Fórmulas, Teoremas y Demostraciones Esenciales (Preparación para la universidad)

Doble Grado Matemática e Informática

Carlos Salazar Castillo

Junio 2021

Índice

1. Análisis Matemático I: Fundamentos

1.1. Sucesiones de Números Reales

Definición 1.1. ***Límite de una Sucesión*** *Decimos que un número $L \in \mathbb{R}$ es el límite de la sucesión $(a_n)_{n\in\mathbb{N}}$ si para todo $\varepsilon > 0$ existe un $N_0 \in \mathbb{N}$ tal que para todo $n \geq N_0$, se verifica:*

$$|a_n - L| < \varepsilon$$

Se denota como $\lim_{n\to\infty} a_n = L$.

Teorema 1.1. ***Convergencia de Sucesiones Monótonas*** *Toda sucesión (a_n) de números reales que es* ***monótona creciente*** *(es decir, $a_n \leq a_{n+1}$ para todo n) y está* ***acotada superiormente*** *converge a su supremo.*

Demostración. Sea $S = \{a_n : n \in \mathbb{N}\}$ el conjunto de términos de la sucesión. Como (a_n) está acotada superiormente, por la propiedad del supremo en $\mathbb{R}$, existe $\sup S = L \in \mathbb{R}$.
Objetivo: Probar que $\lim_{n\to\infty} a_n = L$.

1. Sea $\varepsilon > 0$. Como $L = \sup S$, $L - \varepsilon$ no es cota superior de S.
2. Por lo tanto, existe un término $a_{N_0} \in S$ tal que $L - \varepsilon < a_{N_0}$.
3. Como (a_n) es creciente, para todo $n \geq N_0$, tenemos $a_{N_0} \leq a_n$.
4. Combinando, se tiene $L - \varepsilon < a_{N_0} \leq a_n$.
5. Además, como L es el supremo, $a_n \leq L < L + \varepsilon$.
6. De (4) y (5) se deduce que $L - \varepsilon < a_n < L + \varepsilon$, lo que equivale a $|a_n - L| < \varepsilon$.

Por definición, $\lim_{n\to\infty} a_n = L$. ■

Teorema 1.2. ***Criterio de Cauchy para Sucesiones*** *Una sucesión (a_n) converge en $\mathbb{R}$ si y solo si es una* ***sucesión de Cauchy****. Una sucesión es de Cauchy si para todo $\varepsilon > 0$ existe un $N_0 \in \mathbb{N}$ tal que para todo $m, n \geq N_0$, se verifica:*

$$|a_m - a_n| < \varepsilon$$

2. Matemática Discreta y Algorítmica

2.1. Complejidad Asintótica y Notación $\mathcal{O}$

Definición 2.1. ***Notación $\mathcal{O}$ (Cota Superior Asintótica)*** *Dadas dos funciones $f(n)$ y $g(n)$, decimos que $f(n)$ está en $\mathcal{O}(g(n))$, y lo denotamos $f(n) \in \mathcal{O}(g(n))$, si existen constantes positivas c y n_0 tales que para todo $n \geq n_0$, se cumple:*

$$0 \leq f(n) \leq c \cdot g(n)$$

En informática, $f(n)$ es el tiempo de ejecución y $g(n)$ es la cota de complejidad.

Ejemplo 2.1. *Probar que $f(n) = 3n^2 + 2n + 5 \in \mathcal{O}(n^2)$.*

1. *Necesitamos encontrar $c > 0$ y $n_0 \in \mathbb{N}$ tales que $3n^2 + 2n + 5 \leq cn^2$ para $n \geq n_0$.*
2. *Para $n \geq 1$, se cumple que:*

$$3n^2 + 2n + 5 \leq 3n^2 + 2n^2 + 5n^2 = 10n^2$$

3. *Tomando $c = 10$ y $n_0 = 1$, la desigualdad se cumple.*
4. *Por lo tanto, $3n^2 + 2n + 5 \in \mathcal{O}(n^2)$.*

2.2. Algoritmos de Búsqueda y Ordenación

Proposición 2.1. ***Complejidad de Búsqueda Binaria*** *La complejidad temporal del algoritmo de búsqueda binaria sobre un array ordenado de n elementos es $\mathcal{O}(\log_2 n)$.*

Demostración. Sea n el tamaño del array. En cada paso de la búsqueda binaria, el tamaño del subproblema se reduce a la mitad.
Sea $T(n)$ el tiempo de ejecución. Si $n > 1$, se realiza una comparación y se resuelve un subproblema de tamaño $n/2$.

$$T(n) = T(n/2) + c \quad \text{(donde } c \text{ es una constante)}$$

Expandiendo la recurrencia k veces:

$$T(n) = T(n/2^k) + c \cdot k$$

La recursión finaliza cuando $n/2^k = 1$, es decir, $n = 2^k$, o $k = \log_2 n$. Sustituyendo k:

$$T(n) = T(1) + c \log_2 n$$

Como $T(1)$ es una constante, $T(n)$ está acotada superiormente por un múltiplo de $\log_2 n$. Por lo tanto, $T(n) \in \mathcal{O}(\log_2 n)$. ■