Asymptotic Notation

Asymptotic Upper Bound:

Definition For two functions $f, g : \mathbb{R} \to \mathbb{R}$, we say that $f(x) = O(g(x))$ if there exist $x_0 \in \mathbb{R}$ and $c > 0$ such that for every $x > x_0$, we have:

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

The informal meaning is that the function f grows not faster than g for all sufficiently large x. The same definition holds good for any subdomain of $\mathbb{R}$; in particular, for the analysis of algorithms, we usually consider functions defined on the set of natural numbers.

An example: We shall show that $4n^3 + 100n^2 + 10 = O(n^3)$.

To prove this directly (using the definition), we should find constants n_0 and $c > 0$ such that $4n^3 + 100n^2 + 10 \leq cn^3$ for $n > n_0$. We can easily check that the constants $c = 114$ and $n_0 = 1$ work. Indeed, for $n > 1$, we have $4n^3 = 4n^3$, $100n^2 < 100n^3$ and $10 < 10n^3$. Adding the three inequalities, we get the desired result.

However, such direct proofs and finding explicit constants (c, n_0) is too cumbersome to do all the time, so we will develop a collection of useful results and tricks to compare the growth of two functions.

Asymptotic Notation: Basic Properties

The following properties are useful for making asymptotic comparisons.

1. **Linearity:** If $f(n) = O(h(n))$ and $g(n) = O(h(n))$, then for any constants a, b, we have:

   $$af(n) + bg(n) = O(h(n)).$$

   Similarly, if $t(n) = O(r(n))$ and $t(n) = O(s(n))$, then for $a, b > 0$, we have:

   $$t(n) = O(ar(n) + bs(n)).$$

   Let's consider the earlier example: $4n^3 + 100n^2 + 10 = O(n^3)$. It is easy to see that $4n^3 = O(n^3)$ directly; indeed $4n^3 < cn^3$ for $c = 5$ and all $n > 1$. Similarly, $100n^2 = O(n^3)$ and $10 = O(n^3)$ are easy to show. Thus, we can combine them using linearity to get $4n^3 + 100n^2 + 10 = O(n^3)$.

   Linearity also says that when we have a function which is a sum of a few terms, then the "largest" term dominates the asymptotic growth of that function. Thus, for a polynomial $f(n) = a_0 n^k + a_1 n^{k-1} + \ldots + a_k$, we have $f(n) = O(n^k)$ and if $a_0 > 0$, we also have $n^k = f(n)$.

2. **Transitivity:**

   If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.

   Examples: If $f(n) = 10n^3 - 700n^2 + 5n - 3$, and $g(n) = 1000n^3 + 20n - 6$, then we have: $f(n) = O(n^3)$ and $n^3 = O(g(n))$; thus $f(n) = O(g(n))$.

3. **Multipliying both sides:**

   If $f(n) = O(g(n))$ and $h(n) > 0$ for $n \geq 1$, then $f(n)h(n) = O(g(n)h(n))$. Thus, we can multiply or divide on both sides by positive functions, without changing the validity of the Big-Oh inequality.

Asymptotic Notation: The Little-Oh

When comparing two functions, we are often able to say something stronger than $f(n) = O(g(n))$. We may find that $g(n)$ grows so much faster than $f(n)$ that the ratio $g(n)/f(n)$ goes to ∞ as n goes to infinity. This is captured in the following definition.

Definition We say that $f(n) = o(g(n))$ if $\lim_{n\to\infty} \dfrac{f(n)}{g(n)} = 0$.

For example, we have $5n^2 + 100n - 6 = o(n^3)$. The following connections with the Big-Oh notation are why they are relevant to us.

Proposition 1 *:*

(i) If $f(n) = o(g(n))$, then $f(n) = O(g(n))$.

(ii) If $f(n) = o(g(n))$, then $g(n)$ is NOT $O(f(n))$.

The second part of proposition 1 implies, for example, that $n^3 = O(n^2)$ is false. The implication of the first part is that taking limits of ratios is often a quick and convenient way to compare functions.

The Little-Oh Notation also satisfies the three properties of the Big-Oh, from the second notes: Linearity, Transitivity, and Multiplication on both sides. In fact, it satisfies transitivity even when one of the relations is the weaker Big-Oh.

Proposition 2 *If $f(n) = O(g(n))$ and $g(n) = o(h(n))$, then $f(n) = o(h(n))$.*

If $f(n) = o(g(n))$ and $g(n) = O(h(n))$, then $f(n) = o(h(n))$.

Asymptotic Notation: Important Examples

Proposition 1 *:*

(i) If f, g are two polynomials of degrees $d_1 < d_2$ respectively, then $f(n) = o(g(n))$. This follows easily from the limit definition.

(ii) $n^k = o(e^n)$ for every constant k. To see this, just note that $e^n \geq \frac{n^{k+1}}{(k+1)!}$ (from the Taylor series); thus $n^{k+1} = O(e^n)$ and $n^k = o(n^{k+1})$. Transitivity finishes the proof.

Corollaries:

1. We have $\log n = o(n)$ and $\log\log n = o(\log n)$. To see the first one, substitute $n = e^k$ so that we have $\lim_{n\to\infty} \frac{\log n}{n} = \lim_{k\to\infty} \frac{k}{e^k}$ Now the latter limit is zero from part (ii) above: it is equivalent to $n = o(e^n)$. The second result is one more substitution.

2. For any constants $c > 1$ and $k > 0$, we have $n^k = o(c^n)$. For proof, let $c > 1$ and write $c = e^\alpha$ for $\alpha > 0$. Then c^n can be written as $e^{\alpha n}$, so that the limit of $n^k/e^\alpha n$ is $1/\alpha^k$ times the limit of $(\alpha n)^k/e^{(\alpha n)}$. This last limit is the same as the one in (ii) but for a substitution, so it is still zero.

Asymptotic Notation: Compare Logarithms

Proposition 1 *Suppose that* $\lim_{n\to\infty} g(n) = \infty$ *and* $\log_2 f(n) = o(\log_2 g(n))$. *Then* $f(n)^a = o(g(n)^b)$ *for* $a, b > 0$.

Examples:

1. $(\log_2 n)^{100} = o(n^{0.01})$.

   Proof: Consider their logarithms $f(n) = 100 \log_2 \log_2 n$ and $g(n) = 0.01 \log_2 n$. We have $f(n) = o(g(n))$, and combining this with Proposition 1 gives the result. It also follows directly from Proposition 2.

2. $n^{\log_2 n} = o(2^{\sqrt{n}})$.

   Proof: Consider their logarithms $f(n) = (\log_2 n)^2$ and $g(n) = \sqrt{n}$. We have $f(n) = o(g(n))$ and applying Proposition 1 gives the result. To see why $f(n) = o(g(n))$, substitute $n = e^k$ so that it is equivalent to $k^2 = o(e^{k/2})$, which we know to be true from Note 4.

Asymptotic Notation: Omega and Theta

When we want to say that a function is large, we use the Omega notation. For example, we can say that any comparison-based sorting algorithm must perform $\Omega(n \log n)$ comparisons.

Definition We say that $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$.

For example, $n^3 = \Omega(n^2)$.

Definition We say that $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $g(n) = O(f(n))$.

For example, $56n^2 - 8n + 5 = \Theta(n^2)$ and $\log(n!) = \Theta(n \log n)$.