Some Big O Rules

Just like with limits in Calculus, there are some simple rules for proving that some functions are big O of other functions. Here are a few examples.

Constant Factors Rule

If $f(n)$ is $O(g(n))$ and $a > 0$ is a positive constant, then $af(n)$ is $O(g(n))$.

Proof: There exists a c such $f(n) \leq cg(n)$ for all $n \geq n_0$, and so $af(n) \leq acg(n)$ for all $n \geq n_0$. Thus we use ac and n_0 as constants to show $a\ f(n)$ is $O(g(n))$.

Sum Rule

If $f_1(n)$ is $O(g(n))$ and $f_2(n)$ is $O(g(n)$, then $f_1(n) + f_2(n)$ is $O(g(n))$.

Proof: There exists constants c_1, c_2, n_1, n_2 such $f_1(n) \leq c_1 g(n)$ for all $n \geq n_1$, and $f_2(n) \leq c_2 g(n)$ for all $n \geq n_2$. Thus, $f_1(n) + f_2(n) \leq c_1 g(n) + c_2 g(n)$ for all $n \geq max(n_1, n_2)$. So we can take $c_1 + c_2$ and $max(n_1, n_2)$ as our two constants.

Product Rule

If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n)$, then $f_1(n)f_2(n)$ is $O(g_1(n)\ g_2(n))$.

Proof: We can use similar constants as in the sum rule, except that now we have $g_1(n)$ and $g_2(n)$ instead of $g(n)$. Specifically, there exists constants c_1, c_2, n_1, n_2 such $f_1(n) \leq c_1 g_1(n)$ for all $n \geq n_1$, and $f_2(n) \leq c_2 g_2(n)$ for all $n \geq n_2$. Thus, $f_1(n) * f_2(n) \leq c_1 g_1(n) * c_2 g_2(n)$ for all $n \geq max(n_1, n_2)$. So we can take $c_1 * c_2$ and $max(n_1, n_2)$ as our two constants.

Transitivity Rule

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

Proof: There exists constants c_1, c_2, n_1, n_2 such $f(n) \leq c_1 g(n)$ for all $n \geq n_1$, and $g(n) \leq c_2 h(n)$ for all $n \geq n_2$. Plugging $g(n)$ from the second inequality into $g(n)$ in the first inequality gives that $f(n) \leq c_1 c_2 h(n)$ for all $n \geq max(n_1, n_2)$.

A related observation is that each of the following strict inequalities hold for sufficiently large n:

$$1 < \log_2 n < n < n \log_2 n < n^2 < n^3 < ... < 2^n < n! < n^n$$

Note that each successive pair of inequalities doesn't hold for all n. For example, $2^n < n!$ holds only for $n \geq 4$. Also note $n^3 < 2^n$ only holds for $n \geq 10$.

The point of the above rules is that it allows us to make immediate statements about the $O(\ )$ behavior of some rather complicated functions. For example, we can just look at the following function

$$t(n) = 5 + 8 \log_2 n + 16n + \frac{n(n-1)}{25}$$

and observe it is $O(n^2)$ by noting that the largest term is n^2. The rules above *justify* this observation.

 ©Michael Langer

Big O and sets of functions

Generally when we talk about $O()$ of some function $t(n)$, we use the "tightest" (smallest) upper bound we can. For example, if we observe that a function $f(n)$ is $O(\log_2 n)$, then generally we would not say that that $f(n)$ is $O(n)$, even though technically $f(n)$ *would* be $O(n)$, and it would also be $O(n^2)$, etc.

For a given simple $g(n)$ such as listed in the sequence of inequalities above, there are infinitely many functions $f(n)$ that are $O(g(n))$. So let's think about the set of functions that are $O(g(n))$. Up to now, we have been saying that some function $t(n)$ *is* $O(g(n))$. But sometimes we say that $t(n)$ *is a member of the set of functions that are* $O(g(n))$, or more simply $t(n)$ "belongs to" $O(g(n))$. In set notation, one writes "$t(n) \in O(g(n))$" where $\in$ is notation for set membership. With this notation in mind, and thinking of various $O(g(n))$ as sets of functions, the discussion in the paragraphs above implies that we have strict containment relations on these sets:

$$O(1) \subset O(\log n) \subset O(n) \subset O(n \log n) \subset O(n^2) \ \cdots \subset O(2^n) \subset O(n!) \ldots$$

For example, any function $f(n)$ that is $O(1)$ must also be $O(\log_2 n)$, etc. I will occasionally use this set notation in the course and say "$t(n) \in O(g(n))$" instead of "$t(n)$ is $O(g(n))$".

Big Omega (asymptotic lower bound)

With big O, we defined an asymptotic upper bound. We said that one function grows at most as fast as another function. There is a similar definition for an asymptotic lower bound. Here we say that one function grows *at least* as fast as another function. The lower bound is called "big Omega". [ASIDE: In the slides, I led up to this definition by defining an asymptotic bound. Here I'll just cut to the chase give the definition of big Omega.]

Definition (big Omega): We say that $t(n)$ is $\Omega(g(n))$ if there exists positive constants n_0 and c such that, for all $n \geq n_0$,

$$t(n) \ \geq \ c\, g(n).$$

The idea is that $t(n)$ grows at least as fast as $g(n)$ times some constant, for sufficiently large n. Note that the only difference between the definition of $O()$ and $\Omega()$ is the $\leq$ vs. $\geq$ inequality.

Example

Claim: Let $t(n) = \frac{n(n-1)}{2}$. Then $t(n)$ is $\Omega(n^2)$.

To prove this claim, first note that $t(n)$ is less than $\frac{n^2}{2}$ for all n, so since we want a *lower* bound we need to choose a smaller c than $\frac{1}{2}$. Let's try something smaller, say $c = \frac{1}{4}$.

$$\begin{aligned} \frac{n(n-1)}{2} &\geq \frac{n^2}{4} \\ \iff \quad 2n(n-1) &\geq n^2 \\ \iff \quad n^2 &\geq 2n \\ \iff \quad n &\geq 2 \end{aligned}$$

Note that the "if and only if" symbols $\Longleftrightarrow$ are crucial here. For any n, the first inequality is either true or false. We don't know which, until we check. But putting the $\Longleftrightarrow$ in there, we are say that the inequalities in the different lines have the same truth value.

The last lines says $n \geq 2$, this means that the first inequality is true if and only if $n \geq 2$. Thus, we can use $c = \frac{1}{4}$ and $n_0 = 2$.

Are these the only constants we can use? No. Let's try $c = \frac{1}{3}$.

$$
\begin{array}{rrcl}
 & \dfrac{n(n-1)}{2} & \geq & \dfrac{n^2}{3} \\
\Longleftrightarrow & \dfrac{3}{2}n(n-1) & \geq & n^2 \\
\Longleftrightarrow & \dfrac{1}{2}n^2 & \geq & \dfrac{3}{2}n \\
\Longleftrightarrow & n & \geq & 3
\end{array}
$$

So, we can use $c = \frac{1}{3}$ and $n_0 = 3$.

Finally, a few notes:

- You should be able to easily show that the constant, sum, product, transitivity rules all hold for big Omega also.

- We can say that if $f(n)$ *is* $\Omega(g(n))$ then $f(n)$ is a member of the set of functions that are $\Omega(g(n))$. The set relationship is different from what we saw with $O()$, i.e. note that the set membership symbols are in the opposite direction:

$$\Omega(1) \supset \Omega(\log n) \supset \Omega(n) \supset \Omega(n \log n) \supset \Omega(n^2) \; \cdots \supset \Omega(2^n) \supset \Omega(n!) \ldots$$

  For example, any positive function that is increasing with n will automatically $\Omega(1)$ but there are many increasing functions that are *not* bounded above by a constant i.e. $O(1)$.

Incorrect proofs

When one is first learning to write proofs, it is common to leave out certain important information. Let's look at a few examples of how this happens.

Claim:

For all $n \geq 1$, $\; 2n^2 \leq (n+1)^2$.

If you are like me, you probably can't just look at that claim and evaluate whether it is true or false. You need to carefully reason about it. Here is the sort of incorrect "proof" you might be tempted to write, given the sort of manipulations I've been doing in the course:

$$
\begin{array}{rcl}
2n^2 & \leq & (n+1)^2 \\
 & \leq & (n+n)^2, \text{ where } n \geq 1 \\
 & \leq & 4n^2
\end{array}
$$

which is true, i.e. $2n^2 \leq 4n^2$. Therefore, you might conclude that the claim you started with is true.

Unfortunately, the claim is false. Take $n = 3$ and note the inequality fails since $2 \cdot 3^2 > 4^2$. The proof is therefore wrong. What went wrong is that the first line of the proof *assumes what we are trying to prove.* This is a remarkably common mistake.

Here is another example.

Claim:

For all $n \geq 4$, $2^n \leq n!$

This claim is true, and I asked students to prove it by induction on a midterm exam in a previous year. Many students gave "proofs" by verifying that the base case ($n = 4$) is true, and then trying to prove the induction step as follows:

$$\begin{array}{rcll} 2^{k+1} & \leq & (k+1)! & \\ 2 \cdot 2^k & \leq & (k+1) \cdot k! & \\ 2 \cdot 2^k & \leq & (k+1) \cdot 2^k & \text{(induction hypothesis)} \\ 2 & \leq & k+1 & \end{array}$$

and observing that the last line is indeed true for $k \geq 4$.

However, this proof is *incorrect.* It doesn't specify which statement implies which, or which statements are logically equivalent.

Here is proof of the induction step that is correct:

$$\begin{array}{rcll} 2^{k+1} & = & 2 \cdot 2^k & \\ & \leq & 2 \cdot k! & \text{(by induction hypothesis)} \\ & \leq & (k+1) \cdot k! & \text{when } 2 \leq k+1, \text{ i.e. } k \geq 1 \\ & = & (k+1)! & \end{array}$$

Note that the induction step is proven for $k \geq 1$. But the base case is $n = 4$. So the claim only holds for $n \geq 4$, not $k \geq 1$. Indeed, the inequality of the claim is false for $k = 1, 2, 3$.

 ©Michael Langer