Asymptotic Notations

Definitions and Properties

Definition 1 (Big-O)**.** Let $f = f(n)$ and $g = g(n)$ be two positive functions over integers n. We say $f = O(g)$, if there exists positive number $c > 0$ and integer $N \geq 0$ such that $f(n) \leq c \cdot g(n)$ for all $n \geq N$.

Similarly, we can define Big-O for multiple-variable functions.

Definition 2 (Big-O)**.** Let $f = f(m,n)$ and $g = g(m,n)$ be two positive functions over integers m and n. We say $f = O(g)$, if there exists positive number $c > 0$ and integers $M \geq 0$ and $N \geq 0$ such that $f(m,n) \leq c \cdot g(m,n)$ for all $m \geq M$ and $n \geq N$.

Intuitively, Big-O is analogous to "$\leq$". $f = O(g)$ means "f grows no faster than g".

Example. Let $f(m,n) = 4m + 4n + 5$ and $g(m,n) = m + n$. We now show that $f = O(g)$, using above definition. To show it, we need to find c, M, and N. What are good choices for them? There are lots of choices; one set of it is: $c = 7$, $M = 1$, and $N = 1$. Let's verify: $f(m,n) - c \cdot g(m,n) = 4m + 4n + 5 - 7m - 7n = 5 - 3m - 3n \leq 5 - 3 - 3 = -1 \leq 0$, where we use that $m \geq M = 1$ and $n \geq N = 1$. This proves that $f = O(g)$.

Definition 3 (Big-Omega)**.** Let $f = f(n)$ and $g = g(n)$ be two positive functions over integers n. We say $f = \Omega(g)$, if there exists positive number $c > 0$ and integer $N \geq 0$ such that $f(n) \geq c \cdot g(n)$ for all $n \geq N$.

Similarly, we can define Big-Omega for multiple-variable functions.

Definition 4 (Big-O)**.** Let $f = f(m,n)$ and $g = g(m,n)$ be two positive functions over integers m and n. We say $f = \Omega(g)$, if there exists positive number $c > 0$ and integers $M \geq 0$ and $N \geq 0$ such that $f(m,n) \geq c \cdot g(m,n)$ for all $m \geq M$ and $n \geq N$.

Intuitively, Big-Omega is analogous to "$\geq$". $f = \Omega(g)$ means "f grows at least as fast as g".

Example. Let $f(m,n) = 4m + 4n + 5$ and $g(m,n) = m + n$. We now show that $f = \Omega(g)$, using above definition. To show it, we need to find c, M, and N. We can choose: $c = 1$, $M = 0$, and $N = 0$. Let's verify: $f(m,n) - c \cdot g(m,n) = 4m + 4n + 5 - m - n = 5 + 3m + 3n \geq 5 \geq 0$, where we use that $m \geq M = 0$ and $n \geq N = 0$. This proves that $f = \Omega(g)$.

Claim 1. $f = O(g)$ if and only if $g = \Omega(f)$.

Proof. We have

$$\begin{aligned}
& f = O(g) \\
\Leftrightarrow\quad & \exists\, c > 0, N \geq 0, \text{ s.t. } f(n) \leq c \cdot g(n), \forall n \geq N \\
\Leftrightarrow\quad & \exists\, c > 0, N \geq 0, \text{ s.t. } 1/c \cdot f(n) \leq g(n), \forall n \geq N \\
\Leftrightarrow\quad & \exists\, c' = 1/c > 0, N \geq 0, \text{ s.t. } g(n) \geq c' \cdot f(n), \forall n \geq N \\
\Leftrightarrow\quad & g = \Omega(f)
\end{aligned}$$

□

Definition 5 (Big-Theta)**.** We say $f = \Theta(g)$ if and only if $f = O(g)$ and $f = \Omega(g)$.

Intuitively, Big-Theta is analogous to "$=$". $f = \Theta(g)$ means "f grows at the same rate as g".

Example. Let $f(m,n) = 4m + 4n + 5$ and $g(m,n) = m + n$. We have $f = \Theta(g)$ as we proved that both

$f = O(g)$ and $f = \Omega(g)$.

Definition 6 (small-o)**.** Let $f = f(n)$ and $g = g(n)$ be two positive functions over integers n. We say $f = o(g)$, if for every $c > 0$ there exists integer $N_c \geq 0$, where N_c can be dependent on c, such that $f(n) \leq c \cdot g(n)$ for all $n \geq N_c$.

Example. Let $f(n) = n$ and $g(n) = n^2$. We now show that $f = o(g)$, using above definition. To prove it, we need to show that for any possible $c > 0$ there exists $N_c \geq 0$ such that $f(n) - c \cdot g(n) \leq 0$ when $n \geq N_c$. We write $f(n) - c \cdot g(n) = n - cn^2 = n(1 - cn)$. To let it be ≤ 0, since $n \geq 0$, we can require $1 - cn \leq 0$, leading to $n \geq 1/c$. Therefore, we can choose $N_c = \lceil 1/c \rceil$. This completes the proof.

Intuitively, small-o is analogous to "$<$". $f = o(g)$ means f grows (strictly) slower than g.

Definition 7 (small-omega)**.** Let $f = f(n)$ and $g = g(n)$ be two positive functions over integers n. We say $f = \omega(g)$, if for every $c > 0$ there exists integer $N_c \geq 0$, where N_c can be dependent on c, such that $f(n) \geq c \cdot g(n)$ for all $n \geq N_c$.

Intuitively, small-omega is analogous to "$>$". $f = \omega(g)$ means f grows (strictly) faster than g.

Obviously, if $f = o(g)$ then $f = O(g)$; if $f = \omega(g)$ then $f = \Omega(g)$. This is intuitive, as "$<$" implies "$\leq$" and "$>$" implies "$\geq$". To formally see this, compare the definitions of small-o and big-O. $f = o(g)$ requires that $f(n) \leq c \cdot g(n)$, when $n \geq N_c$, for *every* possible $c > 0$. Therefore of course there exists *one* c and N_c such that $f(n) \leq c \cdot g(n)$, when $n \geq N_c$; this is all we need to prove $f = O(g)$. The same argument can be used for small-omega and big-Omega.

You might found that these asymptotic notations are similar to the (epsilon-delta)-defintions of limit. In fact, they are indeed closely related. Specifically, the limit of $f(n)/g(n)$, if exists (i.e., $f(n)/g(n)$ coverges as $n \to \infty$), or goes to infinity (i.e., $\lim_{n\to\infty} f(n)/g(n) = \infty$), we can conclude a relationship between f and g:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \begin{cases} 0 & \Rightarrow \quad f = o(g) \\ c > 0 & \Rightarrow \quad f = \Theta(g) \\ \infty & \Rightarrow \quad f = \omega(g) \\ \text{oscillate} & \Rightarrow \quad \text{no conclusion} \end{cases}$$

Above claim gives an convinent way to build asymptotic relationship. For the same example where $f(n) = n$ and $g(n) = n^2$. We now can show $f = o(g)$ by calculating $\lim_{n\to\infty} f(n)/g(n)$. In fact, $\lim_{n\to\infty} n/n^2 = \lim_{n\to\infty} 1/n = 0$. Hence, $f = o(g)$.

Another example: $f(n) = n^2$ and $g(n) = 2^n$. We calculate $\lim_{n\to\infty} f(n)/g(n) = \lim_{n\to\infty} n^2/2^n$. Using L-Hopital rule, we have $\lim_{n\to\infty} n^2/2^n = \lim_{n\to\infty} 2n/(2^n \cdot \ln 2) = 2/(2^n \cdot \ln 2 \cdot \ln 2) = 0$. Hence, $f = o(g)$.

Note that when $f(n)/g(n)$ oscillates, as $n \to \infty$, then we cannot conclude anything. Note also that this reasoning is one-side. For example, if $f = \Theta(g)$ then we cannot guarantee that $\lim_{n\to\infty} f(n)/g(n) = c > 0$; for most functions this is correct but exceptions exist.

Commonly-Used Functions in Algorithm Analysis

In theoretical computer science, we often see following categories of functions.

1. logarithmic functions: $\log\log n$, $\log n$, $(\log_n)^2$;

2. polynomial functions: $\sqrt{n} = n^{0.5}$, n, $n\log n$, $n^{1.001}$;
3. exponential functions: 2^n, $n2^n$, 3^n;
4. factorial functions: $n!$;

In above lists, any logarithmic function is small-o of any polynomial function: for example, $(\log n)^2 = o(n^{0.01})$; any polynomial function is small-o of any exponential function: for example, $n^2 = o(2^n)$; any exponential function is small-o of any factorial function: for example, $n2^n = o(n!)$. Within each category, a function to the left is small-o of a function to the right, for example $n\log n = o(n^{1.001})$.