Tabelle 10.2: Anschauliche Erklärung des Wachstumsverhaltens.

Bezeichnung	Sprechweise	Anschauliche Erklärung
$f = O(1)$	konstant	$f(n) \leq c$
$f = \Theta(\ln n)$	logarithmisches Wachstum	$f(2n) \approx f(n) + c$
$f = \Theta(\sqrt{n})$	Wachstum wie die Wurzelfunktion	$f(4n) \approx 2 \cdot f(n)$
$f = \Theta(n)$	lineares Wachstum	$f(2n) \approx 2 \cdot f(n)$
$f = \Theta(n^2)$	quadratisches Wachstum	$f(2n) \approx 4 \cdot f(n)$
$f = \Theta(2^n)$	exponentielles Wachstum	$f(n+1) \approx 2 \cdot f(n)$
$f = \Theta(n!)$	faktorielles Wachstum	$f(n+1) \approx n \cdot f(n)$

D. h. logarithmisches Wachstum ist unwesentlich gegenüber dem Wachstum von Polynomen und polynomielles Wachstum ist unwesentlich gegenüber dem Wachstum von Potenzen.

Um dies zu beweisen, beachten wir zunächst, dass aus $f = o(g)$ auch $f^k = o(g^k)$ für jede Konstante $k > 0$ folgt (Aufgabe 10.2). Es reicht also $\ln x = o(x^c)$ und $x = o(\mathrm{e}^{dx})$ für $c = a/B$ und $d = (a \ln 2)/B$ zu zeigen. Dafür reicht es die Regeln von Bernoulli–l'Hospital anzuwenden:

$$\lim_{x\to\infty} \frac{\ln x}{x^c} = \lim_{x\to\infty} \frac{\frac{1}{x}}{cx^{c-1}} = \frac{1}{c} \cdot \lim_{x\to\infty} \frac{1}{x^c} = 0\,;$$
$$\lim_{x\to\infty} \frac{x}{\mathrm{e}^{dx}} = \lim_{x\to\infty} \frac{1}{d\mathrm{e}^{dx}} = \frac{1}{d} \cdot \lim_{x\to\infty} \frac{1}{\mathrm{e}^{dx}} = 0\,.$$

Beim Wachstumsverhalten von Funktionen, wie zum Beispiel Laufzeiten von Algorithmen, haben sich die in Tabelle 10.2 angegebenen Sprechweisen eingebürgert. So bedeutet zum Beispiel $f = \Theta(\sqrt{n})$, dass $f(n)$ ungefähr auf das doppelte wächst, wenn sich das Argument vervierfacht.

10.6.1 Das Master Theorem

In der Analyse der Laufzeit von Algorithmen tauchen besonders oft die Folgen $z_n = T(n)$ auf, wobei $T : \mathbb{R} \to \mathbb{R}$ eine durch Rekursionsgleichungen der Form

$$T(x) = aT(x/b) + f(x) \quad \text{mit } a \geq 1, b > 1 \tag{10.8}$$

gegebene und $f(x)$ eine monoton wachsende Funktion ist: Aus $x' > x$ folgt $f(x') \geq f(x)$. Eine solche Rekursionsgleichung kann z. B. die Laufzeit eines Algorithmus beschreiben, der die Eingabe der Länge x in a Teilprobleme der Länge x/b zerlegt, diese durch a rekursive Aufrufe löst und aus den erhaltenen Teillösungen die Gesamtlösung zusammensetzt. Hier ist $f(x)$ die zusätzliche Zeit, die man benötigt, um das Problem zu zerlegen und aus den Teillösungen die Gesamtlösung zu berechnen.

In der Regel will man die Funktion $T(x)$ nicht notwendigerweise *exakt* zu bestimmen, es reicht in vielen Fällen, ihr *asymptotisches* Wachstum abzuschätzen. Deshalb sind die Randbedingungen nicht so wichtig. Einfachheitshalber nehmen wir an, dass $T(1) = f(1)$

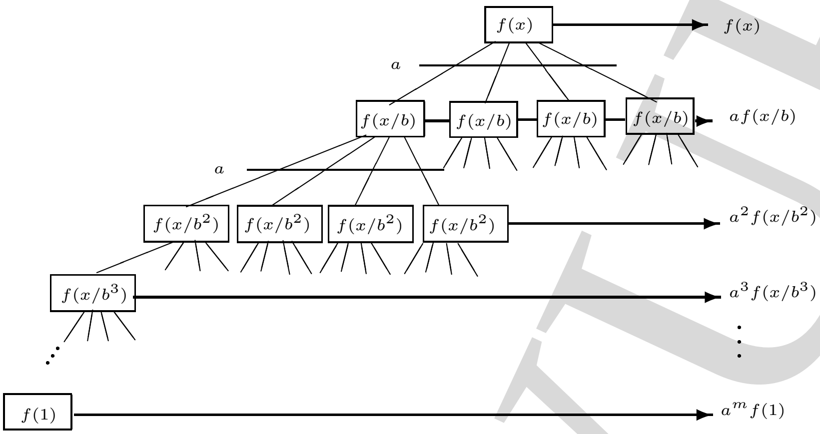

Bild 10.6: Der Rekursionsbaum für $T(x) = aT(x/b) + f(x)$.

und $T(x) = 0$ für $x < 1$ gilt. Wir entwickeln nun die Rekursion (10.8) und erhalten

$$\begin{aligned}
T(x) &= aT(x/b) + f(x) && \text{setze } x := x/b \text{ in (10.8)}\\
&= a\left(aT(x/b^2) + f(x/b)\right) + f(x)\\
&= a^2T(x/b^2) + af(x/b) + f(x) && \text{setze } x := x/b^2 \text{ in (10.8)}\\
&= a^2\left(aT(x/b^3) + f(x/b^2)\right) + af(x/b) + f(x)\\
&= a^3T(x/b^3) + a^2f(x/b^2) + af(x/b) + f(x)\\
&\dots\\
&= f(x) + af(x/b) + a^2f(x/b^2) + \cdots + a^m f(x/b^m)
\end{aligned}$$

mit $m = \log_b x$ (x sei eine Potenz von b). Diese Entwicklung der Rekursion kann man sich am besten als einen Rekursionsbaum vorstellen (siehe Bild 10.6). Das ist ein Wurzelbaum der Tiefe $m = \log_b x$, in dem jeder Knoten genau a Kinder hat. Die Wurzel trägt das Gewicht $f(x)$, deren a Kinder tragen jeweils das Gewicht $f(x/b)$, die a^2 Enkelkinder tragen jeweils das Gewicht $f(x/b^2)$, usw. Alle a^k Knoten auf der k-ten Ebene tragen also gemeinsam das Gewicht $a^k f(x/b^k)$ und die $a^m = a^{\log_b x}$ Blätter tragen jeweils das Gewicht $f(1)$. Das Gesamtgewicht aller Knoten ist dann genau der Wert $T(x)$.

Es gibt einen allgemeingültigen Satz – das sogenannte »Master Theorem« – der die meisten Rekursionsgleichungen der Form (10.8) auf einmal »meistert«. Wir veranschaulichen die Hauptidee dieses Satzes an den einfachsten Fällen. In der Literatur finden sich Versionen des Master Theorems, die weitere Fälle abdecken.

Satz 10.31:

Seien $a, b, K > 1$ Konstanten. Die Rekursionsgleichung $T(x) = a \cdot T(x/b) + f(x)$ mit $T(1) = f(1) = 1$ und $T(x) = 0$ für $x < 1$ kann man wie folgt lösen:

1. gilt $af(x/b) \le f(x)/K$, so folgt $T(x) = \Theta(f(x))$;
2. gilt $af(x/b) \ge K \cdot f(x)$, so folgt $T(x) = \Theta(x^{\log_b a})$;
3. gilt $af(x/b) = f(x)$, so folgt $T(x) = f(x)\log_b(bx)$.

Der dritte Fall tritt zum Beispiel dann ein, wenn $a = b^k$ und $f(x) = x^k$ gilt.

Beweis:

Wir benutzen die folgende einfache Beobachtung: Der Wert einer geometrischen Reihe

$$1 + c + c^2 + c^3 + \cdots + c^m = \frac{1 - c^{m+1}}{1 - c}$$

mit $c > 0$ und $c \neq 1$ kann durch ein ***konstantes Vielfaches*** des größten Terms abgeschätzt werden. Gilt $c < 1$, so ist 1 der größte Term und der Wert der Reihe ist $\Theta(1)$. Gilt $c > 1$, so ist c^m der größte Term und der Wert der Reihe ist dann $\Theta(c^m)$.

Gilt nun $af(x/b) \leq f(x)/K$ für alle x, so ist der erste Term $f(x)$ von

$$T(x) = f(x) + af(x/b) + a^2 f(x/b^2) + \cdots + a^m f(x/b^m)\,. \tag{10.9}$$

auch der größte. Warum? Da die Ungleichung $af(y/b) \leq f(y)/K$ auch für $y = x/b^r$ gelten muss, gilt auch

$$a^{r+1} f(x/b^{r+1}) = a^r\,[af(y/b)] \leq a^r f(y)/K = a^r f(x/b^r)/K$$

für alle $r = 0,1,\ldots,m-1$ und wir erhalten

$$f(x) \leq T(x) \leq f(x)\Big[1 + \frac{1}{K} + \frac{1}{K^2} + \cdots + \frac{1}{K^m}\Big]\,,$$

woraus $T(x) = \Theta(f(x))$ folgt. In diesem Fall ist also das Gewicht des Rekursionsbaumes an der Wurzel konzentriert.

Gilt $af(x/b) \geq K \cdot f(x)$ für alle x, so ist der letzte Term $a^m = a^{\log_b x} = x^{\log_b a}$ in (10.9) nach demselben Argument auch der größte und wir erhalten

$$a^m \leq T(x) \leq a^m\Big[\frac{1}{K^m} + \cdots + \frac{1}{K^2} + \frac{1}{K} + 1\Big]\,,$$

woraus $T(x) = \Theta(a^m) = \Theta(x^{\log_b a})$ folgt. In diesem Fall trägt also jeder innere Knoten des Rekursionsbaumes nur sehr wenig bei und das Gesamtgewicht $T(x)$ ist im Wesentlichen in den $a^m = x^{\log_b a}$ Blättern konzentriert.

Gilt nun $af(x/b) = f(x)$ für alle x, so sind alle $m + 1 = \log_b x + 1 = \log_b(bx)$ Terme gleich $f(x)$ und wir erhalten $T(x) = f(x)\log_b(bx)$. □

Beispiel 10.32:

Die folgenden drei »ähnlich« aussehenden Rekursionsgleichungen mit $f(x) = x^2$ haben verschiedene Lösungen:

$$\begin{array}{lcl@{\quad}c@{\quad}lcl}
T(x) & = & 8 \cdot T(x/3) + f(x) & \Rightarrow & T(x) & = & \Theta(x^2)\,;\\
T(x) & = & 10 \cdot T(x/3) + f(x) & \Rightarrow & T(x) & = & \Theta(x^{\log_3 10}) = \Theta(x^{2{,}09})\,;\\
T(x) & = & 9 \cdot T(x/3) + f(x) & \Rightarrow & T(x) & = & x^2 \log_3(3x)\,.
\end{array}$$

In dem ersten Fall gilt $8f(x/3) = \frac{8}{9}f(x) \leq f(x)/K$ mit $K = 9/8 > 1$. In dem zweiten Fall gilt $10f(x/3) = \frac{10}{9}f(x) \geq Kf(x)$ mit $K = 10/9 > 1$. In dem dritten