

Automat zur Prüfung der ganzzahligen Teilbarkeit einer Dualzahl durch drei

Karsten Brodmann

15. Februar 2017

1 Aufgabenstellung

Schreibe ein Programm, welches eine beliebig lange Dualzahl einliest und bestimmt, ob diese Zahl ganzzahlig durch 3 teilbar ist.

2 Lösung

Der erste, naive Gedanke zur Lösung könnte sein:

- bestimme den Wert eingegebenen Dualzahl,
- berechne den Rest einer ganzzahligen Division durch 3,
- ist der Rest 0, ist die eingegebene Dualzahl ganzzahlig durch 3 teilbar.

Dieser Gedanke ist insoweit nicht zielführend, als alle im Computer darstellbaren Zahlen in ihrem zulässigen Wertebereich eingeschränkt sind. Bei den meisten aktuellen Programmiersprachen sind die größten zulässigen Ganzzahlen als 64-Bit-Werte implementiert. Die Bedingung einer beliebigen Länge des Eingabewertes würde somit verletzt.

w	$w0$	$w1$
$3x + 0$ Rest 0	$6x + 0$ Rest 0	$6x + 1$ Rest 1
$3x + 1$ Rest 1	$6x + 2$ Rest 2	$6x + 3$ Rest 0
$3x + 2$ Rest 2	$6x + 4$ Rest 1	$6x + 5$ Rest 2

Tabelle 1: Zustände bei der ganzzahligen Division durch 3

Eine ganzzahlige Division durch 3 ergibt entweder 0, 1 oder 2 als Rest. Nehmen wir an, wir haben eine Dualzahl, deren Rest bei der ganzzahligen Division durch 3 bekannt ist. Was können wir dann über die Teilbarkeit aussagen, wenn wir unsere Dualzahl um ein Bit (0 oder 1) erweitern, also auf der rechten Seite anfügen?

Betrachten wir Tabelle 1. Augenscheinlich lässt sich bei Kenntnis des aktuellen Restes auf den sich ergebenden Rest schließen, wenn die Dualzahl um ein Bit erweitert wird.

Wir haben also 3 zu unterscheidende Zustände, klare Übergänge beim Ergänzen von Bits, welche mit 0 und 1 die einzig zulässigen Eingabezeichen sind. – Das riecht nach einem endlichen Automaten. Ein endlicher Automat A ist ein 5-Tupel $A = (S, \Sigma, \delta, s_0, F)$ mit

$$\begin{aligned} S &= \text{endliche Zustandsmenge} \\ \Sigma &= \text{endliches Eingabealphabet} \\ \delta : S \times \Sigma &\rightarrow S = \text{Überföhrungsfunktion} \\ s_0 \in S &= \text{Startzustand} \\ F \subseteq S &= \text{Menge der Endzustände} \end{aligned}$$

Ein Wort $w \in \Sigma^*$ wird akzeptiert, falls

$$\begin{aligned} \delta^*(s_0, w) &\in F \\ \delta^*(s_0, x_0x_1x_2 \dots x_{n-1}) &= \delta(\dots \delta(\delta(\delta(s_0, x_0), x_1), x_2) \dots x_{n-1}) \end{aligned}$$

Die Wirkungsweise der Überföhrungsfunktion kann durch Zustandsüberföhrungsgraphen beschrieben werden. In diesem Graphen föhrt eine mit w beschriftete Kante von Knoten s_m zu Knoten s_n , wenn der Wert w des Eingabealphabets vom Zustand s_m zum Zustand s_n föhrt.

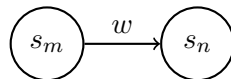


Abbildung 1: Graphische Darstellung einer Überföhrungsfunktion

Auf unsere Aufgabenstellung angewendet: A soll durch 3 teilbare Dualzahlen erkennen. Hierzu spendieren wir drei Zustände mit der Bedeutung „Rest 0“, „Rest 1“ und „Rest 2“. Wenn ein String w , dessen Rest als 0, 1 oder 2 bekannt ist, mit den Zeichen 0 oder 1 verlängert wird, ergeben sich die aus Tabelle 1 bekannten Zustände. Daraus ergibt sich folgender Automat:

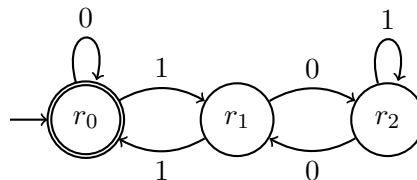


Abbildung 2: Überföhrungsfunktion δ des Automaten

Startzustand ist r_0 . Dazu gehört ein Zustandsüberführungsgraph mit den Knoten r_0 , r_1 und r_2 , welche die Zustände beschreiben, wenn für den bislang eingelesenen String die Division durch 3 den Rest 0, 1 oder 2 ergibt. An der Kante steht das jeweils nächste Bit der Dualzahl, die von links nach rechts abgearbeitet wird. Wenn wir diesen Automaten nun mit einer beliebig langen Folge von Bits füttern, nimmt er seinen jeweils aktuellen Zustand und wendet auf das nächste Bit die Überföhrungsfunktion an, die in Tabelle 1 ablesbar ist. Das Ergebnis ist wieder ein Zustand. Dieses Vorgehen wiederholt der Automat bis alle Eingabezeichen abgearbeitet sind.

Zwar hat der Automat keine Kenntnis über den wahren Wert der eingegebenen Dualzahl, er braucht diese Kenntnis aber auch nicht. Klein aber fleißig prüft er die ganzzahlige Teilbarkeit durch 3.

```

1  import AlgoTools.IO;
2  /**
3   * Endlicher Automat zur Überprüfung, ob eine beliebig
4   * lange Dualzahl durch 3 teilbar ist.
5   */
6  public class Automat {
7
8      public static void main(String[] args) {
9
10         int[][] delta = {                // Überföhrungsfunktion
11             {0, 1},
12             {2, 0},
13             {1, 2}
14         };
15         int s = 0;                        // Startzustand
16         char[] zeile = IO.readChars(); // Eingabezeichen
17
18         // Überföhrungsfunktion zeichenweise anwenden
19         for (int i = 0; i < zeile.length; i++)
20             s = delta[s][zeile[i] - '0'];
21
22         // Endzustand prüfen
23         if (s == 0)
24             IO.println("Eingabe ist durch 3 teilbar.");
25         else
26             IO.println("Eingabe ist NICHT durch 3 teilbar.");
27     }
28
29 }
```

Listing 1: Automat

Ich habe hier Java als Programmiersprache gewählt. Die Programmiersprache ist sehr beliebt und entsprechend weit verbreitet. Der Compiler kann als freie Software im Internet heruntergeladen werden und ist für viele Plattformen verfügbar. Die Objektorientierung von Java spielt hier keine Rolle. Der Code sollte auch Programmierern verständlich sein, die ansonsten andere Programmiersprachen verwenden.

Um die Eingabe einfacher und übersichtlicher zu gestalten, habe die die *AlgoTools* der Universität Osnabrück, von Prof. Dr. Vornberger, verwendet.

Sämtliche Eingabezeichen werden in einem Stück gelesen und im Feld `zeile` gespeichert. Es liegt in der Verantwortung des Anwenders, sich an das Eingabealphabet (0 und 1) zu halten. Weil Zeichen in Java ganze Zahlen sind, wird durch Subtraktion des Zeichens '0' das jeweils aktuelle Eingabezeichen in seinen numerischen Wert umgerechnet. Dieser dient in Verbindung mit dem aktuellen Status als Index, um im Feld `delta`, welches die Überföhrungsfunktion (siehe auch Tabelle 1) darstellt, den sich ergebenden Zustand auszulesen. Die Schleife endet, wenn alle Eingabezeichen verarbeitet wurden.

Hinweis: Der potentiellen Unendlichkeit der Eingabe steht die begrenzte Anzahl der Eingabezeichen am Eingabeprompt sowie der verfügbare Speicher für das Datenfeld `zeile` entgegen. Als Ausweg bietet es sich an, die Eingabezeichen zeichenweise in einer Schleife einzulesen und die zu prüfende Dualzahl nötigenfalls in einer Textdatei zu speichern. Ersteres spart Hauptspeicher. Ist die zu prüfende Zahl in einer Textdatei gespeichert, dann bekommt das Programm die Dualzahl via Eingabeumleitung zugeführt. Da das Programm von der Standardeingabe liest, funktioniert die Eingabeumlenkung auch jetzt bereits.

Eingabe aus Datei durch Umlenkung des Eingabestroms in der Shell:

```
$ java Automat <dualzahl.txt
```

Zum Schluss wird der erreichte Zustand darauf geprüft, ob es sich um den Status 0 handelt. Dieser signalisiert, so wurde es vorher definiert, die ganzzahlige Teilbarkeit durch 3 ohne Rest. Das Ergebnis der Prüfung wird auf dem Bildschirm ausgegeben.

Karsten Brodmann, 15.02.2017