St. Pölten University of Applied Sciences

/ informatik & security

Landau-Symbole (Komplexitätsklassen)

Theoretische Informatik

Dipl.-Ing. Hubert Schölnast, BSc

21. April 2023

Fachhochschule St. Pölten , Campus-Platz 1, 3100 St. Pölten , T: +43 (2742) 313 228, F: +43 (2742) 313 228-339, E: csc@fhstp.ac.at, I: www.fhstp.ac.at

Inhaltsverzeichnis

1 Landau-Symbole (Big O-Notation)

Laundau-Symbole werden verwendet, um Funktionen zu gruppieren und um diesen Gruppen individuelle Namen zu geben. Funktionen, die mit ähnlichen Raten wachsen, wenn ihr Argument größer und größer wird, werden zusammengefasst.

$O(f(x))$, $\Omega(f(x))$, $\mathcal{O}(f(x))$, $\Theta(f(x))$, $o(f(x))$, $\mathcal{o}(f(x))$ und $\omega(f(x))$ sind unterschiedliche Mengen von Funktionen, die später im Detail erklärt werden. Jede dieser Mengen ist ein Container, und innerhalb dieser Container befinden sich Funktionen. Für jeden Container gibt es eine einfache Vorbild-Funktion $f(x)$, die definiert, welche anderen Funktionen in dieser Menge enthalten sind.

$f(x) \in \mathcal{O}(g(x))$ bedeutet, dass die Funktion $f(x)$ eine der Funktionen in der Menge $\mathcal{O}(g(x))$ ist, die durch die Funktion $g(x)$ definiert ist.

Wenn Sie Landau-Symbole verwenden, werden Sie auch sehr oft diese Notation sehen:

$f(x) = \mathcal{O}(g(x))$

Technisch gesehen ist dies eine falsche Schreibweise, denn es würde bedeuten, dass die Funktion $f(x)$ gleich einer Menge von Funktionen ist, was keinen Sinn ergibt. Trotzdem ist es eine gebräuchliche Schreibweise, die jeder als Synonym für $f(x) \in \mathcal{O}(g(x))$ versteht.

Sprich: „f von x ist von der Ordnung g von x".

Es werden unterschiedliche Symbole verwendet ($O, \Omega, \mathcal{O}, \Theta$, usw.). Davon bedeuten manche Buchstaben dasselbe, manche aber nicht.

2 Big O („big oh" oder „big omicron")

Äquivalente Notationen (sie bedeuten alle dasselbe):

$$f(x) \in \mathcal{O}(g(x)) \qquad f(x) \in O(g(x))$$
$$f(x) = \mathcal{O}(g(x)) \qquad f(x) = O(g(x))$$

Formale Definition:

$$\exists k > 0\ \exists x_0 > 0\ \forall x > x_0\colon\ |f(x)| \leq k \cdot g(x)$$

Diese Definition ist wie folgt zu lesen:

$\exists k > 0$	Für mindestens einen (konstanten) Wert k, der größer als 0 ist, und
$\exists x_0 > 0$	für mindestens einen Wert x_0, der größer als 0 ist,
$\forall x > x_0$	gibt es immer einen Wert x, der größer als x_0 ist,
:	sodass
$\lvert f(x)\rvert \leq k \cdot g(x)$	der Absolutbetrag von $f(x)$ kleiner oder gleich ist wie das Produkt von k und $g(x)$.

Oder anders gesagt:
Wenn x größer als ein bestimmter Grenzwert x_0 ist, dann ist der Quotient$\frac{|f(x)|}{g(x)}$ immer kleiner als ein bestimmter konstanter Wert k.

2.1 Beispiel 1:

$f(x) = 5x^2 + 23x + 144$ $\qquad$ $g(x) = x^2$

x	$f(x)$	$g(x)$	$\frac{\lvert f(x)\rvert}{g(x)}$
1	172	1	172,00
2	210	4	52.50
3	258	9	28.67
4	316	16	19.75
5	384	25	15.36
6	462	36	12.83
7	550	49	11.22
8	648	64	10.13
9	756	81	9.33
10	874	100	8.74
11	1002	121	8.28
12	1140	144	7.92
13	1288	169	7.62
14	1446	196	7.38
15	1614	225	7.17
16	1792	256	7.00
17	1980	289	6,85
18	2178	324	6.72
19	2386	361	6.61
20	2604	400	6.51

$f(x) \in \mathcal{O}(g(x))$ trifft zu, denn:

	k			x_0	
Wenn Sie wählen:	$k = 7$	dann gilt	$\lvert f(x)\rvert \leq k \cdot g(x)$ für alle	$x > 16$	oder
wenn Sie wählen:	$k = 200$	dann gilt	$\lvert f(x)\rvert \leq k \cdot g(x)$ für alle	$x > 1$	oder
wenn Sie wählen:	$k = 5.1$	dann gilt	$\lvert f(x)\rvert \leq k \cdot g(x)$ für alle	$x > 237$	oder …

Es spielt keine Rolle, welches Paar von k und x_0 verwendet wird, um diese Beziehung wahr werden zu lassen. Wenn es mindestens ein solches Paar gibt, reicht dies bereits aus.

Daher gilt: $5x^2 + 23x + 144 \in \mathcal{O}(x^2)$

2.2 Beispiel 2:

$f(x) = 5x^2 + 23x + 144$ $\qquad g(x) = x^3$

Jetzt $g(x)$ wächst viel schneller als in Beispiel 1, und daher ist es viel einfacher, den Ausdruck $k \cdot g(x)$ größer als $|f(x)|$ zu machen.

Das bedeutet, dass auch das gilt: $5x^2 + 23x + 144 \in \mathcal{O}(x^3)$

2.3 Gegenbeispiel:

$f(x) = x^3 + 5x^2 + 23x + 144$

$g(x) = x^2$

x	$f(x)$	$g(x)$	$\frac{\lvert f(x)\rvert}{g(x)}$
1	173	1	173.000
2	218	4	54.500
3	285	9	31.667
4	380	16	23.750
5	509	25	20.360
6	678	36	18.833
7	893	49	18.224
7.75	1088	60	18.115
8	1160	64	18.125
9	1485	81	18.333
10	1874	100	18.740
20	10604	400	26.510
50	138794	2500	55.518
100	1052444	10000	105.244
200	8204744	40000	205.119
500	126261644	250000	505.047
1000	1005023144	1000000	1005.023
2000	8020046144	4000000	2005.012
5000	1.25125E+11	25000000	5005.005
10000	1.0005E+12	100000000	10005.002

Der Quotient $\frac{|f(x)|}{g(x)}$ nimmt zu Beginn ab, erreicht dann aber irgendwo in der Nähe von 7,75 ein Minimum und steigt danach an und wächst forthin über alle Schranken.

Das bedeutet: Egal wie groß Sie k wählen, es wird niemals ein dazu passendes x_0 geben, sodass dann die Ungleichung $|f(x)| \leq k \cdot g(x)$ für jedes $x > x_0$ wahr wird.

Und daher gilt:

$x^3 + 5x^2 + 23x + 144 \notin \mathcal{O}(x^2)$

2.4 Fazit:

$f(x) \in \mathcal{O}\big(g(x)\big)$ bedeutet, dass die Wachstumsrate der Funktion $f(x)$ kleiner oder gleich der Wachstumsrate von ist $g(x)$. Oder anders gesagt: $g(x)$ wächst genauso schnell oder sogar schneller als $f(x)$.

3 Big Θ („big theta“)

Äquivalente Notationen (sie bedeuten alle dasselbe):

$$f(x) \in \Theta(g(x))$$
$$f(x) = \Theta(g(x))$$

Formale Definition

$$\exists k_1 > 0\ \exists k_2 > 0\ \exists x_0 > 0\ \forall x > x_0\colon\ k_1 \cdot g(x) \leq |f(x)| \leq k_2 \cdot g(x)$$

Während es in Big O nur eine obere Grenze gab (die hier zu $k_2 \cdot g(x)$) wurde, haben wir in Big Theta jetzt auch eine untere Grenze $k_1 \cdot g(x)$, und beide Grenzen sind konstante Vielfache derselben Funktion $g(x)$.

Das heisst:

$5x^2 + 23x + 144 \ \in\ \mathcal{O}(x^3)$

Aber $5x^2 + 23x + 144 \ \notin\ \Theta(x^3)$

Big O und Big Theta werden Sie ziemlich oft sehen, weil sie die wichtigsten Symbole sind, aber es gibt auch einige andere, die weniger oft verwendet werden:

4 Big Ω („big omega“)

Äquivalente Notationen (sie bedeuten alle dasselbe):

$$f(x) \in \Omega(g(x))$$
$$f(x) = \Omega(g(x))$$

Formale Definition

$$\exists k > 0\ \exists x_0 > 0\ \forall x > x_0\colon\ |f(x)| \geq k \cdot g(x)$$

Big Omega definiert nur eine untere Grenze. Diese Klassifizierung wird in der Komplexitätstheorie sehr selten verwendet.

5 Kleine Symbole

Es gibt auch die Symbole *small o* und *small* ω („small Omega“). Sie sind ähnlich wie ihre großen Cousins definiert, aber restriktiver.

Die Pfeile in den folgenden Aussagen zeigen nur von links nach rechts, nicht in die andere Richtung.

$$f(x) \in o(g(x)) \implies f(x) \in \mathcal{O}(g(x))$$
$$f(x) \in \omega(g(x)) \implies f(x) \in \Omega(g(x))$$

Die kleinen Symbole werden in der Komplexitätstheorie nicht verwendet.

6 Einige wichtige Big-O-Komplexitätsklassen

$f(n) \in \mathcal{O}(1)$ „konstant“
$f(x)$ ist durch einen konstanten Wert begrenzt. Egal wie groß n wird, $f(n)$ wird niemals größer als dieser konstante Wert.
Wenn die Zeitkomplexität eines Algorithmus konstant ist, bedeutet dies, dass der Algorithmus immer innerhalb einer konstanten Zeit terminiert, egal wie groß seine Eingabe war. Beispiel: Teste, ob eine gegebene Dezimalzahl beliebiger Länge ein Vielfaches von 5 ist.

$f(n) \in \mathcal{O}(\log n)$ „logarithmisch“
$f(n)$ wächst ungefähr um einen konstanten Betrag, wenn n verdoppelt wird.
Beispiel: Führen Sie eine binäre Suche in einer sortierten Liste von n Elementen durch.

$f(n) \in \mathcal{O}(\sqrt{n}) = \mathcal{O}\left(n^{\frac{1}{2}}\right)$ „Quadratwurzel“
$f(n)$ verdoppelt sich wenn n mit 4 multipliziert wird.
Beispiel: Anzahl der Divisionen bei der Durchführung eines naiven Primzahltests für die Zahl n.

$f(n) \in \mathcal{O}(n^c),\ 0 < c < 1$ „gebrochene Potenz“
Verallgemeinerte Version der Quadratwurzel

$f(n) \in \mathcal{O}(n)$ „linear“
$f(n)$ verdoppelt sich, wenn n verdoppelt wird.
Beispiel: Suchen eines Elements in einer unsortierten Liste mit n Elementen.

$f(n) \in \mathcal{O}(n \log n)$ „superlinear “, „loglinear“, „n log n“
$f(n)$ wächst schneller als $\mathcal{O}(n)$, aber langsamer als $\mathcal{O}(n^c)$ für jedes $c > 1$
Beispiel: Merge Sort bei einer Liste mit n Elementen.

$f(n) \in \mathcal{O}(n^2)$ „quadratisch“
$f(n)$ multipliziert sich mit 4, wenn Sie n verdoppeln.
Beispiel: Bubble Sort für eine Liste von n Elementen.

$f(n) \in \mathcal{O}(n^c),\ c > 1$ „polynomiell“, „algebraisch“
Verallgemeinerte Version von quadratisch. Beachten Sie, dass die Zahl c keine ganze Zahl sein muss. Auch $f(n) \in \mathcal{O}(n^{1.0001})$ ist polynomiell und somit langsamer als $f(n) \in \mathcal{O}(n \log n)$).
Ein Problem, das mit einem Algorithmus gelöst werden kann, dessen Zeitkomplexität $\mathcal{O}(n^c)$ ist, wird oft als „einfaches“ Problem bezeichnet.

$f(n) \in \mathcal{O}(2^n) = \mathcal{O}(c^n)$ „exponentiell“
$f(n)$ verdoppelt sich (multipliziert mit einem konstanten Faktor), wenn Sie n um 1 erhöhen. Ein Problem, dessen schnellster Lösungsalgorithmus eine Zeitkomplexität von $\Omega(e^n)$ hat (beachten Sie das Omega!), wird oft als „schweres“ Problem bezeichnet.