5. ASYMPTOTIC NOTATIONS AND ITS PROPERTIES

Asymptotic notation is a notation, which is used to take meaningful statement about the efficiency of a program.

The efficiency analysis framework concentrates on the order of growth of an algorithm's basic operation count as the principal indicator of the algorithm's efficiency.

To compare and rank such orders of growth, computer scientists use three notations, they are:

- O - Big oh notation
- Ω - Big omega notation
- Θ - Big theta notation

Let*t(n)*and*g(n)*canbeanynonnegativefunctionsdefinedonthesetofnaturalnumbers. The algorithm's running time t*(n)* usually indicated by its basic operation count C*(n), and* g*(n),* some simple function to compare with the count.

Example 1:

$$n \in O(n^2), \qquad 100n + 5 \in O(n^2), \qquad \frac{1}{2}n(n-1) \in O(n^2).$$

$$n^3 \notin O(n^2), \qquad 0.00001n^3 \notin O(n^2), \qquad n^4 + n + 1 \notin O(n^2).$$

$$n^3 \in \Omega(n^2), \qquad \frac{1}{2}n(n-1) \in \Omega(n^2), \qquad \text{but } 100n + 5 \notin \Omega(n^2).$$

where *g(n)* = n^2.

(I) O - Big oh notation

A function t*(n)* is said to be in *O(g(n))*, denoted (n) ∈ (g(n)), if *t (n)* is bounded above by some constant multiple of *g(n)* for all large *n*, i.e., if there exist some positive constant *c* and some nonnegative integer n_0 such that

$$(n) \leq g(n) \text{ for } n \leq n_0.$$

Where *t(n)* and *g(n)* are nonnegative functions defined on the set of natural numbers.

O = Asymptotic upper bound = Useful for worst case analysis = Loose bound

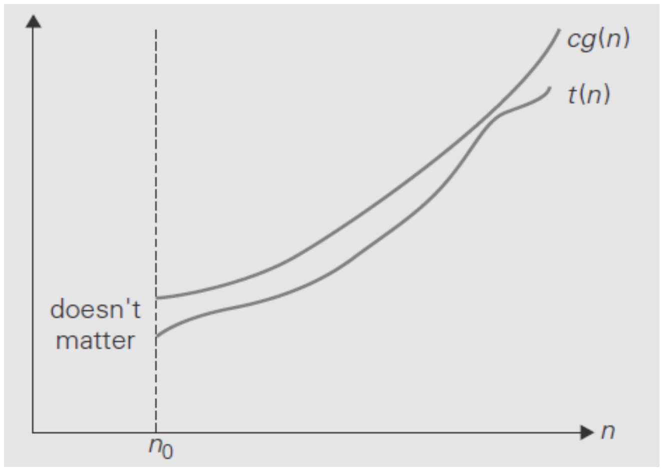

FIGURE 1.5 Big-oh notation: (n) ∈ (g(n)).

Example2: Prove the assertions $100n + 5 \in (n^2)$.

Proof: $100n + 5 \leq 100n + n$ (for all $n \geq 5$)

$= 101n$

$\leq 101n^2$ ($\ddot{Э}$ $n \leq n^2$)

Since, the definition gives us a lot of freedom in choosing specific values for constants **c** and **n_0**. We have c=101 and n_0=5

Example3: Prove the assertions $100n + 5 \in (n)$.

Proof: $100n + 5 \leq 100n + 5n$ (for all $n \geq 1$)

$=105n$

i.e., $100n + 5 \leq 105n$

i.e., $t(n) \leq cg(n)$

ӫ1ӨӨñ ∈ (n) with c=105 andn_0=1

(i) Ω - Big omega notation

A function *t(n)* is said to be in Ω*(g(n)),* denoted *t(n)* ∈ Ω*(g(n)),* if *t(n)* is bounded below by some positive constant multiple of *g(n*) for all large n, i.e., if there exist some positive constant c and some nonnegative integer n_0 such that

$$t(n) \geq cg(n) \text{ for all } n \geq n_0.$$

Where *t(n)* and *g(n*) are nonnegative functions defined on the set of natural numbers.

Ω = Asymptotic lower bound = Useful for best case analysis = Loose bound

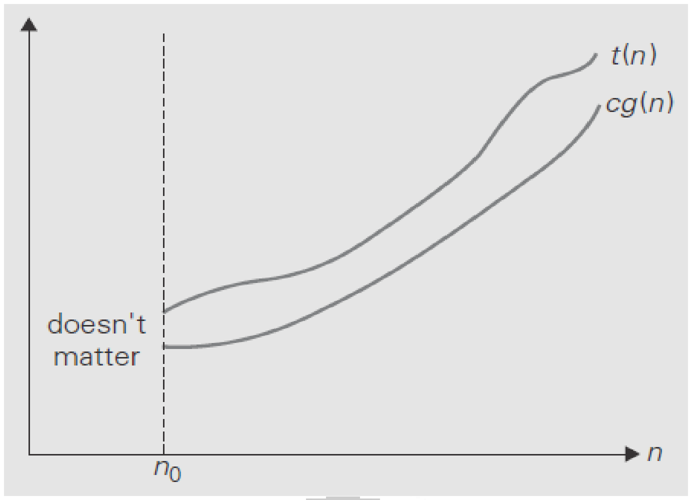

FIGURE 1.6 Big-omega notation: *t (n) ∈ Ω (g(n))*.

Example4: Prove the assertions $n^3+10n^2+4n+2 \in \Omega(n^2)$.

Proof: $n^3+10n^2+4n+2 \geq n^2$ (for all $n \geq 0$)
i.e., by definition $t(n) \geq cg(n)$, where $c=1$ and $n_0=0$

(ii) Θ - Big theta notation

A function t(n) is said to be in *Θ(g(n))*, denoted $t(n) \in \Theta(g(n))$, if t(n) is bounded both above and below by some positive constant multiples of g(n) for all large n, i.e., if there exist some positive constants c_1 and c_2 and some nonnegative integer n_0 such that

$$c_2 g(n) \leq t(n) \leq c_1 g(n) \text{ for all } n \geq n_0.$$

Where *t(n)* and *g(n)* are non-negative functions defined on the set of natural numbers.

Θ = Asymptotic tight bound = Useful for average case analysis

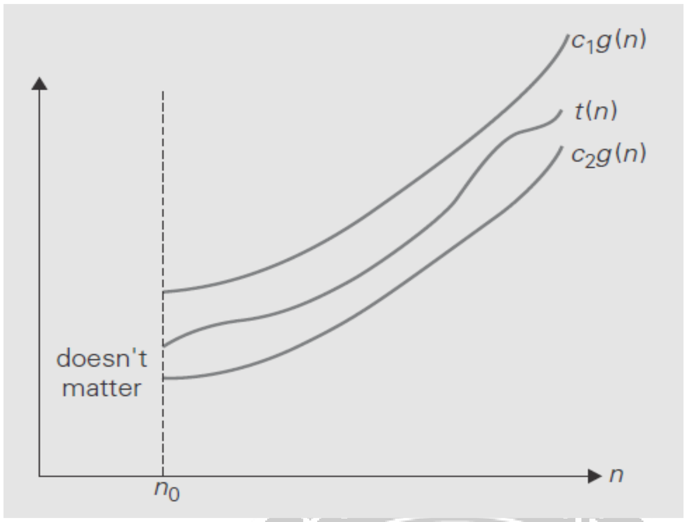

FIGURE 1.7 Big-theta notation: *t (n)* ∈ Θ*(g(n))*.

Example5: Prove the assertions $\frac{1}{2}n(n-1) \in \Theta(n^2)$.

Proof: First prove the right inequality (the upperbound):

$$\frac{1}{2}n(n-1) = \frac{1}{2}n^2 - \frac{1}{2}n \le \frac{1}{2}n^2 \text{ for all } n \ge 0.$$

Second, we prove the left inequality (the lower bound):

$$\frac{1}{2}n(n-1) = \frac{1}{2}n^2 - \frac{1}{2}n \le \frac{1}{2}n^2 - [\frac{1}{2}n][\frac{1}{2}n] \text{ for all } n \ge 2.$$

ӫ $\frac{1}{2}n(n-1) \le \frac{1}{4}n^2$

i.e., $\frac{1}{4}n^2 \le \frac{1}{2}n(n-1) \le \frac{1}{2}n^2$

Hence, $\frac{1}{2}n(n-1) \in \Theta(n^2)$, where $c_2 =$, $c_1 =$, and $n_0 = 2$

Note: asymptotic notation can be thought of as "relational operators" for functions similar to the corresponding relational operators for values.

$= \Rightarrow \Theta()$, $\le \Rightarrow O()$, $\ge \Rightarrow \Omega()$, $< \Rightarrow o()$, $> \Rightarrow \omega()$

Useful Property Involving the Asymptotic Notations

The following property, in particular, is useful in analyzing algorithms that comprise two consecutively executed parts.

THEOREM: If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$, then $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$. (The analogous assertions are true for the Ω and Θ notations as well.)

PROOF: The proof extends to orders of growth the following simple fact about four arbitrary real numbers a_1, b_1, a_2, b_2: if $a_1 \le b_1$ and $a_2 \le b_2$, then $a_1 + a_2 \le 2 \max\{b_1, b_2\}$.

Since $t_1(n) \in O(g_1(n))$, there exist some positive constant c_1 and some nonnegative integer n_1 suchthat

$$t_1(n) \le c_1 g_1(n) \text{ for all } n \ge n_1.$$

Similarly, since $t_2(n) \in O(g_2(n))$,

$$t_2(n) \le c_2 g_2(n) \text{ for all } n \ge n_2.$$

Let us denote $c_3 = \max\{c_1, c_2\}$ and consider $n \ge \max\{n_1, n_2\}$ so that we can use both inequalities. Adding them yields the following:

$$\begin{aligned} t_1(n)+t_2(n) &\le c_1g_1(n) + c_2g_2(n) \\ &\le c_3g_1(n) + c_3g_2(n) \\ &= c_3[g_1(n) + g_2(n)] \\ &\le c_3 2 \max\{g_1(n), g_2(n)\}. \end{aligned}$$

[illegible]

Hence, $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$, with the constants c and n_0 required by the definition O being $2c_3 = 2\max\{c_1, c_2\}$ and $\max\{n_1, n_2\}$, respectively.

The property implies that the algorithm's overall efficiency will be determined by the part with a higher order of growth, i.e., its least efficient part.

ëtɨ(n) $O(g_1(n))$ and $t_2(n) \in O(g_2(n))$, then $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$.

Basic rules of sum manipulation

$$\sum_{i=l}^{u} ca_i = c\sum_{i=l}^{u} a_i, \qquad \textbf{(R1)}$$

$$\sum_{i=l}^{u}(a_i \pm b_i) = \sum_{i=l}^{u} a_i \pm \sum_{i=l}^{u} b_i, \qquad \textbf{(R2)}$$

Summation formulas

$$\sum_{i=l}^{u} 1 = u - l + 1 \quad \text{where } l \le u \text{ are some lower and upper integer limits,} \qquad \textbf{(S1)}$$

$$\sum_{i=0}^{n} i = \sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2} \approx \frac{1}{2}n^2 \in \Theta(n^2). \qquad \textbf{(S2)}$$