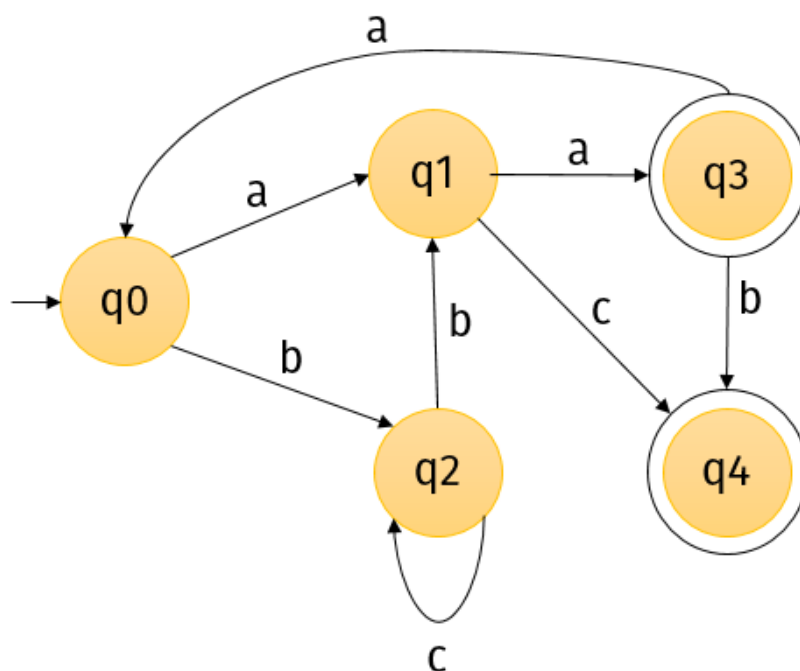


# Einführung in die Automaten theorie



**Verfasser:**

Anne Schüller, Anke Honskamp

**Kurz-Info:**

Informatischer Inhalt: Verwendung formaler Sprachen als  
Mittler zwischen Mensch und  
Maschine und Entwicklung von  
Automaten

Jahrgangsstufe:

EF bis Q2

Vorwissen:

keine speziellen Vorkenntnisse  
erforderlich

# KURZINFORMATION

## FÜR DIE SCHÜLERINNEN UND SCHÜLER

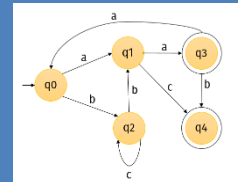
**Titel:** Einführung in die Automatentheorie

**Schulstufe:** Sekundarstufe II

**optimale Jahrgangsstufe:** EF

**geeignete Kursart:** Grundkurs

**Themenbereich:** Formale Sprachen, (endliche) Automaten, Java



### EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN

#### Kernlehrplan NRW:

Inhaltsfeld: Formale Sprachen und Automaten

Kompetenzbereiche: Modellieren, Implementieren, Darstellen und Interpretieren

**Vorgaben zum Zentralabitur:** Formale Sprachen und Automaten, Java

#### Bildungsstandards der GI:

Unterschied zwischen natürlichen und formalen Sprachen

Automaten als zustandsbasierte Systeme, die eine Eingabe zeichenweise lesen und verarbeiten

**Einbindung in den Unterricht:** Das Leitprogramm kann als Einstieg in die Arbeit mit formalen Sprachen und Automaten dienen.

**Vorkenntnisse:** Es sind keine speziellen Vorkenntnisse erforderlich.

**Dauer:** ca. 5 Schulstunden

**Inhaltsbeschreibung:** Im Rahmen dieses Leitprogramms beschäftigen sich die Schülerinnen und Schüler mit einem elementaren Thema der Informatik, das auch im Kernlehrplan der Sekundarstufe II verankert ist: Automaten.

Der Begriff „Automat“ ist fester Bestandteil unserer Alltagssprache. So kennt jeder Getränke-, Bank- oder Parkautomaten. Allerdings gibt es auch Automaten, die nicht intuitiv als solche erkannt werden. So läuft zum Beispiel der Aufladevorgang beim Prepaid-Handy automatisch ab. Eine computergenerierte Stimme leitet den Handynutzer, der durch Tasteneingabe Auswahlen treffen kann, durch ein Menü. „Wenn Sie ihr Guthaben aufladen möchten, drücken Sie die Eins. Möchten Sie das Menü verlassen, drücken Sie die Zwei“. In der Informatik wird dieser geregelte Ablauf auch als Automat bezeichnet. Um zum Beispiel einen solchen Aufladevorgang zu entwickeln und anschließend zu programmieren, muss die Struktur des Vorgangs nachvollzogen werden. Diese Struktur lässt sich gut mit Hilfe von Automaten beschreiben, welche dann die Grundlage für das Programm darstellt.

## INHALT

Kurzinformation für die Schülerinnen und Schüler .....	2
Arbeitsanleitung .....	4
Grundlagen zu Automaten .....	6
Automaten im Alltag .....	6
Allgemeinere Darstellung von Automaten .....	7
Lernfortschrittskontrolle .....	9
Darstellung von Automaten.....	10
Zustände und Übergänge .....	10
Besondere Zustände .....	12
Eingaben .....	13
Additum.....	17
Lernfortschrittskontrolle .....	17
Akzeptanzverhalten von Automaten .....	19
Akzeptanzverhalten.....	19
JFLAP – Testen.....	23
JFLAP – Eigene Automaten entwerfen .....	26
Lernfortschrittskontrolle .....	29
Erstellung von Automaten .....	30
Automaten konstruieren.....	30
Die Sprache eines Automaten .....	32
Additum.....	34
Lernfortschrittskontrolle .....	34
Quellenverzeichnis .....	36
Abbildungsverzeichnis .....	36

## ARBEITSANLEITUNG

In diesem Leitprogramm wirst du in die Welt der Automaten eingeführt. Dabei wirst du die folgenden vier Kapitel eigenständig und in deinem eigenen Lerntempo bearbeiten. Jedes Kapitel besteht aus mehreren Bausteinen:



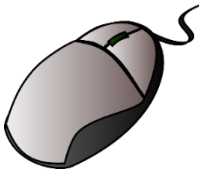
**Übersicht:** Die Übersicht steht vor jedem Kapitel und verschafft einen Eindruck davon, was Gegenstand des Kapitels sein wird.



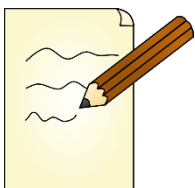
**Lernziel:** Hier werden die Teillernziele des jeweiligen Kapitels definiert.



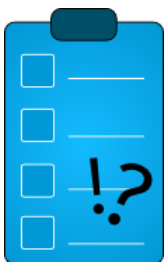
**Definition:** An dieser Stelle werden Begriffe und Konzepte bestimmt und erklärt.



**Übungsaufgaben am Computer:** Die Aufgabe wird am Computer ausgeführt. Bitte arbeite erst dann weiter, wenn dein Programm wirklich läuft. Du darfst diese Aufgaben alleine oder mit einem Partner bearbeiten.



**Übungsaufgaben im Heft:** Diese Übungsaufgaben bearbeitest du in deinem Heft. Anschließend kannst du deine Ergebnisse mit den Musterlösungen, die du von deinem Lehrer/deiner Lehrerin bekommst, vergleichen. Wenn nicht anders angegeben, sollen die Aufgaben in Einzelarbeit erledigt werden.



**Lernkontrolle:** Hier kannst du prüfen, ob du die Inhalte des vorgegangenen Kapitels verstanden hast. Entscheide selbst, wann du dich fit für den Test fühlst. Nur wenn der Test erfolgreich absolviert wurde, darf das folgende Kapitel oder ein Additum in Angriff genommen werden.



**Additum:** Diesen Teil kannst du bearbeiten, wenn du schon alle anderen Aufgaben des Kapitels abgearbeitet hast. Nach der Lernkontrolle des jeweiligen Kapitels, entscheidet die Lehrkraft, ob du mit dem Additum weitermachst. Diese Aufgaben sind bewusst anspruchsvoller gestaltet. Sie können auch unbekanntes Stoff beinhalten.

Wenn du dich nach der Bearbeitung der Lernfortschrittkontrolle fit für den **Kapiteltest** fühlst, kannst du dich bei deinem Lehrer/deiner Lehrerin melden. Du bekommst dann den Kapiteltest. Wenn du diesen fertig bearbeitet hast, gibst du diesen deinem Lehrer/deiner Lehrerin zur Kontrolle. Wenn der Kapiteltest bestanden wurde, kannst du zum nächsten Kapitel übergehen.



### Übersicht

Dieses Kapitel bietet eine Einführung in die Welt der Automaten. Dazu wird anfangs die Funktionsweise eines Parkautomaten genauer betrachtet.



### Teillernziele

Nach der Bearbeitung dieses Kapitels kannst du..

- Beispiele für alltägliche Automaten angeben.
- (einfache) informatische Modelle entwickeln, indem du Automaten graphisch darstellst und dabei einheitliche Notationen (Schreibweisen) verwendest.

## AUTOMATEN IM ALLTAG

Ein Beispiel:

Laura fährt mit ihrem Freund Manfred mit dem Auto in die Stadt, um ein paar Einkäufe zu erledigen. Sie parken auf dem Stadtparkplatz, wo sie auch Parkgebühren bezahlen müssen. Bevor die Shoppingtour losgehen kann, müssen Manfred und Laura erst den Parkautomaten mit Geld füttern.

Der Parkautomat funktioniert wie folgt:

1. Er akzeptiert nur Münzen von 5 Cent bis hin zu 2 Euro.
2. Er gibt kein Wechselgeld zurück.
3. Er zeigt an, wie viele Minuten man für das bis dahin eingeworfene Geld parken darf.
4. Er besitzt einen Knopf, den man drücken muss, um den Parkschein zu bekommen.



Abbildung 1:  
Parkautomat

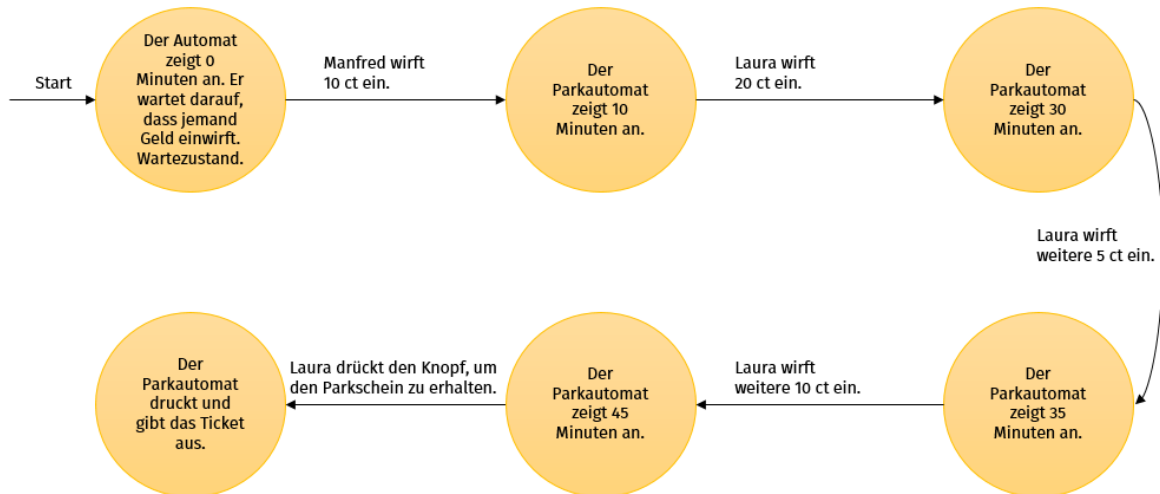
Eine genauere Betrachtung des Innenlebens des Automaten zeigt: Der Automat muss drei verschiedene Situationen unterscheiden:

1. Er ist im Leerlauf, das heißt er hat 0 Euro gespeichert, und keiner wirft gerade Geld ein.
2. Jemand füttert den Automaten mit Geld, will aber noch kein Ticket ziehen.
3. Jemand betätigt den Knopf, damit der Parkschein ausgedruckt wird.

Da Laura und Manfred nur kurz einkaufen wollen, wissen sie schon, dass sie nicht länger als 45 Minuten brauchen werden. Nun stehen sie vor dem Parkautomaten und lesen, dass man für 10 Minuten 10 Cent bezahlen muss. Manfred kramt in seinem Portemonnaie und findet eine 10 Cent Münze, die er einwirft. Unmittelbar danach wechselt die Anzeige des Automaten, die anfangs auf 0 Minuten stand, zu 10 Minuten. Auch Laura findet noch Kleingeld in ihrer Tasche und wirft eine 20 Cent Münze in den Automaten. Die Anzeige erhöht sich auf 30 Minuten. Jetzt fehlen nur noch 15 Cent, damit sie 45 Minuten parken können. Laura wirft eine 5 Cent Münze nach; die Anzeige erhöht sich auf 35 Minuten und nach dem Einwurf von 10 Cent auf 45 Minuten. Super! Jetzt können die beiden endlich einkaufen gehen.

Aber nein! Sie brauchen doch noch den Parkschein. Also drückt Laura schnell auf den Knopf und der Automat fängt an, den Parkschein zu drucken. Anschließend spuckt er das Ticket aus. Die Anzeige des Automaten steht nun wieder auf 0 Minuten.

Die folgende Zeichnung stellt die Situation von Manfred und Laura am Parkautomaten dar:



### Aufgabe 1.1

Du möchtest 30 Minuten parken und musst dafür am oben beschriebenen Parkautomaten einen Parkschein ziehen.

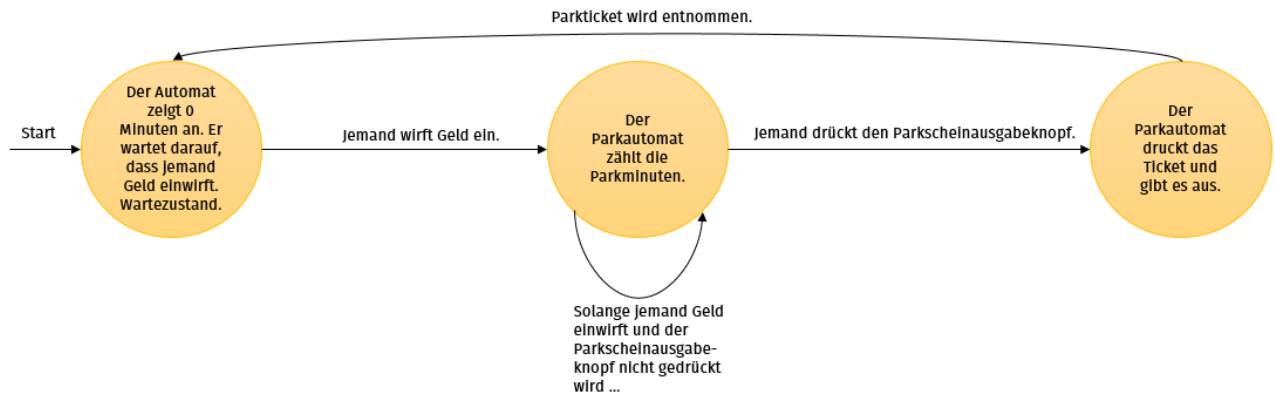
Gib zwei verschiedene Möglichkeiten an, welche Münzkombinationen du in den Automaten einwerfen kannst und skizziere eine Möglichkeit mit einer Zeichnung (s. o.).

## ALLGEMEINERE DARSTELLUNG VON AUTOMATEN

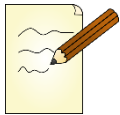
Wie du vielleicht bereits gemerkt hast, sehen alle Zeichnungen, die das Vorgehen von diesem Parkautomaten darstellen, sehr ähnlich aus:

- Am Anfang zeigt der Parkautomat immer 0 Minuten an und wartet, bis jemand Geld einwirft.
- Dann zeigt der Automat die Minuten an, solange jemand weiter Geld einwirft und bis der Parkscheinausgabeknopf gedrückt wird.
- Wurde der Parkscheinausgabeknopf gedrückt, so druckt der Parkscheinautomat das Parkticket und gibt es aus.

Der Parkscheinautomat kann also auch allgemeiner dargestellt werden, so dass er die Situation unabhängig von den Parkminuten widerspiegelt:



Die Kreise im Bild bezeichnet man in der Informatik als **Zustände** und die Pfeile als **Übergänge** von einem Zustand in den anderen. Die genaue Definition der Begriffe wird in Kapitel 2 gegeben.



### Aufgabe 1.2

Stell dir vor, du stehst vor einem Flipperautomaten und möchtest eine Runde spielen. Auf dem Automaten steht: „Pro Spiel 50 Cent!“ und „Ein Spiel = 3 Bälle“.

Beschreibe in deinen eigenen Worten, wie dein Spiel abläuft.

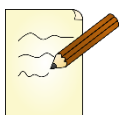


Abbildung 2: Flipperautomat



### Aufgabe 1.3

Fertige eine Zeichnung für den Flipperautomaten an, die die Arbeitsweise darstellt.



### Aufgabe 1.4

Nenne weitere Beispiele für Automaten. Beachte dabei, dass ein Automat etwas beschreibt, was automatisch abläuft, und daher der Begriff „Automat“ in dem Wort nicht immer enthalten sein muss.





**L 1.1**

Welche der folgenden Gegenstände sind Automaten? Kreuze an.

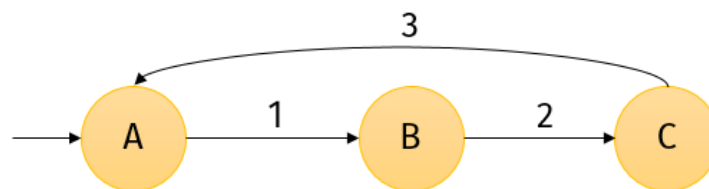
- DVD-Recorder
- MP3-Player
- Scanner
- Fotoapparat
- Kaffeemaschine
- Selbstscan-Kasse
- Waschmaschine
- Verkehrsampel



**L 1.2**

In dieser Aufgabe soll ein Kaugummiautomat mit Hilfe einer Skizze beschrieben werden. Der Kaugummiautomat funktioniert wie folgt:

Man wirft 10 Cent ein und dreht an dem Rad. Dann fällt der Kaugummi in den Auffangbehälter und kann entnommen werden. Die folgende Skizze soll die Funktionsweise des Kaugummiautomaten beschreiben:



Ordne die aufgelisteten Punkte den Buchstaben und Ziffern der Skizze zu:

- Drehknopf wird freigegeben
- Drehknopf wird gedreht
- Automat wartet auf Geldeinwurf
- Automat gibt Kaugummi aus
- Kaugummi wird entnommen
- 10 Cent werden eingeworfen



### Übersicht

Um Abläufe einheitlich und verständlich mit Hilfe von Automaten darstellen zu können, ist es nötig, eine einheitliche Notation (Zeichen, Symbole, Schreibung) für Automaten zu verwenden. Diese lernst du im folgenden Kapitel kennen. Dabei werden die Begriffe **Zustand**, **Übergang** und **Eingabe** von Automaten erläutert. Außerdem wird das Verhalten von Automaten auf Aktionen genauer betrachtet.



### Teillernziele

Nach der Bearbeitung dieses Kapitels kannst du..

- unterscheiden, was in der Automatentheorie unter den Begriffen *Zustand*, *Übergang*, *Eingabe* und *Reaktion* zu verstehen ist.
- Zustände und Übergänge in exemplarischen, graphischen Darstellungen benennen.
- bestimmen, in welchem Zustand sich ein Automat nach einer bestimmten Eingabe befindet.
- das Eingabealphabet eines Automaten bestimmen.

---

## ZUSTÄNDE UND ÜBERGÄNGE

Automaten kann man sich als eine Art „Maschine“ vorstellen, die stur einem festgelegten Schema folgt, so wie zum Beispiel eine Kaffeemaschine. Eine Kaffeemaschine kann sich in verschiedenen **Zuständen** befinden (warten, Kaffee kochen, Kaffee warmhalten). Das festgelegte Schema sagt ihr, dass sie, wenn sie angeschaltet wird, Kaffee kochen soll. Wenn sie damit fertig ist, soll sie den Kaffee warmhalten, solange bis sie ausgeschaltet wird.

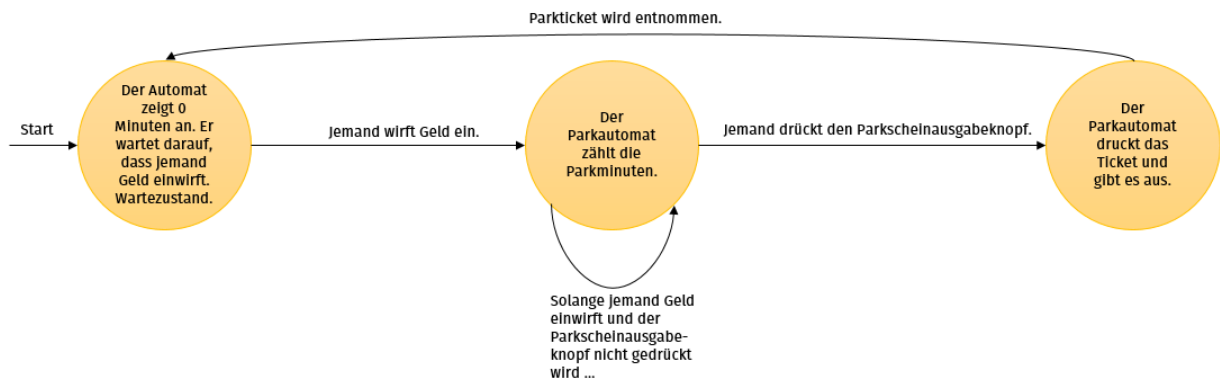
Im Allgemeinen haben alle Automaten ein solch vorgegebenes Schema. Automaten setzen sich zusammen aus **Zuständen** und **Übergängen**. Ein festgelegtes Schema gibt vor, wann ein Automat von einem Zustand in einen anderen übergeht.



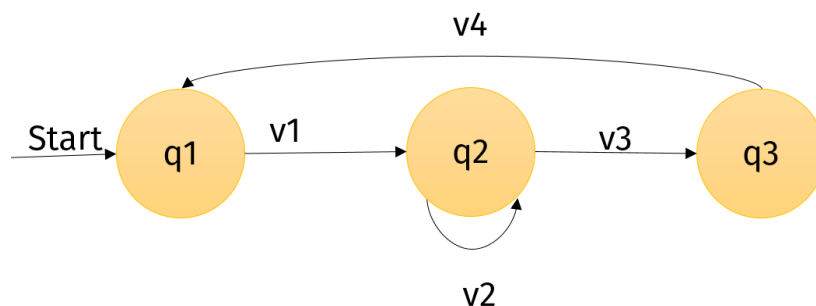
### Definition

Zu jedem Zeitpunkt befindet sich ein Automat in genau einem **Zustand**. **Übergänge** werden anhand einer Übergangsfunktion beschrieben. Eine **Übergangsfunktion** gibt an, mit welchem Zeichen von einem bestimmten Zustand in einen anderen gewechselt werden kann.

Noch einmal zurück zu dem Parkscheinautomaten, vor dem Laura und Manfred stehen:



Beschriftet man die Zustände und Übergänge ein wenig anders, sieht das Ganze so aus:



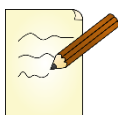
Dieser Automat hat die folgenden **Zustände**:

- $q_1$ . Der Automat wartet auf eine Eingabe. → **Wartezustand**.
- $q_2$ . Der Automat merkt sich, wie viel Geld eingeschmissen wurde.
- $q_3$ . Der Automat drückt das Parkticket und gibt es aus.

Außerdem hat der Automat die folgenden **Übergänge**:

- $v_1$ . Geld wird eingeschmissen. → Der Automat wechselt von  $q_1$  zu  $q_2$ .
- $v_2$ . Es wird mehr Geld eingeworfen. → Der Automat bleibt in  $q_2$  und zählt die Minuten.
- $v_3$ . Der Knopf „Parkschein ausgeben“ wird gedrückt. → Der Automat wechselt in  $q_3$ .
- $v_4$ . Das Parkticket wird entnommen. → Der Automat wechselt zurück in  $q_1$ .

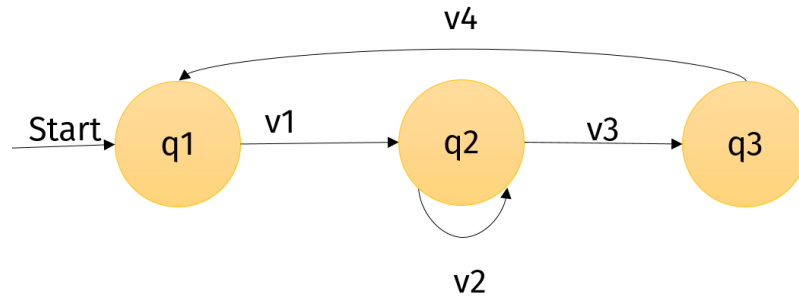
Diese Abstraktion (Verallgemeinerung) hat den Vorteil, dass nun eine gewisse **Vergleichbarkeit** mit anderen Automaten geschaffen wird und so generelle Aussagen und allgemeine Betrachtungen möglich sind.



### Aufgabe 2.1

Eine einfache Supermarktkasse funktioniert folgendermaßen:

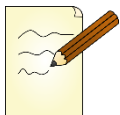
- Wird ein Preis eingegeben, addiert die Kasse diesen zum Gesamtpreis.
- Drückt jemand die Taste „Kassieren“, wird der Gesamtpreis angezeigt und die Geldlade geöffnet.
- Wird die Geldlade geschlossen, wartet die Kasse darauf, dass erneut ein Preis eingegeben wird.



Ordne die verschiedenen Zustände der Kasse und die Aktionen des Kassierers/der Kassiererin den Zuständen und Übergängen der Skizze zu.

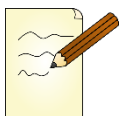
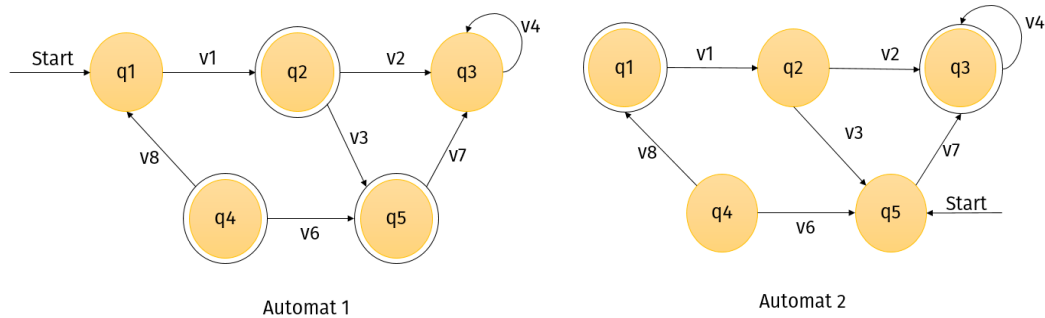
### BESONDERE ZUSTÄNDE

Dir ist vielleicht schon aufgefallen, dass viele Automaten am „ersten“ Zustand einen Pfeil mit der Beschriftung „Start“ haben. Dieser Zustand wird auch **Startzustand** genannt. Komplementär zum Startzustand gibt noch einen weiteren „besonderen“ Zustand: den **Endzustand**. Dieser wird im Allgemeinen durch einen doppelten Kreis gekennzeichnet. Ein Automat kann auch mehrere Endzustände haben. Auf die besondere Bedeutung von Endzuständen wird in Kapitel 3 näher eingegangen.



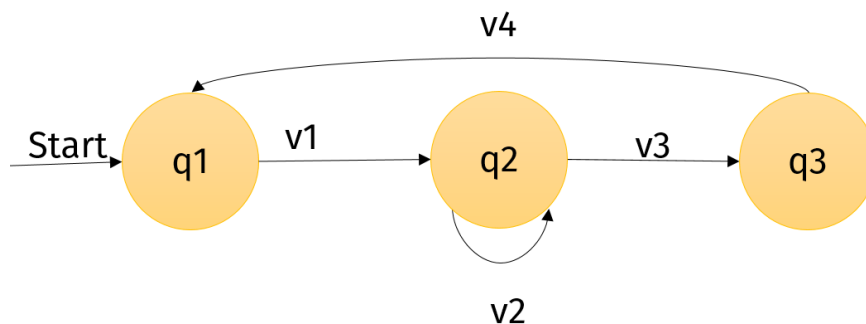
#### Aufgabe 2.2

Benenne die Start- und Endzustände der folgenden zwei Automaten:



#### Aufgabe 2.3

Betrachte noch einmal den Automaten zu der Supermarktkasse aus Aufgabe 2.1.



Die Kasse hat die folgenden **Zustände**:

- *q1*. Wartezustand; die Kasse wartet auf die Eingabe eines Preises.
- *q2*. Ein Preis wurde eingegeben, der Knopf „Kassieren“ wurde allerdings noch nicht gedrückt.
- *q3*. Der Gesamtpreis wird angezeigt, und die Geldlade ist geöffnet.

Die Skizze hat die folgenden **Übergänge**:

- *v1*. Der erste Preis eines Artikels wird eingegeben.
- *v2*. Ein weiterer Preis wird eingegeben.
- *v3*. Der Knopf „Kassieren“ wird gedrückt.
- *v4*. Die Geldlade wird geschlossen.

Welchen Zustand könnte man hier als Endzustand wählen?

---

## EINGABEN

Wie eingangs beschrieben, reagiert ein Automat auf Aktionen. So wechselt der Parkscheinautomat zum Beispiel beim Einwurf von 5 Cent den Zustand. Eine Folge von Aktionen wird **Eingabe** genannt. So sind die Aktionen:

*[5 ct einwerfen – 10 ct einwerfen – 5 ct einwerfen – Knopf drücken – Parkschein entnehmen]*

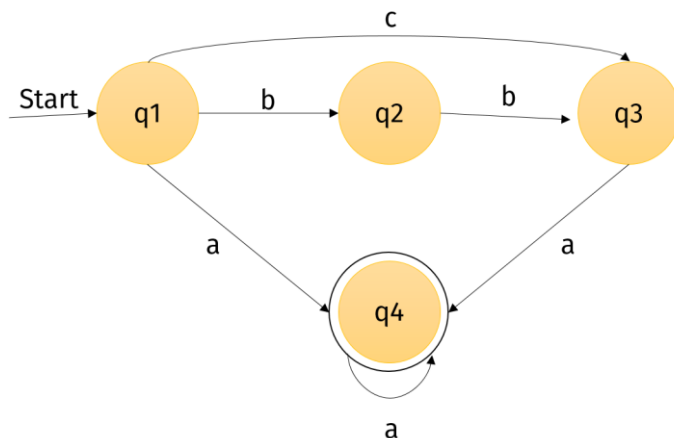
eine Eingabe.

Ein Automat verarbeitet eine Eingabe, indem er die einzelnen Aktionen der Reihe nach betrachtet und entsprechend reagiert. **Reagieren** heißt hier: Der Automat sucht einen Übergang, der vom aktuellen Zustand ausgeht und mit der Aktion, die an der Reihe ist, beschriftet ist.

Eingaben können unterschiedlich aussehen; es können sowohl Folgen von Zahlen, Wörtern, Zeichen oder Buchstaben sein. Bei realen Automaten sind dies Knöpfe, Münzen, Auswahl Tasten etc.

Um Automaten strukturiert und übersichtlich darstellen zu können, werden oft Kürzel an den Übergängen verwendet. Diese Standardisierung (Vereinheitlichung) ermöglicht außerdem die Analyse der Eigenschaften von Automaten. Im folgenden Beispiel werden die Übergänge durch Buchstaben abgekürzt:

## Der Automat



hat die folgenden Zustände:  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$ . Der Zustand  $q_1$  ist der Startzustand,  $q_4$  der Endzustand. Außerdem hat dieser Automat sechs Übergänge, die mit den Buchstaben  $a$ ,  $b$  und  $c$  markiert sind.

Wie reagiert der Automat nun auf die Eingabe  $caa$ ?

Vorgehen:

- Start bei  $q_1$ .
- Wechsel mit  $c$  von  $q_1$  zu  $q_3$ .
- Wechsel mit  $a$  von  $q_3$  zu  $q_4$ .
- Der Automat bleibt im Zustand  $q_4$  mit  $a$ .

Nach der Eingabe  $caa$  befindet sich der Automat also in dem Zustand  $q_4$ .

Die Eingabe  $bbb$  hingegen kann der Automat nicht verarbeiten, da es vom Zustand  $q_3$  aus keinen Übergang gibt, der mit  $b$  beschriftet ist.



### Aufgabe 2.3

Betrachte den Automaten aus dem obigen Beispiel! Wie reagiert der Automat auf die folgenden Eingaben?

- $bbaaa$
- $aa$
- $aab$
- $caaaa$

Wie bereits erwähnt, können Eingaben sehr unterschiedlich sein. Deswegen ist es nötig, zu jedem Automaten anzugeben, aus welchen Zeichen (z. B. Buchstaben) die **Sprache** besteht, die er versteht. Die Menge dieser Zeichen wird **Eingabealphabet** genannt. So ist das Eingabealphabet des Automaten aus dem Beispiel oben  $\{a, b, c\}$ . Das Eingabealphabet muss natürlich nicht aus Buchstaben bestehen, es kann sich auch aus ganzen Wörtern, Sätzen oder Zahlen zusammensetzen.

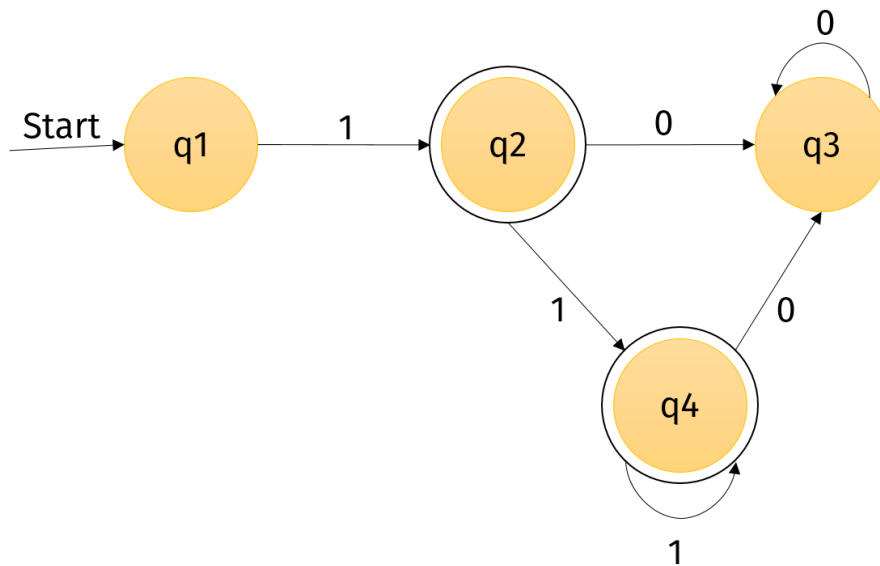


### Definition

Das **Eingabealphabet** eines Automaten ist die Menge an Zeichen (Buchstaben, Zahlen, Wörter, Symbole), auf die der Automat reagieren kann.

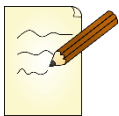


### Aufgabe 2.4

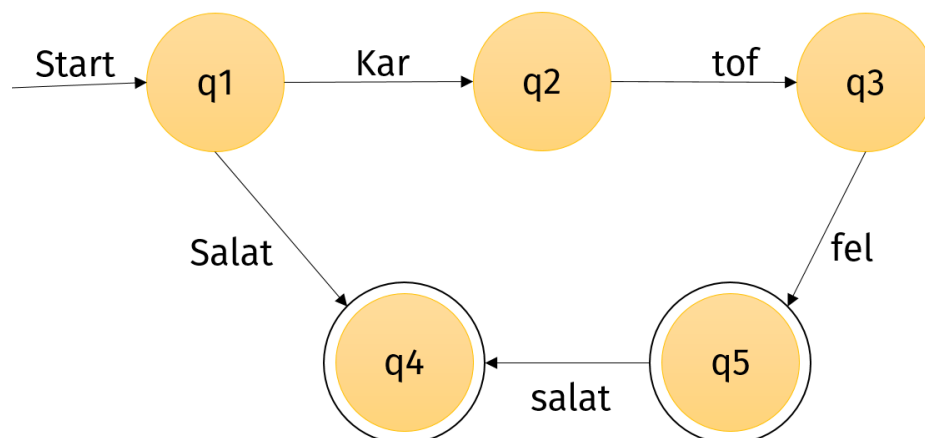


Nenne den Startzustand, den Endzustand und das Eingabealphabet dieses Automaten.

Beschreibe außerdem, wie der Automat sich auf die Eingaben 1111, 1100 bzw. 1001 verhält.



### Aufgabe 2.5



Nenne auch den Startzustand, die Endzustände und das Eingabealphabet dieses Automaten. Beschreibe außerdem das Verhalten des Automaten auf die folgenden drei Eingaben:

- *Kartoffelsalat*
- *Salat*
- *Nudelsalat*



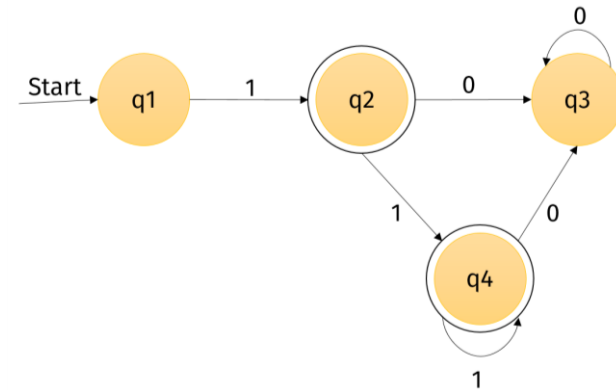
Diese Aufgabe kannst du bearbeiten, wenn du das vorherige Kapitel besonders schnell bearbeitet hast!



**Additum – Java 1**

In diesem Additum schreibst du am Computer ein Programm, das einen Automaten simuliert! Das Programm soll ausgeben, in welchem Zustand sich der Automat nach Bearbeitung einer bestimmten Eingabe befindet.

Betrachte erneut den Automaten aus Aufgabe 2.4:



Das Eingabealphabet ist die Menge {0, 1}.

Du sollst nun ein Javaprogramm schreiben, das diesen Automaten simuliert.

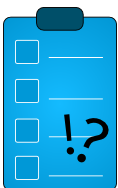
Lege dir dazu ein **Integerfeld** an, in dem die Eingabe gespeichert wird. Die Eingabe kannst du in dem Programm vorgeben, z. B. durch `int[] eingabe = {1, 0, 0, 1};`

Dein Programm soll nun ausgeben, in welchem Zustand sich der Automat befindet, nachdem er die Eingabe vollständig abgearbeitet hat. Die Länge der Eingabe kannst du ebenfalls in deinem Programm selbst festsetzen, so dass sie mit der Länge des Eingabefeldes übereinstimmt. Im Beispiel oben also: `int laenge = 4;`

Wenn du dich beim Programmieren unsicher fühlst, kannst du dir ein **Programmgerüst** bei deinem Lehrer oder deiner Lehrerin abholen.

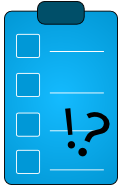
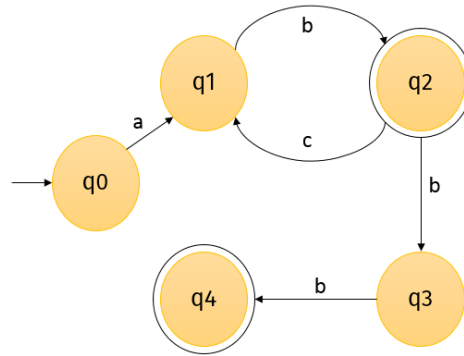
Die Lösung für diese Aufgabe bekommst du ebenfalls von deinem Lehrer bzw. deiner Lehrerin.

LERNFORTSCHRITTSKONTROLLE



**L 2.1**

Betrachte den folgenden Automaten und bestimme die Start- und Endzustände.

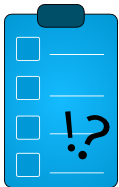


### L 2.2

Fülle folgenden Lückentext aus:

Ein Automat setzt sich aus \_\_\_\_\_ und \_\_\_\_\_ zusammen. Es gibt unterschiedliche Arten von Zuständen. Einer der „besonderen“ Zustände wird mit einem Pfeil gekennzeichnet; diesen Zustand nennt man \_\_\_\_\_. Ein weiterer „besonderer“ Zustand wird dagegen mit einem doppelten Kreis gekennzeichnet; hierbei handelt es sich um den \_\_\_\_\_.

Ein festgelegtes Schema gibt vor, wann ein Automat von einem \_\_\_\_\_ in einen anderen \_\_\_\_\_ übergeht. Allgemein sagt man, dass der Automat eine \_\_\_\_\_ aus dem \_\_\_\_\_ verarbeitet.



### L 2.3

Betrachte noch einmal den Automaten aus Aufgabe 2.1:

- a) In welchem Zustand befindet sich der Automat nach Bearbeitung der folgenden Eingaben?
  1. *abcbb*
  2. *abcbcbbb*
- b) Setze die folgenden Zeichenketten so fort, dass sich der Automat nach Bearbeitung der Zeichenkette in einem Endzustand befindet. Wähle dabei immer die kürzeste Möglichkeit.
  1. *a*
  2. *abc*
  3. *abb*



## Übersicht

Ein Automat soll Eingaben bezüglich bestimmter Eigenschaften unterscheiden. Dies wird realisiert durch das **Akzeptanzverhalten** des Automaten, das in diesem Kapitel genauer unter die Lupe genommen wird. Demonstriert wird dies mit Hilfe des Programms **JFLAP**, das diese Untersuchung vereinfacht.



## Teillernziele

Nach der Bearbeitung dieses Kapitels kannst du...

- die Begriffe „akzeptieren“ und „verwerfen“ verstehen und erläutern.
- untersuchen, ob ein Automat ein Wort akzeptiert oder verwirft.
- Automatenmodelle in JFLAP überführen und testen.

## AKZEPTANZVERHALTEN

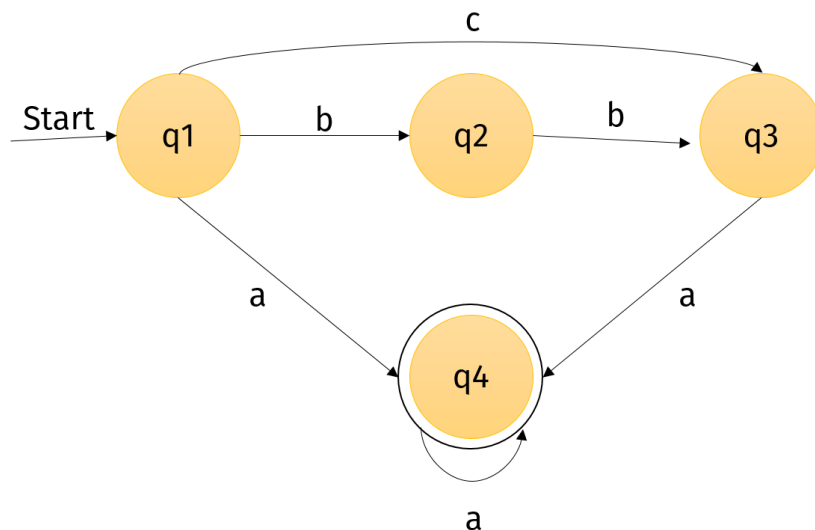
Die Aufgabe eines Automaten besteht oft darin, eine Eingabe auf Korrektheit zu überprüfen. Eine Eingabe besteht aus einer Folge von Zeichen aus dem Eingabealphabet; sie wird genau dann von dem Automaten akzeptiert, wenn der Automat einen Endzustand erreicht.



## Definition

Der Automat **akzeptiert** das Eingabewort genau dann, wenn er sich nach dem Einlesen des ganzen Wortes in einem **Endzustand** befindet. Ansonsten akzeptiert er das Wort nicht. Man sagt deshalb auch, dass der Automat in diesem Fall das Eingabewort **verwirft**.

Betrachte noch einmal den Automaten aus Kapitel 2:



Akzeptiert dieser Automat das Eingabewort *bba*?

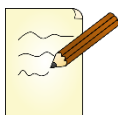
- Start bei  $q_1$ .
- $b$  wird gelesen  $\rightarrow$  Wechsel zu  $q_2$
- zweites  $b$  wird gelesen  $\rightarrow$  Wechsel in  $q_3$
- $a$  wird gelesen  $\rightarrow$  Wechsel zu  $q_4$

Nach dem Einlesen der Zeichenfolge befindet sich der Automat im Endzustand  $q_4$ . Also akzeptiert er das Wort  $bba$ .

Akzeptiert der Automat auch das Wort  $bb$ ?

Nach dem Lesen der Zeichenkette  $bb$  befindet sich der Automat im Zustand  $q_3$ . Da  $q_3$  kein Endzustand ist, akzeptiert er das Wort  $bb$  nicht.

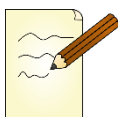
In dem obigen Beispiel hat der Automat erst mit dem Lesen des letzten Zeichens in den Endzustand gewechselt. Das ist nicht immer so. Es können auch **mehrere Endzustände** durchlaufen werden. Wichtig für das Akzeptanzverhalten ist nur, in welchem Zustand sich der Automat befindet, wenn er die Eingabe komplett abgearbeitet hat. Nach kompletter Abarbeitung der Eingabe muss der Automat sich in einem Endzustand befinden.



### Aufgabe 3.1

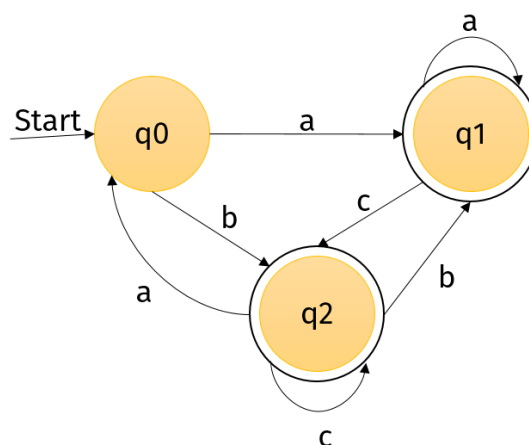
Überprüfe, welche der folgenden Wörter der Automat aus dem obigen Beispiel akzeptiert!

- $aaaaa$
- $b$
- $ca$



### Aufgabe 3.2

Betrachte folgenden Automaten:



Überprüfe, welche der folgenden Wörter der Automat akzeptiert!

- aaaaaac*
- bccb*
- ba*

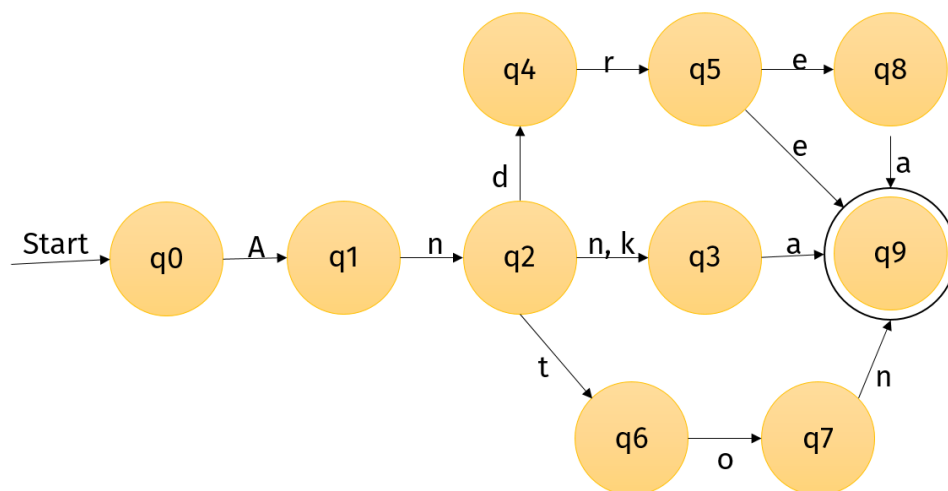
Betrachte ein weiteres Beispiel:

Alle Oberstufenschülerinnen und -schüler des Gymnasiums in Poppelsdorf machen einen Schulausflug. Die Schüler, die sich angemeldet und den Beitrag bezahlt haben, stehen auf einer Liste.

Der Schulcomputer verfügt über ein Programm, das nach der Eingabe eines Namens anzeigt, ob der Schüler bzw. die Schülerin mit auf den Ausflug kommen kann oder nicht. Das Programm simuliert also einen Automaten. Anna ist sich nicht mehr sicher, ob sie wirklich das Geld für den Ausflug bezahlt hat. Deswegen tippt sie ihren Namen in den Computer ein.

Du wirst dich nun mit dem Teil des Automaten beschäftigen, der über die Schüler mit Anfangsbuchstaben *An* Auskunft gibt.

Das Eingabealphabet besteht aus allen Buchstaben des Alphabets: {a, A, b, B, ..., z, Z}. Dieser Automat akzeptiert *Anna* genau dann, wenn er nach Bearbeitung der Eingabe des Wortes in einem Endzustand landet. Verfolge die Reaktion des Automaten:



- Start in *q0*
- *A* wird gelesen → Wechsel zu *q1*
- *n* wird gelesen → Wechsel zu *q2*
- *n* wird gelesen → Wechsel zu *q3*
- *a* wird gelesen → Wechsel zu *q9*

Da *q9* ein Endzustand ist, akzeptiert der Automat die Eingabe *Anna*. Anna darf also mitfahren.



### **Aufgabe 3.3**

Welche Schüler, deren Namen mit *An* beginnt, dürfen auch noch mit auf den Schulausflug fahren? Nenne die Namen.



### **Aufgabe 3.4**

Anne und Anke wollen auch noch mit auf den Ausflug fahren. Sie melden sich deshalb an und bezahlen den Beitrag. Verändere den Automaten so, dass er auch die Eingaben *Anne* und *Anke* akzeptiert.



## Übungsaufgaben

Nun wirst du dich mit der Simulation von Automaten beschäftigen. Das Programm JFLAP ermöglicht dir, Automaten zu entwerfen und ihr Akzeptanzverhalten zu testen. Im folgenden Beispiel lernst du, JFLAP selber zu bedienen. Vollziehe die im Beispiel beschriebenen Schritte nach, um den Umgang mit dem Programm zu lernen.

### JFLAP erkunden und testen

Um das Programm JFLAP kennenzulernen, beschäftigst du dich noch einmal mit dem Beispiel zum Schulausflug.

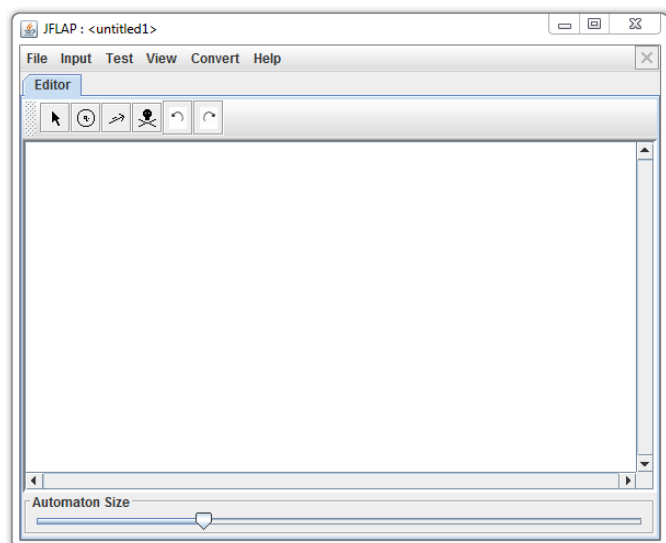
Als Erstes startest du JFLAP. Wenn du dabei Probleme hast, frag einfach deinen Lehrer/deine Lehrerin!

Nach dem Start des Programms erscheint zunächst ein Fenster, in dem man die Art des Automaten auswählen kann.



Da du dich erst einmal nur mit **endlichen Automaten** befasst, wählst du den Button „**Finite Automaton**“.

Es erscheint ein weiteres Fenster zur Erstellung des Automaten.

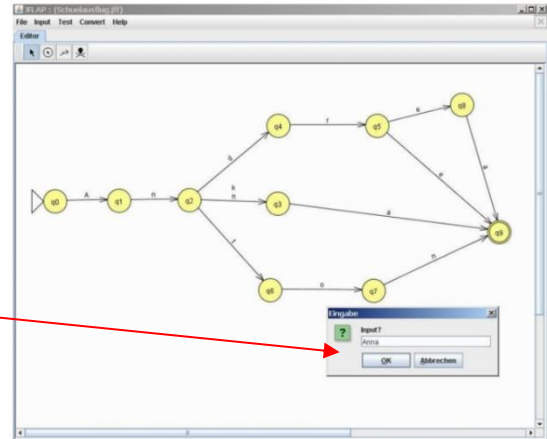


Wie du einen Automaten erstellst, lernst du im nächsten Beispiel. Zunächst wirst du lernen, einen bereits erstellten Automaten zu testen.

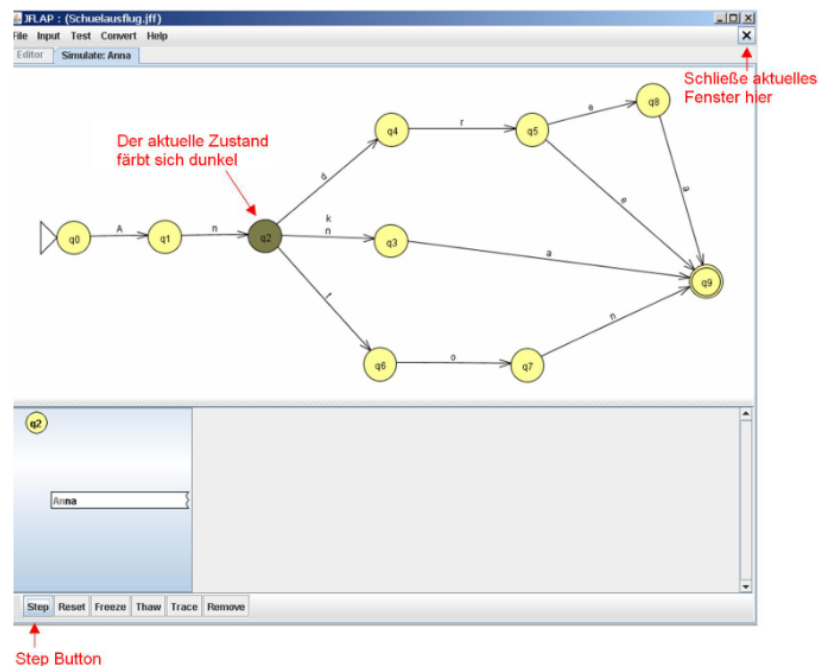
Gehe dazu im Menü auf **File** → **open** und öffne die Datei **Schulausflug.jff**. Nun siehst du den Automaten aus dem Beispiel mit dem Schulausflug.

Jetzt sollst du das Verhalten des Automaten testen. JFLAP bietet dafür zwei Möglichkeiten:

1. Der Automat arbeitet das Wort schrittweise ab. Gehe dazu in der Menüleiste des bereits geöffneten Fensters auf **Input** → **Step by State**. Es erscheint ein Eingabefenster. Gebe dort *Anna* ein.



Ein weiteres Fenster öffnet sich:

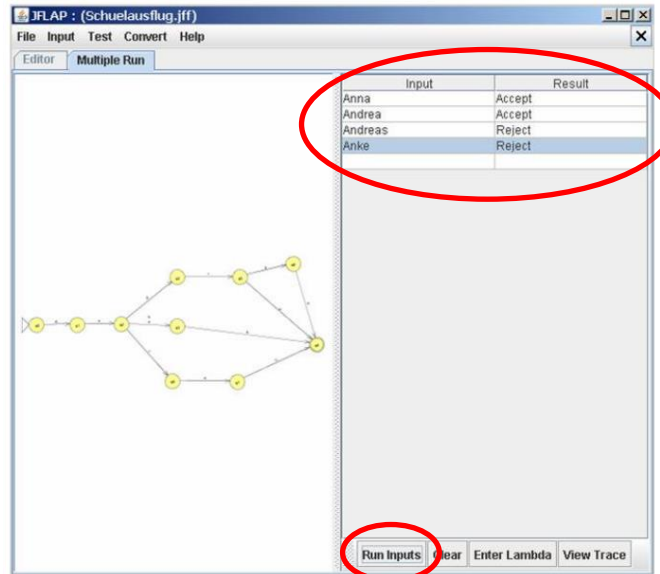


Hier kannst du das Verhalten des Automaten schrittweise nachvollziehen, indem du auf den Button **Step** klickst. Um den Automaten erneut mit der gleichen Eingabe zu starten, musst du **Reset** drücken.

Starte nun auf gleiche Weise den Automaten einmal mit der Eingabe *Anke* und einmal mit der Eingabe *Andreas*. Dabei wirst du feststellen, dass der Automat sich bei der Eingabe *Anke* grün färbt; die Eingabe wird also akzeptiert. Die Eingabe *Andreas* kann der Automat dagegen nicht verarbeiten. Deshalb stoppt er bei  $q_8$ , egal wie oft der Step-Button jetzt noch gedrückt wird.



2. Es werden **mehrere Eingaben gleichzeitig** getestet.  
Gehe dazu auf **Input** → **Multiple Run**.



Das Fenster wird in zwei Unterfenster geteilt. Auf der linken Seite siehst du weiterhin deinen Automaten. Auf der rechten Seite erscheint eine Input- und eine Result- Spalte. In der **Input-Spalte** kannst du nun untereinander mehrere Eingaben eintragen. Jede Eingabe beendest du mit einem Return.

Gebe nun in die Input-Spalte die Eingaben *Anna*, *Andrea*, *Andreas* und *Anke* ein. Durch das Klicken auf den **Run-Inputs-Button** werden deine Eingaben ausgewertet.

Die Auswertung, ob dein Automat die Eingaben akzeptiert oder nicht, kannst du in der **Result-Spalte** ablesen. Steht dort *Accept*, so akzeptiert der Automat die entsprechende Eingabe. Steht dort *Reject*, so akzeptiert der Automat die Eingabe nicht, sondern verwirft sie.



### Aufgabe 3.5

Überprüfe mit Hilfe von JFLAP, ob der Schulausflug-Automat folgende Eingaben akzeptiert:

1. *Anton*
2. *Anika*
3. *Andre*



### Aufgabe 3.6

Nun eine etwas theoretischere Aufgabe. Wie du weißt, werden Zahlen im Rechner als Binärzahlen abgespeichert. Das heißt, jede Zahl wird mit Hilfe von Nullen und Einsen codiert. Mit dem folgenden Automaten lässt sich testen, ob eine binär codierte Zahl gerade oder ungerade ist, denn dieser Automat akzeptiert nur ungerade Zahlen. Beachte dabei,

dass eine Binärzahl genau dann gerade ist, wenn das letzte Bit eine 0 ist. Ist das letzte Bit eine 1, so ist die Zahl ungerade.

Teste das Akzeptanzverhalten des Automaten „ungerade“ mit Hilfe von JFLAP für die folgenden Eingaben:

1. 0010
2. 1000101
3. 1000000

## JFLAP – EIGENE AUTOMATEN ENTWERFEN



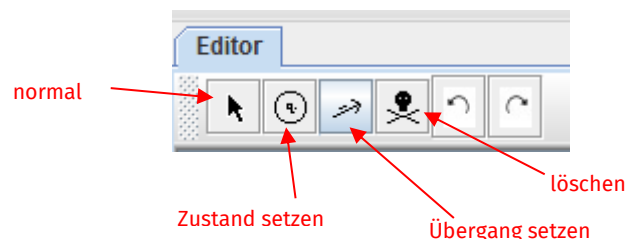
### Übungsaufgaben am Computer

Nachdem du jetzt schon fertige Automaten mit JFLAP testen kannst, wirst du nun lernen, auch eigene Automaten mit der Software zu entwickeln. Vollziehe hierzu zunächst wieder die Schritte des Beispiels nach.

Der Automat, den du erstellst, hat als Eingabealphabet die Menge  $\{0\}$ . Er soll vorerst nur das Wort  $00$  erkennen.

Geh auf JFLAP, **Finite State Automaton**.

Nun erscheint wieder das Fenster, in dem du einen Automaten modellieren kannst. Links oben siehst du mehrere Symbole. Einen Mauspfeil, einen Kreis mit einem eingezeichneten Mittelpunkt, einen langgezogenen Pfeil und einen Totenkopf. Jedes der Symbole stellt einen bestimmten **Modus** dar.



Wenn du mit der linken Maustaste den **Kreis** anklickst, bist du in dem Modus, in dem **Zustände** erzeugt werden können.

Gehe nun zuerst auf diesen Kreis, um in den Modus *Zustand setzen* zu wechseln. Gehe dann auf die **weiße Zeichenfläche** und klicke auf die linke Maustaste. Schon ist der erste Zustand erzeugt, der mit  $q_0$  bezeichnet wird. Setze rechts daneben noch drei weitere Zustände, so dass du schließlich  $q_0$  bis  $q_3$  gesetzt hast.

Eigentlich benötigst du aber nur die Zustände  $q_0$  und  $q_1$ . Wechsle deshalb in den *Lösch-Modus*, indem du den **Totenkopf** anklickst. Nun kannst du die überflüssigen Zustände  $q_2$  und  $q_3$  durch Anklicken **löschen**.

Jetzt fehlen noch die **Übergänge**. Um diese zu zeichnen, musst du in den Modus *Übergänge setzen* gehen, indem du den **länglichen Pfeil** anklickst. Setze einen Übergang von  $q_0$  zu  $q_1$ , indem du den Zustand  $q_0$  anklickst und die linke Maustaste gedrückt hältst. Ziehe dann die Maus zum Zustand  $q_1$  und lasse erst dann die gedrückte Maustaste wieder

los. Es erscheint ein **Eingabefeld**. Klicke dieses an, schreibe eine 0 hinein und drücke *Enter*. Damit ist dein Übergang gesetzt. Setze analog auch noch den zweiten Übergang.

Falls du irrtümlich einen falschen Übergang setzt, kannst du diesen auch löschen, indem du in den *Lösch-Modus* wechselst und den entsprechenden Übergang anklickst.

Möchtest du einen Übergang von einem Zustand zu einem anderen, der mit mehreren Symbolen beschriftet ist, machen, kannst du einfach mehrere Übergänge zwischen den beiden Zuständen erzeugen und jeden dieser Übergänge mit einem Symbol beschriften.

Nun kannst du den Automaten noch etwas mehr in die Mitte **verschieben**. Wechsle dazu in den Modus *normal*. Nun kannst du die Zustände anklicken, und während du die Maus-taste gedrückt hältst, verschieben.

Dieser Modus hat noch eine weitere Eigenschaft: Wenn du dich bei der Beschriftung ver-schrieben hast, kannst du diese in diesem Zustand anklicken und erneut beschriften.

Außerdem kannst du in dem Modus *normal* festlegen, dass  $q_0$  dein **Anfangszustand** sein soll. Klicke dazu mit der rechten Maustaste auf  $q_0$ . Es erscheint ein Menü. Wähle **Initial** (deutsch: Anfangs-) aus. Auf  $q_0$  zeigt jetzt ein großer Pfeil. Analog klickst du nun mit der rechten Maustaste auf  $q_2$  und markierst diesen als **Endzustand**, indem du im Menü auf **Final** (deutsch: End-) klickst.

Schon ist dein erster Automat mit JFLAP fertig und kann getestet werden.

Hier noch einmal ein kurzer Überblick über die verschiedenen **Funktionen von JFLAP**.

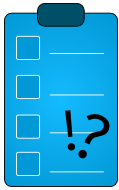
- **Zustände setzen:** Gehe in den Modus *Zustand setzen*, indem du den Kreisbutton anklickst. Dann kannst du beliebig viele Zustände per Mausklick setzen.
- **Übergänge setzen:** In den Modus *Übergänge setzen* gelangst du, indem du den Button mit dem länglichen Pfeil anklickst. Dann kannst du den Zustand, von dem der Übergang ausgehen soll, anklicken und die Maustaste gedrückt halten. Lass sie erst wieder los, wenn du am Zielzustand angekommen bist. So kannst du auch einen Übergang von einem Zustand zu sich selbst setzen.
- **Zustände/Übergänge löschen:** In den *Lösch-Modus* kommst du, indem du den Totenkopf-But-ton anklickst. In diesem Modus kannst du die Zustände/Übergänge anklicken, die du löschen willst.
- **Zustände bearbeiten:** Klickt man einen Zustand mit der rechten Maustaste an, wird ein Menü angezeigt. Bei weiterhin gedrückter rechter Maustaste können z. B. die folgenden Menüpunkte ausgewählt werden:
  - Initial: Anfangszustand setzen
  - Final: Endzustand setzen
  - Über den Menüpunkt **File** → **Save as** kann das Modell gespeichert werden.



### Aufgabe 3.7

Beschäftige dich noch einmal mit dem Schulausflug-Automaten. Lade dir zuerst erneut den Automaten „Schulausflug“ in dein JFLAP-Programm. Wie du bereits weißt, wollen

Anke und Anne auch an dem Ausflug teilnehmen. Ändere den Automaten deshalb so ab, dass er auch *Anke* und *Anne* akzeptiert!

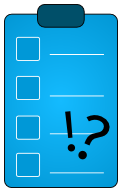


**L 3.1**

Vervollständige den folgenden Lückentext:

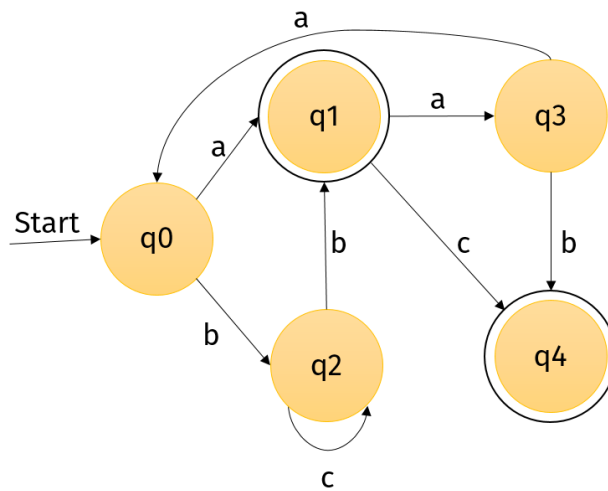
Ein Automat \_\_\_\_\_ das Eingabewort genau dann, wenn er sich nach dem Einlesen des ganzen Wortes in einem Endzustand befindet.

Ansonsten \_\_\_\_\_ er das Wort nicht. Man sagt auch, der Automat \_\_\_\_\_ in diesem Fall das Eingabewort.



**L 3.2**

Betrachte den folgenden Automaten:



Welche der folgenden Eingaben akzeptiert der Automat? Kreuze an.

- aaaaab*
- bcc*
- bccbc*
- aaab*
- bcbab*
- bbaa*



### Übersicht

Automaten werden z. B. als Grundlage für Compiler verwendet. Ein Compiler soll erkennen, ob die Syntax der in den Computer eingegebenen Codes richtig ist. Der zugrundeliegende Automat muss also gewährleisten, dass nur Codebausteine, die programmiersprachenkonform sind, akzeptiert werden. Ein solcher Automat muss somit eine vorgegebene Sprache akzeptieren.



### Lernziel

Nach diesem Kapitel kannst du...

- endliche Automaten als Graphen darstellen.
- natürliche und formale Sprachen unterscheiden und außerdem Begriffe einer formalen Sprache wie z. B. *Sprache des Automaten* oder *Wort* verwenden.
- Sprachen von Automaten beschreiben.
- zu einer gegebenen Sprache einen Automaten konstruieren.

---

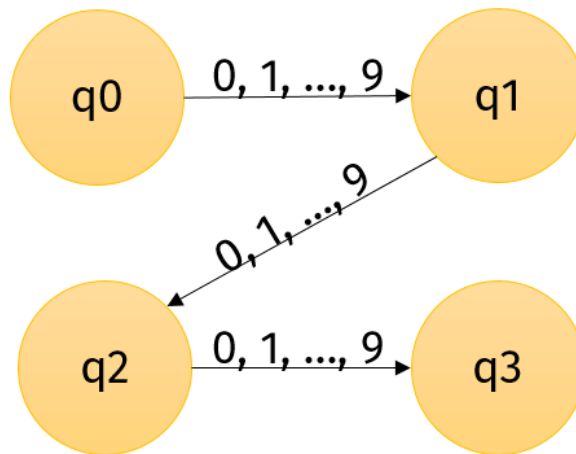
## AUTOMATEN KONSTRUIEREN

Du hast bis jetzt viele verschiedene Automaten kennengelernt. Außerdem hast du im letzten Kapitel gelernt, wie man Automaten verändert und damit auch das Akzeptanzverhalten des Automaten beeinflusst. Auf diesem Wissen baust du nun auf, denn jetzt konstruierst du deinen Automaten von Grund auf selber! Lies dazu zunächst noch ein Beispiel durch:

Es soll ein Automat konstruiert werden, der eine vierstellige Zahl übergeben bekommt. Der Automat soll diese Zahl akzeptieren, wenn sie gerade ist und sie verwerfen, wenn sie ungerade ist. Man kann diesem Automaten außerdem ausschließlich vierstellige Zahlen übergeben; so muss zum Beispiel für die Zahl 100 die vierstellige Zahl 0100 eingegeben werden. Die vorderen Nullen haben dabei keinerlei Bedeutung.

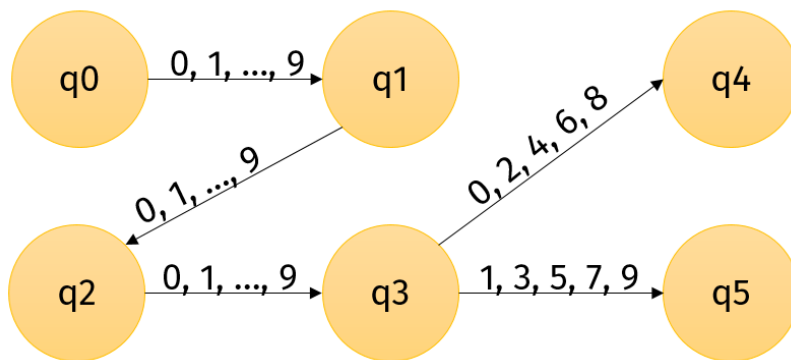
Wie muss aber ein Automat aussehen, der nur gerade Zahlen akzeptiert?

Natürlich ist bei dieser Frage nur die letzte Ziffer der Zahl interessant, denn schon an dieser kann man ablesen, ob eine Zahl gerade ist, oder nicht. Die ersten drei Ziffern der Zahl sind also für den Automaten uninteressant. Das lässt sich folgendermaßen darstellen:

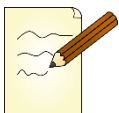
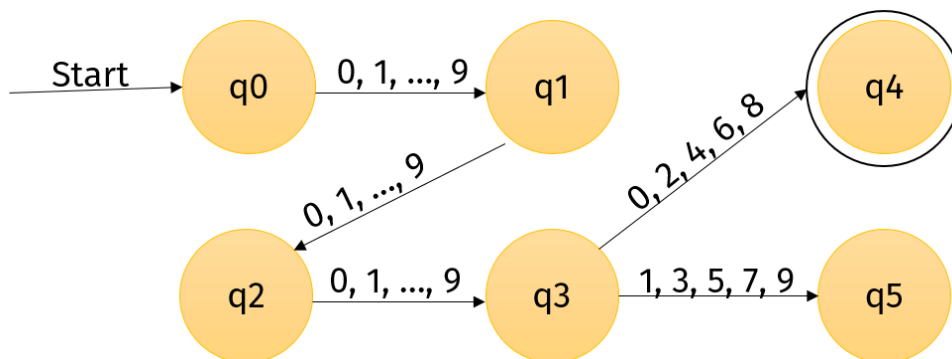


Dieser Automat zählt praktisch nur bis drei und ignoriert alles andere. Nur die letzte, die vierte Ziffer, ist von Bedeutung für unser Beispiel. Denn wenn diese Ziffer gleich 0, 2, 4, 6, oder 8 ist, ist die Zahl gerade. Unser Automat soll in diesem Fall die Zahl somit akzeptieren.

Ist die letzte Ziffer eine 1, 3, 5, 7 oder 9 – ist die Zahl also ungerade – soll der Automat die Zahl verwerfen. Diese Spaltung zwischen geraden und ungeraden Zahlen lässt sich folgendermaßen umsetzen:



Jetzt muss noch der Endzustand richtig gesetzt und der Startzustand markiert werden. Der fertige Automat sieht dann folgendermaßen aus:



#### Aufgabe 4.1

Entwirf nun selber einen Automaten, der als Eingabe Zahlen mit drei Stellen übergeben bekommt und der alle Zahlen, die echt kleiner als 40 (also 040) sind, akzeptiert!

Erstelle diesen Automaten mit Papier und Bleistift!



#### Aufgabe 4.2

Du hast bereits in Kapitel 3 einen Automaten kennengelernt, der nur ungerade binär codierte Zahlen akzeptiert hat. Nun ist es an dir, in JFLAP einen Automaten zu erstellen, der nur gerade binär codierte Zahlen akzeptiert. Das heißt, dass die Binärzahl am Ende eine 0 haben muss.

Entwickle mit Hilfe von JFLAP einen solchen Automaten.

### DIE SPRACHE EINES AUTOMATEN

In Kapitel 2 wurde bei der Einführung von Eingaben den Begriff *Sprache* erwähnt. Mit diesem Begriff wirst du dich nun ein bisschen genauer beschäftigen, denn er kann bei der Erstellung von Automaten sehr behilflich sein!

Zu jedem Automaten lässt sich eine Menge von Wörtern finden, die dieser Automat akzeptiert. Diese Menge wird **Sprache** des Automaten genannt. Man sagt auch: Der Automat erkennt diese Sprache.

Ein **Wort** ist eine Zeichenkette, die aus beliebigen Zeichen bestehen kann. Wörter können zum Beispiel so aussehen:

- *Baum*
- *abbc*
- *011BK5*

Betrachte noch einmal den Automaten aus dem ersten Beispiel dieses Kapitels. Dieser Automat akzeptiert alle geraden Zahlen mit vier Stellen. Man kann diese Zahlen mit der folgenden Menge beschreiben: {0000, 0002, 0004, ..., 9996, 9998}. Diese Menge ist also die Sprache, die dieser Automat erkennt.

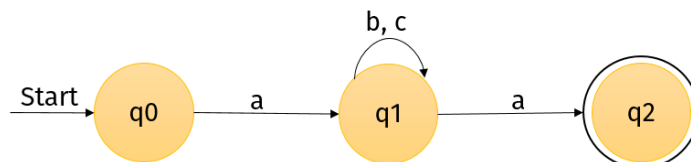


#### Definition

**Wort:** Zeichenkette, die aus beliebigen Zeichen besteht.

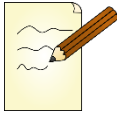
**Sprache eines Automaten:** die Menge von Wörtern, die der Automat akzeptiert. Man sagt, der Automat **erkennt** diese Sprache.

Betrachte noch ein weiteres Beispiel eines Automaten mit dem Eingabealphabet {a, b, c}:



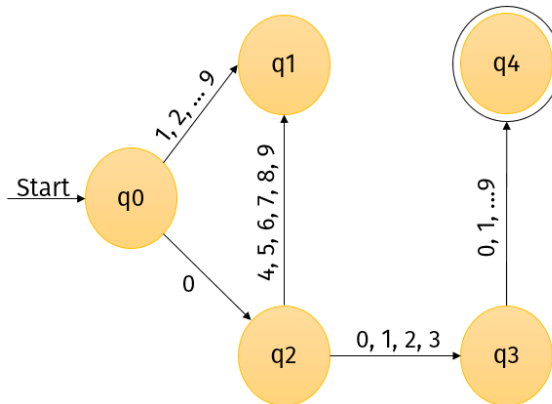
Dieser Automat akzeptiert alle Wörter, die mit *a* anfangen, mit *a* aufhören und in deren Mitte keine weiteren *a*'s vorkommen, sondern nur beliebig viele *b*'s und *c*'s. Er erkennt also die Sprache, die aus der Menge der oben beschriebenen Wörter besteht.





### Aufgabe 4.3

Betrachte noch einmal den Automaten aus Aufgabe 1 dieses Kapitels und ermittle die Sprache, die dieser Automat erkennt:



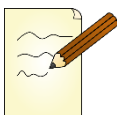
In den folgenden Aufgaben entwickelst du selbst zu einer gegebenen Sprache einen Automaten.



### Aufgabe 4.4

Entwickle einen Automaten mit dem Eingabealphabet  $\{a, b\}$ , der die folgende Sprache erkennt: Die Menge aller Wörter, die vorne und hinten beliebig viele  $a$ 's und in der Mitte irgendwo genau ein  $b$  haben.

Achtung: Beliebige viele  $a$ 's heißt immer, dass auch kein  $a$  an dieser Stelle sein kann!



### Aufgabe 4.5

In einer Firma möchte ein Chef allen Mitarbeitern, die 2006 in der Firma angefangen haben zu arbeiten, einen Bonus auszahlen. Um nicht jede einzelne Personalakte lesen zu müssen, will er ein Programm schreiben, das diese Aufgabe für ihn übernimmt. Dieses Programm soll auf Grundlage eines Automaten arbeiten und alle Datensätze der Mitarbeiterdatenbank überprüfen. Als Eingabe erhält dieser Automat ein Datum. Es werden also grundsätzlich nur Eingaben der Form  $TT.MM.2006$  akzeptiert. Er soll die Eingabe akzeptieren, falls sie ein korrektes Datum aus dem Jahr 2006 ist.

Achtung: Beachte Feinheiten, wie die unterschiedlichen Monatslängen. Die Punkte der Datumseingabe brauchst du aber in deinem Programm zunächst noch nicht zu berücksichtigen.

## ADDITUM



### Additum – Java 2

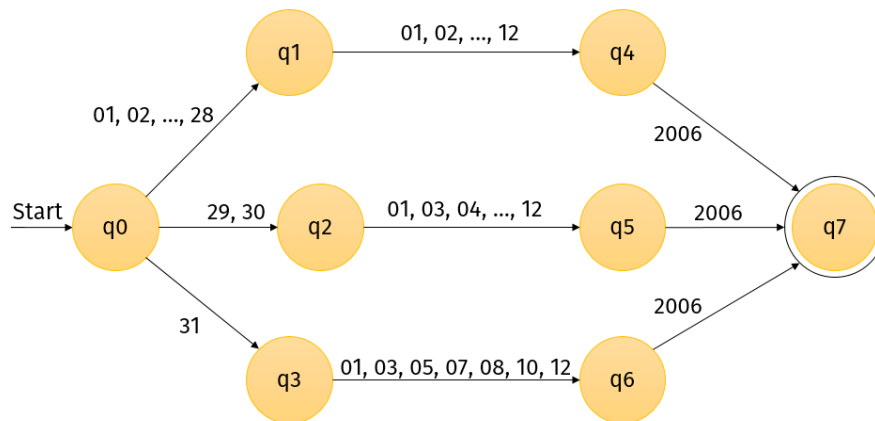
Auch in diesem Additum geht es um die Simulation eines Automaten mit einem Java-Programm.

Du schlüpfst in die Rolle des Chefs, der mit Hilfe eines Automaten die Mitarbeiter ermitteln möchte, die 2006 begonnen haben, in seiner Firma zu arbeiten. Denke daran, dass dieser Automat nur Daten akzeptiert, die in der Form *TT.MM.2006* sind.

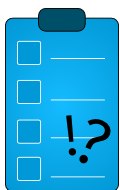
Um das Programmieren des Automaten zu vereinfachen, lässt du vorerst ein Programm über alle Daten laufen, welches die Punkte aus jedem Datum ausliest und die Zahlen in einem Feld mit drei Einträgen abspeichert. So wird aus 04.12.2006 das Feld {4,12, 2006}. Die Eingabe kannst du in dem Programm vorgeben, z. B. durch `int[] datum = {4,12,2006};`

Das Programm soll nun prüfen, ob der Automat die gegebene Eingabe akzeptiert.

Betrachte als Programmierhilfe den folgenden Automaten:



## LERNFORTSCHRITTSKONTROLLE



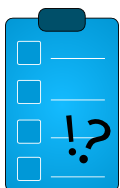
### L 4.1

Vervollständige den folgenden Text!

Eine Zeichenkette, die aus beliebig vielen Zeichen besteht, nennt man auch

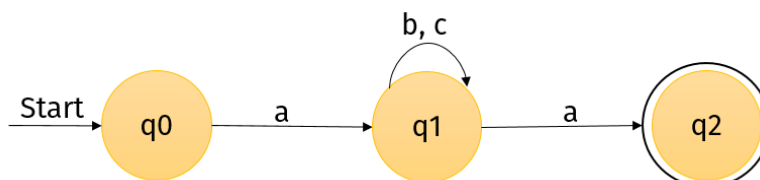
\_\_\_\_\_.

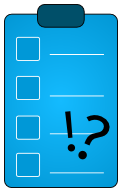
Die Menge von Wörtern, die der Automat akzeptiert, wird \_\_\_\_\_ eines Automaten genannt. Man sagt auch, der Automat \_\_\_\_\_ die Sprache.



### L 4.2

Beschreibe die Sprache des folgenden Automaten!





### L 4.3








Erstelle einen Automaten, der alle Wörter akzeptiert, die eine gerade Anzahl von  $a$ 's enthalten. Das Eingabealphabet ist dabei die Menge  $\{a, b, c\}$ .

Der Automat soll zum Beispiel das Wort  $abacbaa$  akzeptieren, aber das Wort  $abacba$  verwerfen.

## QUELLENVERZEICHNIS

- Gesellschaft für Informatik (GI) e. V.: *Bildungsstandards Informatik für die Sekundarstufe II*. Berlin: LOG IN 2016. Online: [https://www.informatikstandards.de/docs/Bildungsstandards\\_SII.pdf](https://www.informatikstandards.de/docs/Bildungsstandards_SII.pdf) (eingesehen: 25.06.2018).
- Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen: *Kernlehrplan für die Sekundarstufe II Gymnasium/Gesamtschule in Nordrhein-Westfalen. Informatik*. 1. Aufl., Düsseldorf: 2014. Online: [https://www.schulentwicklung.nrw.de/lehrplaene/upload/klp\\_SII/if/KLP\\_GOSt\\_Informatik.pdf](https://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/if/KLP_GOSt_Informatik.pdf) (eingesehen: 25.06.2018).

## ABBILDUNGSVERZEICHNIS

- Abbildung 1: Parkautomat, Quelle: pixabay.org, <https://pixabay.com/de/parkuhr-stra%C3%9Fe-asphalt-altstadt-102361/> (CC0, eingesehen: 25.06.2018).
- Abbildung 2: Flipperautomat, Quelle: pixabay.org, <https://pixabay.com/de/flipper-pin-maschine-jahrgang-296568/> (CC0, eingesehen: 09.07.2018).
- Die Screenshots zur Erklärung der Software wurden durch das InfoSphere-Team m. H. von JFLAP erstellt, <http://www.softpedia.com/get/Others/Home-Education/JFLAP.shtml> (eingesehen: 02.07.2018).
- Alle weiteren Grafiken (, , , , , , ) , Quelle: InfoSphere.