
Grundlagen: Algorithmen und Datenstrukturen

Abgabetermin: Jeweilige Tutorübung in der Woche vom 6. bis 10. Mai

Vereinbarung: Im Kontext der \mathcal{O} -Notation betrachten wir nur Funktionen von \mathbb{N}_0 nach \mathbb{R} . Daher vereinbaren wir, dass in diesem Kontext Ausdrücke wie $\exists n_0, \forall n_0$ und $\exists n \geq n_0$ sowie $\forall n \geq n_0$ zu lesen sind als $\exists n_0 \in \mathbb{N}, \forall n_0 \in \mathbb{N}$ und $\exists n \in \mathbb{N}, n \geq n_0$ sowie $\forall n \in \mathbb{N}, n \geq n_0$. Ausdrücke wie $\exists c > 0$ und $\forall c > 0$ bedeutet hingegen dasselbe wie $\exists c \in \mathbb{R}^+$ und $\forall c \in \mathbb{R}^+$. Diese Vereinbarung führt zu einer prägnanteren und einfacher zu lesenden Schreibweise.

Tutoraufgabe 1

Seien $f, g : \mathbb{N}_0 \rightarrow \mathbb{R}$ Funktionen mit $\exists n_0 \forall n > n_0 : f(n), g(n) > 0$. Beweisen Sie die folgenden Aussagen (a)–(e).

- (a) $f(n) \in \mathcal{O}(g(n)) \Leftrightarrow \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$
- (b) $f(n) \in \Omega(g(n)) \Leftrightarrow 0 < \liminf_{n \rightarrow \infty} \frac{f(n)}{g(n)}$
- (c) $f(n) \in o(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$
- (d) $f(n) \in \omega(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$
- (e) $f(n) \in \Theta(g(n)) \Leftrightarrow 0 < \liminf_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$

Beweisen Sie außerdem die Implikationen (f)–(h).

- (f) $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty \Rightarrow f(n) \in \mathcal{O}(g(n))$
- (g) $0 < \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \Rightarrow f(n) \in \Omega(g(n))$
- (h) $0 < \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty \Rightarrow f(n) \in \Theta(g(n)).$

Die Limes-Definition der \mathcal{O} -Notation ist ein sehr mächtiges Werkzeug, da durch ihre Verwendung für viele in der Praxis auftretenden Funktionen deren relative Beziehung bezüglich der \mathcal{O} -Notation sehr einfach ermittelt werden kann. Von besonderem Interesse

sind hierbei die Aussagen (c)–(d) und (f)–(h), da der Limes $\lim_{n \rightarrow \infty} f(n)/g(n)$ für gewöhnlich existiert. In folgendem (kurzen) Exkurs klären wir zunächst die Begrifflichkeiten. Zunächst halten wir fest, dass jede Funktion $f : \mathbb{N}_0 \rightarrow \mathbb{R}$ als *reellwertige unendliche Folge* $(f(0), f(1), f(2), \dots)$ geschrieben werden kann. Wir werden diese Identifikation im Folgenden stillschweigend verwenden.

Eine Zahl $L \in \mathbb{R}$ nennt man *Grenzwert* einer unendlichen reellwertige Folge $(a_n)_{n \in \mathbb{N}_0} = (a_0, a_1, \dots)$ genau dann, wenn $\forall \epsilon > 0 \exists n_0 \forall n \geq n_0 : a_n \in (L - \epsilon, L + \epsilon)$ erfüllt ist. In diesem Fall schreiben wir $\lim_{n \rightarrow \infty} a_n = L$. Intuitiv nähern sich die Zahlen der Folge dem Grenzwert also mit der Zeit immer weiter an.

Außerdem ist ∞ der (uneigentliche) Grenzwert von $(a_n)_{n \in \mathbb{N}_0}$ genau dann, wenn $\forall x \in \mathbb{R} \exists n_0 \forall n \geq n_0 : a_n > x$. Wir schreiben dann $\lim_{n \rightarrow \infty} a_n = \infty$. Intuitiv überschreitet die Folge dann jede noch so große Zahl irgendwann und bleibt dann über dieser. Analog ist $\lim_{n \rightarrow \infty} a_n = -\infty$ genau dann, wenn $\forall x \in \mathbb{R} \exists n_0 \forall n \geq n_0 : a_n < x$.

Eine Folge $(b_n)_{n \in \mathbb{N}_0}$ ist eine *Teilfolge* von $(a_n)_{n \in \mathbb{N}_0}$ genau dann, wenn $(b_n)_{n \in \mathbb{N}_0} = (a_{i_1}, a_{i_2}, \dots)$ für geeignet gewählte Indizes $i_j \in \mathbb{N}_0$ mit $i_j < i_{j+1}$ für alle $j \in \mathbb{N}$. (D.h., $(b_n)_{n \in \mathbb{N}_0}$ gewinnt man aus $(a_n)_{n \in \mathbb{N}_0}$ durch geeignete Auswahl von Folgengliedern unter Beibehaltung ihrer Anordnung.)

Eine Zahl $H \in \mathbb{R} \cup \{-\infty, +\infty\}$ ist ein *Häufungspunkt* von $(a_n)_{n \in \mathbb{N}_0}$ genau dann, wenn H der Grenzwert einer Teilfolge $(b_n)_{n \in \mathbb{N}_0}$ von $(a_n)_{n \in \mathbb{N}_0}$ ist. Intuitiv gibt es dann in $(a_n)_{n \in \mathbb{N}_0}$ unendlich viele Folgenglieder, die sich immer weiter H annähern.

Während eine unendliche reellwertige Folge nicht notwendigerweise einen Grenzwert haben muss (z.B. die Folge $(0, 1, 0, 2, 0, 3, \dots)$) hat jede solche Folge immer mindestens einen Häufungspunkt (in diesem Beispiel wären 0 und ∞ Häufungspunkte). Der größte bzw. kleinsten Häufungspunkt von $(a_n)_{n \in \mathbb{N}_0}$ (der ebenfalls immer existiert) nennen wir Limes superior bzw. Limes inferior von $(a_n)_{n \in \mathbb{N}_0}$ und schreiben ihn als $\limsup_{n \rightarrow \infty} a_n$ bzw. $\liminf_{n \rightarrow \infty} a_n$.

Tutoraufgabe 2

Ordnen Sie die folgenden Funktionen so an, dass für zwei in der Anordnung aufeinander folgende Funktionen f und g gilt: $f(n) \in o(g(n))$ oder $f(n) \in \mathcal{O}(g(n))$. Beweisen Sie Ihre Anordnung. Benutzen Sie gegebenenfalls die Monotonieeigenschaften elementarer Funktionen ohne Beweis.

$$\sqrt{\text{ld}(n+1)}, \quad 10^{100}, \quad \sqrt{n}, \quad \text{ld} \sqrt{n+1}$$

Anmerkung: Eine solche Anordnung von Funktionen bezüglich ihres asymptotischen Wachstums und der Landau-Symbole $\mathcal{O}, o, \Omega, \omega, \Theta$ ist im Allgemeinen nicht immer möglich, da zwei Funktionen f, g unvergleichbar sein können. Dieses Phänomen (das in der Praxis nur selten auftritt) ist Gegenstand des nächsten Übungsblattes.

Tutoraufgabe 3

Seien $f, g, h : \mathbb{N}_0 \rightarrow \mathbb{R}$ Funktionen mit $\exists n_0 \forall n > n_0 : f(n), g(n), h(n) > 0$. Zeigen Sie die folgenden Transitivitätsregeln.

- (a) $f(n) \in o(g(n)) \Rightarrow f(n) \in \mathcal{O}(g(n))$
- (b) $f(n) \in \mathcal{O}(g(n))$ und $g(n) \in \mathcal{O}(h(n)) \Rightarrow f(n) \in \mathcal{O}(h(n))$
- (c) $f(n) \in o(g(n))$ und $g(n) \in \mathcal{O}(h(n)) \Rightarrow f(n) \in o(h(n))$
- (d) $f(n) \in \mathcal{O}(g(n))$ und $g(n) \in o(h(n)) \Rightarrow f(n) \in o(h(n))$

Zusatzaufgabe 1

Seien $f, g, h : \mathbb{N}_0 \rightarrow \mathbb{R}$ Funktionen mit $\exists n_0 \in \mathbb{N} : \forall n > n_0 : f(n), g(n), h(n) > 0$. Zeigen Sie die folgenden Transitivitätsregeln.

- (a) $f(n) \in \omega(g(n)) \Rightarrow f(n) \in \Omega(g(n))$
- (b) $f(n) \in \Omega(g(n))$ und $g(n) \in \Omega(h(n)) \Rightarrow f(n) \in \Omega(h(n))$
- (c) $f(n) \in \omega(g(n))$ und $g(n) \in \Omega(h(n)) \Rightarrow f(n) \in \omega(h(n))$
- (d) $f(n) \in \Omega(g(n))$ und $g(n) \in \omega(h(n)) \Rightarrow f(n) \in \omega(h(n))$
- (e) $f(n) \in \Theta(g(n))$ und $g(n) \in \Theta(h(n)) \Rightarrow f(n) \in \Theta(h(n))$

Hausaufgabe 1

Betrachten Sie den Algorithmus 1 zur Bestimmung eines minimalen Elements in einem Integer-Array.

Algorithmus 1 : Min

```

Input : int A[], int n
1 int i = 1
2 int min = A[0]
3 while i < n do
4   | if A[i] < min then
5   |   | min = A[i]
6   |   i = i + 1
7 return min

```

Bestimmen Sie die exakte Worst-Case-Laufzeit des Algorithmus, indem Sie die exakte Anzahl der Rechenoperationen, wie sie in der Vorlesung definiert wurden (Vergleiche, Zuweisungen, Additionen, ...), in Abhängigkeit der Eingabegröße berechnen (keine asymptotische Abschätzungen durch Landau-Symbole).

Berechnen Sie nun auch die Worst-Case Laufzeit mit Hilfe der Landau-Symbole.

Hausaufgabe 2

In welcher Relation bezüglich der Landau-Symbole stehen die Funktionen $\text{ld } n = \log_2 n$ und $\ln n = \log_e n$? Beweisen Sie ihre Aussage.

Hausaufgabe 3

Seien $f, g : \mathbb{N}_0 \rightarrow \mathbb{R}$ Funktionen mit $\exists n_0 \in \mathbb{N} : \forall n \in \mathbb{N}, n > n_0 : f(n), g(n) > 0$.

- (a) Zeigen Sie $f(n) \in \Omega(g(n)) \Leftrightarrow g(n) \in \mathcal{O}(f(n))$ ohne Verwendung der in den Tutoraufgaben hergeleiteten Gesetze.
- (b) Zeigen Sie $f(n) \in \omega(g(n)) \Leftrightarrow g(n) \in o(f(n))$ ohne Verwendung der in den Tutoraufgaben hergeleiteten Gesetze.
- (c) Zeigen Sie $\max\{f(n), g(n)\} \in \Theta(f(n) + g(n))$.
- (d) Zeigen Sie, dass im Allgemeinen $\min\{f(n), g(n)\} \in \Theta(f(n) + g(n))$ **nicht** gilt, indem Sie geeignet gewählte Funktionen $f(n)$ und $g(n)$ als Gegenbeispiel verwenden.

Hausaufgabe 4

Ordnen Sie die folgenden Funktionen so an, dass für zwei in der Anordnung aufeinander folgende Funktionen f und g gilt: $f(n) \in o(g(n))$ oder $f(n) \in \mathcal{O}(g(n))$. Beweisen Sie Ihre Anordnung. Benutzen Sie gegebenenfalls die Monotonieeigenschaften elementarer Funktionen ohne Beweis.

$$(\text{ld } n)^{\text{ld } n}, \quad n^2, \quad 2^n, \quad n, \quad \sqrt{n}, \quad n^{\text{ld } n}, \quad n \log n$$