
Grundlagen: Algorithmen und Datenstrukturen

Abgabetermin: Jeweilige Tutorübung in der Woche vom 5. bis 9. Mai

Definition 1.

- $\mathbb{N} := \{1, 2, 3, \dots\}$,
- $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$,
- \mathbb{R} ist die Menge der reellen Zahlen,
- $\mathbb{R}^+ := \{x \in \mathbb{R} : x > 0\}$,
- $\mathbb{R}_0^+ := \mathbb{R}^+ \cup \{0\}$.

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.

Zur Erinnerung wiederholen wir die Definitionen der \mathcal{O} -Notation aus der Vorlesung.

Definition 2.

- $\mathcal{O}(f(n)) = \{g(n) \mid \exists c > 0 : \exists n_0 : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$
- $\Omega(f(n)) = \{g(n) \mid \exists c > 0 : \exists n_0 : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$
- $o(f(n)) = \{g(n) \mid \forall c > 0 : \exists n_0 : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$
- $\omega(f(n)) = \{g(n) \mid \forall c > 0 : \exists n_0 : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$
- $\Theta(f(n)) = \mathcal{O}(f(n)) \cap \Omega(f(n))$

Unter bestimmten Voraussetzungen kann die Beziehung zweier Funktionen bezüglich der \mathcal{O} -Notation mithilfe von Grenzwerten charakterisiert werden.

Lemma 1. *Seien $f, g : \mathbb{N}_0 \rightarrow \mathbb{R}$ Funktionen, die die folgenden beiden Eigenschaften erfüllen:*

- $\exists n_0 \forall n > n_0 : f(n), g(n) > 0$ und

- $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \in \mathbb{R}_0^+ \cup \{\infty\}$. (D.h., der Grenzwert $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$ existiert!)

Dann gelten die folgenden Äquivalenzen.

$$(a) \quad f(n) \in \mathcal{O}(g(n)) \quad \Leftrightarrow \quad 0 \leq \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$$

$$(b) \quad f(n) \in \Omega(g(n)) \quad \Leftrightarrow \quad 0 < \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq \infty$$

$$(c) \quad f(n) \in o(g(n)) \quad \Leftrightarrow \quad \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

$$(d) \quad f(n) \in \omega(g(n)) \quad \Leftrightarrow \quad \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$$

$$(e) \quad f(n) \in \Theta(g(n)) \quad \Leftrightarrow \quad 0 < \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$$

Für Funktionen, die in der Praxis auftreten, existiert der Limes $\lim_{n \rightarrow \infty} f(n)/g(n)$ für gewöhnlich, und kann oft vergleichsweise einfach berechnet werden. Daher ist diese Charakterisierung ein sehr mächtiges und hilfreiches Werkzeug. Die Aussagen (a)–(e) werden in einer Zusatzaufgabe bewiesen. Wir merken der Vollständigkeit halber an, dass für den Fall, dass der Limes $\lim_{n \rightarrow \infty} f(n)/g(n)$ nicht existiert, ähnliche Aussagen gezeigt werden können (welche wir im Rahmen dieser Übung aber nicht thematisieren wollen). Diese Aussagen verwenden statt des Limes den größten Häufungspunkt (limes superior) bzw. kleinsten Häufungspunkt (limes inferior) von $f(n)/g(n)$.

Tutoraufgabe 1

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.

$$(a) \quad f(n) \in o(g(n)) \quad \Rightarrow \quad f(n) \in \mathcal{O}(g(n))$$

$$(b) \quad f(n) \in \mathcal{O}(g(n)) \quad \text{und} \quad g(n) \in \mathcal{O}(h(n)) \quad \Rightarrow \quad f(n) \in \mathcal{O}(h(n))$$

$$(c) \quad f(n) \in o(g(n)) \quad \text{und} \quad g(n) \in \mathcal{O}(h(n)) \quad \Rightarrow \quad f(n) \in o(h(n))$$

$$(d) \quad f(n) \in \mathcal{O}(g(n)) \quad \text{und} \quad g(n) \in o(h(n)) \quad \Rightarrow \quad f(n) \in o(h(n))$$

Anmerkung: Diese Regeln sind beispielsweise in folgendem Szenario hilfreich: Angenommen, man möchte zeigen, dass $f(n) \in \mathcal{O}(h(n))$ gilt. Wenn f und h sehr komplizierte Funktionen sind, für die ein direkter Beweis schwierig ist, kann man einen Zwischenschritt verwenden: Man formuliert eine Funktion g , sodass $f(n) \in \mathcal{O}(g(n))$ und $g(n) \in \mathcal{O}(h(n))$ gilt, was über die Transitivitätsregeln $f(n) \in \mathcal{O}(h(n))$ impliziert. Hat man g geeignet gewählt (f und g sowie g und h unterscheiden sich nur geringfügig), dann kann der Nachweis von $f(n) \in \mathcal{O}(g(n))$ und $g(n) \in \mathcal{O}(h(n))$ deutlich einfacher sein als der direkte Beweis von $f(n) \in \mathcal{O}(h(n))$. (Natürlich kann man diese Methode auch iterativ anwenden und

statt einer einzelnen Zwischenfunktion auch mehrere Zwischenfunktionen g_1, g_2, \dots verwenden.) Analoge Transitivitätsregeln existieren auch für Ω und ω . Diese werden in einer Zusatzaufgabe formuliert und bewiesen.

Tutoraufgabe 2

Ordnen Sie die folgenden Funktionen so an, dass für zwei in der Anordnung aufeinanderfolgende Funktionen f und g gilt: $f(n) \in o(g(n))$ oder $f(n) \in \mathcal{O}(g(n))$. Beweisen Sie Ihre Anordnung. Verwenden Sie gegebenenfalls die Monotonie-, Konvergenz- und Unbeschränktheitseigenschaften elementarer Funktionen ohne Beweis.

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

Anmerkung: Haben wir eine solche Anordnung f_1, f_2, f_3, \dots gefunden, folgt aus den Transitivitätsregeln der vorherigen Tutoraufgabe natürlich unmittelbar, dass z.B. auch $f_1 \in \mathcal{O}(f_3)$ gilt. Weiterhin merken wir an, dass eine solche Anordnung von Funktionen bezüglich ihres asymptotischen Wachstums und der Landau-Symbole $\mathcal{O}, o, \Omega, \omega, \Theta$ im Allgemeinen nicht immer möglich, da es Fälle gibt, bei denen zwei Funktionen f, g unvergleichbar sein können. Dieses Phänomen (das in der Praxis nur selten auftritt) ist Gegenstand des nächsten Übungsblattes.

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) $f(n) \in \omega(g(n)) \quad \Rightarrow \quad f(n) \in \Omega(g(n))$
- (b) $f(n) \in \Omega(g(n)) \quad \text{und} \quad g(n) \in \Omega(h(n)) \quad \Rightarrow \quad f(n) \in \Omega(h(n))$
- (c) $f(n) \in \omega(g(n)) \quad \text{und} \quad g(n) \in \Omega(h(n)) \quad \Rightarrow \quad f(n) \in \omega(h(n))$
- (d) $f(n) \in \Omega(g(n)) \quad \text{und} \quad g(n) \in \omega(h(n)) \quad \Rightarrow \quad f(n) \in \omega(h(n))$
- (e) $f(n) \in \Theta(g(n)) \quad \text{und} \quad g(n) \in \Theta(h(n)) \quad \Rightarrow \quad f(n) \in \Theta(h(n))$

Zusatzaufgabe 2

Seien $f, g : \mathbb{N}_0 \rightarrow \mathbb{R}$ Funktionen, die die folgenden beiden Eigenschaften erfüllen:

- (i) $\exists n_0 \forall n > n_0 : f(n), g(n) > 0$ und
- (ii) $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \in \mathbb{R}_0^+ \cup \{\infty\}$. (D.h., der Grenzwert $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$ existiert!)

Beweisen Sie die folgenden Aussagen (a)–(e) für die Funktionen f und g .

$$(a) \quad f(n) \in \mathcal{O}(g(n)) \Leftrightarrow 0 \leq \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$$

$$(b) \quad f(n) \in \Omega(g(n)) \Leftrightarrow 0 < \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq \infty$$

$$(c) \quad f(n) \in o(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

$$(d) \quad f(n) \in \omega(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$$

$$(e) \quad f(n) \in \Theta(g(n)) \Leftrightarrow 0 < \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$$

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 $\text{ln } n = \log_e n$? 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 aufeinanderfolgende Funktionen f und g gilt: $f(n) \in o(g(n))$ oder $f(n) \in \mathcal{O}(g(n))$. Beweisen Sie Ihre Anordnung. Verwenden Sie gegebenenfalls die Monotonie-, Konvergenz- und Unbeschränktheitseigenschaften elementarer Funktionen ohne Beweis.

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