Datenstrukturen & Algorithmen	2. März, 2016

Anmerkungen

Sebastian Millius, Sandro Feuz

Thema: Asymptotische Analyse, $\mathcal{O}, \Omega, \Theta$-Notation

Asymptotische Analyse (Anmerkungen, Ergänzungen)

Referenz

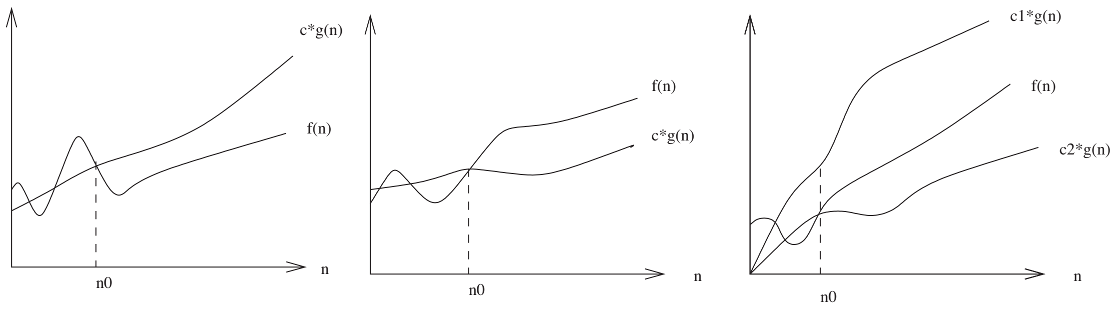
- Datenstrukturen und Algorithmen, Widmayer, Ottman, Kapitel 1.1, 1.2
- Introduction to Algorithms, Cormen et. al, Kapitel 3
- http://www.soe.ucsc.edu/classes/cmps102/Spring04/TantaloAsymp.pdf (http://goo.gl/Wnhly)

Die asymptotische Analyse erlaubt es Algorithmen *maschinenunabhängig* zu vergleichen, in dem man sich

1. auf den *Worst-Case* konzentriert: Welche Laufzeit garantiert der Algorithmus (im schlechtesten Fall)?
2. auf das *Wachstum* der Laufzeit konzentriert: Wie skaliert der Algorithmus mit der Eingabe?

Die *Worst-Case Komplexität/Laufzeit* $T(n)$ eines Algorithmus ist eine Funktion, definiert durch die maximale Anzahl Schritte auf einer Eingabe der Länge n. Um auszudrücken, dass eine Laufzeit $T(n)$ von einer gewissen Grössenordnung ist, definiert man folgende *Mengen von Funktionen*:

- $\mathcal{O}(g(n)) = \{f(n) : \exists c > 0, \exists n_0 : \forall n \geqslant n_0 : 0 \leqslant f(n) \leqslant cg(n)\}$
- $\Omega(g(n)) = \{f(n) : \exists c > 0, \exists n_0 : \forall n \geqslant n_0 : f(n) \geqslant cg(n)\}$
- $\Theta(g(n)) = \{f(n) : \exists c_1, c_2 > 0, \exists n_0 : \forall n \geqslant n_0 : c_1 g(n) \leqslant f(n) \leqslant c_2 g(n)\}$



(a) (b) (c)

Dies formalisiert, was wir unter *obere Schranke* ($\mathcal{O}$, a)), *untere Schranke* (Ω, b)) und *asymptotisch "tight"* (Θ, c)) verstehen. Man schreibt oft, dass $f(n) = \mathcal{O}(g(n))$, wobei eigentlich $f(n) \in \mathcal{O}(n)$ gemeint ist. Das Gleichheitszeichen ist daher nicht mehr symmetrisch. Beachte dazu beispielsweise, dass $n^2 = \mathcal{O}(n^3)$ aber $n^3 \neq \mathcal{O}(n^2)$.

Eine obere Schranke $g(n)$ für $T(n)$ bedeutet, dass $T(n)$ für genügend grosse n beschränkt durch ein konstantes Vielfaches von $g(n)$ ist. Auf eine analoge Weise sprecht man von einer unteren Schranke.

Beobachtung 1. *$f(n) = \mathcal{O}(g(n))$ genau dann wenn $g(n) = \Omega(f(n))$, d.h. ist $g(n)$ eine obere Schranke von $f(n)$, so ist $f(n)$ eine untere von $g(n)$*

Beweis. Ist $f(n) = \mathcal{O}(g(n))$, dann existieren positive Zahlen c, n_0 so dass $0 \leqslant f(n) \leqslant cg(n) \iff g(n) \geqslant \frac{1}{c}f(n)$. Definieren wir $c' := \frac{1}{c}$, so gilt: $\exists c', n_0 : \forall n \geqslant n_0 : g(n) \geqslant c'f(n) \iff g(n) = \Omega(f(n))$.

Ist $T(n)$ in $\mathcal{O}(g(n))$ und zugleich in $\Omega(g(n))$, dann ist in einer gewissen Weise die "richtige" Abschätzung gefunden: $T(n)$ wächst wie $g(n)$ innerhalb konstanter Faktoren (die Abschätzung ist *asymptotisch tight*).

Beobachtung 2. $f(n) = \Theta(g(n)) \iff f(n) = \mathcal{O}(g(n))$ *und* $f(n) = \Omega(g(n))$.

In der Notation, werden einerseits *konstante Faktoren* sowie Terme kleinerer Ordnung (low-order Terms) ignoriert. Also z.b. $\Theta(7n^3 + 5n - 8) \rightsquigarrow \Theta(n^3)$.

Intuitiv ist dies gerechtfertigt, da konstante Faktoren und low-order Terme bei grossen n nicht ins Gewicht fallen. Folgende Betrachtungen werden eine mathematische Rechtfertigung dafür geben.

Desweiteren gilt (Übung)

- Ist $c > 0$ eine positive Konstante, so ist $cf(n) = \Theta(f(n))$
- $f(n) = \Theta(g(n))$ genau dann wenn gilt $g(n) = \Theta(f(n))$

Wir können eine Analogie zwischen dem Vergleich von zwei Funktionen f, g und dem Vergleich zweier Zahlen a, b ziehen:

$$\begin{array}{lcl} f(n) \in \mathcal{O}(g(n)) & \sim & a \leqslant b \\ f(n) \in \Omega(g(n)) & \sim & a \geqslant b \\ f(n) \in \Theta(g(n)) & \sim & a = b \\ f(n) \in o(g(n)) & \sim & a < b \\ f(n) \in \omega(g(n)) & \sim & a > b \end{array}$$

Folgende Regeln gelten, deren Korrektheit einfach bewiesen werden kann, in dem man die Definitionen einsetzt und dann die konkreten Konstanten angibt, ähnlich wie im Beweis von Beobachtung 1.

- *Transitivität*: $f(n) = \Theta(g(n)), g(n) = \Theta(h(n)) \implies f(n) = \Theta(h(n))$
- ebenso für $\mathcal{O}$ und Ω.
- *Reflexivität*: $f(n) = \Theta(f(n)), f(n) = \mathcal{O}(f(n)), f(n) = \Omega(f(n))$.
- Es seien f, g, so dass $f(n) = \mathcal{O}(h(n))$ und $g(n) = \mathcal{O}(h(n))$, dann ist $f + g = \mathcal{O}(h)$

  $f = \mathcal{O}(g(n))$, bedeutet dass ein c, n_0 existiert. Für $g = \mathcal{O}(h(n))$ gibt es c', n'_0. Definiere $c'' = c + c'$ und $n''_0 = \max(n_0, n''_0)$. Dann gilt $\forall n \geqslant n''_0 : (f + g)(n) = f(n) + g(n) \leqslant ch(n) + c'h(n) = (c + c')h(n) = c''h(n)$ und deshalb $f + g = \mathcal{O}(h)$.
- Für ein *fixes* k seien $f_1 \ldots f_k$ and h, so dass $f_1 = \mathcal{O}(h), \ldots, f_k = \mathcal{O}(h)$, dann gilt: $f_1 + \ldots + f_k = \mathcal{O}(h)$

  Es ist wichtig, dass hier k fix ist, also nicht von n abhängig ist, ansonsten gilt die Aussage nicht mehr, da sich die impliziten Konstanten abhängig von n vergrössern (think about it!)
- Seien f, g, so dass $g = \mathcal{O}(f)$, dann ist $f + g = \Theta(f)$.
- Ist für alle n grösser als ein n_0, $f(n) \leqslant h(n)$ und $h(n) = \mathcal{O}(g(n))$, dann gilt $f(n) = \mathcal{O}(g(n))$. Also z.B. ist $n! = \mathcal{O}(n^n)$.
- Ist für alle n grösser als ein n_0, $h_1(n) \leqslant f(n) \leqslant h_2(n)$ und $h_1(n) = \Omega(g(n))$ und $h_2(n) = \mathcal{O}(g(n))$, dann gilt $f = \Theta(g(n))$.

Zu zeigen, dass eine Funktion $f(n) = \Theta(g(n))$, müssen also konkrete c_1, c_2, n_0 angegeben werden, die die Definition erfüllen.

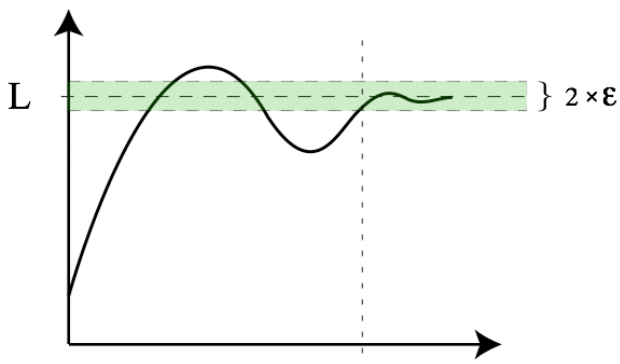
Beispiel: Zeige, dass $\sqrt{n + 10} = \Theta(\sqrt{n})$

Wähle z.B. $c_1 = 1, c_2 = \sqrt{2}, n_0 = 10$. Dann ist für alle $n \geqslant n_0 : c_1\sqrt{n} = \sqrt{n} \leqslant \sqrt{n+10} \leqslant \sqrt{n+n} = \sqrt{2}\sqrt{n}$, und deshalb $\sqrt{n+10} = \Theta(\sqrt{n})$.

Da dies vielfach aufwendig sein kann, eignen sich folgende Implikationen für den Vergleich von Funktionen:

Beobachtung 3. *Ist* $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = L$, *wobei* $0 \leqslant L < \infty$, *dann ist* $f(n) = \mathcal{O}(g(n))$

Beweis.



Es sei $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = L$. Aus der formalen Definition des Grenzwertes folgt, dass $\forall \varepsilon > 0\, \exists n_0 : \forall n \geqslant n_0 : \left|\frac{f(n)}{g(n)} - L\right| < \varepsilon$.

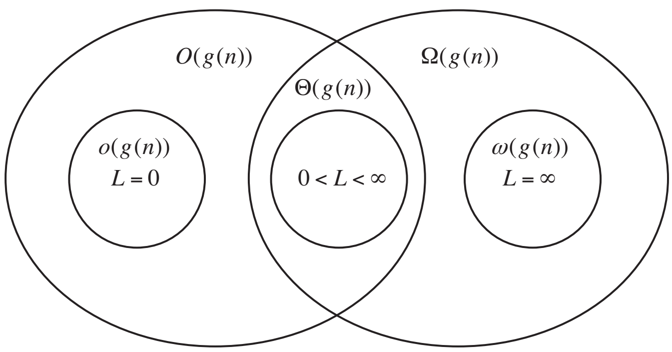
Wähle $\varepsilon := 1$, dann $\exists n_0 : \forall n \geqslant n_0 : \left|\frac{f(n)}{g(n)} - L\right| < 1$, also $-1 < \frac{f(n)}{g(n)} - L < 1$. Daraus schliessen wir, dass $f(n) < (L+1)g(n) = cg(n)$ für $c := (L+1)$ und deshalb ist $f(n) = \mathcal{O}(g(n))$.

Auf ähnliche Weise gilt

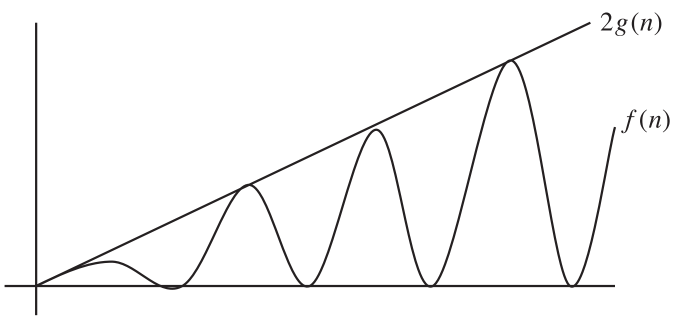
Beobachtung 4. *Ist* $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = L$, *wobei* $0 < L \leqslant \infty$, *dann ist* $f(n) = \Omega(g(n))$

Beobachtung 5. *Ist* $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = L$, *wobei* $0 < L < \infty$, *dann ist* $f(n) = \Theta(g(n))$

Wir erhalten also folgende Übersicht



Als Anmerkung ist zu sagen, dass Beobachtungen 3-5 wirklich Implikationen und keine Äquivalenzen sind. Betrachte dazu beispielsweise: $g(n) = n$ und $f(n) = (1 + \sin n)n$



Es gilt (wie oben im Bild auch eingezeichnet), dass $f(n) = \mathcal{O}(g(n))$, allerdings existiert der Grenzwert von $\frac{f(n)}{g(n)} = 1 + \sin n$ nicht. Man kann allerdings zeigen (vergleiche hierzu Wikipedia), dass

$$\limsup_{n\to\infty} \frac{f(n)}{g(n)} < c \leftrightarrow f(n) = \mathcal{O}(g(n))$$

Mit obigen Beobachtungen können wir nun folgende Wachstumsordnung rechtfertigen

$n^n >> n! >> c^n >> n^3 >> n^2 >> n^{1+\varepsilon} >> n\log n >> n >> \sqrt{n} >> \log^2 n >> \log n >> \frac{\log n}{\log\log n} >> \log\log n >> \alpha(n) >> 1$

Insbesondere gilt, und dies rechtfertigt das Ignorieren von low-order Termen, dass für ein Polynom von Grad k: $f(n) = \sum\limits_{i=0}^{n} a_i n^i = \Theta(n^k)$, da der Grenzwert von $\frac{\sum_{i=0}^{n} a_i n^i}{n^k} = a_k < \infty$ ist.

Desweitern gilt, dass

- Logarithmen wachsen langsamer als jedes Polynom, $\log n = \mathcal{O}(n)$
- Exponentialfunktionen wachsen schneller als jedes Polynom, $\forall c > 1 : n^k = \mathcal{O}(c^n)$

Dies kann wiederum dadurch gesehen werden, dass man den Grenzwert nimmt. Beachte auch, dass $2^{cn} \neq \Theta(2^n)$ (für $c \neq 1$), aber $2^{n+c} = \Theta(2^n)$.

In der Vorlesung wurde eine alternative Definition der Menge $\mathcal{O}(g(n))$ gesehen, nämlich, dass $f(n) = \mathcal{O}^{\text{Widmayer}}(g(n))$, falls c, d, n_0 existieren, so dass $\forall n \geqslant n_0 : f(n) \leqslant cg(n) + d$. Diese beiden Definitionen sind fast äquivalent, unterscheiden sich aber in der Frage, ob z.B. für eine Konstante $c = \mathcal{O}(\frac{1}{n})$ ist.

Es gilt dass $c \neq \mathcal{O}(\frac{1}{n})$, da $\lim\limits_{n\to\infty} \frac{c}{\frac{1}{n}} = \lim\limits_{n\to\infty} cn = \infty$. Andererseit ist $c = \mathcal{O}^{\text{Widmayer}}(\frac{1}{n})$, da man d entsprechend wählen kann (z.b. c).