Más ejemplos de Inducción

Pablo L. De Nápoli

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

Álgebra I - Segundo cuatrimestre de 2020

Recordamos: El Principio de inducción matemática

Para demostrar una propiedad de la forma

$$\forall n \in \mathbb{N} : P(n)$$

podemos usar 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 verdadera para todo $n \in \mathbb{N}$.

$$P(1) \wedge [\forall n : P(n) \Rightarrow P(n+1)] \Rightarrow [\forall n \in \mathbb{N} : P(n)]$$

Este axioma expresa una propiedad fundamental de los números naturales.

Parte I

Sumando los números impares

Enunciado

En nuestro primer ejemplo de hoy $P(n)$ será

Ejercicio 6- Práctica 2

$$\sum_{i=1}^{n}(2i-1) = 1+3+5+7+\ldots+(2n-1) = n^2$$

El caso base

Demostremos $P(1)$. Si $n = 1$,

$$\sum_{i=1}^{n} (2i - 1) = 1$$

mientras que

$$n^2 = 1$$

luego la fórmula es válida cuando $n = 1$.

El paso inductivo

Suponemos que vale $P(n)$, queremos probar entonces que vale $P(n+1)$.

$$\sum_{i=1}^{n+1}(2i-1) = \sum_{i=1}^{n}(2i-1) + [2(n+1)-1]$$

usando la hipótesis inductiva esto es

$$= n^2 + [2(n+1)-1] = n^2 + 2n + 1 = (n+1)^2$$

Por el principio de inducción matemática, concluimos que la fórmula es cierta para todo n natural.

Otra manera de probarla

Un problema de la inducción es que es necesario saber de antemano (conjeturar) qué fórmula queremos demostrar.Otra manera de hacerlo es usando las propiedades de linealidad de la sumatoria

$$\sum_{i=1}^{n}(a_i + b_i) = \left(\sum_{i=1}^{n} a_i\right) + \left(\sum_{i=1}^{n} b_i\right) \quad \text{(propiedad asociativa)}$$

$$\sum_{i=1}^{n} c \cdot a_i = c \cdot \left(\sum_{i=1}^{n} a_i\right) \quad \text{(propiedad distributiva)}$$

y otra sumatoria que vimos en la clase anterior (el ejercicio 3 de la práctica 2):

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

En efecto,

$$\sum_{i=1}^{n}(2i-1) = 2\left(\sum_{i=1}^{n} i\right) - \left(\sum_{i=1}^{n} 1\right) = 2\frac{n(n+1)}{2} - n = n^2 + n - n = n^2$$

Parte II

Un ejemplo con una desigualdad

La desigualdad de Bernoulli

Teorema (Desigualdad de Bernoulli, ejercicio 13 práctica 2)

Si $x \in \mathbb{R}$, $x \geq -1$ *entonces*

$$(1+x)^n \geq 1 + nx$$

Este resultado se emplea en análisis por ejemplo para probar que si $a > 1$,

$$\lim_{n \to \infty} a^n = +\infty$$

En efecto, llamando $x = a - 1$,

$$a^n = (1+x)^n \geq 1 + nx = 1 + n(a-1)$$

y entonces

$$a^n \geq M \text{ si } n \geq \frac{M-1}{a-1}$$

para cualquier $M > 1$.

Prueba de la desigualdad de Bernoulli, por inducción

- caso base: Si $n = 1$, $(1 + x)^n = 1 + x \wedge 1 + nx = 1 + x$ ✓
- paso inductivo: Suponiendo que para un cierto n se cumple la desigualdad

$$(1 + x)^n \geq 1 + nx$$

queremos probar que

$$(1 + x)^{n+1} \geq 1 + (n + 1)x$$

Teniendo en cuenta que

$$(1 + x)^{n+1} = (1 + x)^n \cdot (1 + x)$$

y que $x > -1$ de la hipótesis inductiva deducimos que,

$$(1 + x)^{n+1} \geq (1 + nx) \cdot (1 + x)$$

Ahora

$$(1 + nx) \cdot (1 + x) = 1 + (n + 1)x + x^2 n \geq 1 + (n + 1)x$$

(pues $x^2 \geq 0$) y por la transitividad de la desigualdad

$$(1 + x)^{n+1} \geq 1 + (n + 1)x \checkmark$$

Parte III

Un ejemplo diferente: una aplicación a la lógica matemática

Introducción

La inducción matemática se emplea en todas las ramas de la matemática.Para presentarles un ejemplo relacionado con cosas con las que ya hemos trabajado, voy a demostrar un teorema de la lógica matemática.

Recordamos que el conjunto de valores de verdad $\mathcal{T} = \{F, V\}$ es un álgebra de Boole con los conectivos lógicos $\vee$, $\wedge$ y $\sim$ [Sólo consideraremos hoy estos conectivos, los restantes $\veebar$, $\Rightarrow$, $\Leftrightarrow$ pueden definirse a partir de ellos].

Dadas variables booleanas $p_1, p_2, \ldots, p_n \in \mathcal{T}$ podemos considerar fórmulas booleanas como

$$(\sim p_1 \vee p_2) \wedge p_3$$

$$p_3 \wedge \sim p_1 \wedge F$$

$$(p_1 \wedge p_2) \vee (p_3 \vee \sim p_2)$$

$$p_1 \wedge (p_2 \vee V)$$

Cada fórmula booleana induce una función booleana $f : \mathcal{T}^n \to \mathcal{T}$ (su tabla de verdad).

Enunciado del teorema

El siguiente teorema dice que recíprocamente a cada tabla de verdad o función booleana se la puede representar mediante una fórmula booleana:

Teorema

Cada función $f : \mathcal{T}^n \to \mathcal{T}$ se puede representar como una fórmula boleana en n variables $p_1, p_2, \ldots, p_n$ involucrando a los conectivos lógicos $\vee$, $\wedge$ y $\sim$.

Por supuesto, ¡vamos a probar el teorema por inducción en n !

El caso base $n = 1$

Hay cuatro tablas de verdad posibles con una sóla variable

p_1	$f_0(p_1)$
F	F
V	F

p_1	$f_1(p_1)$
F	V
V	F

p_1	$f_2(p_1)$
F	F
V	V

p_1	$f_3(p_1)$
F	V
V	V

y su representación por medio de fórmulas booleanas es

$$f_0(p_1) = F \quad \text{o} \quad f_0(p) = p_1 \wedge \sim p_1$$

$$f_1(p_1) = \sim p_1$$

$$f_2(p_1) = p_1$$

$$f_3(p_1) = V \quad \text{o} \quad f_3(p) = p_1 \vee \sim p_1$$

El paso inductivo

Suponemos que el teorema es cierto para un determinado n. Consideramos $f : \mathcal{T}^{n+1} \to \mathcal{T}$.

Definimos dos funciones $g, h : \mathcal{T}^n \to \mathcal{T}$ congelando la última variable:

$$g(p_1, p_2, \ldots, p_n) = f(p_1, p_2, \ldots, p_n, F)$$

$$h(p_1, p_2, \ldots, p_n) = f(p_1, p_2, \ldots, p_n, V)$$

Por la hipótesis inductiva, g y h pueden representarse mediante fórmulas booleanas F_g y F_h en las variables $p_1, p_2, \ldots, p_n$ con los conectivos lógicos $\vee$, $\wedge$ y $\sim$.

Pero entonces la fórmula booleana

$$(F_g \wedge \sim p_{n+1}) \vee (F_h \wedge p_{n+1})$$

representa a f. Esto demuestra el teorema para funciones de $n+1$ variables.

Por el principio de inducción matemática, concluimos que el teorema es cierto para funciones booleanas de cualquier número de variables.

Comprobación

Nuestra fórmula propuesta era

$$(F_g \wedge \sim p_{n+1}) \vee (F_h \wedge p_{n+1})$$

Hay dos casos:

- Si $p_{n+1} = F$, entonces al evaluar nuestra fórmula obtenemos

$$\begin{aligned}
&(f(p_1, p_2, \ldots, p_n, F) \wedge V) \vee (f(p_1, p_2, \ldots, p_n, F) \wedge F) \\
&= f(p_1, p_2, \ldots, p_n, F) \vee F \\
&= f(p_1, p_2, \ldots, p_n, F) = f(p_1, p_2, \ldots, p_n, p_{n+1})
\end{aligned}$$

- Si $p_{n+1} = V$, entonces al evaluar nuestra fórmula obtenemos

$$\begin{aligned}
&(f(p_1, p_2, \ldots, p_n, V) \wedge F) \vee (f(p_1, p_2, \ldots, p_n, V) \wedge V) \\
&= F \vee f(p_1, p_2, \ldots, p_n, V) \\
&= f(p_1, p_2, \ldots, p_n, V) = f(p_1, p_2, \ldots, p_n, p_{n+1})
\end{aligned}$$

Parte IV

Recurrencia

Un ejemplo de recurrencia: el factorial

Otra aplicación del principio de inducción matemática es la definición de sucesiones (funciones que salen desde $\mathbb{N}$)

Consideremos por ejemplo la función factorial

$$n! = 1 \cdot 2 \cdot \ldots n = \prod_{k=1}^{n} k$$

Podemos definir esta función para $n \in \mathbb{N}$ recursivamente por

$$n! = \begin{cases} 1 & \text{si } n = 1 \\ (n-1)! \cdot n & \text{si } n > 1 \end{cases}$$

Se define $0! = 1$ de modo que la recurrencia

$$(n-1)! \cdot n = n!$$

siga valiendo cuando $n = 1$.

Formalización de esta idea

Teorema

Si A es un conjunto, $a_1 \in A$ y $f : A \to A$ es una función, podemos definir otra función $g : \mathbb{N} \to A$ por

$$g(n) = \begin{cases} a_1 & si\ n = 1 \\ f(g(n-1)) & si\ n > 1 \end{cases}$$

Por ejemplo, la función factorial $g : \mathbb{N} \to \mathbb{N}, g(n) = n!$ es la función que se obtiene tomando $a_1 = 1$ y $f : \mathbb{N} \to \mathbb{N}$ dada por $f(x) = x \cdot n$.

Otro ejemplo: si tomamos $A = \mathbb{R}$, elegimos $a_1 = a \in \mathbb{R}$, $f(x) = x \cdot a$, obtenemos la definición recursiva de las potencias

$$a^n = \begin{cases} a & \text{si} \quad n = 1 \\ a^{n-1} \cdot a & \text{si} \quad n > 1 \end{cases}$$

Bonus Track: La recursividad en la computadora

$$n! = \begin{cases} 1 & \text{si } n = 0 \\ (n-1)! \cdot n & \text{si } n > 1 \end{cases}$$

Implementación recursiva del factorial en Python 3

```
def factorial(n):
    if n==0:
        return 1
    else:
        return factorial(n-1)*n
```

Sumatorias y productorias

La recurrencia aperece naturalmente en todos los ejemplos con sumatorias y productorias que fuimos considerando hasta ahora, a partir de las relaciones:

$$\sum_{k=1}^{n+1} a_k = \left[\sum_{k=1}^{n} a_k\right] + a_{n+1}$$

$$\prod_{k=1}^{n+1} a_k = \left[\prod_{k=1}^{n} a_k \cdot\right] a_{n+1}$$

Por ejemplo, la suma que consideramos antes

$$S(n) = \sum_{i=1}^{n}(2i-1)$$

satisfacía que

$$S(n+1) = S(n) + [2(n+1) - 1]$$

que también podría escribirse (cambiando el índice $m = n+1$)

$$S(m) = S(m-1) + [2m - 1]$$

y es la clave de que el argumento de inducción funcione.

Bonus track: Sumando los impares por recurrencia

$$S(m) = \sum_{i=1}^{m} (2i - 1)$$

entonces si queremos calcular la suma mediante un programita usando la recurrencia, podríamos escribirlo así:

Implementación recursiva de la suma de los impares en Python 3

```
def S(m):
    if m==1:
        return 1
    else:
        return S(m-1)+(2*m-1)
```