# Automatische Teilbarkeitstests

Andreas Müller*

## 1 Wirklich brauchbare Teilbarkeitstests

Alfred Schreiber hat in seinem Artikel [1] die bekannten Teilbarkeitstests übersichtlich zusammengestellt und versucht, sie bezüglich Brauchbarkeit zu bewerten. Der Fokus lag dabei auf verallgemeinerten Quersummen, die gewöhnlich deutlich kleiner sind als die auf Teilbarkeit zu testende Zahl. Als Kriterium für die Brauchbarkeit wurde verwendet, dass die Quersumme keine allzu langen Blöcke von Ziffern verwenden soll, willkürlich wurden Blöcke mit drei Ziffern als gerade noch zumutbar angesehen. Eine solche Quersumme reduziert das Problem der Teilbarkeitsentscheidung auf arithmetische Operationen mit Zahlen bis 1000 und einen anschliessenden Teilbarkeitstest.

Wenn ein Benutzer zur zuverlässigen Benutzung eines solchen Tests in der Lage ist, ist er rechnerisch offenbar so gewandt, dass sich ein Test wahrscheinlich erübrigt. Und dem Besitzer eines modernen Rechenwerkzeugs stellt sich die Frage wohl erst gar nicht. Wir verschärfen das Brauchbarkeitskriterium daher radikal und verlangen, dass ein wirklich brauchbarer Teilbarkeitstest ganz ohne jegliche rechnerische Fähigkeiten anwendbar sein soll.

Diese Forderung ist bei den bekannten Tests leicht erfüllbar. Zur Bestimmung des Neunerrestes werden die möglichen Reste in einem Kreis angeordnet. Beginnend beim Rest 0 zählt man jetzt die einzelnen Ziffern der Zahl nacheinander ab, am Ende landet man auf dem Neunerrest der Zahl. Für die Bestimmung des Elferrestes geht man gleich vor, wechselt aber bei jeder Ziffer die Abzählrichtung. Für die Durchführung des Tests ist offenbar keine Rechnung erforderlich, Zählen genügt.

Für kleine Testteiler $d$ lässt sich in jedem $B$-adischen Zahlensystem ein solcher wirklich brauchbarer Test konstruieren. Wie in [1] sei $(c_0 \dots c_k)_B$ die $B$-adische

*Hochschule für Technik Rapperswil, Oberseestrasse 10, CH-8640 Rapperswil, Email: andreas.mueller@hsr.ch

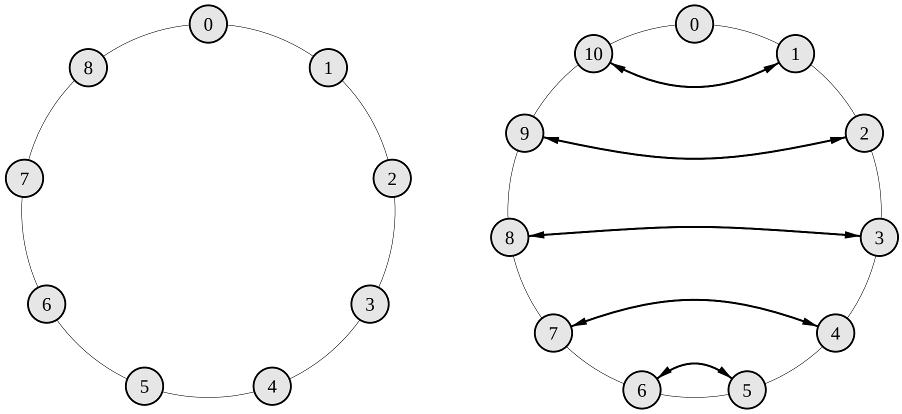


Abbildung 1: Die Testgraphen für die Tests auf Teilbarkeit durch 9 und 11 im Zehnersystem mit Hilfe der Quersumme (links) bzw. der alternierenden Quersumme (rechts).

Darstellung einer natürlichen Zahl $n$, also

$$(c_0 \dots c_k)_B = \sum_{i=0}^{k} c_{k-i} B^i = (\dots((c_0 B + c_1)B + c_2)B + \dots + c_{k-1})B + c_0 \qquad (1)$$

Der Test soll den Rest von $n$ im Restklassenring $R = \mathbb{Z}/d\mathbb{Z}$ berechnen. Statt eine verallgemeinerte Quersumme erst am Schluss des Tests modulo $d$ zu reduzieren, führen wir die Berechnung des Polynoms (1) bereits vollständig in $R$ durch. Dazu dient folgender Algorithmus:

1. Beginne beim Rest $0 \in R$.
2. Für jede Ziffer $c_i$, $0 \leq i \leq k$ führe folgende Schritte durch:
   (a) Multipliziere den aktuellen Rest mit $B$ (“Springen”).
   (b) Addiere (durch “Abzählen”) den Ziffernwert $c_i$.

Die bekannten Quersummentests sind deshalb so brauchbar, weil das “Springen” sehr einfach ist. Beim Neunerrest ist die Multiplikation mit der Basis 10 wegen $10 \equiv 1 \mod 9$ die identische Abbildung, beim Elferrest ist es wegen $10 \equiv -1 \mod 11$ die Multiplikation mit $-1$, also eine Spiegelung. Beides kann man sich leicht merken oder wie in Abbildung 1 visualisieren.

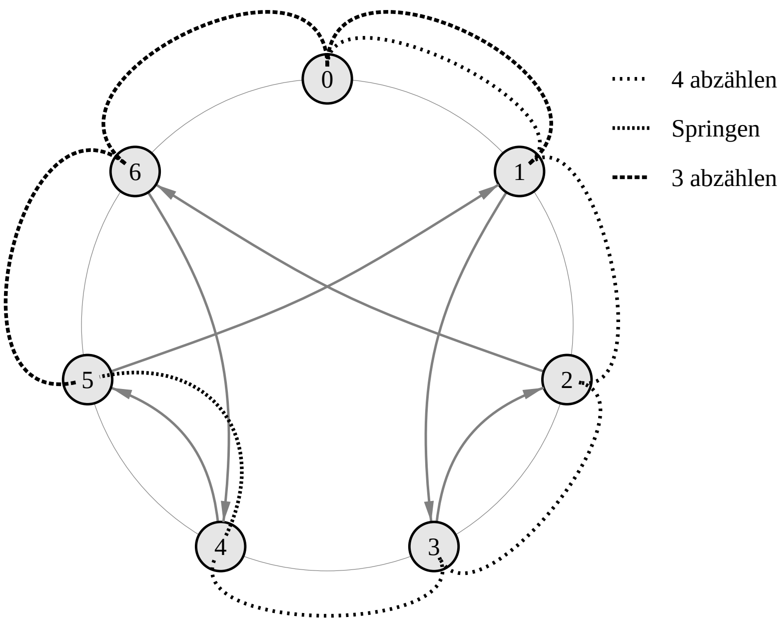


Abbildung 2: “Automatischer” Siebnerrest-Test mit Testgraph $T_{10}(7)$. Die punktierten Pfeile illustrieren die Berechnung des Siebnerrestes von 43.

# 2 Graphen und Automaten

Ein allgemeinerer Test wird dadurch brauchbar, dass das “Springen” jederzeit leicht durchführbar ist. Für den Teiler 7 ist es im Zehnersystem eine Multiplikation mit 3 mit anschliessender Reduktion modulo 7. Dies ist zwar immer noch sehr einfach im Vergleich zu einer alternierenden Quersumme mit Blöcken der Länge 3 wie in [1] vorgeschlagen, aber es entspricht noch nicht unserem Kriterium eines wirklich brauchbaren Tests. Stellt man die Sprungoperationen jedoch als Pfeile zwischen den verschiedenen Resten modulo $d$ wie in Abbildung 2 dar, wird der Algorithmus besonders einfach:

1. Beginne beim Rest 0.
2. Für jede Ziffer $c_i$, $0 \leq i \leq k$ führe folgende Schritte durch:
   (a) Falls vom aktuellen Rest ein Pfeil ausgeht, folge ihm.
   (b) Zähle vom aktuellen Rest $c_i$ Reste ab.

Der Algorithmus endet auf dem Rest modulo $d$ von $n$. Der Algorithmus ist so formuliert, dass alle Pfeile weglassen werden können, die zu Fixpunkten der Multiplikationsabbildung gehören. Wir bezeichnen den so entstehenden gerichteten

Graphen, der Teilbarkeit durch $d$ im $B$-adischen System zu testen erlaubt, mit $T_B(d)$. Die Übersicht der Diagramme auf http://www.othello.ch/teilbar/ hat unabhängig von der darin enthaltenen Mathematik auch rein graphisch ihren Reiz.

Die Bestimmung des Restes modulo $d$ einer natürlichen Zahl ist durch ziffernweise Verarbeitung in $R$ möglich. Das "Springen" entspricht dabei dem Hinzufügen einer 0, das Abzählen dem Erhöhen der 0 auf den tatsächlichen Wert der Ziffer. Man kann diese zwei Operationen auch in einen einzigen Pfeil zusammenfassen. So kann man von jedem Rest $r$ aus für jeden möglichen Ziffernwert $c$ einen mit $c$ beschrifteten Pfeil konstruieren, der zum Rest $Br+c \in R$ führt. Dies definiert einen deterministischen endlichen Automaten über dem Alphabet der möglichen Ziffern $0, \dots, B-1$ mit den Resten $R$ als Zuständen. Bezeichnet man den Rest $r$ als Endzustand, wird der Automat genau diejenigen Ziffernfolgen akzeptieren, die im $B$-adischen System einer Zahl mit dem Rest $r$ entsprechen.

# 3 Varianten

In der vorgestellten Form des Tests werden die Zahlen von links nach rechts gelesen. Wenn $B$ und $d$ teilerfremd sind, dann ist $B$ in $R$ invertierbar, und man kann den Test mit Hilfe der Formel

$$\begin{aligned}(c_0 \dots c_k)_B &= B^k \sum_{i=0}^{k} c_i B^{-i} \\ &= B^k((\dots((c_k B^{-1} + c_{k-1})B^{-1} + c_{k-2})B^{-1} + \dots + c_1)B^{-1} + c_0)\end{aligned}$$

in einen Test umwandeln, der die Zahl von rechts her liest. Die Operation "Springen" entspricht jetzt der Multiplikation mit $B^{-1}$, dazu müssen die Pfeile in umgekehrter Richtung durchlaufen werden. Der Vorfaktor $B^k$ ändert natürlich die Teilbarkeit nicht, aber der vom Test angezeigte Schlusswert ist jetzt der mit $B^{-k}$ multiplizierte Rest. Um den tatsächlichen Rest zu erhalten, muss also noch mit $B^k$ multipliziert werden. Dazu folgt man $k$ Mal den Pfeilen in normaler Richtung.

Ist der Testteiler $d$ prim, dann erhält man durch Verfolgen der Pfeile ausgehend von 1 die von $B$ in $R^*$ erzeugte zyklische Gruppe. Natürlich ist dafür nur die Restklasse von $B$ in $R$ relevant. Die von $B$ erzeugte Untergruppe braucht dabei nicht ganz $R^*$ zu sein, die von $B$ erzeugte Untergruppe in der multiplikativen Gruppe $R^*$ ist die Zusammenhangskomponente des Restes 1 modulo $d$ im Graphen (siehe Abbildung 3).

In einigen Fällen lässt sich die Teilbarkeit aufgrund der letzten ein oder zwei Stellen der Zahl entscheiden. In den Testgraphen äussert sich dies dadurch, dass der Graph eine Baumstruktur hat. Lässt sich die Teilbarkeit allein auf Grund der letzten Ziffer entscheiden, hat der Baum die Tiefe 1, von jedem Rest "springt"

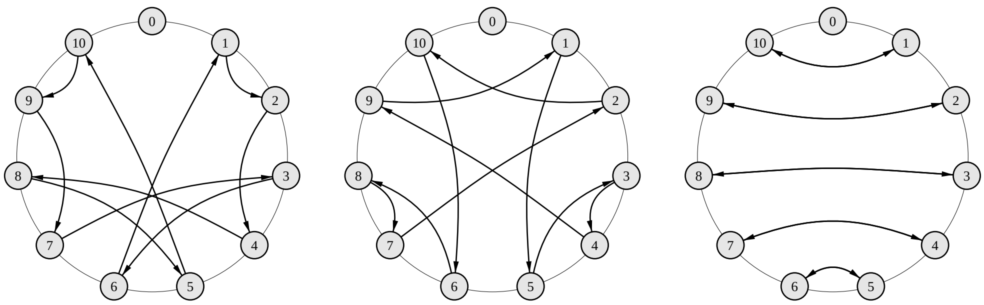


Abbildung 3: Von den Elementen 2 (links), 5 und 10 (rechts) erzeugte Untergruppen in $\mathbb{Z}/11\mathbb{Z}^*$. 2 erzeugt die ganze Gruppe, 5 erzeugt eine Untergruppe der Ordnung 5, 10 eine solche der Ordnung 2.

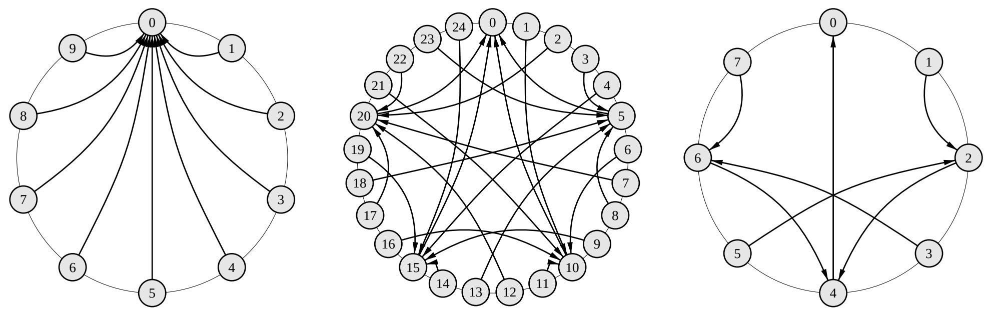


Abbildung 4: Testdiagramme mit Baumstruktur zeigen an, dass Teilbarkeit anhand der Endziffern entschieden werden kann. Von links nach rechts: $T_{10}(10)$ (eine Ziffer, Baumtiefe 1), $T_{10}(25)$ (zwei Ziffern, Baumtiefe 2), $T_{10}(8)$ (drei Ziffern, Baumtiefe 3).

man zum Rest 0, man vergisst also beim Springen immer wieder, was bisher für Ziffern gesehen wurden (siehe Abbildung 4). Testeigenschaften werden damit auf Eigenschaften des Testgraphen zurückgeführt. So erkennt man den in [1] erwähnten Test auf Teilbarkeit durch 7 mit einer alternierenden Quersumme mit Dreierblöcken daran, dass in $T_{10}(7)$ der Reihe nach die Reste 1, 3, 2, −1, −3 und −2 durchlaufen werden.

# 4 Eine Anwendung des Pumping Lemma

Die Theorie der endlichen Automaten hält noch weitere Überraschungen bereit. Das Pumping Lemma [2, p. 78] für einen endlichen Automaten mit $N$ Zuständen besagt, dass jedes Wort $x$, welches länger ist als $N$ Zeichen ist, in drei Teile $x = uvw$ aufgeteilt werden kann mit $|uv| \le N$, so dass $uv^k w$ für alle $k \ge 0$ vom Automaten akzeptiert wird. Es ist dies eine Konsequenz der Tatsache, dass dem Wort $x$ ein Pfad im Graphen des Automaten entspricht. Da der Pfad mehr als $N$ Schritte hat, muss sich innerhalb der ersten $N$ Schritte mindestens ein Zustand wiederholen, der Pfad durchläuft nach einem Anfangsstück $u$ eine Schleife $v$. Die Schleife kann beliebig oft wiederholt oder ganz weggelassen werden, alle resultierenden Wörter werden vom Automaten akzeptiert werden.

In unserem Fall verspricht das Pumping Lemma, dass jede Zahl mit mehr als $d$ Stellen innerhalb der ersten $d$ Stellen eine Folge von Ziffern enthält, die ohne Änderung des Rests weggelassen werden kann. Eine solche Ziffernfolge kann mit Hilfe des Testgraphen leicht identifiziert werden. Da es nur $d$ verschiedene Reste gibt, muss sich spätestens nach $d$ Schritten des Testalgorithmus ein Rest $r$ wiederholen. Die Ziffernfolge zwischen dem ersten und zweiten Vorkommen des Rests $r$ kann weggelassen werden. Beispielsweise durchläuft der Algorithmus bei der Analyse der Zahl 123456789 auf Teilbarkeit durch 7 die Reste

$$0 \xrightarrow{1} 1 \xrightarrow{2} 5 \xrightarrow{3} 4 \xrightarrow{4} 2 \xrightarrow{5} 4 \xrightarrow{6} 4 \xrightarrow{7} 5 \xrightarrow{8} 2 \xrightarrow{9} 1.$$

Die vom Pumping Lemma garantierte entfernbare Ziffernfolge ist jene zwischen den ersten zwei Vorkommen des Restes 4, also der Teil 45, insbesondere hat 1236789 den gleichen Siebnerrest. Man könnte aber auch die Ziffern zwischen den beiden Resten 5 entfernen, somit hat auch 1289 den gleichen Rest bei Teilung durch 7. In der Tat haben beide den Rest 1.

Diese Beispiele zeigen, dass Teilbarkeitstests auf der Basis der Ziffernfolge einer Zahl machbar sind, die keinerlei Rechenfähigkeit erfordern. Es ist bekannt, und wird zum Beispiel in [2] gezeigt, dass die von dem oben beschriebenen endlichen Automaten akzeptierten Wörter, also die Zahlen mit einem gegebenen Rest modulo $d$, auch von einem regulären Ausdruck erkannt werden können. Somit ist

Teilbarkeit etwas, was durch reine Mustererkennung auf den Ziffernfolgen erkannt werden kann, es ist also nicht einmal ein Verständnis dafür erforderlich, was die Ziffern bedeuten.

# Literatur


[1] Alfred Schreiber, *$B$-adische Teilbarkeitstests im Vergleich*, Elem. Math. 64 (2009) 13–20

[2] Michael Sipser, *Introduction to the Theory of Computation*, 2nd edition, Course Technology, 2005.