# Inf 2B: Asymptotic notation

## Lecture 2 of ADS thread

Kyriakos Kalorkoti

School of Informatics
University of Edinburgh

# The Big-$O$ Notation

**Definition**
Let $f, g : \mathbb{N} \to \mathbb{R}$ be functions. We say that $f$ *is* $O(g)$ if there is some $n_0 \in \mathbb{N}$ and some $c > 0$ from $\mathbb{R}$ such that for all $n \geq n_0$ we have

$$f(n) \leq c\, g(n).$$

In other words:

$$O(g) = \big\{ f : \mathbb{N} \to \mathbb{R} \mid \text{there is an } n_0 \in \mathbb{N} \text{ and } c > 0 \text{ in } \mathbb{R} \text{ such that for all } n \geq n_0,\ f(n) \leq cg(n). \big\}$$

Then "$f$ is $O(g)$" means $f \in O(g)$.
Informally, we say "*for sufficiently large* $n$" instead of "there is some $n_0 \in \mathbb{N}$ such that for all $n \geq n_0 \ldots$".

# Intention

"$f$ is $O(g)$" tells us that the growth rate of $f$ is *no worse* than that of $g$. Could be better.

- $c$ allows us to adjust for constants: $n^2$ obviously has same growth rate as $3n^2$, $20n^2$, $100n^2 \ldots$
  - consider $n \to an$ then

$$\begin{aligned} n^2 &\to a^2 \cdot n^2 \\ 3n^2 &\to a^2 \cdot 3n^2 \\ 20n^2 &\to a^2 \cdot 20n^2 \\ &\vdots \end{aligned}$$

- $n_0$ allows a settling in period of atypical behaviour.
- $O$ allows us to concentrate on the big picture rather than details (many being implementation dependent ).

# Notational Convention

Write

$$f = O(g),$$

instead of

$$f \in O(g).$$

- Makes it convenient to have chains reasoning with inequalities etc.
- Notation here is from *left to right*. $f = O(g)$ does *not* mean that $O(g) = f$!
- $f = f_1 + O(g) = f_2 + O(g)$ does *not* imply that $f_1 = f_2$.
- Seems strange but easy to get used to it and *very* useful.

# Examples of $O$

1. $3n^3 = O(n^3)$.
   Need $c$ and $n_0$ so that $3n^3 \leq cn^3$ for all $n \geq n_0$.
   Take $c = 3$, $n_0 = 0$.

2. $3n^3 + 8 = O(n^3)$.
   For a constant $c > 0$ we have

$$3n^3 + 8 \leq cn^3 \iff 3 + \frac{8}{n^3} \leq c \quad \text{provided } n > 0.$$

   As $n$ increases $8/n^3$ decreases. Thus

$$3 + \frac{8}{n^3} \leq 11 \quad \text{for all } n > 0.$$

   So we take $c = 11$, $n_0 = 1$.
   We *can* also take $c = 4$, $n_0 = 2$ or $c = 3 + 8/27$, $n_0 = 3$ etc.

# More Examples of $O$

3. $\lg(n) = O(n)$
   Intuitively: $\lg(n) < n$ for all $n \geq 1$.
   Need a proof. Well

$$\lg(n) < n \iff n < 2^n, \quad \text{for all } n > 0.$$

   Use induction on $n$ to prove rhs.

   - Base case $n = 1$ is clearly true.
   - For induction step assume claim holds for $n$. Then

$$2^{n+1} = 2 \cdot 2^n > 2n, \quad \text{by induction hypothesis.}$$

     To complete the proof just need to show that $2n \geq n + 1$.
     Now

$$2n \geq n + 1 \iff n \geq 1,$$

     and we have finished.
     So we take $c = 1$ and $n_0 = 1$.

# More Examples of $O$

4. $8n^2 + 10n\lg(n) + 100n + 10000 = O(n^2)$.

   We have

$$\begin{aligned} 8n^2 + {} & 10n\lg(n) + 100n + 10000 \\ & < 8n^2 + 10n \cdot n + 100n + 10000, \quad \text{for all } n > 0 \\ & \le 8n^2 + 10n^2 + 100n^2 + 10000n^2 \\ & = (8 + 10 + 100 + 10000)n^2 \\ & = 10118n^2. \end{aligned}$$

   Thus we can take $n_0 = 1$ and $c = 1118$.
   - Value for $c$ seems rather large.
   - Any $c > 8$ will do, the closer $c$ is to 8 the larger $n_0$ has to be.
   - For big-$O$ notation, no point at all in expending more effort just to reduce some constant.

5. $2^{100} = O(1)$.

   Take $n_0 = 0$ and $c = 2^{100}$.

# “Laws” of Big-O

**Theorem:** Let $f_1, f_2, g_1, g_2 : \mathbb{N} \to \mathbb{R}$ be functions. Then:

1. For any constant $a > 0$ in $\mathbb{R}$: $f_1 = O(g_1) \implies af_1 \in O(g_1)$.
2. $f_1 = O(g_1)$ and $f_2 = O(g_2) \implies f_1 + f_2 = O(g_1 + g_2)$.
3. $f_1 = O(g_1)$ and $f_2 = O(g_2) \implies f_1 f_2 = O(g_1 g_2)$.
4. $f_1 = O(g_1)$ and $g_1 = O(g_2) \implies f_1 = O(g_2)$.
5. For any $d \in \mathbb{N}$: $f_1$ polynomial of degree $d \implies f_1 = O(n^d)$.
6. For any *constants* $a > 0$ and $b > 1$ in $\mathbb{R}$: $n^a = O(b^n)$.
7. For any *constant* $a > 0$ in $\mathbb{R}$: $\lg(n^a) = O(\lg(n))$.
8. For any *constants* $a > 0$ and $b > 0$ in $\mathbb{R}$: $\lg^a(n) = O(n^b)$.

# Example (using Laws for $O$)

$$871n^3 + 13n^2 \lg^5(n) + 18n + 566 = O(n^3).$$

$$\begin{aligned}
&871n^3 + 13n^2 \lg^5(n) + 18n + 566 \\
&\qquad = 871n^3 + 13n^2 O(n) + 18n + 566 \qquad \text{by (8)} \\
&\qquad = 871n^3 + O(n^3) + 18n + 566 \qquad \text{by (3)} \\
&\qquad = 871n^3 + 18n + 566 + O(n^3) \\
&\qquad = O(n^3) + O(n^3) \qquad \text{by (5)} \\
&\qquad = O(n^3) \qquad \text{by (2) \& (1)}
\end{aligned}$$

# Big-Ω and Big-Θ

## Definition

Let $f, g : \mathbb{N} \to \mathbb{R}$ be functions.

1. We say that $f$ *is* $\Omega(g)$ if there is an $n_0 \in \mathbb{N}$ and $c > 0$ in $\mathbb{R}$ such that for all $n \geq n_0$ we have
$$f(n) \geq c\,g(n).$$
2. We say that $f$ *is* $\Theta(g)$, or $f$ *has the same asymptotic growth rate as* $g$, if $f$ is $O(g)$ and $\Omega(g)$.

**Observation:**
$$f = \Omega(g) \iff g = O(f).$$

**Prove** this.

# Examples of $f = \Omega(g)$

1. Let $f(n) = 3n^3$ and $g(n) = n^3$.
   (combining this with Ex 1. for $O$ gives $3n^3 = \Theta(n^3)$)

   Let $n_0 = 0$ and $c = 1$. Then for all $n \geq n_0$,
   $f(n) = 3n^3 \geq cg(n) = g(n)$.
2. Let $f(n) = \lg(n)$ and $g(n) = \lg(n^2)$.

   Well
   $$\lg(n^2) = 2\log(n).$$
   So take $c = 1/2$ and $n_0 = 1$. Then for every $n \geq n_0$ we have,
   $$f(n) \;=\; \lg(n) \;=\; \frac{1}{2} 2\lg(n) \;=\; \frac{1}{2}\lg(n^2) \;=\; \frac{1}{2} g(n).$$

# Quick quiz: True or False ?

$$\sqrt{n^3} = O(n^2)?$$

**True.** $\sqrt{n^3} = n^{3/2} \leq n^2$.

$$2^{\lfloor \lg n \rfloor} = O(n)?$$

**True.** $2^{\lfloor \lg n \rfloor} \leq 2^{\lg n} = n$.

$$2^{\lfloor \lg n \rfloor} = \Omega(n)?$$

**True:** Let $n \geq 2$. Then $\lfloor \lg n \rfloor \geq (\lg n) - 1 \geq 0$. Hence $2^{\lfloor \lg n \rfloor} \geq 2^{(\lg n)-1} = n/2$. So take $n_0 = 2, c = 1/2$.

$$n \lg n = \Theta(n^2)?$$

**False:** We do have $n \lg n = O(n^2)$ but $n \lg n$ is **not** $\Omega(n^2)$.

# Further Reading

- Lecture notes 2 (handed out).
- If you have Goodrich & Tamassia [GT]:
  All of the chapter on “Analysis Tools” (especially the “Seven functions” and “Analysis of Algorithms” sections).

  NB: the title of the book is as given in slides of lecture 1, not as in note 1.
- If you have [CLRS]:
  Read chapter 3 on “Growth of Functions”.
- `Wikipedia` has a page about asymptotic notation:
  `en.wikipedia.org/wiki/Asymptotic_notation`