



openHPI

Einführung in die Mathematik der Algorithmik

– O-Notation

Thomas Bläsius, Pascal Lenzner

Hasso-Plattner-Institut

Universität Potsdam

Pascal und Thomas haben Algorithmen entwickelt

- ihre Laufzeiten hängen von der Problemgröße n ab
- Pascals Algorithmus benötigt $f(n) = n^2 + 2$ Schritte
- Thomas Algorithmus benötigt $g(n) = 4n + 1$ Schritte

Egal wie schnell Pascals Rechner ist...

- ... für ausreichend großes n ist Thomas Algorithmus schneller

■ denn:

$$\begin{aligned} g(n) &= 4n + 1 \\ \text{setze } n &= 5c + \varepsilon \text{ mit } \varepsilon > 0 \\ &= 4(5c + \varepsilon) + 1 \\ &= 20c + 4\varepsilon + 1 \end{aligned}$$
$$\begin{aligned} \frac{f(n)}{c} &= \frac{n^2 + 2}{c} \\ &= \frac{(5c + \varepsilon)^2 + 2}{c} \\ &= 25c + 10\varepsilon + \frac{\varepsilon^2 + 2}{c} \end{aligned}$$

$\Rightarrow$ für $n > 5c$ gilt $\frac{f(n)}{c} \geq g(n)$

Formal: $\forall c \exists n_0 \forall n > n_0 f(n) > c \cdot g(n)$

- wir sagen: $f(n)$ wächst **asymptotisch schneller** als $g(n)$
- und schreiben: $f(n) \in \omega(g(n))$

Notation	Asymptotischer Vergleich (für großes n)	Formale Definition
$f(n) \in \omega(g(n))$ (egal wieviel schneller der Rechner für $f(n)$, für genügend großes n ist $f(n)$ größer als $g(n)$)	$f(n)$ wächst schneller als $g(n)$	$\forall c \exists n_0 \forall n > n_0 f(n) > c \cdot g(n)$
$f(n) \in \Omega(g(n))$ (wenn der Rechner für $f(n)$ langsam genug und n groß genug ist, dann ist $f(n)$ größer als $g(n)$)	$f(n)$ wächst min. so schnell wie $g(n)$	$\exists c \exists n_0 \forall n > n_0 c \cdot f(n) \geq g(n)$
$f(n) \in \Theta(g(n))$ (es hängt vom Rechner ab, welche Laufzeit für großes n schneller wächst)	$f(n)$ und $g(n)$ wachsen gleich schnell	$f(n) \in O(g(n)) \wedge f(n) \in \Omega(g(n))$
$f(n) \in O(g(n))$ (wenn der Rechner für $f(n)$ schnell genug und n groß genug ist, dann ist $f(n)$ kleiner als $g(n)$)	$f(n)$ wächst max. so schnell wie $g(n)$	$\exists c \exists n_0 \forall n > n_0 f(n) \leq c \cdot g(n)$
$f(n) \in o(g(n))$ (egal wieviel langsamer der Rechner für $f(n)$, für genügend großes n ist $f(n)$ kleiner als $g(n)$)	$f(n)$ wächst langsamer als $g(n)$	$\forall c \exists n_0 \forall n > n_0 c \cdot f(n) < g(n)$

Zusammenfassung

- mit dieser Notation kann man Laufzeiten unabhängig vom ausführenden Computer beschreiben
- Bezeichnungen: O-Notation, Landau-Symbole
- Laufzeiten von Algorithmen werden meist als $O(g(n))$ angegeben (man ist hauptsächlich daran interessiert, wie lang es maximal dauert)