Digital Logic Design: a rigorous approach ©

Chapter 7: Asymptotics

part 1: big-O, big-Ω

Guy Even Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

April 6, 2020

Book Homepage:
http://www.eng.tau.ac.il/~guy/Even-Medina

The functions we study

We study functions that describe the number of gates in a circuit, the delay of a circuit (length of longest path), the running time of an algorithm, number of bits in a data structure, etc. In all these cases it is natural to assume that

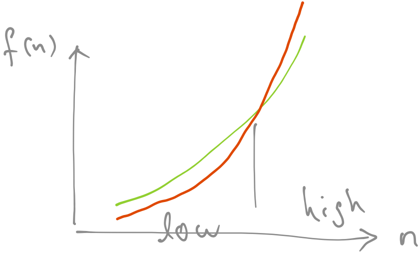
$$\forall n \in \mathbb{N} : \quad f(n) \geq 1.$$

Assumption

The functions we study are functions $f : \mathbb{N} \to \mathbb{R}^{\geq 1}$.

Order of Growth Rates

- We want to compare functions asymptotically (how fast does $f(n)$ grow as $n \to \infty$).
- Ignore constants (not because they are not important, but because we want to focus on "high order" terms).



big-O, big-Omega, big-Theta

Definition

Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq 1}$ denote two functions.

1. We say that $f(n) = O(g(n))$, if there exist constants $c \in \mathbb{R}^+$ and $N \in \mathbb{N}$, such that,
$$\forall n > N : f(n) \leq c \cdot g(n) \, .$$
2. We say that $f(n) = \Omega(g(n))$, if there exist constants $c \in \mathbb{R}^+$ and $N \in \mathbb{N}$, such that,
$$\forall n > N : f(n) \geq c \cdot g(n) \, .$$
3. We say that $f(n) = \Theta(g(n))$, if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

What does "$=$" actually mean here?!

Warning on Notation

What does the equality sign in $f = O(g)$ mean?

- $O(g)$ in fact refers to a set of functions:

$$O(g) \triangleq \{h : \mathbb{N} \to \mathbb{R}^{\geq 1} \mid \exists c \exists N \forall n > N : h(n) \leq c \cdot g(n)\}$$

- Would have been much better to write $f \in O(g)$ instead of $f = O(g)$.
- But we want to abuse notation and write expressions like:

$$\underbrace{(2n^3 + 3n) \cdot 5 \log(n^2)}_{f} = O(\overbrace{n^2 \cdot \log n^2}^{g})$$
$$= O(\underbrace{n^2 \cdot \log n}_{h}) \,.$$

$f \in O(g)$

$g \in O(h)$

Justification: transitivity.

$\cancel{O(g) = f}$ ← does <u>not</u> make sense!

big-O, big-Omega, big-Theta

Definition

Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq 1}$ denote two functions.

1. We say that $f(n) = O(g(n))$, if there exist constants $c \in \mathbb{R}^+$ and $N \in \mathbb{N}$, such that,
$$\forall n > N : f(n) \leq c \cdot g(n) \,.$$
2. We say that $f(n) = \Omega(g(n))$, if there exist constants $c \in \mathbb{R}^+$ and $N \in \mathbb{N}$, such that,
$$\forall n > N : f(n) \geq c \cdot g(n) \,.$$
3. We say that $f(n) = \Theta(g(n))$, if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

If $f(n) = O(g(n))$, then "asymptotically, $f(n)$ does not grow faster than $g(n)$".
If $f(n) = \Omega(g(n))$, then "asymptotically, $f(n)$ grows as least as fast as $g(n)$".
Finally, if $f(n) = \Theta(g(n))$, then "asymptotically, $f(n)$ grows as fast as $g(n)$".

Remark

When proving that $f(n) = O(g(n))$, it is not necessary to find the "smallest" constant c.

Example

Suppose you want to prove that $n + \sqrt{n} = O(n^{1.1})$. Then, it suffices to prove that for $n > 2^{100}$:

$$n + \sqrt{n} \leq 10^6 \cdot n^{1.1} .$$

Any other constants you can prove the statement for are just as good!

Examples

$$n = O(n^2)$$

$\exists c > 0 \quad \exists N \quad \forall n > N:$

$$n \leq c \cdot n^2$$

$c = 100$

$N = 556$

Examples

$$\log(n) = O(n)$$

$\exists c > 0 \quad \exists N \quad \forall n > N$

$\lg n \leq c \cdot n$

$c = 100$

$N = 55$

Examples

$$\underbrace{10}_{c}n = O(n), \underbrace{10^2}_{c}n = O(n), \ldots, \underbrace{10^{100}}_{c}n = O(n)$$

$$c \cdot n \leq c \cdot n$$

Examples

$$n \cdot \underbrace{\log\log\log n}_{\xrightarrow[n\to\infty]{} \infty} \neq O(n)$$

$$\cancel{n} \cdot \underbrace{\lg\lg\lg n} \overset{?}{\leq} \underbrace{c} \cdot \cancel{n}$$

Constant Function

Claim

$f(n) = O(1)$ iff there exists a constant c such that $f(n) \leq c$, for every n.

proof:

$(\Leftarrow)$ by def.

$(\Rightarrow)$ $\exists c' \; \exists N \; \forall n > N: \quad f(n) \leq c'$

$\Rightarrow \quad \forall n: \quad f(n) \leq \max\{c', f(1), \dots, f(N)\}$

$\boxtimes$

Claim if $f_i = O(g)$ for $i \in \{1,2\}$

then $f_1 + f_2 = O(g)$

Proof: $\exists c_i \, \exists N_i \; \forall n > N_i :$

$$f_i(n) \leq c_i \cdot g(n)$$

$$\Rightarrow \quad f_1(n) + f_2(n) \leq (c_1 + c_2) \cdot g(n)$$

if $n \geq N_1 + N_2$.

conseq: $n^2 + n + 1 = O(n^2)$ $\boxtimes$

Asymptotic Algebra (big-O)

Abbreviate: $f_i = O(h)$ means $f_i(n) = O(h(n))$.

k is a constant

Claim

Suppose that $f_i = O(g_i)$ for $i \in \{1, \ldots, k\}$, then:

$$\max\{f_i\}_i = O(\max\{g_i\}_i)$$
$$\sum_i f_i = O(\sum g_i)$$
$$\prod_i f_i = O(\prod_i g_i) \, .$$

Consequences:

$$2n = O(n) \qquad \text{mult. by constant}$$
$$50n^2 + 2n + 1 = O(n^2) \qquad \text{polynomial with positive leading coefficient}$$

$O(n^2 + n + 1)$

claim: $f_i = O(g_i) \Rightarrow \max_i f_i = O(\max_i g_i)$

proof: $\forall i \; \exists c_i \; \exists N_i \; \forall n \geq N_i$

$$f_i(n) \leq c_i \cdot g_i(n)$$

define $c \triangleq \max\{c_1, \ldots, c_k\}$

$N \triangleq \max\{N_1, \ldots, N_k\}$

$\forall n \geq N$: $\max_i f_i(n) \leq \max_i c \cdot g_i(n)$

$$\leq c \cdot \max_i g_i(n)$$

$$= O(\max_i g_i(n))$$

$\boxtimes$

claim: $f_i = O(g_i) \Rightarrow \sum_{i=1}^{k} f_i = O\left(\sum_{i=1}^{k} g_i\right)$

proof: use same notation:

$$\forall n \geq N: \quad \sum_{i=1}^{k} f_i(n) \leq \sum_{i=1}^{k} c_i \cdot g_i(n)$$

$$\leq c \cdot \sum_{i=1}^{k} g_i(n)$$

$$= O\left(\sum_{i=1}^{k} g_i(n)\right)$$

$\boxtimes$

Claim $f_i = O(g_i) \Rightarrow \prod_i f_i = O(\prod g_i)$

proof: using same notation except

$$\tilde{c} \triangleq c_1 \cdot c_2 \cdot \ldots \cdot c_k$$

$\forall n \geq N$: $\prod_i f_i(n) \leq \prod_i c_i \cdot g_i(n)$

$$\leq \tilde{c} \cdot \prod_i g_i(n)$$

$$= O\left(\prod_i g_i(n)\right)$$

$\boxtimes$

Asymptotic Algebra (big-Omega)

Claim

Suppose that $f_i = \Omega(g_i)$ for $i \in \{1, \ldots, k\}$, then:

$$\min\{f_i\}_i = \Omega(\min\{g_i\}_i)$$
$$\sum_i f_i = \Omega(\sum g_i)$$
$$\prod_i f_i = \Omega(\prod_i g_i) \, .$$

Consequences:

$$2n = \Omega(n) \qquad \text{mult. by constant}$$
$$10^{-6} \cdot n^2 + 2n + 1 = \Omega(n^2) \qquad \text{polynomial with positive leading coefficient}$$

Asymptotics of Arithmetic Series

Claim

If $\{a_n\}_n$ is an arithmetic sequence with $a_0 \geq 0$ and $d > 0$, then $\sum_{i \leq n} a_i = \Theta(n \cdot a_n)$.

Consequence:

$$\sum_{i=1}^{n} i = \Theta(n^2) \ .$$

proof:

$$S_n = a_0(n+1) + d \cdot \frac{n(n+1)}{2}$$

(algebra) $= \underbrace{a_0}_{\Theta(1)} + \underbrace{\left(a_0 + \frac{d}{2}\right)n}_{\Theta(n)} + \underbrace{\frac{d}{2} n^2}_{\Theta(n^2)}$

(sum of Θ) $= \Theta(1 + n + n^2) = \Theta(n^2)$

(claim)

Claim

If $\{b_n\}_n$ is a geometric sequence with $b_0 \geq 1$ and $q > 1$, then $\sum_{i \leq n} b_i = \Theta(b_n)$.

Consequence: If $q > 1$, then $\sum_{i=1}^{n} q^i = \Theta(q^n)$.

proof:

$$S_n = b_0 \cdot \frac{q^{n+1} - 1}{q - 1}$$

$$= \frac{b_0 \cdot q}{q-1} \cdot q^n - \frac{b_0}{q-1}$$

$$= O(q^n)$$

for large enough n: $S_n \geq \frac{1}{2} \cdot \frac{b_0 q}{q-1} \cdot q^n = \Omega(q^n)$ $\boxtimes$

Asymptotics as an Equivalence Relation

Claim

$$
\begin{aligned}
f &= O(f) && \textit{reflexivity} \\
f = O(g) &\not\Longrightarrow g = O(f) && \textit{no symmetry} \\
(f = O(g)) \wedge (g = O(h)) &\Longrightarrow f = O(h) && \textit{transitivity}
\end{aligned}
$$

What about Ω?

claim: $f = O(g) \not\Rightarrow g = O(f)$

proof: suffices to show a counter example.

$f(n) = 1$

$g(n) = n$

<u>claim</u> $f = O(g)$ & $g = O(h)$

$$\Longrightarrow f = O(h)$$

<u>proof:</u> $\exists c_1 \, \exists N_1 \, \forall n \geq N_1 : f(n) \leq c_1 \cdot g(n)$

$\exists c_2 \, \exists N_2 \, \forall n \geq N_2 : g(n) \leq c_2 \cdot h(n)$

$$\Rightarrow \forall n \geq \underbrace{N_1 + N_2}_{N} : \; f(n) \leq c_1 g(n)$$

$$\leq \underbrace{c_1 \cdot c_2}_{c} \cdot h(n)$$

$\boxtimes$

Claim

Assume $f(n), g(n) \geq 1$, *for every* n. *Then,*

$$f(n) = O(g(n)) \quad \Leftrightarrow \quad g(n) = \Omega(f(n)).$$

proof: $(\Rightarrow)\ \exists c\ \exists N\ \forall n > N \quad f(n) \leq c \cdot g(n)$

$\Rightarrow \exists c\ \exists N\ \forall n: \quad g(n) \geq \frac{1}{c} f(n)$

$\Rightarrow \quad g = \Omega(f).$

$(\Leftarrow)$ exercise.