Computer Science 331

Asymptotic Notation

Mike Jacobson

Department of Computer Science
University of Calgary

Lecture #9

Outline

Properties and Application

Asymptotic Notation . . .

- provides information about the *relative rates of growth* of a pair of functions (of a single integer or real variable)
- ignores or hides other details, including
  - behaviour on *small* inputs — results are most meaningful when inputs are extremely *large*
  - multiplicative constants and lower-order terms — which can be implementation or platform-dependent anyway
- permits classification of algorithms into classes (eg. linear, quadratic, polynomial, exponential, etc...)
- is useful for giving the kinds of bounds on running times of algorithms that we will study in this course

Big-Oh Notation

Definition: Suppose $f, g : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$.

$f \in O(g)$:

There exist constants $c > 0$ *and* $N_0 \geq 0$ *such that*

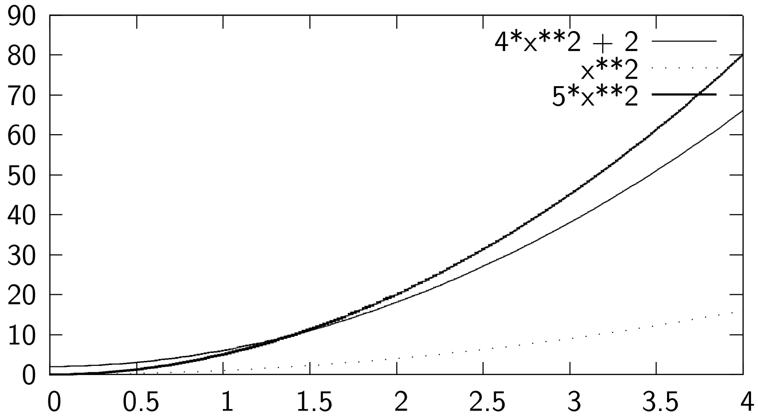
$$f(n) \leq c \cdot g(n)$$

for all $n \geq N_0$.

Intuition:

- growth rate of f is at most (same as or less than) that of g
- Eg. $4n + 3 \in O(n)$ — definition is satisfied using $c = 5$ and $N_0 = 3$
- sometimes written $f = O(g)$ (also OK)

Example: $4n^2 + 2 \in O(n^2)$



Proof that $4n^2 + 2 \in O(n^2)$

Theorem 1

$4n^2 + 2 \in O(n^2)$

Proof.

4n^2+2 < cn^2.

show 4n^2+2 <= 5n^2 for all n >=2

□

Note: this proof is *constructive* in that it determines the appropriate constants. Also OK to find constants by any means and simply prove that they satisfy the definition.

Big-Omega Notation

Definition: Suppose $f, g : \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$.

$f \in \Omega(g)$:

There exist constants $c > 0$ and $N_0 \geq 0$ such that

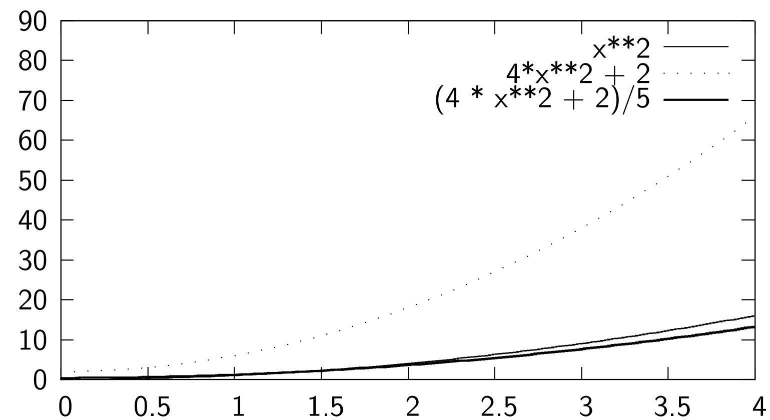
$$f(n) \geq c \cdot g(n)$$

for all $n \geq N_0$.

Intuition:

- growth rate of f is at least (the same as or greater than) that of g
- $4n + 3 \in \Omega(n)$ — definition is satisfied using $c = N_0 = 1$

Example: $n^2 \in \Omega(4n^2 + 2)$



Transpose Symmetry

Theorem 2

Suppose $f, g : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$. Then $f \in O(g)$ if and only if $g \in \Omega(f)$.

Proof.

then there is a c >0, N0 >= 0,
f(n) <=cg(n).
Then g(n) >= 1/c(f(n)), c is >
0. QED. reverse.

$\square$

Big-Theta Notation

Definition: Suppose $f, g : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$.

$f \in \Theta(g)$:

There exist constants $c_L, c_U > 0$ and $N_0 \geq 0$ such that

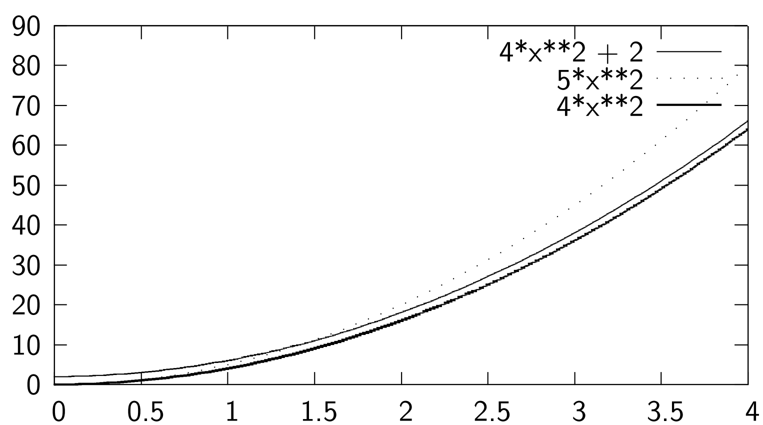
$$c_L g(n) \leq f(n) \leq c_U \cdot g(n)$$

for all $n \geq N_0$.

Intuition:

- f has the same growth rate as g
- $4n + 3 \in \Theta(n)$ — definition is satisfied using $c_L = 1$, $c_U = 5$, $N_0 = 3$

Example: $4n^2 + 2 \in \Theta(n^2)$



An Equivalent Definition

Theorem 3

Suppose $f, g : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$. Then $f \in \Theta(g)$ if and only if

$$f \in O(g) \text{ and } f \in \Omega(g)$$

Exercise: Prove that the two definitions of "$f \in \Theta(g)$" are *equivalent*.

How To Solve This:

- Work from the definitions, as in previous example!

A Common Mistake

People sometimes write "f is $O(g)$" (which is yet another way to write "$f \in O(g)$" or "$f = O(g)$") when they actually mean "$f \in \Theta(g)$."

Please note that if $f \in O(g)$ then it is *not* necessarily true that $f \in \Theta(g)$ as well

- For example, as functions of n, $n \in O(n^2)$ but $n \notin \Theta(n^2)$.

So: If you want people to understand that "$f \in \Theta(g)$" then this is what you should write!

Little-oh Notation

Definition: Suppose $f, g : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$.

$f \in o(g)$:

For every *constant* $c > 0$ **there exists a constant** $N_0 \geq 0$ *such that*

$$f(n) \leq c \cdot g(n)$$

for all $n \geq N_0$.

Intuition:

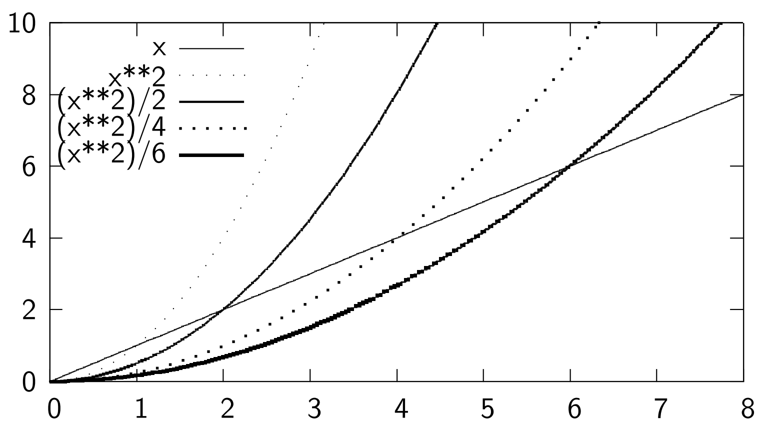
- f grows strictly slower than g

x <= cx^2

1/c <= x

Big-Oh versus Little-Oh: notice how the constant c is quantified!

Example: $x \in o(x^2)$



Little-omega Notation

Definition: Suppose $f, g : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$.

$f \in \omega(g)$:

For every *constant* $c > 0$ **there exists a constant** $N_0 \geq 0$ *such that*

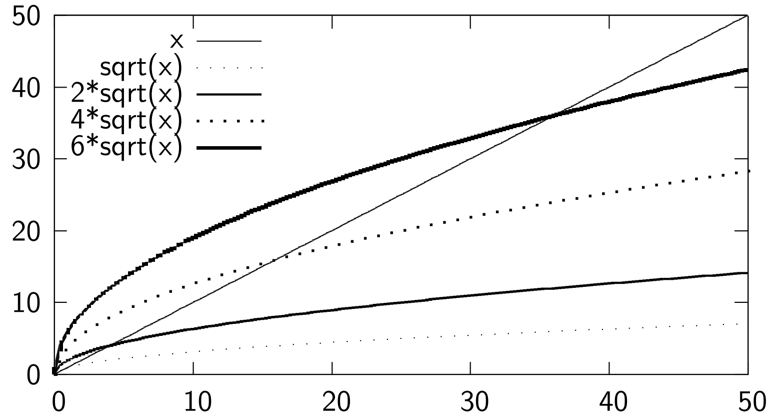
$$f(n) \geq c \cdot g(n)$$

for all $n \geq N_0$.

Intuition:

- f grows strictly faster than g

Example: $x \in \omega(\sqrt{x})$



Useful Properties

Suppose $f, g : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$.

Useful properties:

- $f \in o(g) \Rightarrow f \in O(g)$
- $f \in \omega(g) \Rightarrow f \in \Omega(g)$
- *Transpose Symmetry:*
  $$f \in o(g) \iff g \in \omega(f)$$
- *Limit Test:*
  $$f \in o(g) \iff \lim_{x \to +\infty} \frac{f(x)}{g(x)} = 0$$
- *Limit Test:*
  $$f \in \omega(g) \iff \lim_{x \to +\infty} \frac{f(x)}{g(x)} = +\infty$$

Some Standard Functions

Polynomial (degree d): $p(n) = a_d n^d + a_{d-1} n^{d-1} + \cdots + a_1 n + a_0$

- $p(n) \in \Theta(n^d)$

Exponentials: a^n, $a \in \mathbb{R}^{\geq 0}$ (increasing if $a > 1$)

- if $a > 1$, then $a^n \in \omega(p(n))$ for every polynomial $p(n)$

Logarithms: $\log_a n$, $a \in \mathbb{R}^{\geq 0}$

- $(\log_a n)^k \in o(p(n))$ whenever $a > 1$, $k \in \mathbb{R}^{\geq 0}$, and $p(n)$ is a polynomial with degree at least one

Recommended Reading

Chapter 3 of *Introduction to Algorithms*. Especially useful:

- Comparing functions, Exercises (pp. 51–53)
- Standard Notation and Common Functions (Section 3.2):
  - Floors and Ceilings
  - Modular Arithmetic
  - Standard Functions: Polynomials, Exponentials, Logarithms, and Their Properties