# Automaten Entwerfen II - formale Sprachen

## 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.

# Los gehts…

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 die ersten drei Ziffern drei und ignoriert alle anderen Eigenschaften der Eingaben. 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 Unterscheidung 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

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 zunächst mit Papier und Bleistift, übertrage ihn dann in JFLAP und teste ihn!

# Die Sprache eines Automaten

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 des Eingabealphabets 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.

**Wort:** Zeichenkette, die aus beliebigen Zeichen des Eingabealphabets besteht. **Sprache eines Automaten:** die Menge von Wörtern, die der Automat akzeptiert.

Man sagt, der Automat erkennt diese Sprache.

# Beispiel

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.

# Aufgaben

Entwerfe die Automaten auf Papier, teste mit JFLAP

---

**(A1)**

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: Beliebig viele a’s heißt immer, dass auch kein a an dieser Stelle sein kann!

---

**(A2)**

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.

—-

## (A3)

Erstelle einen Automaten, der alle Wörter akzeptiert, die eine gerade Anzahl von a’s ent- halten. Das Eingabealphabet ist dabei die Menge `{a,b,c}`.

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

From:
https://wiki.qg-moessingen.de/ - **QG Wiki**

Permanent link:
**https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:automaten:lepro:erstellung2:start?rev=1652944766**

Last update: **19.05.2022 09:19**