Landau-Symbole

Es ist nicht praktisch die Laufzeit von Algorithmen durch die Zeit in Sekunden anzugeben, die der Algorithmus auf einem speziellen Computer benötigt. Einerseits ist die Laufzeit auf anderen Systemen wahrscheinlich anders, andererseits ist man insb. daran interessiert, die Laufzeit in Abhängigkeit der Eingabegröße zu analysieren, um bspw. Algorithmen, die die selbe Aufgabe lösen, auf einfache Art und Weise miteinander vergleichen zu können.

Dazu versucht man zunächst die Laufzeit eines Algorithmus als Funktion der Eingabegröße zu beschreiben, d. h. man sucht eine Funktion

$$f : \mathbb{N} \to \mathbb{R}_+ ,$$

die in Abhängigkeit der Länge/ Größe der Eingabe n die Anzahl der Operationen (also der einzelnen Schritte), die der Algorithmus durchführt, berechnet ($\mathbb{R}_+$ steht hier für die positiven reellen Zahlen, d. h. $\mathbb{R}_+ = (0, \infty)$). Angenommen, man hat für einen Algorithmus eine solche Funktion gefunden, diese sei bspw. $f(n) = n^2 + 2n$. Ist man nun besonders daran interessiert, wie sich die Laufzeit für sehr große Eingaben verhält (also für sehr große n), so kann man abschätzen

$$f(n) \approx n^2 \quad \text{für } n \text{ sehr groß} .$$

Da man sehr häufig an solchen *asymptotischen* Aussagen interessiert ist, führt man nun Definitionen ein, die diese Abschätzungen etwas präzisieren, die sog. *Landau-Symbole*.

Wir definieren diese Symbole hier explizit als Menge von Funktionen, was in der Vorlesung nur implizit in der Definition steckt.

Die Klasse $\mathcal{O}(\cdot)$

Es sei $g : \mathbb{N} \to \mathbb{R}_+$ irgendeine Funktion. Dann definiert man

$$\mathcal{O}(g) := \{f : \mathbb{N} \to \mathbb{R}_+ \mid \exists c > 0, N_0 \in \mathbb{N} : \quad f(n) \leqslant c\, g(n) \quad \text{für alle } n \geqslant N_0\} .$$

Die Menge $\mathcal{O}(g)$ ("Groß O von g", engl. "Big O") enthält also alle Funktionen die durch die Funktion g "beschränkt" sind. Beschränkt ist wegen der Konstante c natürlich nicht ganz der richtige Begriff. Man sagt auch, die Funktionen in $\mathcal{O}(g)$ "wachsen ähnlich wie g" oder "wachsen nicht wesentlich schneller als g" oder "wachsen höchstens so schnell wie g". Ist $f : \mathbb{N} \to \mathbb{R}_+$ eine Funktion in $\mathcal{O}(g)$, so schreibt man

$$f \in \mathcal{O}(g) \qquad \text{oder} \qquad f = \mathcal{O}(g) ,$$

wobei das Gleichheitszeichen hier nicht als übliches "ist gleich" zu verstehen ist, sondern als "gehört zu der Menge".

Am einfachsten macht man sich die Definition an ein paar Beispielen klar:

- Sei $f(n) = n$. Dann ist $f \in \mathcal{O}(n^2)$. Denn: für alle $n \geqslant 2$ gilt

$$n \leqslant n^2 ,$$

  d. h. die Bedingung in der obigen Definition ist erfüllt mit $c = 1$ und $N_0 = 2$. Analog argumentiert man, dass $f \in \mathcal{O}(n^3)$, $f \in \mathcal{O}(n^4)$ etc. Manchmal schreibt man auch einfach $n \in \mathcal{O}(n^2)$ und versteht hier n^2 implizit als Funktion $f(n) = n^2$.

- Sei nun $f(n) = n^2$. Frage: Gilt $f \in \mathcal{O}(n)$? Um die Frage zu beantworten, setzen wir in die Definition ein und versuchen geeignete c und N_0 zu finden: Es müsste also (mit noch unbekannten c, N_0) gelten

$$n^2 \leqslant cn \quad \text{für alle } n \geqslant N_0 .$$

  Wenn wir durch n teilen, können wir die Ungleichung umformen zu

$$n \leqslant c \quad \text{für alle } n \geqslant N_0 .$$

  Das bedeutet, für festes c kann die Ungleichung immer nur für alle n, die kleiner als c sind, erfüllt sein. Wir können c zwar immer weiter erhöhen, für alle $n > c$ ist die Aussage dann aber immer falsch. Wir schließen daraus

$$n^2 \notin \mathcal{O}(n) .$$

- Sei nun $f \equiv k$ für eine konstante $k \in \mathbb{N}$, d h. f hat immer den Wert k. Frage: Gilt $f \in \mathcal{O}(n)$? Wir betrachten wieder die Definition: Angenommen es gibt c und N_0, sodass die Behauptung gilt. Dann

$$k \leqslant cn \quad \text{für alle } n \geqslant N_0 .$$

  Das ist bspw. erfüllt für $c = k$ und $N_0 = 1$, wir erhalten also $f \in \mathcal{O}(n)$. Natürlich folgt dann auch direkt $f \in \mathcal{O}(n^2)$, $f \in \mathcal{O}(n^3)$ etc.

- Oft kann man die Laufzeit eines Algorithmus als Polynom darstellen, also als Funktion der Form

$$p(n) = a_k n^k + a_{k-1} n^{k-1} + \cdots + a_1 n + a_0 .$$

  Ist k der höchste auftretende Exponent, dann sagt man p ist vom Grad k. Betrachten wir als Beispiel die Funktion $f(n) = 2n^2 + n - 1$. Frage: Liegt diese Funktion in $\mathcal{O}(n^2)$. Ja! Denn falls $n \geqslant 2$ ist, dann gilt die Ungleichung

$$2n^2 + n - 1 < 2n^2 + n < 2n^2 + n * n = 3n^2 ,$$

d. h. die Bedingung in der Definition ist erfüllt mit $c = 3$ und $N_0 = 2$. Dieses Beispiel lässt sich verallgemeinern. Betrachten wir allgemein Polynome vom Grad k wie bspw. $p(n)$ von oben, dann kann man zeigen (indem man einfach das n^k ausklammert), dass

$$p(n) \in \mathcal{O}(n^k)\,,$$

d. h. lässt sich die Laufzeit eines Algorithmus durch ein Polynom beschreiben, so hängt die asymptotische Laufzeit (also die Laufzeit, wenn n sehr groß wird) nur vom Grad des Polynoms ab!

Wir haben im dritten Beispiel schon gesehen, dass bspw. aus $f \in \mathcal{O}(n)$ schon folgt, dass auch $f \in \mathcal{O}(n^2)$ und $f \in \mathcal{O}(n^3)$ etc. Wenn man nun so etwas sagt, wie "Der Algorithmus f hat eine Laufzeit von $\mathcal{O}(n^2)$", dann meint man damit (meistens) implizit auch, dass $f \notin \mathcal{O}(n)$. Konkreter kann man solche Aussage machen, indem man weitere solcher Klassen einführt:

Die Klasse $\Omega(\cdot)$

Analog wie oben definiert man für eine Funktion $g : \mathbb{N} \to \mathbb{R}_+$ die Menge

$$\Omega(g) \coloneqq \{f : \mathbb{N} \to \mathbb{R}_+ \mid \exists c > 0, N_0 \in \mathbb{N} : \quad c\, f(n) \geqslant g(n) \quad \text{für alle } n \geqslant N_0\}\,.$$

Diese Klasse enthält also alle Funktionen, die mindestens so schnell wachsen wie g. Man sieht leicht ein, dass für zwei Funktionen $f, g : \mathbb{N} \to \mathbb{R}_+$ gilt, dass

$$f \in \mathcal{O}(g) \qquad \text{genau dann, wenn} \qquad g \in \Omega(f)\,,$$

da die Definitionen sich ja im Wesentlichen nur im Ungleichheitszeichen unterscheiden. Anhand der Beispiele oben, findet man also bspw.

- $n^2 \in \Omega(n)$ oder
- $n \in \Omega(k)$, für $k \in \mathbb{R}$ beliebig.

Man kann aber auch zeigen, dass für Polynome $p(n)$ vom Grad k gilt, dass

$$p \in \Omega(n^k)\,.$$

Es gilt also sowohl $p \in \mathcal{O}(n^k)$ als auch $p \in \Omega(n^k)$. Für solche Fälle führt man noch eine weitere Klasse ein.

Die Klasse $\Theta(\cdot)$

Mit den Definitionen der Klassen $\mathcal{O}(\cdot)$ und $\Omega(\cdot)$ als Mengen von Funktionen, kann man die Klasse $\Theta(g)$ für eine Funktion $g : \mathbb{N} \to \mathbb{R}_+$ einfach definieren als

$$\Theta(g) = \mathcal{O}(g) \cap \Omega(g) \,,$$

also alle Funktionen die sowohl in der Klasse $\mathcal{O}(g)$ als auch in der Klasse $\Omega(g)$ liegen. Man kann sich natürlich trotzdem eine explizite Definition wie für die anderen beiden Klassen hinschreiben (kann man ja mal als Übung versuchen).

Abschließende Bemerkungen

Es gibt noch weitere Klassen, die man alle auf ähnliche Art und Weise definieren kann. Die Klasse $\mathcal{O}(\cdot)$ ist jedoch die mit Abstand wichtigste und begegnet einem nicht nur in der Informatik, sondern bspw. auch in der Numerik oder der Zahlentheorie.

Man kann diese Klassen auch auf eine völlig andere Art und Weise definieren, in dem man den Quotienten zweier Funktionen betrachtet und dann den Grenzwert für $n \to \infty$ bildet. Mit dieser Definition erhält man in der Regel die selben Aussagen wie mit “unserer” Definition, sie sind jedoch nicht vollständig äquivalent und lassen sich nicht 1-zu-1 ineinander überführen (siehe auch Wikipedia-Artikel *Landau-Symbole* für mehr Infos).