4.9 Das Wachstumsverhalten von Funktionen

Die Groß-O-Notation wurde von D. E. Knuth in der Algorithmenanalyse eingeführt. Sie wurde ursprünglich von Paul Bachmann (1837–1920) entwickelt und von Edmund Landau (1877–1938) in seinen Arbeiten verbreitet.

Definition 39 (Groß-O-Notation)

- $f(n) \in \mathcal{O}\big(g(n)\big)$ (für $n \to \infty$) genau dann, wenn $\exists c > 0,\ n_0 \in \mathbb{N}$, so dass
$$(\forall n \geq n_0) \quad \big[|f(n)| \leq c \cdot g(n)\big]$$
„f wächst bis auf einen konstanten Faktor nicht schneller als g"

- $f(n) \in o\big(g(n)\big)$ (für $n \to \infty$) genau dann, wenn $\forall\ c > 0\ \exists\ n_0 \in \mathbb{N}$, so dass
$$(\forall n \geq n_0) \quad \big[|f(n)| < c \cdot g(n)\big]$$
„f wächst echt langsamer als g"

- $f(n) \in \Omega\big(g(n)\big)$ (für $n \to \infty$) genau dann, wenn $\exists c > 0,\ n_0 \in \mathbb{N}$, so dass

$$(\forall n \geq n_0)\ \big[|f(n)| \geq c \cdot g(n) \geq 0\big]$$

„f wächst bis auf einen konstanten Faktor nicht langsamer als g“

- $f(n) \in \omega\big(g(n)\big)$ (für $n \to \infty$) genau dann, wenn $\forall\ c > 0\ \exists\ n_0 \in \mathbb{N}$, so dass

$$(\forall n \geq n_0)\ \big[|f(n)| > c \cdot g(n) \geq 0\big]$$

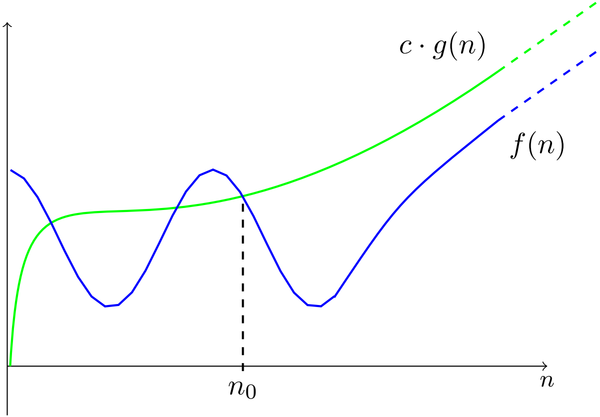
„f wächst echt schneller als g“

- $f(n) \in \Theta\big(g(n)\big)$ (für $n \to \infty$) genau dann, wenn

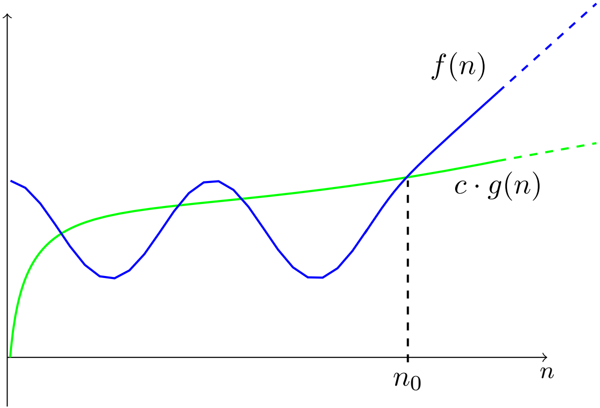
$$f(n) \in \mathcal{O}\big(g(n)\big) \text{ und } f(n) \in \Omega\big(g(n)\big)$$

„f wächst (bis auf konstante Faktoren) genauso schnell wie g“

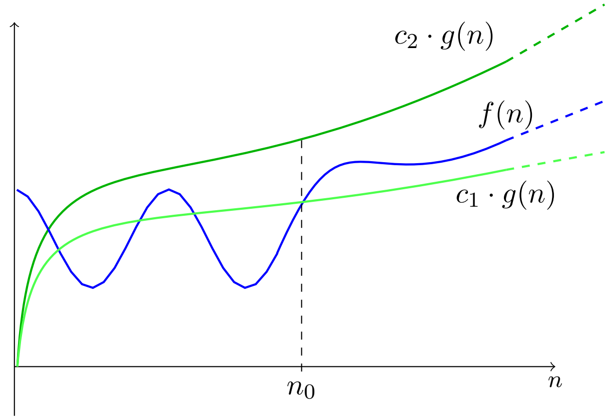
Graphische Darstellung von $\mathcal{O}$



Graphische Darstellung von ω



Graphische Darstellung von Θ



- $f(n) \in \Omega_\infty(g(n))$ genau dann, wenn $\exists\ c > 0$, so dass für unendlich viele $n \in \mathbb{N}$

$$|f(n)| \geq c \cdot g(n) \geq 0\,.$$

- $f(n) \in \omega_\infty(g(n))$ genau dann, wenn $\forall\ c > 0\ \exists$ unendlich viele $n \in \mathbb{N}$ mit

$$|f(n)| > c \cdot g(n) \geq 0\,.$$

Bemerkungen:

1. Man schreibt oft, aber logisch unsauber $f(n) = \mathcal{O}(g(n))$.
2. Oft werden nur Funktionen $\mathbb{N}_0 \to \mathbb{N}_0$ betrachtet (oder $\mathbb{N} \to \mathbb{N}_0$); dann sind die Absolutbeträge überflüssig.
3. Manchmal werden auch Funktionen $\mathbb{R} \to \mathbb{R}$ oder das Verhalten für $x \to a$ betrachtet.
4. Achtung: Die Notation für Ω und Ω_∞ ist in der Literatur nicht eindeutig; im Zweifelsfall muss auf die jeweilige Definition geachtet werden!

Rechenzeit in Abhängigkeit von der Problemgröße

Problemgröße	Zeitbedarf					
n	$\log n$	n	$n \log n$	n^2	2^n	$n!$
10	3×10^{-9} s	10^{-8} s	3×10^{-8} s	10^{-7} s	10^{-6} s	3×10^{-3} s
10^2	7×10^{-9} s	10^{-7} s	7×10^{-7} s	10^{-5} s	4×10^{13} yr	*
10^3	$1,0 \times 10^{-8}$ s	10^{-6} s	1×10^{-5} s	10^{-3} s	*	*
10^4	$1,3 \times 10^{-8}$ s	10^{-5} s	1×10^{-4} s	10^{-1} s	*	*
10^5	$1,7 \times 10^{-8}$ s	10^{-4} s	2×10^{-3} s	10 s	*	*
10^6	2×10^{-8} s	10^{-3} s	2×10^{-2} s	17 min	*	*

Annahme: eine Operation dauert 10^{-9} Sekunden, $\log n = \log_2 n$

Bezeichnung von Wachstums-Größenordnungen

$o(1)$	konvergiert gegen 0
$\mathcal{O}(1)$	beschränkt durch Konstante
$\mathcal{O}(\log n)$	logarithmische Funktion
$\mathcal{O}(\log^k n)$	polylogarithmische Funktion
$\mathcal{O}(n)$	linear beschränkte Funktion
$\bigcup_{k\geq 0} \mathcal{O}(n^k)$	polynomiell beschränkte Funktion
$\bigcup_{c\geq 0} \Omega(2^{cn})$	(mindestens) exponentielle Funktion