COMP2054-ADE:

ADE Lec04a: The Big-Oh family

Lecturer: Andrew Parkes

andrew.parkes ‘at’ Nottingham.ac.uk

http://www.cs.nott.ac.uk/~pszajp/

Relatives of Big-Oh

A close family:

- **big-Oh** **'O'**
- **big-Omega** **'Ω'**
- **big-Theta** **'Θ'**

- **little-oh** **'o'**
  - note this is not in the main text-book but is required for the module
- **little-omega** **'ω'**
  - Not required, as not used a lot, though mention it for completeness

Big-Omega: Definition

Definition: Given functions $f(n)$ and $g(n)$, we say that

$$f(n) \text{ is } \Omega(\, g(n)\,)$$

if there are (strictly) positive constants c and n_0 such that

$$f(n) \geq c\, g(n) \qquad \text{for all } n \geq n_0$$

Big-Omega: Definition

Definition: Given functions $\boldsymbol{f}(\boldsymbol{n})$ and $\boldsymbol{g}(\boldsymbol{n})$, we say that

$$\boldsymbol{f}(\boldsymbol{n}) \text{ is } \boldsymbol{\Omega}(\ \boldsymbol{g}(\boldsymbol{n})\ )$$

if there are (strictly) positive constants $\boldsymbol{c}$ and $\boldsymbol{n_0}$ such that

$$\boldsymbol{f}(\boldsymbol{n}) \geq \boldsymbol{c}\ \boldsymbol{g}(\boldsymbol{n}) \qquad \text{for all } \boldsymbol{n} \geq \boldsymbol{n_0}$$

- Spot the difference from big-Oh?
- “greater than” rather than “less than”
- Note that need c > 0, and we not allowed c=0
- Note that c must be **constant (cannot depend on n)**

Exercise (online):

- Show n is $\Omega(n)$

Need c > 0, n0 such that

n >= c n forall n >= n0

Try c = 1

n >= n forall n >= n0

Pick n0=1.

DONE.

Exercise (online):

- Show n^2 is $\Omega(n)$

Need $c > 0$, n0 such that

$n^2 >= c\ n$ forall n >= n0

Try c = 1

n >= 1 forall n >= n0

Pick n0=1

Also works if try c=2

n >= 2 forall n >= n0

n0 = 2

DONE.

Exercise (online):

- Show $n^3 - n$ is $\Omega(n^3)$

Need $c > 0$, $n0$ such that

$$n^3 - n >= c\, n^3 \quad \text{forall } n >= n0$$

Try $c = 1$

$$n^3 - n >= n^3 \quad \text{forall } n >= n0$$

$$- n >= 0 \text{ forall } n >= n0$$

FAILS

Try $c = 1 / 2$

$$2\, n^3 - 2\, n >= n^3 \quad \text{forall } n >= n0$$

$$n^2 >= 2 \text{ forall } n >= n0$$

$n0 = 2$ DONE.

Exercise (online):

- Is it true that: 1 is $\Omega(n)$?

Need $c > 0$, $n0$ such that

$$1 >= c\, n \text{ forall } n >= n0$$

Note: No need for c to be integer

Try c = … FAILS

Instead use $c > 0$ to get

$$(1/c) >= n \text{ forall } n >= n0$$

FAILS as eventually $n > (1/c)$

FAILS.

1 is NOT $\Omega(n)$

$c = 1/n$ is NOT allowed – need c constant

Big-Omega Examples

- We have
  - n is $\Omega(1)$
  - n is $\Omega(n)$
  - n is not $\Omega(n^2)$
- This is "the opposite of Big-Oh"
- **f(n) is $\Omega(g(n))$ says that "f(n) grows at least as fast as g(n) at large n"**
- Compared to
  - f(n) is O(g(n)) says that "f(n) grows no faster than g(n) at large n"

Big-Omega properties

- Similarly to big-Oh, Big-Omega is
  - Reflexive
  - NOT symmetric
  - Transitive

- Exercise (offline): Prove these.

Linking big-Oh and Big-Omega

- Suppose that we are give "f is O(g)"
- What can we say about big-Omega?

Linking big-Oh and Big-Omega

- Suppose that we are give "f is O(g)"
  - We know there exist c n_0 such that

    $f(n) \leq c\, g(n) \quad \forall\, n \geq n_0$
  - We can assume $c > 0$ (Why?). Hence
    - $g(n) \geq \left(\frac{1}{c}\right) f(n) \quad \forall\, n \geq n_0$
  - Hence, $g \text{ is } \Omega(f)$.
- That is: $f \in O(g) \rightarrow g \in \Omega(\mathrm{f})$
- Similarly: $f \in \Omega(g) \rightarrow g \in \mathrm{O}(\mathrm{f})$.
- Note: similar to: x <= y → y >= x

Usage of big-Omega

- Once familiar with big-Oh then big-Omega is very similar but "upside down". Same rules apply:
  - Multiplication rule still applies
  - Can still drop smaller terms
- E.g. $n^3 - n$ is $\Omega(n^3)$
- 'Ω ' expresses "grows as least as fast as"
- 'O' expresses "grows at most as fast as"

‘Paradox’ of big-Omega

  - “Can still drop smaller terms”
- E.g. $n^3 - n$ is $\Omega(n^3)$

- Note that this might seem counter-intuitive
  - thinking “big Omega does `at least’ and so need the smallest term” is incorrect!
  - instead think of the large n behaviour being dominated by the n^3, and that this grows at least as fast as n^3.

Omega Usage

- A standard usage might be to capture a limitation on the best one can hope for,
  e.g. "*The best case for algorithm X is $\Omega(n^3)$ and so it will not scale well*"
- However, if the worst case behaviour of an algorithm is a 'bizarre' or 'partially unknown' function of n, then one might say,
  e.g. "*The worst case for algorithm X is not precisely known but we know it is $\Omega(n^3)$ and $O(n^4)$*"
- Gives a lot more flexibility than doing ratios.

Big-Theta: Definition

Definition: Given functions $f(n)$ and $g(n)$, we say that

$f(n)$ is $\Theta(\ g(n)\ )$

if there are positive constants c' , c'' and n_0 such that

$$f(n) \leq c' g(n)$$

$$f(n) \geq c'' g(n)$$

for all $n \geq n_0$

- *What does this say about the growth rate of f(n) ?*
- *How does it connect to O and Omega?*

Big-Theta

Definition: Given functions $f(n)$ and $g(n)$, we say that

$f(n)$ is $\Theta(\ g(n)\ )$

if there are positive constants c' , c'' and n_0 such that

$$f(n) \le c'\, g(n)$$

$$f(n) \ge c''\, g(n)$$

for all $n \ge n_0$

- *f(n)* is *Θ(g(n))* if and only if
  f(n) is *O(g(n))* and also *f(n)* is *Ω(g(n))*
- Example: *2n+1* is *Θ(n)*
- *Θ expresses "grows 'exactly' as fast as"*

EXERCISE

- Consider the function:

$$f(n) = \begin{cases} n & \text{if } n \text{ is even} \\ 2n & \text{if } n \text{ is odd} \end{cases}$$

What is its big-Omega behaviour?

Clearly is $\Omega(n)$.

Just take c=1 n0=1 then

$f(n) >= 1\ n$ forall $n >= 1$

EXERCISE

- Consider the function:

$$f(n) = \begin{cases} n & \text{if } n \text{ is even} \\ 2n & \text{if } n \text{ is odd} \end{cases}$$

What is its big-Theta behaviour?

Clearly is $\Theta(n)$. As is both $O(n)$ and $\Omega(n)$.

But note that there is not a single value for ratio $f(n)/n$.

$\Theta(n)$ expresses the growth rate is linear but does not tightly sandwich the specific values.

- This is very useful in CS !
- Sometimes people can really mean Big-Theta when they say Big-Oh

Big-Theta is reflexive and transitive

- Trivial as both O and Ω are.

Is Big-Theta symmetric?

- If $f \in \Theta(g)$ then it is (automatically and always) true that $g \in \Theta(f)$
- Exercise (offline): (Follows quickly from previous results).

Θ is an equivalence relation

- Any relation that is
  - Reflexive & Symmetric & Transitive
- is an "equivalence relation"
  - Roughly speaking: it behaves like a "equality":
  - $\Theta(g(n))$ is "the equivalence class of all functions whose large n behaviour is bounded above and below by constants times g(n)"
- It is reasonable to write "$f = \Theta(g)$"
  - (pedantically, arguably, could be "{f} = $\Theta(g)$" but dropping "{}" on singleton sets is a common abuse of notation. But also it is somewhat of an abuse as it is not a set inequality, rather it is an "equivalence relation", i.e. closer to "{f} == $\Theta(g)$")
  - "Reasonable" is not same as "recommended" ☺

BREAK

- Next lecture “little-oh”