



[Saltar al contenido principal](#)



Notación O grande y similares



Introducción

En la entrada pasada hablamos del modelo RAM. Nos convencimos de que es una buena forma de medir la cantidad de pasos que toma un algoritmo. Además, platicamos de las dificultades e inconveniencias de contar de manera totalmente exacta los pasos que se toman. Vimos que en muchos casos lo más conveniente es dar simplemente una aproximación.

Sin embargo, no podemos quedarnos con una idea muy vaga de si nuestras aproximaciones son buenas o no. Lo que haremos ahora es dar algunas definiciones que nos permitirán comparar distintos algoritmos entre sí, sin tener que meternos en las complicaciones de dar cuentas exactas. Al inicio esto requiere de aspectos un poco técnicos, pero en la siguiente entrada demostraremos algunas propiedades que harán que el cálculo asintótico sea mucho más práctico.

Notación de O grande

Sea \mathbb{P} el conjunto de los reales no negativos. En toda esta discusión pensaremos únicamente en funciones f que van de \mathbb{P} a \mathbb{P} , pues usualmente $f(n)$ denotará la cantidad de pasos que toma un algoritmo (en el peor de los casos) en resolver instancias de tamaño n de un problema.

Definición. Sean $f: \mathbb{P} \rightarrow \mathbb{P}$ y $g: \mathbb{P} \rightarrow \mathbb{P}$ dos funciones. Diremos que $f(n) = O(g(n))$ si existen constantes $c > 0$ y n_0 tales que para toda $n \geq n_0$ se cumpla que

$$f(n) \leq cg(n).$$

El determinar si se cumple o no que $f(n) = O(g(n))$ suele ser un juego de determinar constantes que sirvan para la definición, o no. Es muy parecido al tipo de trabajo que se hace cuando en cálculo se trabaja con la definición épsilon-delta de límites.

Veamos algunos ejemplos.

Ejemplo. Si $f(n) = 2n^2 + 4n - 1$, entonces $f(n) = O(n^2)$. Verifiquemos esto mediante la definición. Para ello, debemos de dar los valores de c y de n_0 que funcionen. Nuestra propuesta será tomar $c = 3$ y $n_0 = 4$. Para verificar que funcionan, notemos que la igualdad

$$f(n) \leq 3n^2$$

es

$$2n^2 + 4n - 1 \leq 3n^2$$

y se da si y sólo si

$$4n - 1 \leq n^2.$$

Pero esto se cumple si $n \geq 4$ pues en ese caso $n^2 \geq 4n \geq 4n - 1$.

(square)

Es fácil verificar que también se vale «el otro lado» en el ejemplo anterior. Si $g(n) = n^2$, entonces $g(n) = O(2n^2 + 4n - 1)$. Puedes verificar que basta tomar $c = 1$ y $n_0 = 1$. La intuición que debe dejarte este ejemplo es que n^2 y $2n^2 + 4n - 1$ «se comportan igual cuando n es grande». Más adelante formalizaremos esto diciendo que estas funciones tienen el mismo **orden de magnitud** o el mismo **comportamiento asintótico**.

Ejemplo. Si $f(n) = 3^n$, entonces no es cierto que $f(n) = O(2^n)$. Para ver que esto no sucede, supongamos que sí con el fin de encontrar una contradicción. De ser el caso, tendríamos variables $c > 0$ y n_0 tales que $3^n \leq c \cdot 2^n$ para $n \geq n_0$. Esto querría decir que

$$\left(\frac{3}{2}\right)^n \leq c$$

para cualquier $n \geq n_0$). Pero esto es imposible, pues $\lim_{n \rightarrow \infty} \left(\frac{3}{2}\right)^n = \infty$, de modo que para n suficientemente grande la expresión $\left(\frac{3}{2}\right)^n$ supera a c . \square

En este ejemplo aunque no se tenga que $3^n = O(2^n)$, es relativamente fácil verificar que $2^n = O(3^n)$ tomando $c=1$ y $n_0=1$. La intuición que debe quedarte es que 3^n crece mucho más rápido que 2^n . Tan rápido que es imposible compensar este crecimiento con una constante multiplicativa. Después introduciremos el concepto de **dominancia** para formalizar esta idea.

Ejemplo. Si $f(n) = n^{100}$, entonces $f(n) = O(2^n)$. ¿Cómo le hacemos para encontrar los valores de c y de n_0 que funcionen? Podemos trabajar hacia atrás, como cuando en cálculo buscamos un delta que funcione para un ϵ . Lo que nos gustaría que pasara es que $n^{100} \leq c2^n$ para n suficientemente grande. Intentemos ver si podemos lograr lo buscado con $c=1$. Para ello necesitaríamos dar una n_0 tal que $n^{100} \leq 2^n$ cuando $n \geq n_0$.

Si n_0 es (100) , esto todavía no es suficiente, pues del lado izquierdo tenemos 100^{100} y del derecho 2^{100} . ¿Qué pasa si n_0 es (1000) ? Del lado izquierdo tenemos $(1000)^{100} = 10^{300}$ y del derecho $2^{1000} = (2^4)^{250} = 16^{250} \geq 10^{250}$. Esto ya se ve un poco más prometedor, pues el lado derecho ya va acercándose al izquierdo. Agrandando en otro factor de (10) a n_0 logramos que el lado derecho exceda el izquierdo pues

$$2^{10000} = 16^{2500} > 10^{2500} > 10^{400} = 10000^{100}.$$

Resulta que este valor de n_0 es bueno, y es posible dar una demostración inductiva de ello, la cual queda de tarea moral. \square

En este último ejemplo la intuición que queda es que 2^n crece por lo menos tan rápido como n^{100} , o bien, que n^{100} se puede «acotar por arriba asintóticamente» por 2^n . Puedes verificar que lo contrario no sucede, es decir, que no se cumple que $2^n = O(n^{100})$. Así, intuitivamente tenemos que 2^n crece mucho más rápido que n^{100} .

Por ahora hemos elegido a nuestras constantes c y n_0 de manera muy artesanal. Más adelante hablaremos de otras técnicas para traducir resultados de la notación O a resultados de límites.

Notación de Omega

Si tenemos que $f(n) = O(g(n))$, lo que nos está diciendo la notación O grande intuitivamente es que « $f(n)$ es más chica que algún múltiplo de $g(n)$ para n suficientemente grande». En ocasiones queremos expresar la idea contraria: que $f(n)$ le gane a algún múltiplo de $g(n)$ si n es suficientemente grande. Es por ello que introducimos la siguiente definición.

Definición. Sean $f: P \rightarrow P$ y $g: P \rightarrow P$ dos funciones. Diremos que $f(n) = \Omega(g(n))$ si existen constantes $c > 0$ y n_0 tales que

$$f(n) \geq cg(n)$$

para toda $n \geq n_0$.

La notación de Ω la introducimos más que nada por conveniencia, pues en realidad es muy sencillo demostrar lo siguiente.

Proposición. $f(n) = O(g(n))$ si y sólo si $g(n) = \Omega(f(n))$.

La idea clave de la demostración es que las constantes c y n_0 sirven para probar lo primero si y sólo si $(1/c)$ y n_0 son constantes que sirven para probar lo segundo.

Veamos un par de ejemplos de la notación Ω .

Ejemplo. Se tiene que $f(n) = \frac{\sqrt{n}}{2}$ cumple que $f(n) = \Omega(\log n)$. Para verificar esto, hay que dar una c y una n_0 que cumplan con la definición. Tomaremos $c = \frac{1}{4}$, de manera tal que la desigualdad que queremos que se cumpla es

$$2\sqrt{n} \geq \log n$$

para n suficientemente grande. Afirmamos que con tomar $n_0=1$ basta.

Hay varias formas de demostrar esta desigualdad. Pero usemos un argumento de integrales. Para $x \geq 1$ tenemos que $\frac{1}{x} \geq \frac{1}{x^2}$

Hay varias formas de demostrar esta desigualdad, pero usamos un argumento de integrales. Para $x \geq 1$ tenemos que $\frac{1}{\sqrt{x}} \leq \frac{1}{\sqrt{x}}$. Así, usando que la integral es monótona tenemos que:

$$\int_1^n \frac{1}{\sqrt{x}} dx \leq \int_1^n \frac{1}{\sqrt{x}} dx, \quad \int_1^n \frac{1}{\sqrt{x}} dx = 2\sqrt{x} \Big|_1^n = 2\sqrt{n} - 2 < 2\sqrt{n}.$$

Así, la desigualdad que queremos se cumple con $c = \frac{1}{4}$ y para toda $n \geq n_0 = 1$. (\square)

En algunas ocasiones dentro de la notación de $O()$ grande, de $\Omega()$ o de otras que veremos más adelante se pone simplemente el número (1) .

Ejemplo. Veamos que si $f(n) = \log(\log(n))$, entonces $f(n) = \Omega(1)$. La idea es exactamente la misma. Queremos encontrar constantes c y n_0 tales que

$$\log(\log(n)) \geq c$$

para $n \geq n_0$.

Es muy fácil argumentar que dicha constante c debe existir pues la función $\log(\log n)$ es creciente (aunque crece muy muy lento). Así, una vez que supere, digamos, el valor (1) , entonces siempre será mayor que (1) . Podemos formalizar esto mediante el siguiente argumento de cálculo.

Tomemos $c = 1$ y $n_0 = e^e$. En n_0 tenemos que $f(n_0) = f(e^e) = \log(\log(e^e)) = \log(e) = 1$. Luego, pensando a f como una función de variable real tenemos que $f'(x) = \frac{1}{\log x} \cdot \frac{1}{x} > 0$. Así, f es creciente y por lo tanto $f(n) \geq f(n_0) = 1$ para $n \geq n_0$. (\square)

Notación de Theta

En algunas ocasiones dos funciones son prácticamente iguales asintóticamente, salvo por una constante multiplicativa. Cuando eso sucede, usamos la notación $\Theta()$.

Definición. Sean $f: P \rightarrow P$ y $g: P \rightarrow P$ dos funciones. Diremos que $f(n) = \Theta(g(n))$ si existen constantes $c > 0$, $d > 0$ y n_0 tales que

$$cg(n) \leq f(n) \leq dg(n)$$

para toda $n \geq n_0$.

En otras palabras, tenemos que $f(n) = \Theta(g(n))$ si y sólo si $f(n) = O(g(n))$ y $g(n) = O(f(n))$. O bien, si y sólo si $f(n) = O(g(n))$ y $f(n) = \Omega(g(n))$. Veamos algunos ejemplos.

Ejemplo. Veamos que $(n+1)^2 = \Theta(n^2)$. Basta con demostrar que $(n+1)^2 = O(n^2)$ y $(n+1)^2 = \Omega(n^2)$.

Para la primer parte, basta con tomar $c_1 = 2$ y $n_1 = 3$. Verifiquemos que funciona. Para $n \geq 3$ tenemos que $n^2 \geq 2n+1$ (se puede probar inductivamente o con cálculo). De este modo, $2n^2 \geq n^2 + 2n + 1 = (n+1)^2$, como queremos.

Para la segunda parte, basta con tomar $c_2 = 1$ y $n_2 = 1$ pues de manera inmediata obtenemos $n^2 \leq n^2 + 2n + 1 = (n+1)^2$.

Si quisiéramos dar de manera explícita las constantes que sirven para la notación $\Theta()$, tomaríamos $c = c_2 = 1$, $d = c_1 = 2$ y $n_0 = \max(n_1, n_2) = 3$. (\square)

Ejemplo. Si a y b son reales mayores que (1) , entonces $\log_a n = \Theta(\log_b n)$. En otras palabras, asintóticamente «no nos importa la base del logaritmo». La demostración es sencilla pues usando propiedades de logaritmos tenemos que:

$$\log_a n = \frac{\log n}{\log a} = \frac{\log n}{\log b} \cdot \frac{\log b}{\log a} = \log_b n \cdot \frac{\log b}{\log a},$$

es decir, de hecho las funciones difieren únicamente por una constante multiplicativa, así que podemos tomar $n_0 = 1$ y $c = d = \frac{\log a}{\log b}$ en la definición de $\Theta()$. (\square)

Los siguientes problemas te ayudarán a practicar lo visto en esta entrada. Para resolverlos, necesitarás usar herramientas matemáticas, computacionales o ambas.

1. Demuestra inductivamente que si $(n \geq 10000)$, entonces $(n^{100} \leq 2^n)$.
2. Muestra que lo siguiente se cumple:
 - $(\sqrt[3]{n} = O(\sqrt{n}))$.
 - $(n^5 + 3n - 8 = O(n^6))$.
 - $(\log(\log n) = O(\log n))$.
 - $(\frac{1}{n} = O(1))$.
 - Para $(k > 1)$, $(x^k = O(x^n))$.
3. Muestra que lo siguiente se cumple:
 - $(\sqrt[3]{n} = \Omega(\sqrt[4]{n}))$.
 - $(n \log n = \Omega(\log(n^{100})))$.
 - $(2^n + 3^n + 4^n = \Omega(2^n))$.
 - $(\log(n!) = \Omega(n \log n))$.
 - $(n! = \Omega(3^n))$.
4. Muestra que lo siguiente se cumple:
 - $(100n^3 + 10n^2 + n + 1 = \Theta(n^3))$.
 - $(\sqrt{2n+1} = \Theta(\sqrt{n}))$.
 - $(\binom{n}{3} = \Theta(n^3))$.
 - $(1^4 + 2^4 + 3^4 + \dots + n^4 = \Theta(n^5))$.
 - $(\log(n!) = \Theta(n \log n))$.
5. Sea (P) un polinomio cualquiera que mande reales positivos a reales positivos. Muestra que $(P(n) = O(2^n))$, pero que no sucede que $(2^n = O(P(n)))$.



[Anterior](#)
[Modelo RAM y pensamiento asintótico](#)

[Siguiendo](#)

[Dominancia de funciones, orden de crecimiento y propiedades](#)

