Chapter 3
Growth of Functions

C.K. Huang

Overview

- A way to describe behavior of functions in the limit. We're studying asymptotic(漸近) efficiency.
- Describe growth of functions.
- Focus on what's important by abstracting away low-order terms and constant factors.
- How we indicate running times of algorithms.
- A way to compare "sizes" of functions:

$$
\begin{array}{ccc}
O & \approx & \leq \\
\Omega & \approx & \geq \\
\Theta & \approx & = \\
o & \approx & < \\
\omega & \approx & >
\end{array}
$$

Graphic Examples

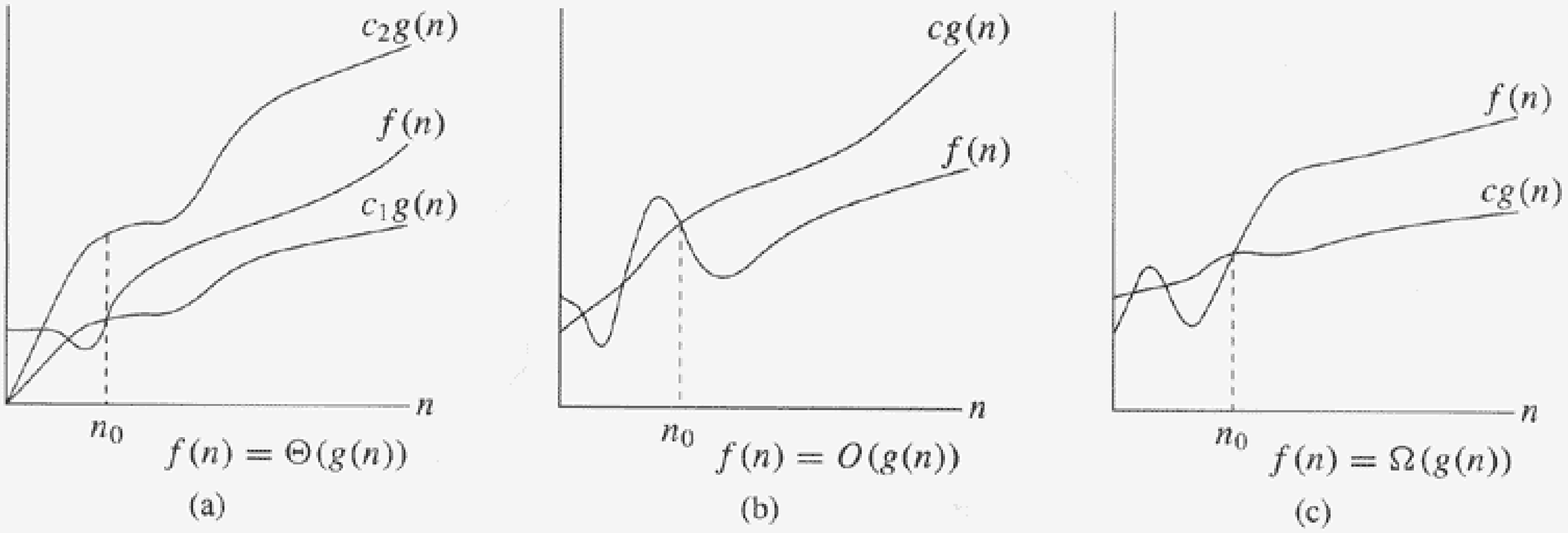
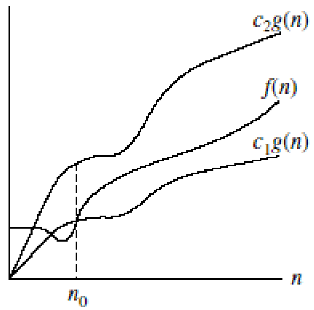


Figure 3.1 Graphic examples of the Θ, O, and Ω notations. In each part, the value of n_0 shown is the minimum possible value; any greater value would also work. **(a)** Θ-notation bounds a function to within constant factors. We write $f(n) = \Theta(g(n))$ if there exist positive constants n_0, c_1, and c_2 such that to the right of n_0, the value of $f(n)$ always lies between $c_1g(n)$ and $c_2g(n)$ inclusive. **(b)** O-notation gives an upper bound for a function to within a constant factor. We write $f(n) = O(g(n))$ if there are positive constants n_0 and c such that to the right of n_0, the value of $f(n)$ always lies on or below $cg(n)$. **(c)** Ω-notation gives a lower bound for a function to within a constant factor. We write $f(n) = \Omega(g(n))$ if there are positive constants n_0 and c such that to the right of n_0, the value of $f(n)$ always lies on or above $cg(n)$.

Θ-notation

$$\Theta(g(n)) = \left\{ f(n) \,\middle|\, \begin{array}{l} \text{there exist positive constants } c_1, c_2, \text{ and } n_0 \\ \text{such that } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge n_0 \end{array} \right\}$$

$$f(n) \in \Theta(g(n)) \text{ or } f(n) = \Theta(g(n))$$



- $g(n)$ is an asymptotically tight bound for $f(n)$
- $f(n)$ is asymptotically nonnegative

$$f(n) = an^2 + bn + c = \Theta(n^2), \text{ where } a > 0$$

$$f(n) = \sum_{i=0}^{d} a_i n^i = \Theta(n^d)$$

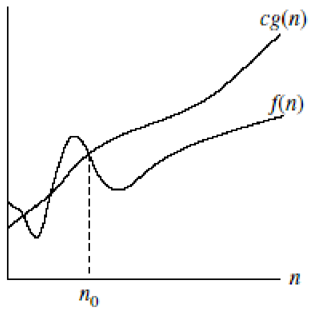
$$\Theta(n^0) = \Theta(1), \text{ a constant or a constant function.}$$

O-notation

$$O(g(n)) = \left\{ f(n) \middle| \begin{array}{l} \text{there exist positive constants } c \text{ and } n_0 \\ \text{such that } 0 \le f(n) \le cg(n) \text{ for all } n \ge n_0 \end{array} \right\}$$

$$f(n) = \Theta(g(n)) \xrightarrow{\textit{implies}} f(n) = O(g(n))$$

$$\Theta(g(n)) \subseteq O(g(n))$$



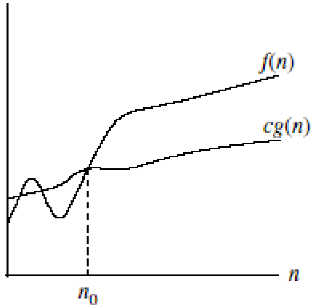
- "big-oh of g of n"
- Asymptotically upper bound → worst-case running time
- Θ-notation is a stronger notion than O-notation
- Using O-notation, we ca often describe the running time of an algorithm merely by inspecting the algorithm's overall structure.
  - Insertion-sort

$$an^2 + bn + c = O(n^2)$$

$$an + b = O(n^2)$$

Ω-notation

$$\Omega(g(n)) = \left\{ f(n) \,\middle|\, \begin{array}{l} \text{there exist positive constants } c \text{ and } n_0 \\ \text{such that } 0 \le cg(n) \le f(n) \text{ for all } n \ge n_0 \end{array} \right\}$$



- "big-omega of g of n"
- Asymptotically lower bound → best-case running time
- **Theorem:**

  $f(n) = \Theta(g(\mathrm{n}))$ if and only if $f(n) = \Omega(g(\mathrm{n}))$ and $f(n) = O(g(\mathrm{n}))$

Asymptotic Notation in Equations

- On right-hand side:
  - $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$
  - $2n^2 + 3n + 1 = 2n^2 + f(n)$
    for some $f(n) \in \Theta(n)$
    in this case, $f(n) = 3n + 1$

$$\sum_{i=1}^{n} O(i) \qquad \text{OK}$$

$$O(1) + O(2) + \cdots + O(2) \qquad \text{not OK}$$

- On left-hand side:
  - No matter how the anonymous functions are chosen on the left-hand side, there is a way to choose the anonymous functions on the right-hand side to make the equation valid.
  - $2n^2 + \Theta(n) = \Theta(n^2)$
  - $2n^2 + f(n) = g(n)$
    for any function $f(n) \in \Theta(n)$,
    there exists a function $g(n) \in \Theta(n^2)$

$$\begin{aligned} 2n^2 + 3n + 1 &= 2n^2 + \Theta(n) \\ &= \Theta(n^2) \end{aligned}$$

o-notation

$$o(g(n)) = \left\{ f(n) \,\middle|\, \begin{array}{l} \text{for any constant } c > 0, \text{ there exist a constants } n_0 > 0 \\ \text{such that } 0 \le f(n) < cg(n) \text{ for all } n \ge n_0 \end{array} \right\}$$

$$2n = o(n^2), \text{ but } 2n^2 \ne o(n^2)$$

- “little-oh of g of n”
- An upper bound that is not asymptotically tight.

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

ω-notation

$$\omega(g(n)) = \left\{ f(n) \;\middle|\; \begin{array}{l} \text{for any constant } c > 0, \text{ there exist a constant } n_0 > 0 \\ \text{such that } 0 \le cg(n) < f(n) \text{ for all } n \ge n_0 \end{array} \right\}$$

$$f(n) \in \omega(g(n)) \text{ if and only if } g(n) \in o(f(n))$$

$$\frac{n^2}{2} = \omega(n), \text{ but } \frac{n^2}{2} \ne \omega(n^2)$$

- "little-omega of g of n"
- A lower bound that is not asymptotically tight.

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$$

Comparison of Functions – Relational Properties

- Transitivity

  $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$
  (same for O, Ω, o, ω)

- Reflexivity

  $f(n) = \Theta(f(n))$ (same for O, Ω)

- Symmetry

  $f(n) = \Theta(g(n))$ if and only if $g(n) = \Theta(f(n))$

- Transpose symmetry

  $f(n) = O(g(n))$ if and only if $g(n) = \Omega(f(n))$

  $f(n) = o(g(n))$ if and only if $g(n) = \omega(f(n))$

Comparison of Functions – Comparisons

$$f(n) = O(g(n)) \quad \approx \quad a \le b$$
$$f(n) = \Omega(g(n)) \quad \approx \quad a \ge b$$
$$f(n) = \Theta(g(n)) \quad \approx \quad a = b$$
$$f(n) = o(g(n)) \quad \approx \quad a < b$$
$$f(n) = \omega(g(n)) \quad \approx \quad a > b$$

- $f(n)$ is asymptotically smaller than $g(n)$
  $g(n)$ is asymptotically larger than $f(n)$
- **No trichotomy**(三一律)

$$a < b, a = b, a > b$$

Standard Notations – Monotonicity

- $f(n)$ is monotonically increasing if $m \leq n \Rightarrow f(m) \leq f(n)$.
- $f(n)$ is monotonically decreasing if $m \geq n \Rightarrow f(m) \geq f(n)$.
- $f(n)$ is strictly increasing if $m < n \Rightarrow f(n) < f(n)$.
- $f(n)$ is strictly decreasing if $m > n \Rightarrow f(n) > f(n)$.

Standard Notations – Floors and Ceilings

For all real x, $x-1<\lfloor x\rfloor \le x \le \lceil x\rceil < x+1$

For any integer n, $\left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor = n$

For any real number $n \ge 0$, and integers $a, b > 0$,

$\lceil \lceil n/a \rceil / b \rceil = \lceil n/ab \rceil$

$\lfloor \lfloor n/a \rfloor / b \rfloor = \lfloor n/ab \rfloor$

$\lceil a/b \rceil \le (a + (b-1))/b$

$\lfloor a/b \rfloor \ge (a - (b-1))/b$

$f(x) = \lfloor x \rfloor$ and $f(x) = \lceil x \rceil$ are monotonically increasing.

Standard Notations – Modular Arithmetic

- Remainder or residue

For any integers a, and any positive integer n,

$$a \bmod n = a - \lfloor a/n \rfloor n$$

$$(a \bmod n) = (b \bmod n) \Rightarrow a \equiv b \pmod{n}$$

$$a \equiv b \pmod{n} \Leftrightarrow n \text{ is a divisor of } |b - a|$$

Standard Notations – Polynomials

Given a nonnegative integer d, a polynomial in n of degree d,

$p(n) = \sum_{i=0}^{d} a_i n^i$, where $a_1, a_2, \cdots, a_d$ are the coefficients.

$a_d > 0 \Leftrightarrow$ A polynomial is asymptotically positive.

$p(n) = \Theta(n^d)$

$a \geq 0$, n^a is monotonically increasing.

$a \leq 0$, n^a is monotonically decreasing.

$f(n)$ is polynomially bounded if $f(n) = O(n^k)$ for some consnstant k.

Standard Notations – Exponentials

For all real $a > 0$, m, and n,

$$a^0 = 1$$

$$a^1 = a$$

$$a^{-1} = 1/a$$

$$(a^m)^n = a^{mn} = (a^n)^m$$

$$a^m a^n = a^{m+n}$$

For all real constants $a > 1$ and b,

$$\lim_{n\to\infty} \frac{n^b}{a^n} = 0 \Rightarrow n^b = o(a^n)$$

- For all real x,

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

$$e^x > 1 + x$$

- When $|x| \le 1$,

$$1 + x \le e^x \le 1 + x + x^2$$

- When $x \to 0$,

$$e^x = 1 + x + \Theta(x^2)$$

$$\lim_{n\to\infty}\left(1 + \frac{x}{n}\right)^n = e^x$$

Standard Notations – Logarithms

$$\lg n = \log_2 n$$
$$\ln n = \log_e n$$
$$\lg^k n = (\lg n)^k$$
$$\lg \lg n = \lg(\lg n)$$

For $b > 1, n > 0,$

$\log_b n$ is strictly increasing

For all real $a > 0, b > 0, c > 0,$ and n

$$a = b^{\log_b a}$$
$$\log_c(ab) = \log_c a + \log_c b$$
$$\log_b a^n = n \log_b a$$
$$\log_b a = \frac{\log_c a}{\log_c b}$$
$$\log_b \frac{1}{a} = -\log_b a$$
$$\log_b a = \frac{1}{\log_a b}$$
$$a^{\log_b c} = c^{\log_b a}$$

Standard Notations – Logarithms (2)

- When $|x| < 1$,

$$\ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \cdots.$$

- For $x > -1$,

$$\frac{x}{1 + x} \le \ln(1 + x) \le x$$

$f(n)$ is polylogarithmically bounded if $f(n) = O(\lg^k n)$.

- For $a > 0$,

$$\lim_{n\to\infty} \frac{\lg^b n}{(2^a)^{\lg n}} = \lim_{n\to\infty} \frac{\lg^b n}{n^a} = 0$$

$$\lg^b n = o(n^a).$$

Standard Notations – Factorials

$$n! = \begin{cases} 1 & \text{if } n = 0, \\ n \cdot (n-1)! & \text{if } n > 0. \end{cases}$$

$$n! = o(n^n)$$

$$n! = \omega(2^n)$$

$$\lg(n!) = \Theta(n \lg n)$$

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right), \text{Stirling's Approximation}$$

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\alpha_n}, \text{ where} \frac{1}{12n+1} < \alpha_n < \frac{1}{12n}$$

Standard Notations – etc…

- Functional iteration
- Iterated logarithm function
- Fibonacci numbers