

Un ejemplo de Inducción con los números de Fibonacci

Pablo L. De Nápoli

Departamento de Matemática
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Álgebra I - Primer cuatrimestre de 2020

Introducción

En esta clase, veremos cómo resolver un ejercicio consistente en probar una afirmación por **inducción matemática**.

Como sabemos la **inducción matemática**, es un procedimiento para demostrar una afirmación (proposición) que involucra **un número natural**.

La inducción matemática también nos permite definir sucesiones en forma **recursiva**. Es decir, donde la definición de cada término depende los anteriores. Un ejemplo paradigmático de ello es la **sucesión de Fibonacci**.

Los números de Fibonacci

Los **números de Fibonacci** son un ejemplo de una **sucesión definida por recurrencia**. Están definidos en la sección 2.5.2 del apunte de la materia (de la profesora Teresa Krick), por medio de la recurrencia

$$\begin{cases} F_0 & = & 0 \\ F_1 & = & 1 \\ F_{n+2} & = & F_n + F_{n+1} \text{ (para todo } n \in \mathbb{N}_0\text{)}. \end{cases}$$

Sus primeros términos son:

$$F_0 = 0, F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5, F_6 = 8, F_7 = 13, F_8 = 21$$

$$F_9 = 34, F_{10} = 55, F_{11} = 89, F_{12} = 144, F_{13} = 233, F_{14} = 377, F_{15} = 610$$

Bonus track: cómo calculé la tabla anterior

Es extremadamente fácil traducir la definición de una sucesión definida recursivamente en un programa en su lenguaje de programación favorito (¡el mío es **Python!**) usando una **llamada recursiva** a una **función**.

Programa para calcular los números de Fibonacci en Python 3

```
def fibo(n):  
    if n==0:  
        return 0  
    elif n==1:  
        return 1  
    else:  
        return fibo(n-1)+fibo(n-2)
```

Notamos que aquí hemos reescrito la recurrencia de los números de Fibonacci como

$$F_n = F_{n-1} + F_{n-2} \text{ para todo } n \geq 2$$

El enunciado de nuestro ejercicio

Ahora enunciamos el teorema que vamos a demostrar por inducción. Fue demostrado originalmente por Édouard Lucas en 1876, y se refiere a la suma de los primeros n números de Fibonacci.

Ejercicio

$$F_1 + F_2 + \dots + F_n = F_{n+2} - 1$$

Podríamos escribir esto de manera equivalente (más precisa) utilizando la **notación de sumatoria**

Ejercicio

Para todo $n \in \mathbb{N}$,

$$\sum_{k=1}^n F_k = F_{n+2} - 1$$

Recurrencia de la sumatoria

Antes de resolver nuestro ejercicio por inducción, será conveniente trabajar un poquito con la sucesión definida por la sumatoria que aparece en él. Por lo tanto **definamos una nueva sucesión** S_n por

$$S_n = \sum_{k=1}^n F_k$$

Consideremos ahora S_{n+1} , Separando el último término vemos que:

$$S_{n+1} = \sum_{k=1}^{n+1} F_k = \left(\sum_{k=1}^n F_k \right) + F_{n+1} = S_n + F_{n+1}$$

Así también pues S_n puede ser **definida recursivamente** para $n \in \mathbb{N}$ por

$$\begin{cases} S_1 & = & F_1 \\ S_{n+1} & = & S_n + F_{n+1} \end{cases}$$

Bonus track 2: comprobando que la fórmula es cierta (para n en un rango dado)

Antes de intentar resolver nuestro ejercicio, es conveniente ensayar si la fórmula es correcta para algunos valores de n . En este ejemplo, esto se podría hacer a mano. Pero en general una forma conveniente de hacerlo es mediante un pequeño programita.

Para ello comenzamos **traduciendo** la definición recursiva de S_n :

Traducción de la definición recursiva de S_n en Python 3

```
def s(n):  
    if n==1:  
        return fibo(1)  
    else:  
        return s(n-1)+fibo(n)
```

Bonus track 3: comprobando que la fórmula es cierta (para n en un rango dado)

Ahora pueden escribir fácilmente un programita que comprueba si la fórmula que nos proponen es correcta para n variando en un determinado rango mediante un ciclo `for`:

Programita de comprobación de la fórmula en Python 3

```
def comprobar(n):  
    return S(n) == fibo(n+2) - 1  
  
for n in range(1,10):  
    print("n=",n," comprobar(n)=",comprobar_suma(n))
```

Bonus track 4: comprobando que la fórmula es cierta (para n en un rango dado)

La salida de este programa es

Salida del programa para comprobar que la fórmula es cierta

```
n= 1  comprobar(n)= True
n= 2  comprobar(n)= True
n= 3  comprobar(n)= True
n= 4  comprobar(n)= True
n= 5  comprobar(n)= True
n= 6  comprobar(n)= True
n= 7  comprobar(n)= True
n= 8  comprobar(n)= True
n= 9  comprobar(n)= True
```

Por lo que vemos que la fórmula es cierta para $0 \leq n \leq 9$. ¿Pero cómo la comprobamos para todo n ? ¡Los números naturales son **infinitos!**

Parte I

Alcanzando el infinito por pequeños pasos

El Principio de inducción matemática

Es ahí donde entra la **inducción matemática**.

Tenemos una propiedad $P(n)$ que depende de un número natural $n \in \mathbb{N} = \{1, 2, 3, \dots\}$. Técnicamente se denomina una **función proposicional** de n (tenemos una proposición para cada n). Podemos formular entonces el

Principio de Inducción Matemática

Si $P(1)$ es verdadera, y para todo $n \in \mathbb{N}$ se verifica que

$$P(n) \Rightarrow P(n + 1)$$

entonces $P(n)$ es cierta para todo $n \in \mathbb{N}$.

En nuestro ejemplo

$P(n)$:

$$S_n = F_{n+2} - 1$$

El caso base de la inducción $n = 1$

Comprobamos que la proposición $P(1)$ es cierta. Para ello **sustituimos** n por el valor 1 en la fórmula $P(n)$.

$$S_n = F_{n+2} - 1$$

Queda

$$S_1 = F_{1+2} - 1$$

o sea:

$$S_1 = F_3 - 1$$

Como $S_1 = F_1 = 1$ y $F_3 - 1 = 2 - 1 = 1$ esta afirmación **es verdadera**.

El paso inductivo

Ahora debemos probar que para todo $n \in \mathbb{N}$, $P(n) \Rightarrow P(n+1)$. O sea en nuestro caso

$$S_n = F_{n+2} - 1 \Rightarrow S_{n+1} = F_{(n+1)+2} - 1$$

o equivalentemente:

$$S_n = F_{n+2} - 1 \Rightarrow S_{n+1} = F_{n+3} - 1$$

Para hacer esto **suponemos** que para cierto n , $S_n = F_{n+2} - 1$. Esto se denomina **hipótesis de inducción** o **hipótesis inductiva**.

Usando la **recurrencia** para S_n tenemos entonces que:

$$S_{n+1} = S_n + F_{n+1}$$

y usando la **hipótesis de inducción**, tenemos que

$$S_{n+1} = F_{n+2} - 1 + F_{n+1}$$

pero por la **recurrencia** que define los números de Fibonacci

$$F_{n+2} + F_{n+1} = F_{n+3}$$

en consecuencia **sustituyendo** obtenemos la conclusión buscada:

$$S_{n+1} = F_{n+3} - 1$$

Conclusión

Como hemos demostrado la afirmación $P(1)$, y para cada $n \in \mathbb{N}$ hemos demostrado que $P(n)$ implica $P(n + 1)$, usando el **principio de inducción matemática**, concluimos que $P(n)$ es verdadera **para todo n natural**.

Esto concluye la resolución de nuestro ejercicio.

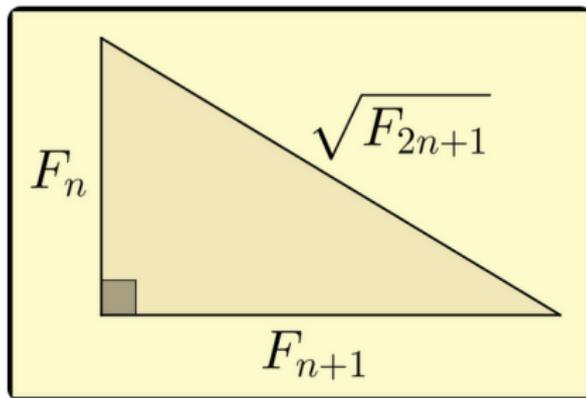
Otro problema con los números de Fibonacci

Ahora les dejo otro problema interesante, de un tweet de un tweet de @gaussianos (una cuenta de twitter dedicada a curiosidades matemáticas).



gaussianos
@gaussianos

Sea F_k el k -ésimo número de Fibonacci. Entonces se cumple que si un triángulo rectángulo tiene por catetos dos números de Fibonacci consecutivos, F_n y F_{n+1} , su hipotenusa es la raíz cuadrada de otro número de Fibonacci, exactamente del que está en la posición $2n+1$



3:37 p. m. · 11 abr. 2020 · Twitter Web App

¿Porqué es interesante?

La afirmación del tweet es (usando el **teorema de Pitágoras**) que para todo $n \in \mathbb{N}$,

$$F_n^2 + F_{n+1}^2 = F_{2n+1}$$

Pueden ver cómo demostrarlo por inducción en

http://nextlevelmaths.com/resources/wow/pythag_fibonacci/

Es más sutil que lo que uno espera, porque hay que probar por inducción que

$$F_{2n} = F_n(F_{n-1} + F_{n+1}) \wedge F_{2n+1} = F_{n+1}^2 + F_n^2$$

Aunque el tweet de @gaussianos sólo contiene la segunda afirmación, ¡ambas se entrecruzan en el argumento inductivo!

Es un lindo ejemplo de que no siempre es tan fácil demostrar afirmaciones por inducción.

Bonus Track Final

El código del programita en Python de esta clase lo pueden descargar de mi repositorio de [GitHub](#)

<https://github.com/pdenapo/programitas-algebraI/blob/master/Python/2020/fibo.py>

Les dejo un pequeño **ejercicio** para terminar. ¿Qué pasa si sumamos los números de Fibonacci de índice par? Tenemos otra fórmula de Lucas

$$\sum_{k=1}^n F_{2k} = F_{2n+1} - 1$$

El ejercicio consiste en hacer un programita para verificarla, y probarla por inducción para todo $n \in \mathbb{N}$.

¿Se animan a ver qué pasa sumando los de índice impar?

$$\sum_{k=1}^n F_{2k-1}$$