Properties of O-notation
COMS20010 2020, Video lecture 1-4

John Lapinskas, University of Bristol

The key to working with O-notation

Last time about comparing functions using the definitions of O-notation. You should almost never actually do this!

Your life will be much happier if you work mostly based on **intuition**.

Usually (not always!) if something is true for $\leq$, it is true for O.
For example, if $x \leq y$ and $y \leq z$ then $x \leq z$;
likewise, if $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$ then $f(n) \in O(h(n))$.

The same goes for $\geq$ and Ω, $=$ and Θ, $<$ and o, and $>$ and ω.
For example, if $x \leq y$ and $x \geq y$ then $x = y$;
likewise, if $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$, then $f(n) \in \Theta(g(n))$.

This, combined with the following rough hierarchy, will let you solve most problems without thinking about C's or n_0's:

$$n! \in \omega(3^n) \subseteq \omega(2^n) \subseteq \omega(n^2) \subseteq \omega(n) \subseteq \omega(\log^2 n) \subseteq \omega(\log n) \subseteq \omega(1).$$

When you *should* work formally

The time to fall back to definitions is when you need to confirm your intuition — when you're not sure if a general principle holds or not.

Example: Is it true that if $f(n) \in \Omega(g(n))$, then $f(n)^2 \in \Omega(g(n)^2)$?

Think back to the definitions.
We have: There exist c, $n_0 > 0$ such that $f(n) \geq cg(n)$ for all $n \geq n_0$.
We want: There exist c', $n_0' > 0$ such that $f(n)^2 \geq c'g(n)^2$ for all $n \geq n_0'$.

So we can just take $c' = c^2$ and $n_0' = n_0$ to prove $f(n)^2 \in \Omega(g(n)^2)$. ✓

Example: Is it true that if $f(n) < g(n)$ for all n, then $f(n) \in o(g(n))$?

We want: For all $C > 0$, there exists n_0 such that $f(n) < Cg(n)$ for all $n \geq n_0$.

Since we only have $f(n) < g(n)$, this looks dubious when $C \ll 1$...
One counterexample is $f(n) = n/2$, $g(n) = n$ (taking $C = 1/4$). ✓

L'Hôpital's rule

This is like a more powerful form of the racetrack principle from last year.

L'Hôpital's rule: Suppose $f, g\colon \mathbb{R} \to \mathbb{R}$ are differentiable and that $f(n),\ g(n) \in \omega(1)$. Then:

- $f(n) \in \omega(g(n))$ if and only if $f'(n) \in \omega(g'(n))$; and
- $f(n) \in o(g(n))$ if and only if $f'(n) \in o(g'(n))$.

Intuitively: This makes sense since f' and g' are the *rates of change* of f and g — if f grows much faster than g, then f' should grow much faster than g', and vice versa.

I won't prove it, though! (It's also a weaker form of the "real" result.)

Example: Prove that $n \in o(b^n)$ for all constants $b > 0$.

By L'Hôpital's rule, this holds if and only if $1 \in o(b^n \ln b) = o(b^n)$.
For any $C > 0$, we have $1 \leq C \cdot b^n$ for all $n \geq \log_b(1/C)$, so this is true.

Example: Proving that exponential beats polynomial

Theorem: For all polynomial functions $f(n) = \sum_i a_i n^{x_i}$ and all $y > 1$, we have $f(n) \in o(y^n)$.

Proof: By the hierarchy, we have $n^{x_i} \in o(n^{x_j})$ whenever $x_i < x_j$.

Fact: If $g(n) \in o(f(n))$, then $f(n) + g(n) \in \Theta(f(n))$. (Why?)

Hence $f(n) \in \Theta(n^x)$ for some $x > 0$, and we must show $n^x = o(y^n)$.

We have that $f(n)^x \in o(g(n)^x)$ if and only if $f(n) \in o(g(n))$, so it is enough to show $n \in o(y^{n/x}) = o((y^{1/x})^n)$.

We already saw this is true via L'Hôpital, so we're done. $\square$

Notice the overall process here: rather than working with definitions directly, we reduce the question to one we know how to solve.

Example: Dealing with unpleasant exponentials

Example: Prove that $2^{(\log\log n)^2} \in o(n)$ and $2^{(\log\log n)^2} \in \omega(\log n)$.

Problems like this are much easier if you give the two things you're trying to compare a common base.

Here, we have $n = 2^{\log n}$ and $\log n = 2^{\log\log n}$.

We have $(\log\log n)^2 \in o(\log n)$ and $(\log\log n)^2 = \omega(\log\log n)$, so "clearly" $2^{(\log\log n)^2} \in o(n)$ and $2^{(\log\log n)^2} \in \omega(\log n)$.

All we need is that if $f(n) = o(g(n))$, then $2^{f(n)} \in o(2^{g(n)})$, which is true... as long as $g(n) \in \omega(1)$. (Exercise!) ✓

(In practice, if you see a running time like this, you should be very careful even though it's theoretically fast — the constants are probably massive...)