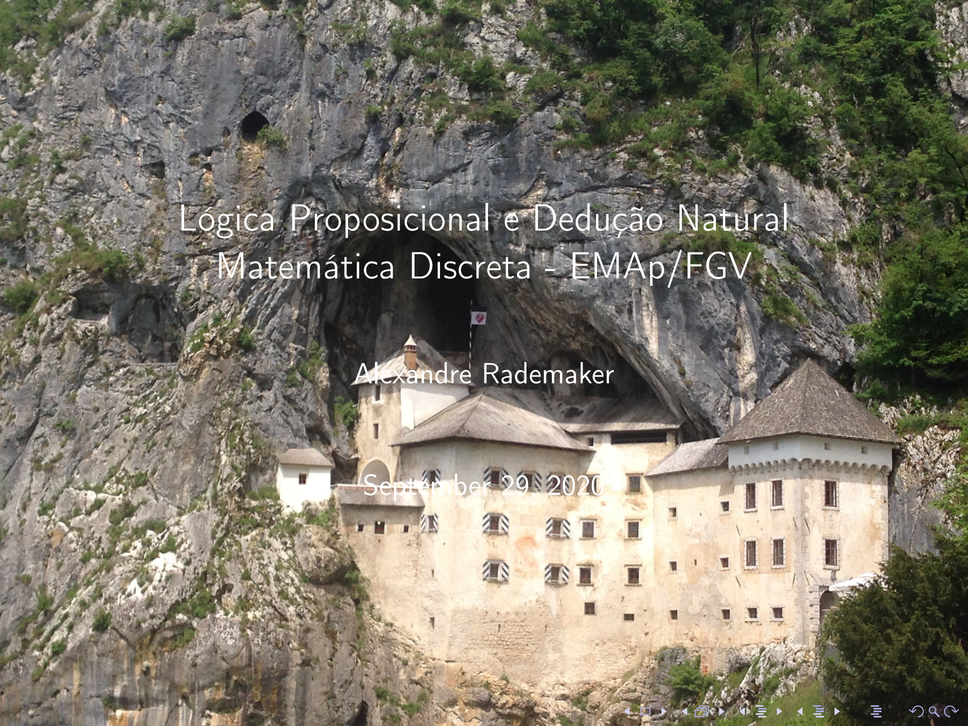

Lógica Proposicional e Dedução Natural
Matemática Discreta - EMAp/FGV

Alexandre Rademaker

September 29, 2020

Roadmap

Asymptotic notation

Asymptotic notations

Θ, O, and Ω ('big theta', 'big omicron', and 'big omega').

$f = \Theta(g)$ f is of order of g.

$f = O(g)$ f is of order at most g.

$f = \Omega(g)$ f is of otder at least g.

Big Theta

We say that f is of order g, and write $f = \Theta(g)$, if there are positive constants C and D and a number n_0 such that, for all $n > n_0$,

$$Cg(n) \leq f(n) \leq Dg(n)$$

Examples

$$n(n+1)/2 = \Theta(n^2)$$
$$n^3 + n^2 + nlogn = \Theta(n^3)$$
$$n(1 + 1/2 + 1/3 + \ldots) = \Theta(nlogn)$$

We write $f(n) = \Theta(g(n))$ or $f = \Theta(g)$ or $f \in \Theta(g)$.

Example: $head(list)$, it is $\Theta(1)$.

Big Omicron

We say that f is of order at most g, and write $f = O(g)$, if there are positive constants C and a number n_0 such that, for all $n > n_0$,

$$f(n) \leq Cg(n)$$

In particular, $O(1)$ stands for an anonymous function whose values are bounded above by some positive constant.

The running time of *takeWhile* on a list of length n is $O(n)$ steps, assuming the test takes constant time.

In the worst case the running time is $\Theta(n)$ steps but in the best case, when the first element does not pass the test, the running time is $\Theta(1)$ steps.

Big Omega

A running time of $O(n^2)$ does not imply that the running time is not also $O(n)$.

We say that f is of order at least g, and write $f = \Omega(g)$, if there is a positive constant C and a natural number n_0 such that

$$f(n) \geq Cg(n)$$

It follows that $f = \Theta(g)$ if and only if $f = O(g)$ and $f = \Omega(g)$.