Informatik II: Algorithmen und Datenstrukturen SS 2013

Vorlesung 3, Dienstag, 30. April 2013
(O-Notation, Groß-O, Omega, Theta, usw.)

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

UNI FREIBURG

Blick über die Vorlesung heute

- Organisatorisches
  - Ihre Erfahrungen mit dem Ü2 (Induktion + Laufzeit QuickSort)
  - Morgen (Mittwoch) ist Feiertag = keine Vorlesung
- O-Notation
  - Motivation
  - Klassische Definition mit C und n_0
  - Einfachere Bestimmung über den $\lim_{n \to \infty}$
  - $f = O(g)$ heißt nicht immer f ist besser
  - Übungsblatt 3: ein paar Beweise / Rechenaufgaben dazu

    So was in der Art kommt meistens auch in der Klausur !

Erfahrungen mit dem Ü2 (Induktion)

- Zusammenfassung / Auszüge Stand 30. April 15:00
  - Aufgaben 1 und 2 waren machbar, wenn auch knifflig
  - Aufgabe 3 für manche gut machbar mit Vorlage aus der VL

    andere hätten sich mehr Hilfestellung zu der Aufgabe gewünscht !
  - Die Mathe StudentInnen haben sich gefreut über das Blatt
  - Das Blatt war einfach nur Scheiße
  - Immer noch nicht einverstanden mit 10 h / Übungsblatt

    "Die Kultusministerkonferenz berechnet für eine 30-ECTS-Woche übrigens 32 - 39 Stunden."

    Das ist aber, wenn man 46 Wochen / Jahr so arbeitet !

    Tatsächlich gibt es nur ca. 25 Übungsblätter pro Jahr

    Kompromissvorschlag: **9 Stunden / Übungsblatt**

Arbeitsaufwand / ECTS

- Zweiter Versuch ...

Vorlesung : 3h / Woche im Semester

Übungen : 9h / Woche " "

12h / Woche " "

1.5h / ECTS im Semester

≈ 30 ECTS ⇒ 45h / Woche i. S.

$$\frac{25 \cdot 45h + 21 \cdot 30h}{46} = 38h \,/\, \text{Woche}$$

Wochen pro Jahr

Arbeits

O-Notation — Motivation

- Primär interessiert uns oft
  - das "Wachstum" einer Funktion, z.B. einer Laufzeit $T(n)$
  - Die Werte der Konstanten (z.B. A) sind dabei oft sekundär
  - Und auch, wenn die Schranken erst ab $n \geq \ldots$ gelten
  - Zum Beispiel war beim Sortieren interessant, dass
    - die Laufzeit von MinSort "wächst wie" n^2
    - aber die Laufzeit von QuickSort "wächst wie" $n \cdot \log n$
  - Das wollen wir jetzt formaler machen, damit wir in Zukunft etwas schreiben bzw. sagen können wie:
    - Die Laufzeit des Algorithmus ist $O(n)$ "O von n"
    - Die Laufzeit des Algorithmus ist $\Omega(n)$ "Omega von n"
    - Die Laufzeit des Algorithmus ist $\Theta(n)$ "Theta von n"

O-Notation — Definition 1/7

- Vorweg
  - Wir betrachten Funktionen $f : \mathbf{N} \rightarrow \mathbf{R}$
    - **N** = die natürlichen Zahlen ... typisch: Eingabegröße
    - **R** = die reellen Zahlen ... typisch: Laufzeit
  - Zum Beispiel
    - $f(n) = 3 \cdot n + 3$
    - $f(n) = 2 \cdot n \cdot (\log_2 n - 5)$
    - $f(n) = n^2 / 10$
    - $f(n) = n^2 + 3 \cdot n \cdot \log_2 n - 4 \cdot n$

O-Notation — Definition 2/7

- Groß-O, Definition
  - Seien g und f zwei Funktionen $\mathbf{N} \to \mathbf{R}$
  - **Intuitiv:** Man sagt g ist Groß-O von f ...

    wenn g "höchstens so stark wächst wie" f

    es zählt die "Wachstumsrate", nicht die absoluten Werte!
  - **Informal:** Man schreibt $g = O(f)$ ...

    wenn ab irgendeinem Wert n_0 für all $n \geq n_0$

    $g(n) \leq C \cdot f(n)$ für irgendeine Konstante C
  - **Formal:** für eine Funktion $f : \mathbf{N} \to \mathbf{R}$ ist ...

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

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

es kann durchaus $g = O(f)$ sein aber trotzdem $g(n) > f(n)$

zum Beispiel:

$g(n) = 5 \cdot n$

$f(n) = 2 \cdot n$

$g = O(f)$

$g(n) > f(n)$

O-Notation — Definition 3/7

- Groß-O, Beispiel
  - Sei $g(n) = 5 \cdot n + 7$ und $f(n) = n$
  - Dann ist $g = O(f)$ bzw. man schreibt $5 \cdot n + 7 = O(n)$
  - **Intuitiv:** $5 \cdot n + 7$ wächst höchstens "linear"
  - Beweis unter Verwendung der Definition von O :

eigentlich müsste man schreiben $g \in O(f)$

auch gilt übrigens $g(n) > f(n) \ \forall n.$

$$
\begin{aligned}
g(n) &= 5 \cdot n + 7 \\
&\le 5 \cdot n + 7 \cdot n \quad \forall n \ge 1 \\
&\le 12 \cdot n \quad \forall n \ge 1 \qquad C := 12, \ n_0 := 1 \\
&\le C \cdot \underbrace{n}_{g(n)} \quad \forall n \ge n_0 \qquad \blacksquare
\end{aligned}
$$

Es gibt viele Möglichkeiten; z.B. auch

$$g(n) = 5 \cdot n + \underbrace{7}_{\le n} \le 5 \cdot n + n \le \underbrace{6}_{=:C} \cdot n \qquad \forall n \ge \underbrace{7}_{=: n_0}$$

O-Notation — Definition 4/7

auch hier gilt:
es kann sein
$g = \Omega(f)$ aber $g < f$

wie O, nur hier
$\geq$ statt $\leq$

- Groß-Omega, Definition + Beispiel
  - **Intuitiv:** Man sagt g ist Groß-Omega von f ...

    ... wenn g "mindestens so stark wächst wie" f

    Also wie Groß-O, nur mit "mindestens" statt "höchstens"
  - **Formal:** Für eine Funktion $f : N \rightarrow R$ ist

    $\Omega(f) = \{ g : N \rightarrow R \mid \exists n_0 \in N \;\; \exists C>0 \;\; \forall n \geq n_0 \;\; g(n) \geq C \cdot f(n) \}$
  - Zum Beispiel $5 \cdot n + 7 = \Omega(n)$
  - Beweis unter Verwendung der Definition von Ω :

$5 \cdot n + 7 \geq 5 \cdot n \qquad C := 5, \; n_0 = 1$

$\vdots$

$\geq C \cdot n \qquad \forall n \geq n_0$

O-Notation — Definition 5/7

wenn $g = \Theta(f)$
kann sowohl
$g < f$ als auch $g > f$ sein
(oder auch weder noch)

- Groß-Theta, Definition + Beispiel
  - **Intuitiv:** Man sagt g ist Theta von f ...

    ... wenn g "genauso so stark wächst wie" f
  - **Formal:** Für eine Funktion $f : N \rightarrow R$ ist

    $\Theta(f) = O(f) \cap \Omega(f) =$ die Schnittmenge von $O(f)$ und $\Omega(f)$
  - Zum Beispiel $5 \cdot n + 7 = \Theta(n)$
  - Beweis unter Verwendung der Definition von Θ :

$5 \cdot n + 7 \in O(n)$ Folie 8

$5 \cdot n + 7 \in \Omega(n)$ Folie 9

$\Rightarrow 5 \cdot n + 7 \in O(n) \wedge \Omega(n) = \Theta(n)$ ∎

Funktionen in $\Theta(n)$ z.B.:

$3 \cdot n - 2$, $4 \cdot n + 7$, $5 \cdot n - 14$, ...

O-Notation — Definition 6/7

- Es gibt auch noch o (Klein-O) und ω (Klein-Omega)
  - Die braucht man nicht ganz so oft
  - Hier kurz die Definitionen für $f : N \rightarrow R$

    $o(f) = \{ g : \forall \varepsilon > 0 \;\; \exists n_0 \in N \;\; \forall n \geq n_0 \;\; g(n) < \varepsilon \cdot f(n) \}$

    $\omega(f) = \{ g : \forall C > 0 \;\; \exists n_0 \in N \;\; \forall n \geq n_0 \;\; g(n) > C \cdot f(n) \}$
  - Intuitiv:

    $g = o(f)$: g wächst (strikt) langsamer als f

    $g = \omega(f)$: g wächst (strikt) schneller als f

    Insbesondere : $o(g) \cap \omega(g) = \emptyset$

O-Notation — Definition 7/7

- Intuitive Zusammenfassung
  - Die Operatoren O , Ω , Θ , o , ω sind auf Funktionen, was die Operatoren $\leq$, $\geq$, $=$, $<$, $>$ auf Zahlen sind:

    O entspricht $\leq$

    Ω entspricht $\geq$

    Θ entspricht $=$

    o entspricht $<$

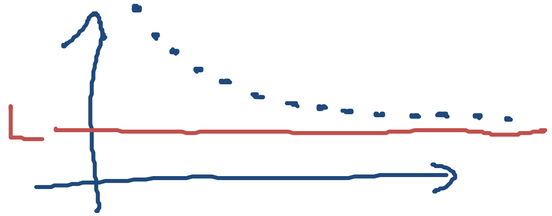
    ω entspricht $>$
  - Viele Eigenschaften übertragen sich auch

    z.B. Transitivität: $f = o(g) \wedge g = O(h) \Rightarrow f = o(h)$

    z.B. Additivität: $f_1 = O(g_1) \wedge f_2 = O(g_2) \Rightarrow f_1+f_2 = O(g_1+g_2)$

O-Notation — Grenzwerte 1/4

- In den bisherigen Beispielen ...
  - ... haben wir die Zugehörigkeit zu $O(\ldots)$ 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 $f_1, f_2, f_3, \ldots$ hat einen Grenzwert L, wenn für alle $\varepsilon > 0$ ein $n_0 \in N$ existiert so dass für alle $n \geq n_0$ gilt dass $|f_n - L| \leq \varepsilon$
  - In Symbolen schreibt man dann $\lim_{n\to\infty} f_n = L$
  - Eine Funktion $f : N \to R$ kann man genauso gut als Folge $f(1), f(2), f(3), \ldots$ auffassen und schreibt $\lim_{n\to\infty} f(n) = L$

O-Notation — Grenzwerte 2/4

- Beispiel für einen Beweis von einem Grenzwert
  (sollten Sie eigentlich in Mathe 1 schon mal gesehen haben)

zu beweisen: $\lim_{n \to \infty} \frac{1}{n} = 0$

umgangssprachlich $\frac{1}{\infty} = 0$

$L := 0$; $\varepsilon > 0$ beliebig (klein)

$n_0 := \lceil \frac{1}{\varepsilon} \rceil \geq \frac{1}{\varepsilon}$

$$\left| \frac{1}{n} - 0 \right| = \frac{1}{n}$$

$$\leq \frac{1}{n_0} \qquad \forall n \geq n_0$$

$$\leq \frac{1}{1/\varepsilon}$$

$$= \varepsilon \qquad \forall n \geq n_0$$

O-Notation — Grenzwerte 3/4

- Satz
  - Seien $f, g : \mathbf{N} \to \mathbf{R}$ und der Grenzwert $\lim_{n\to\infty} f(n)/g(n)$ existiert (evtl. ist er ∞)
  - Dann gelten

    (1) $f = O(g) \Leftrightarrow \lim_{n\to\infty} f(n)/g(n) < \infty$

    (2) $f = \Omega(g) \Leftrightarrow \lim_{n\to\infty} f(n)/g(n) > 0$

    (3) $f = \Theta(g) \Leftrightarrow \lim_{n\to\infty} f(n)/g(n) > 0$ und $< \infty$

    (4) $f = o(g) \Leftrightarrow \lim_{n\to\infty} f(n)/g(n) = 0$

    (5) $f = \omega(g) \Leftrightarrow \lim_{n\to\infty} f(n)/g(n) = \infty$

O-Notation — Grenzwerte 4/4

$|x| \le D$

$\Rightarrow x \le D$

und $-x \le D$

- Beweis von (1) ... die anderen Beweise gehen analog

zu zeigen: $f = O(g) \iff \lim_{n\to\infty} f(n)/g(n) < \infty$

„$\Rightarrow$": $f = O(g) \Rightarrow \exists C>0\ \exists n_0\ \forall n \ge n_0\ \underbrace{f(n) \le C \cdot g(n)}_{\iff \frac{f(n)}{g(n)} \le C}$

$\Rightarrow \lim_{n\to\infty} f(n)/g(n) \le C < \infty$

könnte man jetzt auch noch beweisen

„$\Leftarrow$": $\lim_{n\to\infty} f(n)/g(n) = L < \infty$

nach Def. Grenzwert

$\varepsilon := 1 \underset{\exists n_0}{\Rightarrow} |f(n)/g(n) - L| \le 1 \quad \forall n \ge n_0$

$\Rightarrow f(n)/g(n) \le L + 1 \quad \forall n \ge n_0$

$\Rightarrow f(n) \le \underbrace{(L+1)}_{=:C} \cdot g(n) \quad \forall n \ge n_0$

$\Rightarrow f = O(g)$

Rechnen mit Grenzwerten 1/2

natürliche Logarithmus $= \log_e$

- Variante 1: "zu Fuß"
  - Dafür hatten wir gerade das Beispiel

    $\lim_{n\to\infty} 1/n = 0$
- Variante 2: Regel von L'Hôpital
  - Seien $f, g : \mathbb{N} \to \mathbb{R}$ wie gehabt
  - Es existieren die ersten Ableitungen f' und g', sowie der Grenzwert $\lim_{n\to\infty} f'(n)/g'(n)$ ... dann gilt

    $\lim_{n\to\infty} f(n)/g(n) = \lim_{n\to\infty} f'(n)/g'(n)$
- Variante 3: göttliche Inspiration
  - Erst mit Promotion erlaubt ...

$f(n) = \ln n$

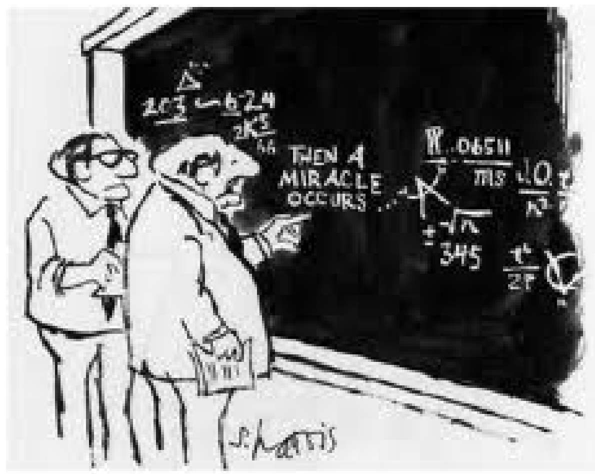
$g(n) = n$

$\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{\infty}{\infty} = ?$

$f'(n) = \frac{1}{n}, \quad g'(n) = 1$

$\frac{f'(n)}{g'(n)} = \frac{1}{n} \xrightarrow[n\to\infty]{} 0$



"I think you should be more explicit here in step two."

from *What's so Funny about Science?* by Sidney Harris (1977)

Rechnen mit Grenzwerten 2/2

- Was darf man ohne Beweis annehmen?
  - Gute Frage!
  - Da gibt es keine klare Regel
  - Im Zweifelsfall immer mehr beweisen als weniger !
  - **Beispiel 1:** $\lim_{n\to\infty} 1/n = 0$

    Brauchen Sie nicht mehr weiter zu beweisen
  - **Beispiel 2:** $\lim_{n\to\infty} 1/n^2 = 0$

    Einfach auf so was wie Beispiel 1 zurückführen
  - **Beispiel 3:** $\lim_{n\to\infty} (\log n)/n = 0$

    Hier ist ein Argument angebracht, z.B. mit L'Hôpital

O-Notation — Diskussion 1/2

- Sprechweise
  - Die O-Notation schaut sich das Verhalten der Funktionen an, wenn $n \to \infty$ geht (es interessieren nur die $n \geq n_0$)
  - Wenn man Laufzeiten o.ä. als $O(...)$, $\Omega(...)$, $\Theta(...)$, $o(...)$ oder $\omega(...)$ 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 das n_0 ist nicht immer so klein ... nächste Folie

O-Notation — Diskussion 2/2

- Beispiel

  und sogar $f = o(g)$ weil $\lim_{n\to\infty} \frac{f}{g} = 0$

  – Algorithmus A hat Laufzeit $f(n) = 80 \cdot n$

  – Algorithmus B hat Laufzeit $g(n) = 2 \cdot n \cdot \log_2 n$

  – Dann ist $f = O(g)$ aber **nicht** $f = \Theta(g)$

  – Das heißt, A ist asymptotisch schneller als B

  d.h. ab irgendeinem n_0 ist für alle $n \geq n_0$ $f(n) \leq g(n)$

  – Allerdings:

$$n_0 = 2^{40} \approx 1000^4 = 10^{12} = 1 \text{ Billionen (Tera)}$$

$$\Rightarrow \forall n < n_0 : g(n) = 2 \cdot n \cdot \underbrace{\log_2 n}_{< 40 \text{ für } n < 2^{40}}$$

$$< 80 \cdot n$$

$$= f(n)$$

Literatur / Links

- O-Notation / Ω-Notation / Θ-Notation
  - In Mehlhorn/Sanders:

    2.1 Asymptotic Notation
  - In Cormen/Leiserson/Rivest

    I 2.1 Asymptotic Notation
  - In Wikipedia

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

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