CSE 241 Class 3

Jeremy Buhler

August 31, 2015

Time to stop playing fast and loose with asymptotic notation. Previously, we said that "$\Theta(f(n))$" means "roughly a constant times $f(n)$." How can we formalize this notion?

1 Definitions

Let $f(n)$ and $g(n)$ be two functions of n defined to be non-negative on positive integers (e.g., running times!).

- $f(n) = O(g(n))$ iff there exists a constant c and a value n_0 such that for every $n \geq n_0$,
  $$f(n) \leq c \cdot g(n).$$
  [Draw a picture!]

- $f(n) = \Omega(g(n))$ iff there exists a constant $c > 0$ and a value n_0 such that for every $n \geq n_0$,
  $$f(n) \geq c \cdot g(n).$$
  [Draw a picture!]

- $f(n) = \Theta(g(n))$ iff $f(n)$ is both $O(g(n))$ and $\Omega(g(n))$.
  [Draw a picture!]

Let's think about what these definitions mean. Do they match the intuitive notion of asymptotic complexity we had before?

- **Why allow an $n_0 > 1$?** Think of fast and slow closest-pair. For small n, quadratic algo dominates, but eventually $n \log n$ algorithm wins. (Remember **asymptopia**.)
- **Ignores constants**. Does $\Omega(3n)$ makes sense? Yes, but it's the same as $\Omega(n)$, so we never write the "3".
- **Ignores lower-order terms**. **Example**:

  $$n^2 + 7n + 5 = O(n^2).$$

  Proof: Let $c = 2$ and $n_0 = 8$. Consider that

  $$n^2 + 7n + 5 - 2n^2 = 0$$

  for $n \approx 7.65$. Easy to check that derivative of difference is negative, so $n^2 + 7n + 5 \leq 2n^2$ for $n \geq 8$. QED.
- Can you prove the Ω relation for the above pair of functions? [**trivial:** $n_0 = 1$, $c = 1$].

Let's do some more examples.

- **Example**:

  $$n^2 = \Omega(n \log n).$$

  Proof: Let $c = 1$. $n^2 > n \log n$ for $n_0 = 1$. Observe that continuous functions n^2 and $n \log n$ never cross for any $n > 0$, so n^2 must always be larger for every positive n. QED.
- **Counterexample**:

  $$\text{Does } 6n^2 = \Omega(n^3) \text{ ?}$$

  Contradiction Proof: Suppose yes. Then for some $c > 0$ and all sufficiently large n,

  $$6n^2 \geq cn^3.$$

  but this implies that $6 \geq cn$, or equivalently $n \leq 6/c$, which is surely false for any fixed c and sufficiently large n. QED.

Technically, the "=" is poor notation. $O(g(n))$ means "the set of all functions that are at most a constant times $g(n)$ for all sufficiently large n," so we should say $f(n) \in O(g(n))$. Oh well.

2 Quick and Easy Comparisons of Growth

Is there an easy way to tell which of two functions $f(n)$ and $g(n)$ grows faster? Yes! **Consider their ratio as n gets big.** This procedure is practically mechanical, unlike direct proof that relies on various functional analysis tech.

- Suppose
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$
Which function grows faster? [wait] ($g(n)$, of course.)
  - Does $f(n) = O(g(n))$? Recall definition of limit... for all large enough n,
$$\frac{f(n)}{g(n)} - 0 < 1.$$
Hence, for large enough n, $f(n) \leq g(n)$. QED.
  - Can $f(n) = \Omega(g(n))$? If so, then $f(n) \geq cg(n)$ for some c and all sufficiently large n. But for all large enough n,
$$\frac{f(n)}{g(n)} - 0 < c,$$
which contradicts claim. Hence, $f(n)$ is *not* $\Omega(g(n))$.
  - When $f(n)$ is $O(g(n))$ but not $\Omega(g(n))$, we write $f(n) = o(g(n))$.

- Suppose
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty.$$
Which function grows faster? [wait] ($f(n)$, of course.)
  - By similar arguments, can say that $f(n) = \Omega(g(n))$ but is never $O(g(n))$.
  - When $f(n)$ is $\Omega(g(n))$ but not $O(g(n))$, we write $f(n) = \omega(g(n))$.

- Suppose
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$$
for constant $c > 0$. What can we say? **[wait]** (functions grow at same rate.) For all large enough n and any $\varepsilon > 0$,
$$\left|\frac{f(n)}{g(n)} - c\right| < \varepsilon,$$
which implies $(c - \varepsilon)g(n) \leq f(n) \leq (c + \varepsilon)g(n)$, or equivalently $f(n) = \Theta(g(n))$.

Let's do some examples:

- n^3 versus n^2? Well,

$$\frac{n^3}{n^2} = n,$$

so the limit is ∞, and so n^3 grows faster.

- n versus $\log n$? Hmmm... what is

$$\lim_{n\to\infty} \frac{n}{\log n} \ ?$$

Recall l'Hôpital's Rule:

$$\text{If } \lim_{n\to\infty} \frac{f(n)}{g(n)} \text{ is ill-defined, compute } \lim_{n\to\infty} \frac{\frac{d}{dn}f(n)}{\frac{d}{dn}g(n)}.$$

Now it's obvious:

$$\begin{aligned} \lim_{n\to\infty} \frac{n}{\log n} &= \lim_{n\to\infty} \frac{c}{1/n} \\ &= \lim_{n\to\infty} cn \\ &= \infty. \end{aligned}$$

so n grows faster than $\log n$.

- n^2 versus $3n^2 + 5n + 7$?

$$\begin{aligned} \lim_{n\to\infty} \frac{n^2}{3n^2+5n+7} &= \lim_{n\to\infty} \frac{2n}{6n+5} \\ &= \lim_{n\to\infty} \frac{2}{6} \\ &= 1/3 \end{aligned}$$

so the functions grow at the same rate.

- n^4 versus 2^n?

$$\begin{aligned} \lim_{n\to\infty} \frac{n^4}{2^n} &= \lim_{n\to\infty} \frac{4n^3}{(\ln 2)2^n} \\ &= \lim_{n\to\infty} \frac{12n^2}{(\ln 2)^2 2^n} \\ &= \lim_{n\to\infty} \frac{24n}{(\ln 2)^3 2^n} \\ &= \lim_{n\to\infty} \frac{24}{(\ln 2)^4 2^n} \\ &= 0. \end{aligned}$$

so 2^n grows faster than n^4. **Inductive proof along same lines shows that a^n grows faster than n^b for all positive constants a and b.**

3 Relationships Among O, Ω, Θ

Some useful facts:

1. $f(n) = O(g(n))$ iff $g(n) = \Omega(f(n))$. (Proof by definitions of O, Ω).
2. Above implies $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$.
3. More useful facts in your text: beginning of Chapter 3.

Here's something to ponder. Define $f(n)$ as follows:

$$f(n) = \begin{cases} 1 & \text{if } n \text{ even} \\ n^2 & \text{if } n \text{ odd} \end{cases}$$

- Clearly, $f(n)$ is *not* $O(n)$.
- Does $f(n) = \Omega(n)$? *No!* No matter how large n gets, there is always *some* larger n such that $f(n) < cn$ for any fixed c.
- Shows that given two functions $f(n)$ and $g(n)$, they may be asymptotically **incomparable**.
- Does this answer seem "right" if $f(n)$ describes the running time of an algorithm? Some people use a different notion of Ω because of this phenomenon.

4 Importance for Running Time

Which algorithm is fastest? (Which running time is smallest, i.e. grows most slowly?) Assume worst-case times given as follows:

Algorithm	Time
A1	$\Theta(n^2)$
A2	$\Theta(n \log n)$
A3	$\Omega(n^2)$
A4	$\Theta(n^3)$
A5	$\Theta(n \log n)$

Clearly, A1, A3, A4 are slower than A2, A5. How do we choose between the last two?

- Constants of abstract reps, if sufficiently different.
- Otherwise, code up implementations and test!
- What if we added $A6 = O(n^2)$?
- Cannot tell – could be either faster *or* slower than $\Theta(n \log n)$.