

Automat

zur Prüfung der ganzzahligen Teilbarkeit einer Dualzahl durch drei

Karsten Brodmann

10. März 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

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

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.

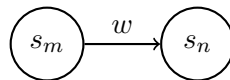


Abbildung 1: Graphische Darstellung einer Überföhrungsfunktion

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:

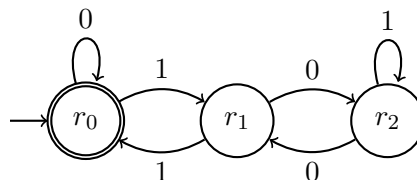


Abbildung 2: Überföhrungsfunktion δ des Automaten

Startzustand ist r_0 . Dazu gehört ein Zustandsüberföhrungsgraph 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  program Automat;
2      (* Endlicher Automat mit delta: Zustand x Eingabe -> Zustand.
3      *   - Ueberprueft, ob ein eingegebener Binaerstring durch 3
4      *   teilbar ist.
5      *   - Syntaktisch korrekte Eingabe wird vorausgesetzt.
6      *   - Zeichen '0' und '1' werden umgewandelt in Integer 0 und 1.
7      *
8      * 2017-01-03 (kb)
9      *)
10 var
11   c: char;                                (* Eingabezeichen *)
12   d: array [0..2,0..1] of integer = ((0,1), (* Ueberfuehrungsfkt. *)
13                                       (2,0),
14                                       (1,2));
15   s: integer;                                (* aktueller Zustand *)
16 begin
17   s:=0;                                       (* Startzustand *)
18   while not Eoln do begin
19       read(c);                                (* naechstes Zeichen *)
20       s:=d[s,Ord(c)-Ord('0')]                (* delta anwenden *)
21   end;
22   if (s=0) then                                (* Endzustand pruefen *)
23       writeln('Eingabe ist durch 3 teilbar.')
24   else
25       writeln('Eingabe ist NICHT durch 3 teilbar.')
26 end.
```

Listing 1: Automat

Ich habe hier Free Pascal als Programmiersprache gewählt. Der Compiler kann als freie Software im Internet heruntergeladen werden und ist für viele Plattformen verfügbar. Als didaktische Programmiersprache ist die Pascal-Syntax auch dem Nicht-Programmierer zuzumuten und verständlich.

Da das obige Programm keinerlei Anleihen an spezielle Features des Free Pascal Compilers macht, ist der Programmcode von jedem Pascal-Compiler zu übersetzen¹.

In einer Schleife werden einzelne Zeichen eingelesen. Es liegt in der Verantwortung des Anwenders, sich an das Eingabealphabet (0 und 1) zu halten. Mit `Ord(c)-Ord('0')` wird

¹Sollte jemand einen ISO-Compiler verwenden, ist der Programmcode der Kopfzeile des Programms wie folgt abzuändern: `PROGRAM Automat(Input, Output);`

das aktuelle Eingabezeichen in seinen Wert umgerechnet. Dieser dient in Verbindung mit dem aktuellen Status als Index, um im Feld d, welches die Überföhrungsfunktion (siehe auch Tabelle 1) darstellt, den sich ergebenden Zustand auszulesen. Die Schleife endet, wenn als Eingabezeichen ein *End of Line* gelesen wird, der Anwender also die Return-Taste betätigt.

Hinweis: Der potentiellen Unendlichkeit der Eingabe steht die begrenzte Anzahl der Eingabezeichen am Eingabeprompt entgegen. Als Ausweg bietet es sich an, die zu prüfende Dualzahl in einer Textdatei zu speichern. Das Programm bekommt die Dualzahl dann via Eingabeumleitung zugeführt.

Eingabe aus Datei durch Umlenkung des Eingabestroms in der Shell:

```
$ 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, 10.03.2017