Der euklidische Algorithmus — Lösung

Die Definition des ggT

Der ggT von zwei ganzen Zahlen a und b ist die grösste natürliche Zahl t, die a und b ohne Rest teilt. Formal:

$$\mathrm{ggT}(a,b) = \max\{k \in \mathbb{N} \colon k \mid a \wedge k \mid b\}$$

Dabei bedeutet $k \mid a$ [sprich: „k teilt a“], dass k die ganze Zahl a ohne Rest teilt, d. h. dass es eine ganze Zahl m gibt, so dass $a = k \cdot m$ gilt. Umgekehrt bedeutet $k \nmid a$ [sprich: „k teilt a nicht“], dass die natürliche Zahl k die ganze Zahl a *nicht* teilt. Das Symbol „$\wedge$“ ist das logische UND und verlangt, dass k sowohl a als auch b ohne Rest teilt.

Die englische Bezeichnung für den ggT lautet *greatest common divisor* und wird entsprechend mit *gcd* abgekürzt.

Aufgaben

(a) $\mathrm{ggT}(12,18) = 6$

(b) $\mathrm{ggT}(18,12) = 6$

(c) $\mathrm{ggT}(-18,12) = 6$

(d) $\mathrm{ggT}(18,-12) = 6$

(e) $\mathrm{ggT}(-18,-12) = 6$

(f) $\mathrm{ggT}(31,24) = 1$

(g) $\mathrm{ggT}(1245,1) = 1$

(h) $\mathrm{ggT}(371,371) = 371$

(i) $\mathrm{ggT}(371,0) = 371$

Eigenschaften des ggTs

Die obigen Beispiele machen die unten formulierten Eigenschaften der Funktion $\mathrm{ggT}(a,b)$ plausibel. Insbesondere können wir uns wegen (c)–(e) darauf beschränken, dass a und b nicht negative ganze Zahlen sind.

(1) $\mathrm{ggT}(a,b) = \mathrm{ggT}(b,a)$

(2) $\mathrm{ggT}(a,a) = a$

(3) $\mathrm{ggT}(a,1) = 1$

(4) $\mathrm{ggT}(a,0) = a$

(5) $\mathrm{ggT}(a \cdot c, b \cdot c) = c \cdot ggt(a,b)$

Die Berechnung des ggTs durch Primfaktorzerlegung

Sind die Primfaktorzerlegungen der natürlichen Zahlen a und b bekannt, so ist der $\mathrm{ggT}(a, b)$ das Produkt aller Primfaktoren, die in *beiden* Zerlegungen vorkommen.

$$\begin{array}{l} a = 32\,340 = 2 \cdot 2 \quad \cdot 3 \cdot 5 \quad \cdot 7 \cdot 7 \cdot 11 \\ b = 18\,200 = 2 \cdot 2 \cdot 2 \quad \cdot 5 \cdot 5 \cdot 7 \quad\quad \cdot 13 \\ \hline \mathrm{ggT}(a, b) = 2 \cdot 2 \quad\quad \cdot 5 \cdot \quad 7 \quad\quad \cdot 13 = 1820 \end{array}$$

Da die Zerlegung einer natürlichen Zahl in ihre Primfaktoren eine sehr aufwändig Aufgabe ist (man muss im Grunde alle Primzahlteiler durchprobieren), ist diese Methode sehr ineffizient und nur für kleine Operanden sinnvoll anwendbar.

Der euklidische Algorithmus

Euklid von Alexandria war ein griechischer Mathematiker, der wahrscheinlich im 3. Jahrhundert v. Chr. in Alexandria gelebt hat. In seinem Werk *Die Elemente* hat er unter anderem beschrieben, wie man den grössten gemeinsamen Teiler zweier positiver ganzer Zahlen berechnet. Vermutlich stammt das Verfahren nicht von ihm, da Euklid in dem oben genannten Buch die Erkenntisse früherer Mathematiker zusammenfasste. Die Kern des Verfahrens beruht auf folgendem mathematischen Lehrsatz:

Ist die ganze Zahl k ein Teiler der ganzen Zahlen a und b, dann ist k auch ein Teiler von $a - b$.

Beweis: Da k ein Teiler von a ist, muss es eine ganze Zahl p geben, so dass $a = k \cdot p$. Da k auch ein Teiler von b ist, muss es eine ganze Zahl q geben, so dass $b = k \cdot q$.

Daraus folgt $a - b = k \cdot p - k \cdot q = k(p - q)$. Da k ein Teiler von $k(p - q)$ ist, muss k auch ein Teiler von $a - b$ sein, was zu beweisen war. $\square$

Der klassische Algorithmus von Euklid (Beispiel 1)

Wiederhole die folgenden Schritte in dieser Reihenfolge:

(1) Falls $b = 0$, gib a aus und stoppe den Algorithmus.

(2) Falls $a < b$, ersetze $\mathrm{ggT}(a, b)$ durch $\mathrm{ggT}(b, a)$.

(3) Ersetze $\mathrm{ggT}(a, b)$ durch $\mathrm{ggT}(a - b, b)$.

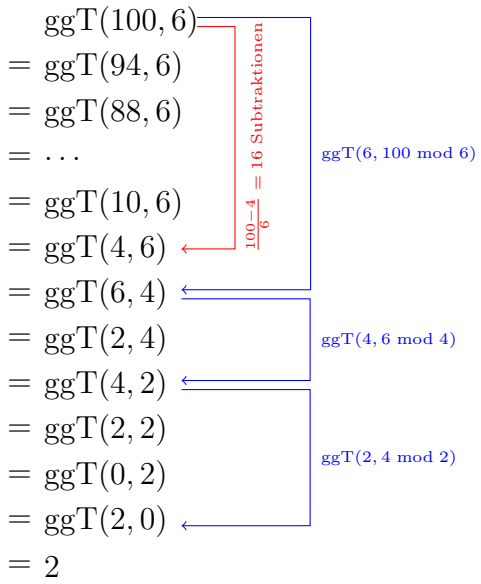
$$\begin{aligned} \mathrm{ggT}(27, 6) &\overset{(3)}{=} \mathrm{ggT}(21, 6) \overset{(3)}{=} \mathrm{ggT}(15, 6) \overset{(3)}{=} \mathrm{ggT}(9, 6) \\ &\overset{(3)}{=} \mathrm{ggT}(3, 6) \overset{(2)}{=} \mathrm{ggT}(6, 3) \overset{(3)}{=} \mathrm{ggT}(3, 3) \\ &\overset{(3)}{=} \mathrm{ggT}(0, 3) \overset{(2)}{=} \mathrm{ggT}(3, 0) \overset{(3)}{=} \mathrm{ggT}(3, 0) \\ &\overset{(1)}{=} 3 \end{aligned}$$

Pseudocode des klassischen euklidischen Algorithmus

```
EuklidClassic(a, b)
1  a ← |a|
2  b ← |b|
3  while b > 0
4        if b > a
5              tausche a und b
6        a ← a − b
7  return a
```

Ein Nachteil der Subtraktionsmethode

Problem: Viele Subtraktionen, wenn $a - b$ sehr gross ist.



Lösung: Subtraktionen durch $\mathrm{ggT}(a, b) \to \mathrm{ggT}(b, a \bmod b)$ ersetzen.

Die Divisionsrest-Methode (Beispiel 2)

Wiederhole die folgenden Schritte in dieser Reihenfolge:

(1) Falls $b = 0$, gib a aus und stoppe den Algorithmus.

(2) Berechne den Rest $r = a \bmod b$, ersetze a durch b und ersetze b durch r.

$$\mathrm{ggT}(27, 6) \overset{(2)}{=} \mathrm{ggT}(6, 27 \bmod 6) = \mathrm{ggT}(6, 3)$$
$$\overset{(2)}{=} \mathrm{ggT}(3, 6 \bmod 3) = \mathrm{ggT}(3, 0)$$
$$\overset{(1)}{=} 3$$

Vergleich der beiden Verfahren:

klassisch (Beispiel 1): 9 Schritte
modern (Beispiel 2): 3 Schritte (ohne Zwischenrechungen)

Pseudocode des „modernen“ euklidischen Algorithmus

```
EuklidModern(a, b)
1  a ← |a|
2  b ← |b|
3  while b > 0
4        r ← a mod b
5        a ← b
6        b ← r
7  return a
```

Der Worst Case

Grundsätzlich ist das moderne Verfahren mit dem Divisionsrest viel effizienter als das klassische Verfahren mit dem Subtrahieren und Tauschen. Für ganz bestimmte Werte von a und b muss das moderne Verfahren jedoch einen vergleichbaren Aufwand wie das klassische Verfahren betreiben. Dabei handelt es sich um jeweils zwei aufeinanderfolgende Werte der Fibonacci-Folge.

f_n: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, ...

klassisch	modern
$\mathrm{ggT}(8,5)$	$\mathrm{ggT}(8,5)$
$= \mathrm{ggT}(3,5) = \mathrm{ggT}(5,3)$	$= \mathrm{ggT}(5,3)$
$= \mathrm{ggT}(2,3) = \mathrm{ggT}(3,2)$	$= \mathrm{ggT}(3,2)$
$= \mathrm{ggT}(1,2) = \mathrm{ggT}(2,1)$	$= \mathrm{ggT}(2,1)$
$= \mathrm{ggT}(0,1) = \mathrm{ggT}(1,0)$	$= \mathrm{ggT}(1,0)$
$= 1$	$= 1$