# Computerorientierte Mathematik
# ***Landau*** -Notation

15. Dezember 2015

Diese Definitionen und Erklärungen wurden aus dem Buch:
Th.H.Cormen | Ch.E.Leiserson | R.Rivest |C.Stein " Algorithmen - Eine Einführung"(ISBN: 978-3-486-58262) entnommen (Kapitel 3.1)

## Θ-Notation

Für eine gegebene Funktion $g$ bezeichnet $\Theta(g)$ die Menge der Funktionen:

$$\theta(g(n)) = \{f(n): \text{ es existieren positive Konstante } c_1, c_2 \text{ und } n_0 \text{ sodass } 0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \quad \forall n \geq n_0\}$$

Eine Funktion $f$ gehört zur Menge $\Theta(g)$, wenn positive Konstanten $c_1$ und $c_2$ existieren, sodass $f(n)$ zwischen $c_1 \cdot g(n)$ und $c_2 \cdot g(n)$ für ein hinreichend großes $n$ eingeschlossen werden kann.

## $\mathcal{O}$-Notation

Die Θ-Notation beschränkt eine Funktion asymptotisch von oben **und** unten. Wenn wir nur die ***obere asymptotische Schranke*** betrachten wollen, verwenden wir die $\mathcal{O}$-Notation. ($\mathcal{O} = O$)
Wir definieren analog zur Θ-Notation folgende Menge:

$$\mathcal{O}(g(n)) = \{f(n): \text{ es existieren positive Konstante } c \text{ und } n_0 \text{ sodass } 0 \leq f(n) \leq c \cdot g(n) \quad \forall n \geq n_0\}$$

Die $\mathcal{O}$-Notation kann verwendet werden die obere Schranke einer Funktion bis auf einen konstanten Faktor anzugeben.

### Bemerkung

Technisch gesehen ist es falsch zu sagen, dass die Laufzeit von Bubblesort in $\mathcal{O}(n^2)$ liegt, da die tatsächliche Laufzeit eines Algorithmus von der spezifischen Eingabegröße $n$ abhängt und damit variieren kann.
Wenn man nun sagt "Die Laufzeit ist in $\mathcal{O}(n^2)$"bedeutet das eigentlich:
Es gibt eine Funktion $f(n)$ in $\mathcal{O}(n^2)$, sodass für jeden Wert von $n$, egal wie die spezielle Eingabe der Größe $n$ aussieht, die Laufzeit für diese Eingabe von oben durch den Wert $f(n)$ beschränkt ist. Entsprechend meinen wir, dass die Laufzeit im schlechtesten Fall (*Worst-Case*) in $\mathcal{O}(n^2)$ liegt.

## *o*-Notation

Die von der $\mathcal{O}$-Notation definierte obere Schranke kann scharf sein, **muss** es aber **nicht**!
Die Schranke $2n^2 = \mathcal{O}(n^2)$ ist asymptotisch scharf, während $2n = \mathcal{O}(n^2)$ nicht ist.
Die $o$-Notation wir verwendet, um eine *nicht asymptotische scharfe* obere Schranke zu definieren.
Sei die Menge wie folgt definiert:

$$o(g(n)) = \{f(n): \text{ für jedes positive } c > 0 \text{ existiert ein konstantes } n_0 > 0 \text{ ,sodass } 0 \leq f(n) < c \cdot g(n) \quad \forall n \geq n_0\}$$

Die Definitionen der $\mathcal{O}$-Notation und der $o$-Notation sind sich sehr ähnlich. Der große Unterschied liegt aber darin, dass $f(n) = \mathcal{O}(g(n))$ die Schranke $0 \leq f(n) \leq c \cdot g(n)$ für (irgend)**eine** Konstante $c > 0$ verwendet. In der $o$-Notation wird dagegen die Schranke $0 \leq f(n) < c \cdot g(n)$ für **alle** Konstanten $c > 0$ verwendet. (Beachte jeweils $<$ in der $o$-Notation und $\leq$ $\mathcal{O}$-Notation).
Man kann beobachten, dass in der $o$-Notation die Funktion $f(n)$ gegenüber $g(n)$ unbedeutend wird, wenn $(n \to \infty)$.
Das führt zu der Grenzwertdefinition:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

### Beispiel:

Es gilt: $2n = 0(n^2)$ und $2n^2 \neq o(n^2)$