7 Endliche Automaten

Ganz im Sinn der vorangegangenen Kapitel machen wir wieder Anleihen in einem wohl etablierten Gebiet. Dieses Mal ist es aus der theoretischen Informatik das Gebiet der formalen Sprachen. So ganz neu verwenden wir dieses Gebiet nicht, denn die Begriffe „Alphabet“ und „Wort“, die wir im Kapitel 1 eingeführt haben, stammen von dort. Ganz wesentliche Operationen in formalen Sprachen sind das korrekte Generieren von Wörtern und Texten nach festgelegten Regeln (Grammatik), die Überprüfung der Texte auf Korrektheit und das Übersetzen von Texten in eine (andere) formale Sprache (Compilieren). Für die übersichtliche Behandlung dieser Operationen wurden die endlichen Automaten entwickelt, die neben einer exakten, mathematischen Definition die Operationen leicht nachvollziehbar darstellen, dieses vor allem wegen ihrer graphischen Darstellung.

Endliche Automaten sind eine Sammlung von Zuständen und Eingabe-, Übergangs- und Ausgabeaktionen. Ein endlicher Automat arbeitet so, dass er sich zunächst in einem Anfangszustand befindet. Der Automat wird dadurch betrieben, dass er ein Wort zeichenweise einliest. Durch jedes eingelesene Zeichen geht er vom augenblicklichen Zustand in einen Folgezustand über. Ist das Wort eingelesen, wird je nach erreichtem letzten Zustand ein Zeichen ausgegeben.

Ein erstes Beispiel

Wir wollen diese abstrakte Beschreibung mit einem Beispiel erläutern. Dafür sei das Alphabet A = {a, b, c} und A* die Menge aller Wörter, die aus diesen drei Buchstaben gebildet werden können. Nach unserer Grammatik sollen die Wörter korrekt sein, in denen nicht zwei gleiche Buchstaben aufeinander folgen. Also wären „abac“ oder „bcbcbac“ korrekte Wörter, während „abbac“ kein korrektes Wort wäre, da auf ein „b“ ein zweites folgt. Wir wollen nun einen endlichen Automaten konstruieren, der Wörter auf diese Korrektheit überprüft. Dazu soll er nach dem Einlesen des gesamten Wortes „+“ ausgeben, wenn es korrekt ist und „–“, wenn es nicht korrekt ist.

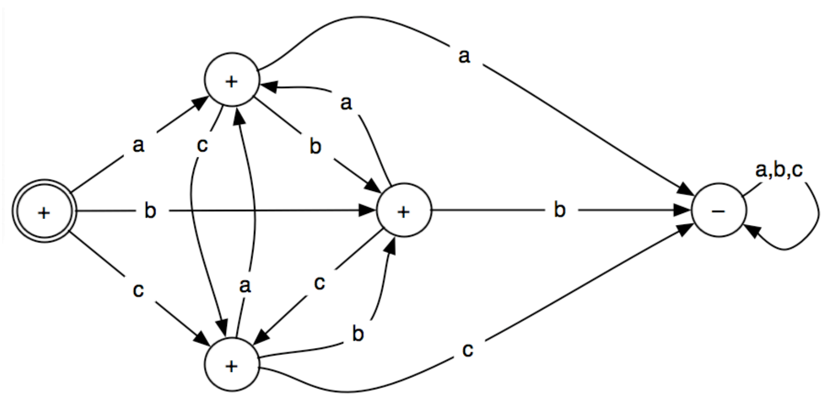

Abb. 7.1: Der Graph eines endlichen Automaten für das Testen der Wörter des Beispiels

Mit der graphischen Darstellung arbeitet man nun so, dass man mit dem doppelt umkreisten Zustand beginnt und das zu testende Wort Buchstabe für Buchstabe einliest. Zu jedem Zustand und Buchstaben ist in der grafischen Darstellung mit einem Pfeil der Übergang zum Folgezustand dargestellt. Diesen Pfeilen folgt man entsprechend der gelesenen Buchstaben. Für jeden Zustand ist notiert, welches Zeichen am Ende ausgegeben wird, wenn der Automat an diesem Zustand anhält. In Abb. 7.1 ist das ein „+“ für „korrekt“ bzw. „–“ für „nicht korrekt“.

Definition 7.1

Die formale Definition eines endlichen Automaten

Die exakte Definition[1] eines (determinierten) endlichen Automaten als Akzeptor für die Wörter einer Sprache ist ein 5-Tupel (A, S, s_0, δ, F) mit folgenden Bedeutungen:

- A ist ein Alphabet, also eine endliche, nicht leere Menge von Zeichen, die als Eingabe gelesen werden können
- S ist eine endliche, nicht leere Menge von Zuständen
- s_0 ist ein Element von S, der Anfangszustand
- δ ist die Übergangsfunktion $S \times A \to S$, mit der die Übergänge von einem Zustand zum nächsten in Abhängigkeit vom eingelesenen Zeichen definiert werden
- F ist eine Teilmenge von S, die Menge der akzeptierenden Zustände. Hält der Automat in einem Zustand von F an, so ist das Wort akzeptiert.

Für unsere Zwecke reicht diese Definition nicht aus und wir erweitern sie zu

Definition 7.2

Endlicher Automat mit Ausgabe

Ein (determinierter) endlicher Automat mit Ausgabe ist ein 6-Tupel (A, S, s_0, δ, B, ω) mit folgenden Bedeutungen:
Die Elemente A, S, s_0 und δ haben die Bedeutung wie in Definition 7.1 .

- B ist ein Alphabet, also eine endliche, nicht leere Menge von Zeichen, die nach Beendigung der Eingabe ausgegeben werden können
- ω ist die Ausgabefunktion $S \to B$, die jedem Zustand aus S ein Zeichen aus B als Ausgabe zuordnet

Beispiel

Der in Abb. 7.1 graphisch dargestellte endliche Automat ist folgendermaßen definiert:
$A = \{\text{a, b, c}\}$, $S = \{s_0, s_a, s_b, s_c, s_f\}$, $s_0 = s_0$

Die Zustände sind in Abb. 7.1 einfach nur Kreise. Dabei ist s_0 der linke, doppelt umkreiste Zustand und s_f der ganz rechte. Bei den zentralen drei Zuständen ist s_a der obere, s_b der mittlere, rechte und s_c der untere Zustand.
Die durch die Pfeile dargestellten Übergänge schreiben wir in einer Zuordnungstabelle.

δ	Eingabe		
Zustand	a	b	c
s_0	s_a	s_b	s_c
s_a	s_f	s_b	s_c
s_b	s_a	s_f	s_c
s_c	s_a	s_b	s_f
s_f	s_f	s_f	s_f

Das Alphabet für die Ausgabe ist $B = \{+, -\}$.
Die Zuordnung der Ausgabesymbole durch die Funktion ω geben wir ebenfalls in einer Tabelle an.

ω	Zustand	s_0	s_a	s_b	s_c	s_f
	Ausgabe	+	+	+	+	–

[1] Nach R.Verbeek in [S05]

Für unsere Zwecke arbeiten wir nur mit endlichen Automaten mit Ausgabe. Da eine grafische Darstellung alle Informationen eines solchen Automaten beinhaltet, beschränken wir uns darauf und sagen, dass der endliche Automat dann definiert ist, wenn er in der grafischen Darstellung vollständig angegeben ist.

Automatische Folgen

Mit Hilfe eines endlichen Automaten kann man nun die Elemente einer Folge definieren. Wenn ein endlicher Automat gegeben ist, der als Eingabe die Ziffern einer Zahl akzeptiert, so ist das Folgenelement a_n definiert als Ausgabe des endlichen Automaten nachdem er den Index n gelesen hat. Dabei kann der Index als Zahl im Dezimalsystem gelesen werden, es kann aber auch ein anderes, dem Problem angepasstes Stellenwertsystem sein.

Beispiel:

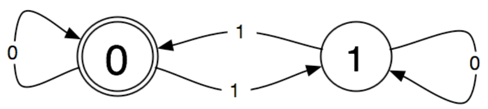

Abb. 7.2: Der Graph eines endlichen Automaten, der Zahlen im Binärsystem verarbeitet

Die Zahlenfolge $(a_n)_{n\in\mathbb{N}_0}$ ist hier so definiert, dass der Index eines Folgenelementes als Binärzahl verarbeitet wird und die Ausgabe 0 oder 1 das Folgenglied definiert.

Damit ist $a_0 = 0$, $a_1 = 1$ und z.B. $a_{18_{10}} = a_{10010_2} = 0$. Für diesen Automaten ist es egal, ob die Binärzahl von links (wie ein Text) oder von rechts (entsprechend der Wertigkeit der Stellen) gelesen wird. Die so definierte Zahlenfolge ist als Thue-Morse-Folge bekannt und man kann sie auch so definieren, dass a_n die Quersumme der binären Darstellung von n modulo 2 ist.

Wir wollen nun zur Papierfaltungsfolge zwei endliche Automaten konstruieren und so aufzeigen, dass die Papierfaltungsfolge eine automatische Folge ist. In beiden Fällen wird der Index als Binärzahl eingelesen. Im ersten Fall soll die Zahl von rechts nach links eingelesen werden, also beginnend mit der Einerziffer fortschreitend zu höheren Stellen. Im zweiten Fall wollen wir genau umgekehrt vorgehen. In beiden Fällen müssen wir beachten, dass die Zahldarstellung nach links mit beliebig vielen führenden Nullen fortgesetzt werden kann. So kann z.B. $9_{10}=1001_2$ auch als 000001001_2 geschrieben werden. Beide Darstellungen sollen von beiden endlichen Automaten korrekt verarbeitet werden.

Der endliche Automat für das Lesen von rechts nach links

Das Eingabealphabet ist $A = \{0,1\}$, da Binärzahlen eingelesen werden. Das Ausgabealphabet ist $B = \{L,R\}$.

Die Übergänge zwischen den Zuständen stehen in direktem Zusammenhang mit dem Inflations-/ Deflationsgesetz. Beim Bestimmen des Symbols auf dem Platz k nach dem Inflationsgesetz (siehe Kapitel 3) haben wir k zerlegt in $k = 2^a(4b+c)$, mit $a,b,c \in \mathbb{N}$ und $0 \le c \le 3$. Diese Zerlegung finden wir direkt in der Binärdarstellung von k wieder:

$$k = 2^a(4b+c) = \underbrace{d_{a+m}...d_{a+2}}_{\text{Ziffern für b}}\ \underbrace{d_{a+1}d_a}_{\text{Ziffern für c}}\ \underbrace{0...00}_{\text{a Nullen}} \quad , m \in \mathbb{N}$$

Am Ende steht für $a = 0$ keine Null oder ($a > 0$) es stehen in den Ziffern d_0 bis d_{a-1} Nullen. Die Ziffer d_a ist immer 1. Die Zahl c finden wir dann in den beiden Ziffern d_a und d_{a+1}, nämlich 01_2 für 1 und 11_2 für 3. Davor stehen die Binärziffern, die die Zahl b bilden. Diese haben auf das Bestimmen des Symbols „L“ oder „R“ keinen Einfluss.

Nach diesen Überlegungen kann der endliche Automat konstruiert werden.

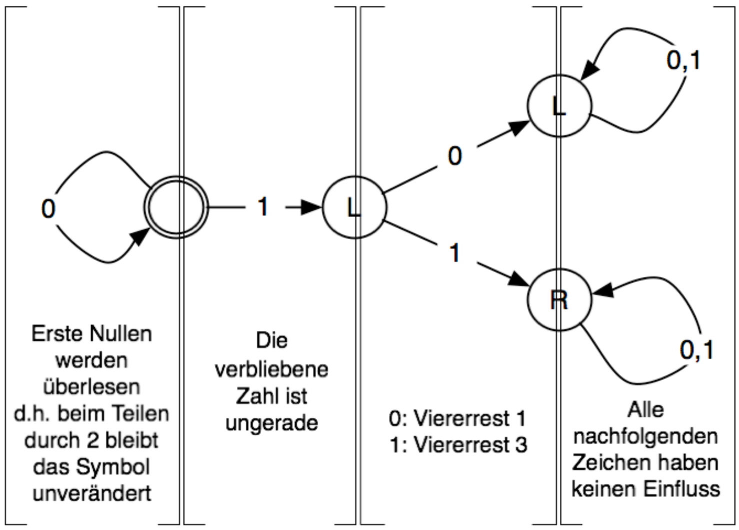

Abb. 7.3: Der Graph des endlichen Automaten für das Einlesen der Binärzahlen von rechts

Dieser endliche Automat ist sehr einfach und übersichtlich, so dass wir das Lesen der Binärzahlen als gut zu handhabende Regel für das Bestimmen des Symbols auf dem entsprechenden Platz in der Papierfaltungsfolge formulieren können:

Ein ganz einfacher Algorithmus

Suche in der Binärzahl von rechts nach links die erste 1. Steht davor eine 0, so ist das Symbol ein „L“, steht davor eine 1, so ist es ein „R“.

Damit bekommt die in den Kapiteln 2 und 3 aufgeworfene Frage, wie man für eine gegebene Stelle aus der Nummer das Symbol berechnen kann, letztlich eine überraschend einfache Antwort. Dieses ist ein schönes Beispiel, wie die Betrachtung ein und desselben Problems von verschiedenen Seiten zu immer tieferer Einsicht führt.

Die Tabelle zeigt, wie leicht unsere letzte Regel zu handhaben ist.

Platz	Binär	Symbol		Platz	Binär	Symbol		Platz	Binär	Symbol
1	000000**01**	L		11	000010**11**	R		21	000101**01**	L
2	00000**01**0	L		12	0000**11**00	R		22	00010**11**0	R
3	000000**11**	R		13	000011**01**	L		23	000101**11**	R
4	0000**01**00	L		14	00001**11**0	R		24	000**11**000	R
5	000001**01**	L		15	000011**11**	R		25	000110**01**	L
6	00000**11**0	R		16	00**01**0000	L		26	00011**01**0	L
7	000001**11**	R		17	000100**01**	L		27	000110**11**	R
8	000**01**000	L		18	00010**01**0	L		28	0001**11**00	R
9	000010**01**	L		19	000100**11**	R		29	000111**01**	L
10	00001**01**0	L		20	0001**01**00	L		30	00011**11**0	R

Abb. 7.4: Tabelle für die Platznummer als 8-stellige Binärzahl und das zugehörige Symbol

Der endliche Automat für das Lesen von links nach rechts

Wir wollen nun einen endlichen Automaten konstruieren, bei dem der Index des Folgenelementes wieder als Binärzahl eingelesen wird, nun aber von links nach rechts, also von den hohen Stellen bis zur Einerstelle, die als letzte eingelesen wird. Die Ziffernfolge wird also wie ein Text gelesen.

Das Eingabealphabet ist also wieder $A = \{0,1\}$und das Ausgabealphabet $B = \{L,R\}$.

Für die Übergänge von einem Zustand zum anderen sind die Veränderungen einer Zahl maßgeblich, die durch das Anhängen einer Ziffer an der rechten Seite verursacht werden.

- Anhängen einer 0: Die Zahl wird verdoppelt. Wegen $\sigma(n) = \sigma(2n)$ wird das Symbol nicht verändert. Aus einer ungeraden Zahl wird eine gerade Zahl und eine gerade Zahl bleibt gerade.

- Anhängen einer 1: Die Zahl wird verdoppelt und dann 1 addiert. Aus einer geraden Zahl wird eine ungerade, die beim Teilen durch 4 den Rest 1 lässt. Aus einer ungeraden Zahl entsteht eine Zahl, die beim Teilen durch 4 den Rest 3 lässt.

Beide Überlegungen legen nahe, vier Zustände zu unterscheiden: „L" auf einer geraden bzw. ungeraden Position und „R" auf einer geraden bzw. ungeraden Position.

Nach diesen Überlegungen können wir nun den endlichen Automaten konstruieren.

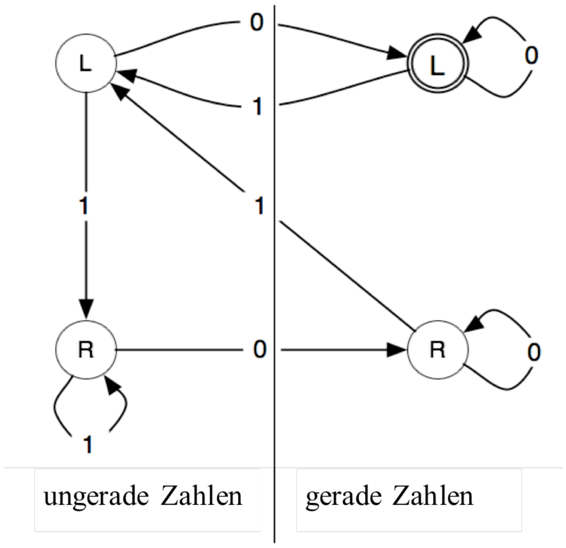

Abb. 7.5: Der Graph des endlichen Automaten für das Einlesen der Binärzahlen von links

Der Startzustand ist das doppelt eingekreiste „L" rechts oben. Führende, also überflüssige Nullen halten diesen Startzustand fest. Die linken beiden Zustände erreicht man, wenn die bisher eingelesenen Ziffern eine ungerade Zahl darstellen. Sie werden immer durch eine 1 erreicht. Rechts stehen die Zustände, die nach einer geraden Zahl erreicht werden. Zu diesen Zuständen führen eingelesene Nullen.

Hinweise für eine unterrichtliche Behandlung

Praktische Erfahrungen[2] zeigen, dass sich Schüler der Oberstufe oder Studenten überraschend schnell in die zu endlichen Automaten gehörende Denkweise hineinfinden und bereits nach wenig Übung Automaten konstruieren können. Günstig ist, wenn dieses neue Thema mit alten, bekannten verknüpft wird. So lassen sich beispielsweise endliche Automaten einsetzen, um die Teilbarkeitsregeln darzustellen. Dieses Übungsfeld ist besonders ergiebig, wenn man die Teilbarkeisregeln in verschiedenen Stellenwertsystemen behandelt hat. Insbesondere bei kleinen Basen kann man übersichtliche, endliche Automaten konstruieren. Das Zehnersystem hat mit zehn Zeichen bereits ein recht umfangreiches Alphabet.

Unterrichtsvorschlag (Teilbarkeitsregel im Zehnersystem)

angesprochene schulmathematische Themen

Teilbarkeitsregeln, Teilen mit Rest

Aufgabenstellung

Die Schülerinnen und Schüler sollen einen endlichen Automaten konstruieren, der eine bekannte Teilbarkeitsregel darstellt. Dabei sollen die Ziffern der Zahl von rechts nach links eingelesen werden. Die Ausgabe am Ende liefert als Ergebnis, ob die Zahl durch die betrachtete Zahl teilbar ist oder nicht.

Ablauf

Als Beispiel wollen wir hier die Teilbarkeitsregel für die 4 behandeln: *Eine Zahl (geschrieben im Zehnersystem) ist durch 4 teilbar, wenn die aus den letzten beiden Ziffern gebildete Zahl durch 4 teilbar ist.*

1. Vorüberlegung

Wir werden den Rest beim Teilen durch 4 berechnen. Diese Reste liefern dann das gewünschte Ergebnis. Wir überlegen uns zunächst die Reste beim Teilen durch 4 für die Einerziffern.

$0 = 0 \cdot 4 + 0$	$1 = 0 \cdot 4 + 1$	$2 = 0 \cdot 4 + 2$	$3 = 0 \cdot 4 + 3$
$4 = 1 \cdot 4 + 0$	$5 = 1 \cdot 4 + 1$	$6 = 1 \cdot 4 + 2$	$7 = 1 \cdot 4 + 3$
$8 = 2 \cdot 4 + 0$	$9 = 2 \cdot 4 + 1$		

Entsprechend gehen wir bei den Zehnerziffern vor:

	$10 = 2 \cdot 4 + 2$	$20 = 5 \cdot 4 + 0$	$30 = 7 \cdot 4 + 2$
$40 = 10 \cdot 4 + 0$	$50 = 12 \cdot 4 + 2$	$60 = 15 \cdot 4 + 0$	$70 = 17 \cdot 4 + 2$
$80 = 20 \cdot 4 + 0$	$90 = 22 \cdot 4 + 2$		

Die Hunderter- und höheren Ziffern haben keinen Einfluss auf die Teilbarkeit durch 4, so dass wir dafür keine Tabelle aufstellen müssen.

[2] Eigene Erfahrungen in einem Seminar im Sommersemester 2005 und Vortrag und Übungen im Vorsemester Bremen 2005

2. Erstellen des Automaten selbst

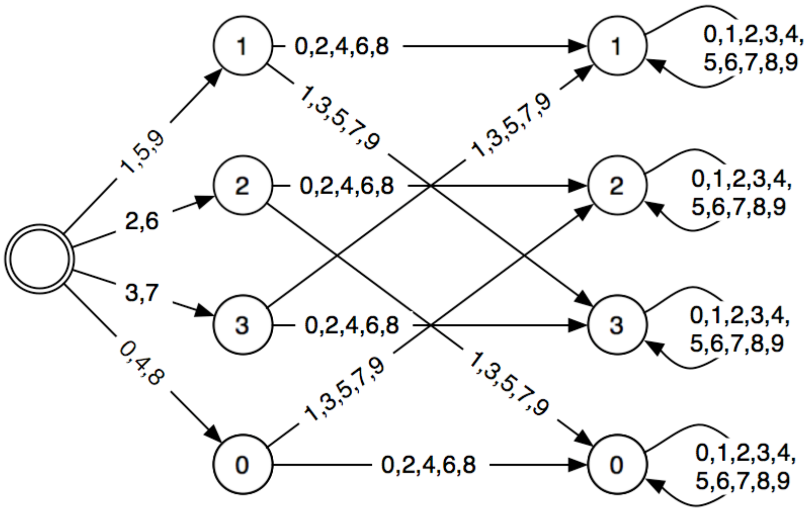

Abb. 7.6: Der Graph des endlichen Automaten zur Bestimmung des Viererrestes

Die Ausgabe dieses Automaten liefert den Rest beim Teilen durch 4. Ist der Rest 0, so ist die Zahl durch 4 teilbar, in den anderen Fällen nicht. Ist man also wirklich nur an der Teilbarkeit durch 4 interessiert, so kann man den Automaten vereinfachen, indem man die rechten Zustände mit der Ausgabe 1, 2 und 3 zu einem Zustand zusammenfasst. Die linken beiden Zustände für 1 und 3 müssen dann auch nicht unterschieden werden. Der vereinfachte endliche Automat ist dann:

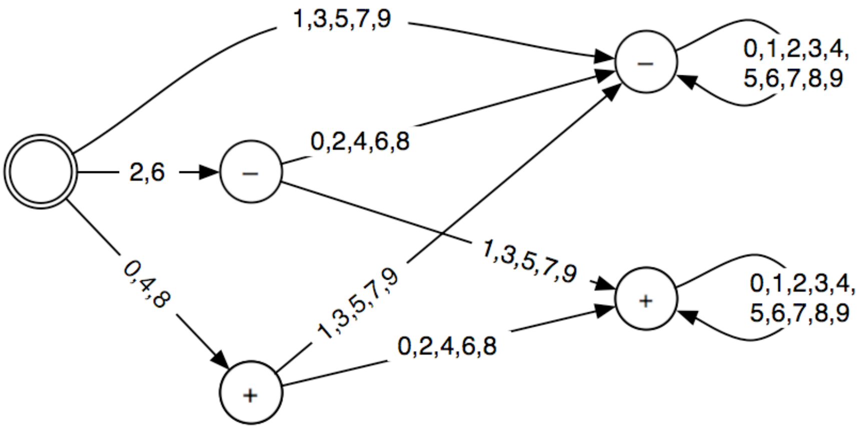

Abb. 7.7: Der Graph des endlichen Automaten zur Bestimmung der Teilbarkeit durch 4