Conjuntos inductivos, recursión y principio de inducción

Esquemas de recursión no primitivos

Vimos en la clase pasada el esquema de inducción (que nos permite definir conjuntos), el esquema de recursión primitiva (que nos permite definir funciones con conjuntos inductivos como dominio) y el principio de inducción primitiva (que nos permite probar propiedades en conjuntos inductivos).

Hemos llamado primitivo al esquema de recursión que vimos, porque basa los pasos recursivos en el objeto inmediatamente anterior según las reglas de definición del conjunto.

Podríamos modificar el esquema de recursión (primitiva) de la forma:

> Sea A un conjunto definido inductivamente.
>
> Para definir una función $f : A \to B$, alcanza con definir a f mediante ecuaciones que determinen:
>
> - El valor de f para los objetos de A obtenidos de aplicar cláusulas base
> - El valor de f para los objetos de A obtenidos de aplicar cláusulas inductivas, utilizando el valor de f en **objetos anteriores** (llamada recursiva).

Un ejemplo de recursión no primitiva en N (notar que el conjunto de los naturales puede ser definido de forma inductiva), es la conocida función recursiva de los números de Fibonacci:

$$fib : N \to N /$$

$$fib(0) := 1$$
$$fib(1) := 1$$
$$fib(n+2) := fib(n+1) + fib(n)$$

La recursión primitiva era sin duda más limitada y no nos hubiera permitido definir una función como la de Fibonacci. Sin embargo, el esquema primitivo nos ofrecía ciertas seguridades que ahora no están tan claras, por ejemplo:

- ¿Será exhaustiva esta función? ¿Todo elemento tendrá imagen?
- ¿No habrá superposición de casos? ¿Más de una forma de calcular la misma imagen?
- ¿Terminará la recursión?

No sabemos la respuesta a estas preguntas a partir de la definición general del esquema de recursión primitiva, y por lo tanto deberemos probar para cada caso particular:

- Que las ecuaciones son **exhaustivas**
- Que **no** hay **superposición de casos**
- Que la definición **termina**

En el caso del ejemplo de Fibonacci, basta observar que:

- Todo natural es 0, 1 o de la forma $n+2$ **(exhaustividad)**
- Se cumple: $0 \neq 1 \neq n+2$ **(no superposición de casos)**
- N es bien ordenado, $n+2 > n$ y $n+2 > n+1$ **(terminación)**

Veamos otro interesante ejemplo de recursión no primitiva:

$$div : N \times N^* \rightarrow N /$$

$$div(n,m) := 0 \quad \text{si } n < m$$
$$div(n,m) := div(n-m,m)+1 \quad \text{si } n \geq m$$

Recapitulando, tenemos un conjunto de ecuaciones y debemos probar que definen una función:

- Exhaustividad: Para todo par $n,m \in N$ se cumple $n < m$ o $n \geq m$
- Superposición: Para todo par $n,m \in N$ no puede pasar que $n < m$ y $n \geq m$
- Terminación: Para todo par $n,m \in N$ si $0 < m \leq n$ se cumple que $n-m < n$

Ejercicio: ¿Cuál es esta función?

Note el lector en este punto que hemos definido la función div con dominio $N \times N^*$. Para nosotros, este dominio es un producto cartesiano de dos conjuntos que conocemos, pero… ¿es este producto cartesiano en sí mismo un conjunto inductivo?

Veamos la siguiente definición inductiva:

Definición inductiva de $N \times N$:

i. $(0,0) \in N \times N$
ii. Si $(n,0) \in N \times N$, entonces $(n+1,0) \in N \times N$
iii. Si $(n,m) \in N \times N$, entonces $(n,m+1) \in N \times N$

Queda como ejercicio para el lector definir inductivamente el conjunto $N \times N^*$, dominio de la función div.

Consideremos un momento el conjunto $N \times N$, y el siguiente esquema de recursión sobre él:

$$sum : N \times N \rightarrow N /$$

$$sum(n,0) := n$$
$$sum(n,m+1) := sum(n,m)+1$$

Es muy importante notar en este caso, que si bien existen dos cláusulas recursivas, estamos ante un caso de recursión primitiva, ya que cada una de las cláusulas recursivas especifica el valor de un elemento del dominio en función del elemento inmediatamente anterior de acuerdo a las reglas del conjunto inductivo.

En este caso, sum define una función y no es necesario probar nada.

Esto lo podríamos formalizar definiendo el esquema de recursión primitiva para $N \times N$.

Un conjunto de ecuaciones como el que sigue define una única función con dominio $N \times N$:

i. $f(0,0) := ...$
ii. $f(n+1,0) := ...n...f(n,0)...$
iii. $f(n,m+1) := ...n...m...f(n,m)...$

Como vemos, la recursión primitiva no implica que se tenga sólo un paso recursivo, sino que cada elemento se defina en función del valor del elemento anterior según las reglas del conjunto inductivo.

Con el mismo criterio podríamos definir el principio de inducción primitiva para $N \times N$.

Sea P una propiedad sobre los objetos de $N \times N$ tal que:

i. $P(0,0)$ se cumple
ii. Si se cumple $P(n,0)$, entonces se cumple $P(n+1,0)$
iii. Si se cumple $P(n,m)$, entonces se cumple $P(n,m+1)$

Entonces se cumple $P(n,m)$ para todo elemento de $N \times N$

Veamos ahora un ejemplo de recursión no primitiva con el mismo dominio:

$mcd : N^+ \times N^+ \to N^+ /$

$$mcd(n,m) := n \quad \text{si } n = m$$
$$mcd(n,m) := mcd(n,m-n) \quad \text{si } n < m$$
$$mcd(n,m) := mcd(n-m,m) \quad \text{si } n > m$$

Queda como ejercicio al lector mostrar la exhaustividad, no superposición de casos y terminación para este ejemplo.