Asymptotics & Analysis of Alg.

Chapter Overview

1. Introduction
2. Growth Rate of Functions
3. Asymptotic Notations (O, Ω, Θ)
   - Definitions and examples
   - Properties & Theorems
   - Alternative Definition using Limits
   - Different Methods of Proving
4. Efficiency Considerations [in another document]
5. Time & Space Complexity of Algorithms [… same …]

Reading Assignments

[1] [CLRS], Chapter 1-2
[2]* [Weiss92], Ch. 2, pp. 15-40,

My Comments:

This chapter on asymptotic and analysis of algorithm is an important topic, not just for IT111, but also for CS103, CS104, and CS203, CS305, and many other courses in DISCS. It is perhaps one of the most fundamental topic in all of computer science and it laid the foundations for rapid and fruitful advancement in the design and analysis of algorithms for all areas of algorithmic (which is *everywhere* and *anywhere*) since the late sixties. I would highly recommend all of you to read [1] and [2]* both of which contain excellent introduction to both topics covered in this chapter.

Asymptotics and analysis of algorithms is NOT a hard topic, but the student should set out by thinking about what to do if we want to analyze algorithms (and programs) and also compare one program with another. Then, you will appreciate the beauty of these notations and also the shortcuts they provide. Enjoy!

Note added in 2013: These are VERY old notes. CS103, CS203, CS305 existed before DISCS moved to the modular curriculum in 1994/95 AY.

(Started: 1990; Revised: 1992, Jan 2007, Aug 2013)

Asymptotics - Introduction

Reading: [CLR90], Chapter 2, pp. 23-41.
*[Weiss92], Ch. 2, pp. 15-40,

AIM:

- establish relative order among functions
- based on their $\boxed{\text{rate of growth}}$.
- interested in asymptotic growth rate (for large n)

$\boxed{\text{apply them to analysis of algorithms}}$

Typical Rates of Growth: (what if input size *doubles*)

Function	Growth Rate	How-fast?
$k(n) = \text{c}$	Constant	$k(2n) = k(n)$
$\lg(n) = \lg n$	logarithmic	$\lg(2n) = \lg(n) + 1$
$L(n) = n$	Linear	$L(2n) = 2\ L(n)$
$L_2(n) = n \lg n$	n-log-n	$L_2(2n) \approx 2\ L_2(n)$
$Q(n) = n^2$	Quadratic	$Q(2n) = 4\ Q(n)$
$C(n) = n^3$	Cubic	$C(2n) = 8\ C(n)$
$E(n) = 2^n$	Exponential	$E(2n) = E(n) \cdot E(n)$

Note:

- We will be dealing with $\boxed{\text{nonnegative}}$ functions.
- Used to represent running time of algorithms.

Asymptotic Growth of Functions

Comparisons of Rate-of-Growth of functions

- Logarithmic, Linear, quadratic, exponential

$f(n) = 10000 \lg n$	$g(n) = 100n$
$q(n) = n^2$	$h(n) = 2^n$

$g(10) = 1000$	$q(10) = 100$	$h(10) = 1024$
$g(100) = 10000$	$q(100) = 1000000$	$h(100) = ??$

- We ask how fast $f(n)$ grows with-respect-to n?

  For small n, $100n$ is larger than n^2,
  But, n^2 has a faster growth rate than $100n$, and
  Eventually, n^2 will be the larger function.

Asymptotic Rate-of-Growth

- For large n, constant multipliers are not important
- Can establish following relative ordering

$$\boxed{10000 \lg n < 100n < n^2 < 2^n}$$

Asymptotic Notations - Defn

 Big-Oh Notation (Upper Bound)

$$\{\, T(n) = O(f(n)) \} \quad \Leftrightarrow \quad \left\{ \begin{array}{c} \exists\, c > 0 \text{ and } n_0 > 0 \text{ s.t.} \\ T(n) \le c\, f(n) \text{ for all } n \ge n_0 \end{array} \right\}$$

Eg: $10000n = O(n)$, $20n^2 = O(n^3)$, $20n^2 + 8n = O(n^2)$,

 Big-Omega Notation (Lower Bound)

$$\{\, T(n) = \Omega(g(n)) \} \quad \Leftrightarrow \quad \left\{ \begin{array}{c} \exists\, c > 0 \text{ and } n_0 > 0 \text{ s.t.} \\ T(n) \ge c\, g(n) \text{ for all } n \ge n_0 \end{array} \right\}$$

Eg: $10000n = \Omega(n)$, $0.01n^3 = \Omega(n^2)$, $0.5n^2 + 7n = \Omega(n^2)$,

 Big-Theta Notation (Exact Bound)

$$\{\, T(n) = \Theta(h(n)) \} \quad \Leftrightarrow \quad \left\{ \begin{array}{l} T(n) = O(h(n)) \ \& \\ T(n) = \Omega(h(n)) \end{array} \right\}$$

Eg: $10000n = \Theta(n)$, $2n^3 \ne \Theta(n^2)$, $3n^2 + 7n = \Theta(n^2)$,

 Little-Oh Notation (True upper bound)

$$\{\, T(n) = o(p(n)) \} \quad \Leftrightarrow \quad \left\{ \begin{array}{l} T(n) = O(p(n)) \ \& \\ T(n) \ne \Theta(h(n)) \end{array} \right\}$$

Eg: $10000\text{n} = o(n^2)$, $2n^2 = o(n^3)$, $3n^2 + 7n = o\,(n^3)$,

Remarks:

- These gives various $\boxed{\text{asymptotic}}$ bounds on $T(n)$, $\boxed{\text{to within a constant factor}}$

Asymptotic Notations

Summary

Rate	$g(n)$ is asymptotic..	$T(n)$ grows...	analog
$T(n) = O(g(n))$	**upper bound‡** of $T(n)$	**at most** as fast as $g(n)$	$a \le b$
$T(n) = \Omega(g(n))$	**lower bound** of $T(n)$	**at least** as fast as $g(n)$	$a \ge b$
$T(n) = \Theta(g(n))$	**exact bound** of $T(n)$	**exactly** as fast as $g(n)$	$a = b$
$T(n) = o(g(n))$	**true upper bound** of $T(n)$	**slower** than $g(n)$	$a < b$

‡ all bounds in table above are to within a constant factor

Simple Properties:

- **Reflexitivity Theorem: O, Ω, Θ are reflexive**

  $\boxed{f(n) = O(f(n)) f(n) = \Omega(f(n)) f(n) = \Theta(f(n))}$

- **Transitivity-Theorem: O, Ω, Θ, o are transitive**

  $\boxed{f(n) = O(g(n)) \;\&\; g(n) = O(h(n))} \Rightarrow \boxed{f(n) = O(h(n))}$

  $\boxed{f(n) = \Omega(g(n)) \;\&\; g(n) = \Omega(h(n))} \Rightarrow \boxed{f(n) = \Omega(h(n))}$

  $\boxed{f(n) = \Theta(g(n)) \;\&\; g(n) = \Theta(h(n))} \Rightarrow \boxed{f(n) = \Theta(h(n))}$

- **Symmetry-Theorem: Θ is symmetric, O, Ω are transpose symmetric**

  $\boxed{f(n) = \Theta(g(n))} \Leftrightarrow \boxed{g(n) = \Theta(f(n))}$

  $\boxed{f(n) = O(g(n))} \Leftrightarrow \boxed{g(n) = \Omega(f(n))}$

Three Theorems

Theorem: (Addition and multiplication)

> If $T_1(n) = O(f(n))$ & $T_2(n) = O(g(n))$, then
>
> (i) $T_1(n) + T_2(n) = O(\max\{f(n), g(n)\})$
>
> (ii) $T_1(n) \cdot T_2(n) = O(f(n) \cdot g(n))$

Eg:

$4n + 5n^2 = O(n^2)$, $\quad 5 + 6n + 7n^4 = O(n^4)$

$(2n-3)\cdot(4n^2 + 19n - 23) = O(n^3)$ $\quad (2n + 3)^3 = O(n^3)$

$O(1) \cdot O(n) = O(n)$ $\quad O(n) \cdot O(n^2) = (n^3)$

Theorem: (Polynomials)

> $P(n) = a_k n^k + a_{k-1} n^{k-1} + \ldots + a_1 n + a_0 = O(n^k)$

Proof:

$$P(n) \le |a_k| \cdot n^k + |a_{k-1}| \cdot n^{k-1} + \ldots + |a_1| \cdot n + |a_0|$$
$$\le (|a_k| + |a_{k-1}| + \ldots + |a_1| + |a_0|) \cdot n^k$$

for all $n \ge 1$. ■

Eg: $5 + 6n + 7n^4 = O(n^4)$ $\quad (2n + 3)^m = O(n^m)$

Theorem: (Logarithms) {logs grow very slowly}

> $\lg^k(n) = O(n)$, for any constant k

Eg: $20 \lg n = O(\sqrt{n})$ $\quad 4n + 5n \lg n = O(n^2)$

$\lg_c n = O(\lg n)$, for any constant c

Alternative Definitions

Use limits to Compare Growth Rates

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

Use L'Hospital's rule if necessary

Three Cases: The limit...

limit	Conclusion	Growth Rates	Example
is 0	$f(n) = o(g(n))$	$f(n)$ grows slower	$5n+3 = o(n^2)$
is $c \neq 0$	$f(n) = \Theta(g(n))$	same growth rate	$8n+7\lg n = \Theta(n)$
is ∞	$f(n) = \Omega(g(n))$	$f(n)$ grows faster	$n^2 + n \lg n = \Omega(n)$

Note:

$$\boxed{f(n) = o(g(n))} \quad \Rightarrow \quad \boxed{f(n) = O(g(n)) \quad \& \quad \neq \Omega(g(n))}$$

$$\boxed{f(n) = \Theta(g(n))} \quad \Rightarrow \quad \boxed{f(n) = O(g(n)) \quad \& \quad = \Omega(g(n))}$$

Examples:

eg: $\lim_{n\to\infty} \frac{5n+3}{n^2} = 0$, and so $5n+3 = O(n^2)$ & $\neq \Omega(n^2)$

eg: $\lim_{n\to\infty} \frac{9n+2\lg n}{n} = 9$, and so $9n+2\lg n = \Theta(n)$

eg: $\lim_{n\to\infty} \frac{3n^2+2n}{n\lg n} = \infty$, and so $3n^2 + 2n = \Omega(n\lg n)$ & $\neq O(n\lg n)$

Proving Asymptotic Notations

- **To prove $T(n) = O(f(n))$,**

  1. **Can use the definitions**

     Eg: $T(n) = 20n^2 + 10n - 18 \le 20n^2 + 10n$

     $\le 30n^2$ for all $n \ge 1$.

     Therefore, c=30, n_0=1, and $T(n) = O(n^2)$. (by definition)

  2. **Can apply Three Theorems**

     Eg: $T(n) = 111n^2 + 112n \lg n = \max\{O(n^2), O(n \lg n)\} = O(n^2)$

  3. **Can Alternative Definition**

     Eg: $\lim_{n\to\infty} \dfrac{2.5n^2 + 3n \lg n}{n^2} = 2.5, \Rightarrow 2.5n^2 + 3n\lg n = \Theta(n^2)$

- **Homework:**

  Order these functions by their growth-rates.

n	$\lg n$	$n(\lg^2 n)$	n^2	$200n$
$\sqrt{n}$	$n^{1.5}$	$2/n$	$n(\lg(n^2))$	n^3
2^n	$n!$	$n^2(\lg n)$	$n(\lg\lg n)$	$\sqrt{\lg n}$
$\sqrt{n(\lg n)}$	e^n			