Definition

- equation (inequailty) describing function in terms of its value for smaller inputs
  - Example: factorial function
    $f(n) = n \cdot f(n-1)$ for $n > 1$, $f(1) = 1$

    Example: fibonacci numbers
    $f(n) = f(n-1) + f(n-2)$ for $n > 2$, $f(1) = f(2) = 1$
- relevant for [[Laufzeitanalyse]]
  - runtime and consumption analysis

Asymptotic Bounds

- [[O-Notation]] and [[Θ-Notation]]
- Limit theorem
  - $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \begin{cases} 0 & f(n) = \mathcal{O}(g(n)) \\ 0 < c < \infty & f(n) = \Theta(g(n)), \\ & \qquad \mathcal{O}(g(n)), \Omega(g(n)) \\ \infty & f(n) = \Omega(g(n)) \end{cases}$$

    $\liminf_{n\to\infty} \frac{f(n)}{g(n)} > 0$ $\qquad f(n) = \Omega(g(n))$

    $\limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$(and ≥ 0) $\qquad f(n) = \mathcal{O}(g(n))$

    if both are true: $\qquad f(n) = \Theta(g(n))$
- computation
  - • Addition:
    $\Theta(f(n)) + \Theta(g(n)) = \Theta(\max\{f(n), g(n)\})$
    $\sum_i \Theta(f_i(n)) = \Theta\big(\sum_i f_i(n)\big)$

    Attention: not iteratively!

    • Mulitplication:
    $c \cdot \Theta(f(n)) = \Theta(f(n))$ for constant $c > 0$
    $\Theta(f(n)) \cdot \Theta(g(n)) = \Theta(f(n) \cdot g(n))$

    • Attention: equations always "from left to right":
    $f(n) = \Theta(g(n))$ vs. $\Theta(g(n)) = f(n)$

Methods to solve recurrence relations

- [[Iterative Method]]
- [[Master Theorem]]
- [[Substitution Method]]