

Algorithmen und Datenstrukturen (für ESE) WS 2010 / 2011

Vorlesung 3, Montag, 8. November 2010
(Asymptotische Analyse / O-Notation)

Prof. Dr. Hannah Bast
Lehrstuhl für Algorithmen und Datenstrukturen
Institut für Informatik
Universität Freiburg

Blick über die Vorlesung heute

■ Organisatorisches

- Keine Übungsgruppe morgen am 9. November
- Sollen die Übungen aufgenommen werden?
- Ihre Erfahrungen mit dem 2. Übungsblatt

■ Asymptotische Analyse / O-Notation

- Motivation
- Definition O , Ω , Θ
- Beispiele
- Bezug zu Grenzwerten
- Vorsicht

Der Übungstermin am Dienstag

■ Morgen ...

- ... am 9. November ist keine Übung
(weil letzte Woche keine Vorlesung war und es auch kein neues Übungsblatt gab)

■ Aufnahme

- Ich hatte die letzte Übung aufnehmen lassen, weil es nur einen Übungstermin gibt, zu dem vielleicht nicht alle können
- Die Frage ist, ob es überhaupt jemand anschaut
- Oder ob sich nicht manche davon sogar gestört / gehemmt fühlen
- Ohne ein starkes Votum dafür lassen wir es in Zukunft

Ihre Erfahrungen mit dem 2. Ü-Blatt

■ Zusammenfassung von Ihrem Feedback

- Die meisten fanden den Beweis viel schwieriger als das Programm
- Problem dabei war nicht die Mathematik sondern die Beweistechnik (hier Induktionsbeweis)
- "Beweisen ist nervig, langweilig und macht keinen Spaß!"
- Manche fanden aber auch das Programm schwierig, insbesondere die Rekursion + keine Erklärung dazu in VL
- Einige hatten Probleme mit Zeigern / Pointern und / oder waren generell außer Übung beim Programmieren
- Zeitaufwand lag bei den meisten unter 10 Stunden (was als viel empfunden wurde aber ok ist für 2 Wochen) bei einigen aber auch deutlich darüber

- In der letzten Vorlesung (Sortieren) ...
 - ... hatten wir so Aussagen wie:
 - Die Laufzeit des Algorithmus ist $\leq A_2 \cdot n \cdot \log n$ für $n \geq 2$
 - Die Laufzeit des Algorithmus ist $\geq A_1 \cdot n \cdot \log n$ für $n \geq 2$
 - Dabei waren uns die Konstanten A_1 und A_2 erstmal egal
 - Und auch, dass die Schranken erst ab $n \geq 2$ gelten
 - Uns hat interessiert, dass die Laufzeit "wächst wie" $n \cdot \log n$
 - Das wollen wir in dieser Vorlesung formaler machen
 - Wir wollen in Zukunft so Aussagen machen wie
 - Die Laufzeit des Algorithmus ist $O(n \cdot \log n)$ O von ...
 - Die Laufzeit des Algorithmus ist $\Omega(n \cdot \log n)$ Omega von ...
 - Die Laufzeit des Algorithmus ist $\Theta(n \cdot \log n)$ Theta von ...

O-Notation — Definition 1/4

■ Vorweg

- Wir betrachten Funktionen $f : \mathbb{N} \rightarrow \mathbb{R}$
 - $\mathbb{N}$ = die natürlichen Zahlen (oft: die Eingabegrößen)
 - $\mathbb{R}$ = die reellen Zahlen (oft: irgendwelche Kosten)
- Zum Beispiel
 - $f(n) = 3 \cdot n$
 - $f(n) = 2 \cdot n \cdot \log n$
 - $f(n) = n^2 / 10$
 - $f(n) = n^2 + 3 \cdot n \cdot \log n - 4 \cdot n$

O-Notation — Definition 2/4

■ Groß-O

- Definition: Für eine Funktion $f : \mathbb{N} \rightarrow \mathbb{R}$ ist

$$O(f) = \{ g : \mathbb{N} \rightarrow \mathbb{R} \mid \exists n_0 \in \mathbb{N} \exists C > 0 \forall n \geq n_0 \ g(n) \leq C \cdot f(n) \}$$

dabei heißt $\exists$ = "es existiert ..." und $\forall$ = "für alle ..."

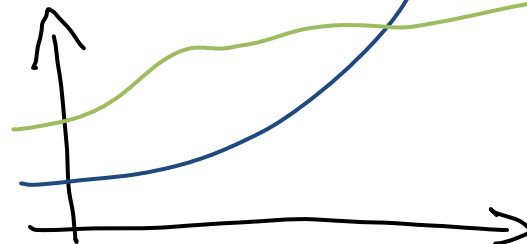
- Zum Beispiel $f(n) = n \cdot \log n$, $g(n) = 5 \cdot n + 7$, dann $g \in O(f)$

Man schreibt das meistens einfach $5 \cdot n + 7 = O(n \cdot \log n) \underset{\in}{=} O(n^2)$

- Beweis:

zu zeigen: $\exists n_0 \in \mathbb{N}, C > 0 \forall n \geq n_0 \ g(n) \leq C \cdot f(n)$

$$g(n) = 5 \cdot n + 7 \stackrel{n \geq 1}{\leq} 12 \cdot n \stackrel{n \geq 2}{\leq} 12 \cdot n \cdot \log_2 n = 12 \cdot f(n) \quad [\text{dann ist } g \in O(f)]$$



O-Notation — Definition 3/4

■ Groß-Omega

– Definition: Für eine Funktion $f : \mathbb{N} \rightarrow \mathbb{R}$ ist

$$\Omega(f) = \{ g : \mathbb{N} \rightarrow \mathbb{R} \mid \exists n_0 \in \mathbb{N} \exists C > 0 \forall n \geq n_0 \ g(n) \geq C \cdot f(n) \}$$

– Zum Beispiel $n \cdot \log n = \Omega(n)$

– Beweis:

$$g(n) = n \cdot \log_2 n, \quad f(n) = n$$

$$\text{zu zeigen: } \exists n_0 \in \mathbb{N}, \exists C > 0 \forall n \geq n_0 \ g(n) \geq C \cdot f(n)$$

[dann $g \in \Omega(f)$]

$$g(n) = n \cdot \log_2 n \geq \underbrace{\dots \geq 1}_{\geq 1 \text{ für } n \geq 2} \cdot n \geq C \cdot n$$

$$\geq n \text{ für } n \geq 2 \quad \mathbb{D}$$

■ Groß-Theta

- Definition: Für eine Funktion $f : \mathbb{N} \rightarrow \mathbb{R}$ ist

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

$X \cap Y$: die Schnittmenge von X und Y

- Zum Beispiel $3 \cdot n^2 + 5 \cdot n = \Theta(n^2)$

zu zeigen : (1) $3 \cdot n^2 + 5 \cdot n = O(n^2)$

(2) $3 \cdot n^2 + 5 \cdot n = \Omega(n^2)$

@ (1) : $3 \cdot n^2 + 5 \cdot n^{\leq n^2} \leq 8 \cdot n^2$ für alle $n \geq 1$

@ (2) : $3 \cdot n^2 + 5 \cdot n^{\geq 0} \geq 3 \cdot n^2$

- In den bisherigen Beispielen ...
 - ... haben wir die Zugehörigkeit zu $O(\dots)$ etc. gewissermaßen "zu Fuß" bewiesen, indem wir explizit das n_0 und das C bestimmt haben
 - Die Definitionen erinnern aber sehr an den Grenzwertbegriff aus der Analysis
 - Definition: Eine unendliche Folge $f_1, f_2, f_3, \dots$ hat einen Grenzwert L , wenn für alle $\varepsilon > 0$ ein $n_0 \in \mathbb{N}$ existiert so dass für alle $n \geq n_0$ gilt dass $|f_n - L| \leq \varepsilon$
 - In Symbolen schreibt man dann $\lim_{n \rightarrow \infty} f_n = L$
 - Eine Funktion $f : \mathbb{N} \rightarrow \mathbb{R}$ kann man genauso gut als Folge $f(1), f(2), f(3), \dots$ auffassen und schreibt $\lim_{n \rightarrow \infty} f(n) = L$

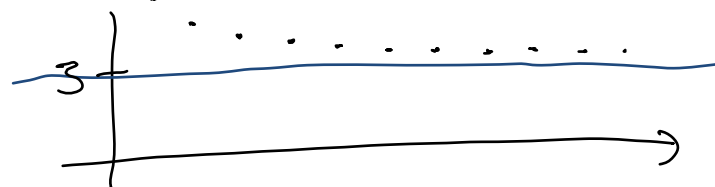
O-Notation — Grenzwerte 1/4

■ Beispiel für einen (bewiesenen) Grenzwert

(sollten Sie eigentlich in Mathe 1 schon mal gesehen haben)

$$f_n = 3 + \frac{1}{n} \quad ; \quad 4, 3.5, 3.33\ldots, 3.25, 3.2, \ldots$$

$$f(n) = 3 + \frac{1}{n}$$



Bekanntung: $\lim_{n \rightarrow \infty} f(n) = 3$

Zu zeigen $\forall \varepsilon > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 |f(n) - 3| \leq \varepsilon$

Also, sei $\varepsilon > 0$ beliebig.

z.B. $\varepsilon = 0.1$ dann $\forall n \geq 10 |f(n) - 3| = \frac{1}{n} \leq \frac{1}{10} = \varepsilon$

ε beliebig: $n_0 = \lceil 1/\varepsilon \rceil$

dann $\forall n \geq n_0 |f(n) - 3| = \frac{1}{n} \leq \frac{1}{\lceil 1/\varepsilon \rceil} \leq \frac{1}{1/\varepsilon} = \varepsilon \quad \square$

■ Satz

- Seien $f, g : \mathbb{N} \rightarrow \mathbb{R}$ und der Grenzwert $\lim_{n \rightarrow \infty} f(n)/g(n)$ existiert (evtl. ist er ∞)
- Dann gelten
 - (1) $f(n) = O(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} f(n)/g(n) < \infty$
 - (2) $f(n) = \Omega(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} f(n)/g(n) > 0$
 - (3) $f(n) = \Theta(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} f(n)/g(n) \in (0, \infty)$

O-Notation — Grenzwerte 3/4

- Beweis

- Beweis (Fortsetzung)

■ Sprechweise

- Die O-Notation schaut sich das Verhalten der Funktionen an, wenn $n \rightarrow \infty$ geht (es interessieren nur die $n \geq n_0$)
- Wenn man Laufzeiten, Kosten, etc. als $O(\dots)$ oder $\Omega(\dots)$ oder $\Theta(\dots)$ ausdrückt, spricht man daher von

asymptotischer Analyse

■ Vorsicht

- Asymptotische Analyse sagt nichts über das Laufzeitverhalten bei kleinen Eingabegrößen ($n < n_0$) aus
- Für $n < 2$ oder $n < 10$ ist das egal, da wird schon nichts Schlimmes passieren
- Aber ...

O-Notation — Diskussion 2/2

■ Beispiel

- Algorithmus A hat Laufzeit $T_A(n) = 80 \cdot n$
- Algorithmus B hat Laufzeit $T_B(n) = 2 \cdot n \cdot \log n$
- Dann ist $T_A(n) = O(T_B(n))$ aber nicht $\Theta(T_B(n))$
- Das heißt, A ist asymptotisch schneller als B
- Allerdings

$$\begin{aligned} \frac{T_A(n)}{T_B(n)} &= \frac{80 \cdot n}{2 \cdot n \cdot \log n} \\ &= \frac{40}{\log n} \xrightarrow{n \rightarrow \infty} 0 \end{aligned}$$

$$\text{also } \lim_{n \rightarrow \infty} \frac{T_A(n)}{T_B(n)} = 0$$

$$T_B(n) \leq T_A(n)$$

$$\Leftrightarrow 2 \cdot n \cdot \log_2 n \leq 80 \cdot n$$

$$\Leftrightarrow \log_2 n \leq 40$$

$$\Leftrightarrow n \leq 2^{40} = (2^{10})^4 \approx 1000^4 = 10^{12} = 1000 \text{ Milliard.} = 1 \text{ Tera}$$

d.h. für Eingabegrößen bis zu 1 Tera
ist Algorithmus B schneller (trotz schlechterer
Asymptotik)

- O-Notation / Ω -Notation / Θ -Notation

- In Mehlhorn/Sanders:

- 2.1 Asymptotic Notation

- In Cormen/Leiserson/Rivest

- 2.1 Asymptotic Notation

- In Wikipedia

- http://en.wikipedia.org/wiki/Big_O_notation

- <http://de.wikipedia.org/wiki/Landau-Symbole>

