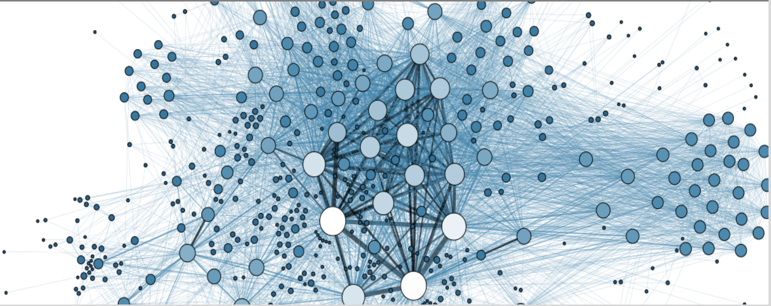
Karlsruher Institut für Technologie

Grundbegriffe der Informatik
Tutorium 38

O-Notation

Patrick Fetzer, `uxkln@student.kit.edu` | 10.01.2018



Patrick Fetzer,
uxkln@student.kit.edu

Komplexitätstheorie

Wichtige Komplexitätsmaße:

- Speicherplatzbedarf
- Rechen- bzw. Laufzeit

Unterscheidung in

- Best Case (oft uninteressant)
- Average Case (schwierig zu finden, deswegen selten angegeben)
- Worst Case (meistens angegeben)

Ignorieren konstanter Faktoren

Patrick Fetzer,
uxkln@student.kit.edu

Definition

Seien $g, f : \mathbb{N}_0 \to \mathbb{R}_0^+$ Funktionen. Dann wächst g asymptotisch genauso schnell wie f genau dann, wenn gilt:
$\exists c, c' \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : c \cdot f(n) \leq g(n) \leq c' \cdot f(n)$

Notation

$f \asymp g$ oder $f(n) \asymp g(n)$ (äsymptotisch gleich")

Bemerkung

$\asymp$ ist eine Äquivalenzrelation

Theta

Patrick Fetzer,
uxkln@student.kit.edu

Karlsruher Institut für Technologie

Definition

$\Theta(f) = \{g | g \asymp f\}$

Satz

$\forall a, b \in \mathbb{R}_+ : \Theta(a \cdot f) = \Theta(b \cdot f)$

Obere und untere Schranke

Karlsruher Institut für Technologie

Obere Schranke (Worst-Case Approximation)

$$O(f) = \{g \mid \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

Untere Schranke (Best-Case Approximation)

$$\Omega(f) = \{g \mid \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$$

Notation

- $g \preccurlyeq f$ falls $g \in O(f)$ bzw. g wächst asymptotisch höchstens so schnell wie f
- $g \succcurlyeq f$ falls $g \in \Omega(f)$ bzw. g wächst asymptotisch mindestens so schnell wie f

Bemerkung

Es gilt $\Theta(f) = O(f) \cap \Omega(f)$

Patrick Fetzer,
uxkln@student.kit.edu

Lemma

$log_a n \in \Theta(log_b n)$

Beispiel

$log_2 n \in \Theta(log_8 n)$

Beweis

$\frac{1}{3} log_2 n = \frac{1}{log_2 8} log_2 n = \frac{log_2 n}{log_2 8} = log_8 n \leq log_2 n$

Patrick Fetzer,
uxkln@student.kit.edu

Aufgabe

Gilt $log_2(n^{20}) \in \Theta(logn)$

Lösung

Ja, denn $log_2(n^{20}) = 20 \cdot log_2 n$

Patrick Fetzer,
uxkln@student.kit.edu

Algorithmus von Warshall

Karlsruher Institut für Technologie

$$
\begin{array}{l}
\textbf{for}\ \ i \leftarrow 0\ \ \textbf{to}\ \ n-1\ \ \textbf{do} \\
\quad \textbf{for}\ \ j \leftarrow 0\ \ \textbf{to}\ \ n-1\ \ \textbf{do} \\
\quad\quad W_{ij}^{[0]} \leftarrow \begin{cases} 1 & \text{falls } i = j \\ A_{ij} & \text{falls } i \neq j \end{cases} \\
\quad \textbf{od} \\
\textbf{od}
\end{array}
$$

$$
\begin{array}{l}
\textbf{for}\ \ k \leftarrow 0\ \ \textbf{to}\ \ n-1\ \ \textbf{do} \\
\quad \textbf{for}\ \ i \leftarrow 0\ \ \textbf{to}\ \ n-1\ \ \textbf{do} \\
\quad\quad \textbf{for}\ \ j \leftarrow 0\ \ \textbf{to}\ \ n-1\ \ \textbf{do} \\
\quad\quad\quad W_{ij}^{[k+1]} \leftarrow \max(W_{ij}^{[k]}, W_{ik}^{[k]} \cdot W_{kj}^{[k]}) \\
\quad\quad \textbf{od} \\
\quad \textbf{od} \\
\textbf{od}
\end{array}
$$

Wie schnell ist das ganze?

Patrick Fetzer,
uxkln@student.kit.edu

Gelten folgende Approximationen?

- $4n^2 + \pi n + 2\sqrt{n} \in \Theta(n^2)$? Ja.
- $4n^{2,1} + \pi n + 2\sqrt{n} \in \Theta(n^2)$? Nein.

Es sind immer nur die höchsten Faktoren interessant!

- $4n^4 + 3c^6 \in \Theta(n^4)$? Ja, c ist eine Konstante, $3c^6 = (3c^6)n^0$ hat eine kleinere Potenz als n^4.
- $\log_{4213}(n) \in \Theta(\log_2(n))$ Ja, die Basis des Logarithmus ist im O-Kalkül egal.
  - Grund: $\mathcal{O}(\log_b n) = \mathcal{O}(\frac{\log_a n}{\log_a b}) = \mathcal{O}(\frac{1}{\log_a b}\log_a n) = \mathcal{O}(\log_a n)$.
- $n! \in \Theta(n^{\pi e 2000})$ Nein, Fakultät wächst asymptotisch schneller als fast alles andere.

Patrick Fetzer,
uxkln@student.kit.edu

Aufgabe

Übungsaufgabe

Entscheide für jede Zelle, ob die Formel der Zeile in der Menge der Spalte liegt.

	$\mathcal{O}(n^3)$	$\mathcal{O}(n)$	$\Theta(c!)$	$\Theta(n^\pi)$	$\Omega(n^6)$	$\Omega(n!)$
$2n^2 + 4n$	$\in$	$\notin$	$\notin$	$\notin$	$\notin$	$\notin$
π	$\in$	$\in$	$\in$	$\notin$	$\notin$	$\notin$
$\log(n)$	$\in$	$\in$	$\notin$	$\notin$	$\notin$	$\notin$
$n\log(n)$	$\in$	$\notin$	$\notin$	$\notin$	$\notin$	$\notin$
n^π	$\notin$	$\notin$	$\notin$	$\in$	$\notin$	$\notin$
$12n^3 + 7000n^2$	$\in$	$\notin$	$\notin$	$\notin$	$\notin$	$\notin$
n^3	$\in$	$\notin$	$\notin$	$\notin$	$\notin$	$\notin$
$n!$	$\notin$	$\notin$	$\notin$	$\notin$	$\in$	$\in$

Schnitt

Patrick Fetzer,
uxkln@student.kit.edu

- $\mathcal{O}(n^2) \cap \mathcal{O}(n) = \mathcal{O}(?)$? $= \mathcal{O}(n)$.
- $\mathcal{O}(n^2) \cap \Omega(n^3) = \emptyset$

Grundlegende Reihenfolge von Größen

Patrick Fetzer,
uxkln@student.kit.edu

$$1 \preccurlyeq \log n \preccurlyeq n \preccurlyeq n \log n \preccurlyeq n^2 \preccurlyeq n^3 \preccurlyeq n^{10000} \preccurlyeq 2^n \preccurlyeq 3^n \preccurlyeq 1000^n \preccurlyeq n! \preccurlyeq n^n$$

Mathematische Definitionen

Patrick Fetzer,
uxkln@student.kit.edu

$$f(n) \in \Omega(g(n)) \Leftrightarrow 0 < \liminf_{n\to\infty} \frac{f(n)}{g(n)} \leq \infty$$

$$f(n) \in \Theta(g(n)) \Leftarrow 0 < \lim_{n\to\infty} \frac{f(n)}{g(n)} = c < \infty$$

$$f(n) \in \mathcal{O}(g(n)) \Leftrightarrow 0 \leq \limsup_{n\to\infty} \frac{f(n)}{g(n)} = c < \infty$$

Zeige

- $3n^2 + 14n + 159 \in \Theta(n^2)$
- $\log n^2 \in \Theta(\log n^3)$
- $\log^2 n \in \mathcal{O}(\log^3 n)$

Patrick Fetzer,
uxkln@student.kit.edu

Größenordnung	Bezeichnung
$\mathcal{O}(1)$	konstante Laufzeit
$\mathcal{O}(\log n)$	logarithmische Laufzeit
$\mathcal{O}(\log^2 n)$	quadratisch logarithmische Laufzeit
$\mathcal{O}(n)$	lineare Laufzeit
$\mathcal{O}(n^2)$	quadratische Laufzeit
$\mathcal{O}(n^3)$	kubische Laufzeit
$\mathcal{O}(n^k)$	polynomielle Laufzeit

Grundbegriffe der Informatik

Patrick Fetzer,
uxkln@student.kit.edu

Komplexitätstheorie

O-Notation