_1 = n$ (1)

n $\leq$ q $\Rightarrow$ ($\exists$ k_2:: N) $n + k_2 = q$ (2)

Tomando $k = k_1 + k_2$, tenemos que:

$m + (k_1 + k_2)$
= (asociatividad de +)
$(m + k_1) + k_2$
= (1)
$n + k_2$
= (2)
q

por asociatividad de +.

Con estas propiedades del ≤ podemos demostrar varias propiedades que involucran a las funciones definidas anteriormente, sin necesidad de encontrar el k de la definición matemática en casa caso.

Antes de comenzar, observemos que **≤ es reflexiva y transitiva, pero no simétrica**.
Por esto último, no podremos razonar por reducción a la misma expresión, sino que deberemos realizar las demostraciones en forma directa, o sea, saliendo de una expresión, y llegando, por medio de igualdades y relaciones de menor igual a la otra.

Ahora sí, veamos algunos ejemplos.

Empecemos con la función doble:

doble :: N → N
doble = λn→ case n of { 0 → 0 ; S x → S (S (doble x)) }

La siguiente propiedad de **doble** se demuestra por inducción:

Lema$_{\leq doble}$: (∀ n :: N) n ≤ doble n
Dem. Por inducción en n :: N
Caso n = 0: 0 ≤$^{?}$ doble 0
Dem.
0
≤ (Lema$_{0\leq}$)
doble 0
Caso n = S x con x :: N
HI) x ≤ doble x
TI) S x ≤$^{?}$ doble (S x)
Dem
S x
≤ (Lema$_{\leq S}$)
S (S x)
≤ (Lema$_{\leq SS}$ e HI)
S (S (doble x))
= (def. doble, β, case)
doble (S x)

Observemos que en la demostración, debimos utilizar la definición de **doble** “al revés”.
Esto ocurre muy a menudo cuando demostramos desigualdades, ya que como podemos avanzar en un solo sentido, nos vemos forzados a deshacer las definiciones de las funciones para llegar a la expresión deseada.
También podríamos haber demostrado la desigualdad directamente, usando el **Lema$_{doble+}$**, que decía que (∀ n :: N) doble n = n + n. Con este lema, podemos tomar k = n, y trivialmente se cumple la condición para afirmar el ≤.

Ahora demostraremos una desigualdad que involucra a las funciones **sumi** y **sumpi**.

La función **sumi :: N → N** ya fue definida y calcula la sumatoria de todos los naturales entre 0 y un número dado:

sumi = λn→ case n of { 0 → 0; S x → S x + sumi x }

La función **sumpi** :: (N → Bool) → N → N fue dejada como ejercicio.
La misma recibe un predicado p y un natural n, y calcula la suma de los naturales entre 0 y n para los cuales p es True:

$$\sum_{i=0}^{n} i \quad cuando\ se\ cumpla\ p(i)$$

La definición de esta función es la siguiente:

sumpi = λp n→ case n of { 0 → 0 ;
S x → case p (S x) of { False → sumpi p x ;
True → S x + sumpi p x }}

La propiedad a demostrar es la siguiente: **(∀ p :: N → Bool)(∀ n :: N) sumpi p n ≤ sumi n.**
O sea, sumar todos los naturales entre 0 y uno dado que cumplen cierta condición, siempre dará como resultado un número menor o igual que sumar a todos los naturales entre 0 y el número dado.

Lema$_{\leq sumpi}$: (∀ p :: N → Bool)(∀ n :: N) sumpi p n ≤ sumi n
Dem. Sea p :: N → Bool cualquiera. *{Podemos considerar un p genérico ya que la demostración no depende de p}*

Por inducción en n :: N
Caso n = 0: sumpi p 0 ≤? sumi 0

Dem.
sumpi p 0
= (def. sumpi, βx2, case)
0
≤ (Lema$_{0\leq}$)
sumi 0

Caso n = S x con x :: N cualquiera
HI) sumpi p x ≤ sumi x
TI) sumpi p (S x) $\leq^{?}$ sumi (S x)
{Si miramos la definición de sumpi, para saber el resultado de sumpi p (S x), debemos saber cuánto vale p (S x). Luego, la demostración deberá hacerse por casos en p (S x)}

Dem. Por casos en p (S x) :: Bool

Caso p (S x) = False

sumpi p (S x)
= (def. sumpi, βx2, case x2)
sumpi p x
≤ (HI)
sumi x
≤ ($Lema_{\leq +}$ y conmutatividad de +)
S x + sumi x
= (def. sumi, β, case)
sumi (S x)

Caso p (S x) = True

sumpi p (S x)
= (def. sumpi, βx2, case x2)
S x + sumpi p x
≤ ($Lema_{\leq ++}$, $Lema_{\leq refl}$, HI)
S x + sumi x
= (def. sumi, β, case)
sumi (S x)

Observar que la demostración no es tan intuitiva cuando está involucrada la relación ≤.
Es importante en estos casos tener claro a qué expresión se quiere llegar al final, e ir trabajando en ambas direcciones, aplicando las propiedades de ≤ vistas anteriormente.

Ejercicios

Demostrar que:

1. **- (∀ n :: N) doble n ≤ triple n**
2. **- (∀ p :: N → Bool) (∀ n :: N) contar p n ≤ S n**