CSC110 Lecture 21: Asymptotic Notation for Function Growth

Print this handout

Exercise 1: Practice with Big-O

As a reminder, here is the formal definition "g is Big-O of f":

$$g \in \mathcal{O}(f) : \exists c, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow g(n) \leq c \cdot f(n)$$

Note: In Big-O expressions, it will be convenient to just write down the "body" of the functions rather than defining named functions all the time. We'll always use the variable n to represent the function input, and so when we write "$n \in \mathcal{O}(n^2)$," we really mean "the functions defined as $f(n) = n$ and $g(n) = n^2$ satisfy $f \in \mathcal{O}(g)$."

1. Our first step in comparing different types of functions is looking at different powers of n. Consider the following statement, which generalizes the idea that $n \in \mathcal{O}(n^2)$:

$$\forall a, b \in \mathbb{R}^+, a \leq b \Rightarrow n^a \in \mathcal{O}(n^b)$$

   a. Rewrite the above statement, but with the definition of Big-O expanded.

$$\forall a, b \in \mathbb{R}^+, \ a \leq b \Rightarrow \left(\exists c, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, \ n \geq n_0 \Rightarrow n^a \leq c \cdot n^b\right)$$

   b. Prove the above statement. **Hint**: you can actually pick c and n_0 to both be 1, and have the proo work. Even though this is pretty simple, take the time to understand and write down the proof, as a good warm-up.

Let $a, b \in \mathbb{R}^+$.

Let $c = 1$ and $n_0 = 1$.
Let $n \in \mathbb{N}$ and assume $n \geq n_0$.

Assume $a \le b$.

We want to show
$\exists c, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N},$
$n \ge n_0 \Rightarrow n^a \le c \cdot n^b$

We want to show $n^a \le c \cdot n^b$
Since $a \le b$, $0 \le b-a$,
and $0 \le n^0 \le n^{b-a}$.
Also, $n^a \le n^a$
so $n^a \cdot n^0 \le n^{b-a} \cdot n^a$
or $n^a \le n^b$
$= c \cdot n^b$
as req'd.

2. In this question, we'll investigate what it means to show that a function *isn't* Big-O of another.

   a. Express the statement $g \notin \mathcal{O}(f)$ in predicate logic, using the expanded definition of Big-O. Simplify so that all negations are pushed as far "inside" as possible.

$g \notin O(f)$ means $\neg\, g \in O(f)$ or
$\neg(\exists c, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \ge n_0 \Rightarrow g(n) \le c f(n))$
or $\boxed{\forall c, n_0 \in \mathbb{R}^+, \exists n \in \mathbb{N}, n \ge n_0 \land g(n) > c f(n)}$
① ②

$\neg(a \Rightarrow b)$
$\Leftrightarrow a \land \neg b$

always not bounded above for some $n \ge n_0$

   b. Prove that for all positive real numbers a and b, if $a > b$ then $n^a \notin \mathcal{O}(n^b)$.

      Hints:

      - In your negated statement from part (a), you should find that you can only pick the value n. Work backwards from the target inequalities to find what value of n works.

Let $a, b \in \mathbb{R}^+$ and assume $a > b$. We want to prove $n^a \notin O(n^b)$.
Let $c, n_0 \in \mathbb{R}^+$ and let $n = \lceil n_0 + c^{1/a-b} \rceil$

roughwork:
want $n^a > c \cdot n^b$ (isolate n)
$\times$ both sides by n^{-b}
$n^{a-b} > c$

① show $n \ge n_0$
since $n = \lceil n_0 + c^{\frac{1}{a-b}} \rceil$,
$n \ge n_0$

② show $n^a > c \cdot n^b$

take a-bth root

$(n^{a-b})^{\frac{1}{a-b}} > c^{1/a-b}$

$n > c^{1/a-b}$

since $n = \lceil n_0 + c^{\frac{1}{a-b}} \rceil$,

$n > c^{\frac{1}{a-b}}$

take a-b power

then $n^{a-b} > c$

or $n^a > cn^b$

as req'd

Exercise 2: Omega and Theta

1. Consider the following statement: for all $a, b \in \mathbb{R}^+$, $an^2 + b \in \Theta(n^2)$.

   $g(n)$ $f(n)$

   a. Expand this statement using the definition of Theta involving c_1 and c_2.

   $\forall a, b \in \mathbb{R}^+, \exists c_1, c_2, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow c_1 n^2 \leq an^2 + b \leq c_2 n^2$

   b. You should see two different inequalities that you need to prove: $c_1 n^2 \leq an^2 + b$ and $an^2 + b \leq c_2 n^2$.

   Which of these variables are universally quantified? Which are existentially quantified?

   n, a, b — c_1, c_2, n_0

   c. Choose values for the existentially-quantified variables from (b) to make the two inequalities true (you might need to do some rough calculations). Remember that when choosing values for these variables, you can define them in terms of any variables that appear to the *left* of them in the statement you wrote in part (a).

   $c_1 n^2 \leq an^2 + b$

   since $b > 0$, $c_1 = a$ works

   since $c_1 n^2 = an^2 \leq an^2 + b$

   $an^2 + b \leq c_2 n^2$

   provided $n \geq 1$,

   $an^2 + b \leq an^2 + bn^2 = (a+b)n^2$

   so $c_2 = (a+b)$ works

   d. Prove your statement from part (a).

   Let $a, b \in \mathbb{R}^+$.

Let $c_1 = a$, $c_2 = a+b$, $n_0 = 1$.

Let $n \in \mathbb{N}$ and assume $n \geq n_0$.

Then $c_1 n^2 = an^2$

$\leq an^2 + b$

and $c_2 n^2 = (a+b)n^2$

$= an^2 + bn^2$

$\geq an^2 + b$ (since $n^2 \geq 1$, $bn^2 \geq b$)

or $c_1 n^2 \leq an^2 + b \leq c_2 n^2$,

as req'd.

Additional exercises

1. One slight oddness about the definition of Big-O is that it treats all logarithmic functions "the same." Your task in this question is to investigate this, by proving the following statement:

$$\forall a, b \in \mathbb{R}^+,\ a > 1 \land b > 1 \Rightarrow \log_a n \in \Theta(\log_b n)$$

We won't ask you to expand the definition of Theta, but if you aren't quite sure, then we highly recommend doing so before attempting even your rough work!

Hint: look up the "change of base rule" for logarithms, if you don't quite remember it.

2. Prove that for all $a, b \in \mathbb{R}^+$, if $a < b$ then $a^n \in \mathcal{O}(b^n)$ and $a^n \notin \Omega(b^n)$.

3. As we discuss in the Course Notes, *constant functions* like $f(n) = 100$ will play an important role in our analysis of running time of algorithms. For now let's get comfortable with the notation.

   a. Let $g : \mathbb{N} \to \mathbb{R}^{\geq 0}$. Show how to express the statement $g \in \Theta(1)$ by expanding the definition of Theta. (Here 1 refers to the constant function $f(n) = 1$.)

   b. Prove that $100 + \frac{77}{n+1} \in \Theta(1)$.