

# Übungen mit JFLAP

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. ([Download<sup>1\)</sup>](#))

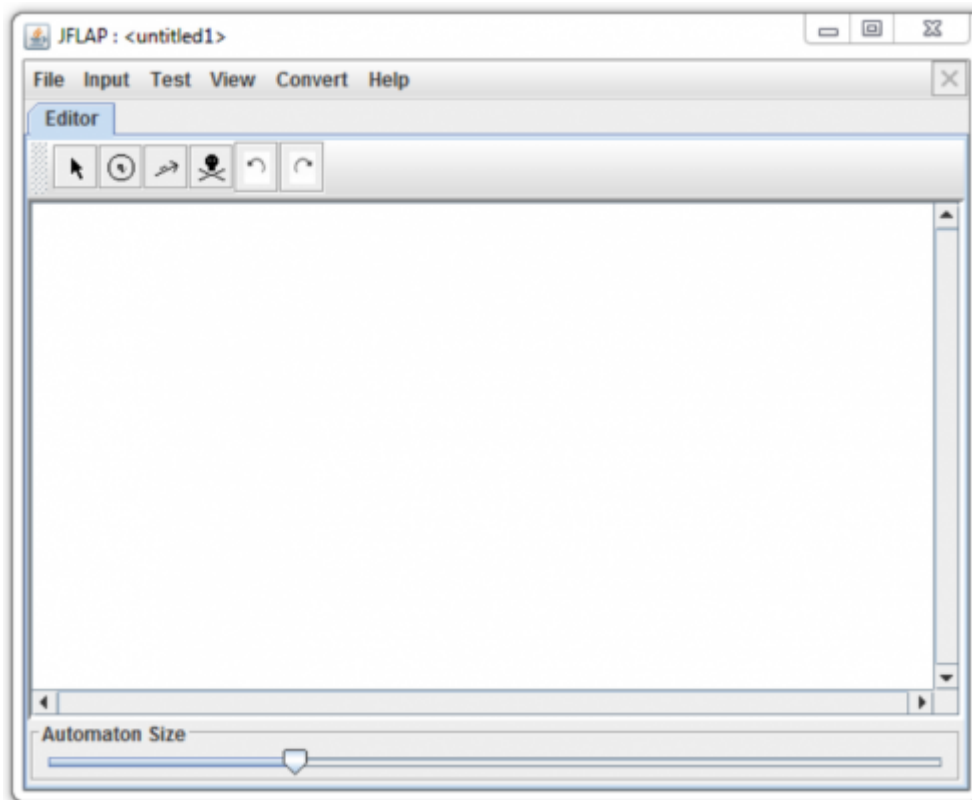
## Erster Start

JFLAP kommt als .jar-Datei, diese kann man auf einer Konsole ausführen mit `java -jar JFLAP7.1.jar`.

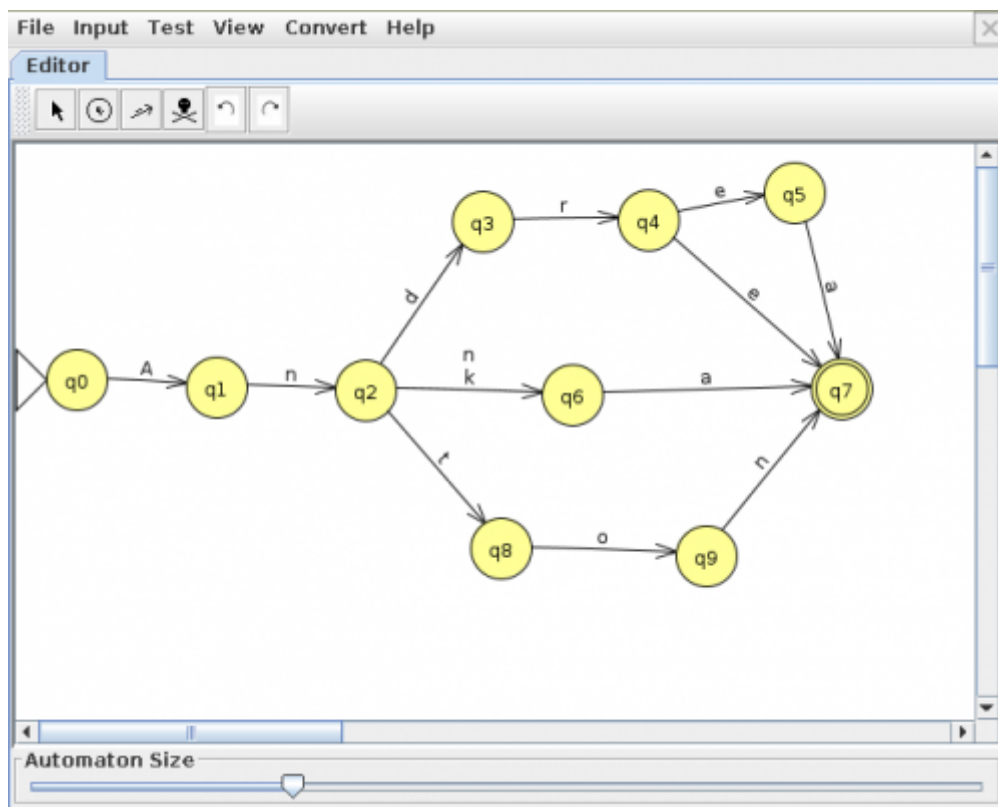
Nach dem Start des Programms erscheint zunächst ein Fenster, in dem man die Art des Automaten auswählen kann. Da du dich erst einmal nur mit **endlichen Automaten** befasst, wählst du den Button **Finite Automaton**.



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.



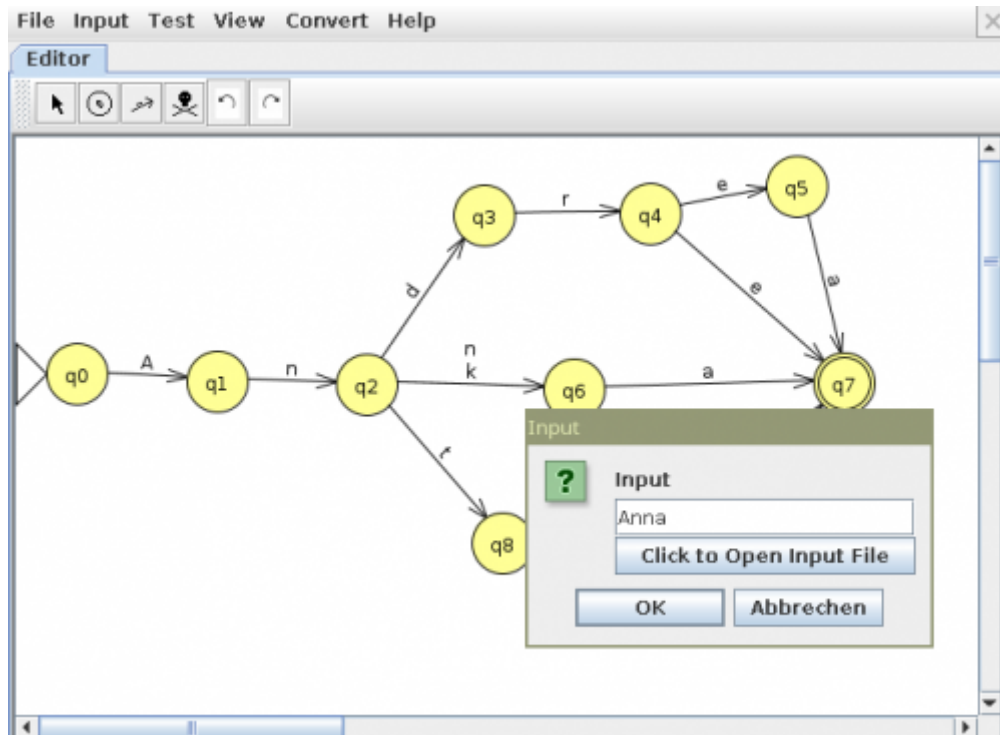
Gehe dazu im Menü auf File → open und öffne die Datei Schulausflug.jff, diese findest du in [diesem ZIP-Archiv](#). Nun siehst du den Automaten aus dem Beispiel mit dem Schulausflug.



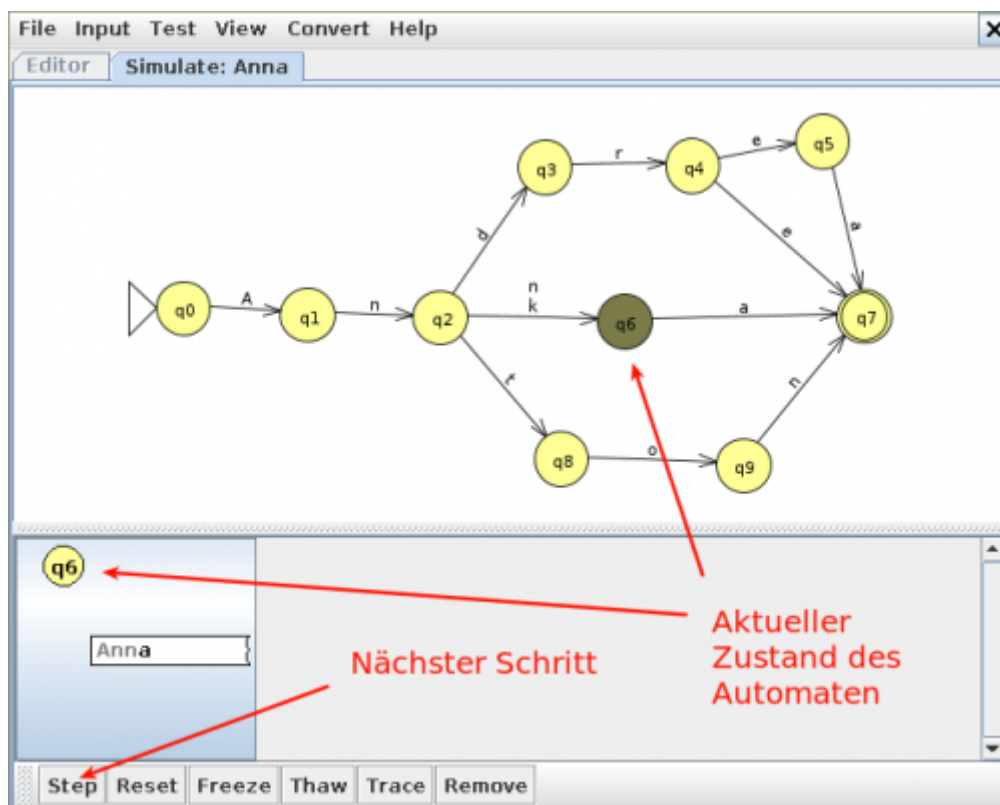
Jetzt sollst du das Verhalten des Automaten testen. JFLAP bietet dafür zwei Möglichkeiten:

## Möglichkeit 1: Eine Eingabe schrittweise verarbeiten

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



Dann öffnet sich ein Fenster, in dem du das Verhalten des Automaten schrittweise nachvollziehen kannst, indem du auf den Button Step klickst. Um den Automaten erneut mit der gleichen Eingabe zu starten, musst du Reset drücken. Möglichkeit 2: Mehrere Eingaben verarbeiten



Starte nun auf gleiche Weise den Automaten einmal mit der Eingabe Anka und einmal mit der Eingabe Andreas. Dabei wirst du feststellen, dass der Automat sich bei der Eingabe Anka grün färbt; die Eingabe wird also akzeptiert. Die Eingabe Andreas kann der Automat dagegen nicht verarbeiten. Deshalb stoppt er bei q7, egal wie oft der Step-Button jetzt noch gedrückt wird.

## Möglichkeit 2: Mehrere Eingaben verarbeiten

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.

The screenshot shows the JFLAP software interface. On the left, a finite state automaton is displayed with states q0 through q9. q0 is the start state, and q7 is the final state. Transitions are labeled with letters: 'A' from q0 to q1, 'n' from q1 to q2, 'r' from q2 to q3, 'k' from q2 to q6, 'e' from q3 to q4, 'a' from q4 to q7, 'o' from q6 to q7, 'r' from q8 to q2, 'a' from q8 to q9, and 'n' from q9 to q7. On the right, a table titled 'Table Text Size' shows the results of multiple runs. The table has two columns: 'Input' and 'Result'. The results are as follows:

| Input   | Result |
|---------|--------|
| Anna    | Accept |
| Andreas | Reject |
| Anke    | Reject |
| Anni    | Reject |
| Andrea  | Accept |

At the bottom of the interface, there are buttons: 'Load Inputs', 'Run Inputs' (highlighted with a red arrow), 'Clear', 'Enter Lambda', and 'View Trace'.

## Aufgaben

---



### (A1)

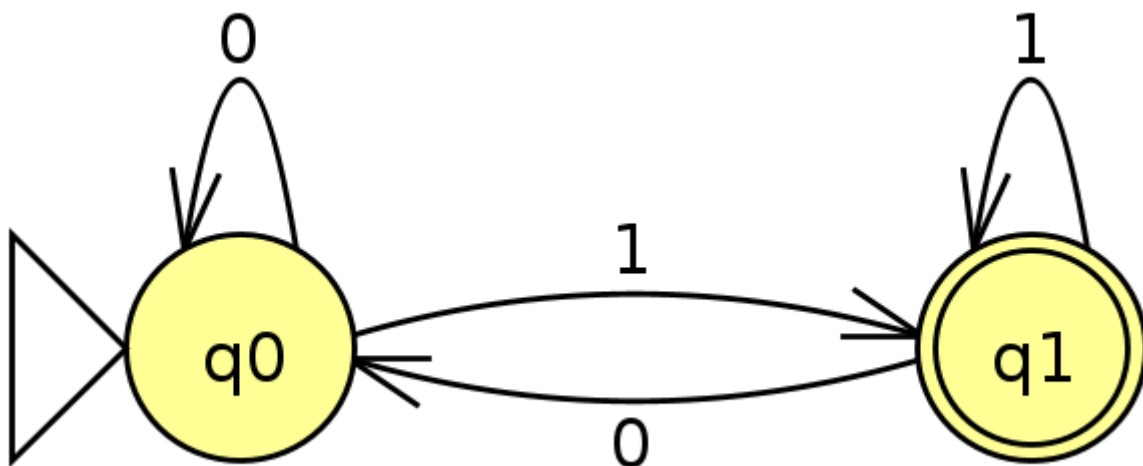
Überprüfe mit Hilfe von JFLAP, ob der Schulausflug-Automat folgende Eingaben akzeptiert:

- Anton
  - Anika
  - Andre
- 



### (A2)

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<sup>2)</sup> Ist das letzte Bit eine 1, so ist die Zahl ungerade.

Teste das Akzeptanzverhalten des Automaten „ungerade“ von Hand für die folgenden Eingaben:

1. 0010

2. 1000101

3. 1000000

1)

Lokale Sicherungskopie

2)

Begründe!?

From:  
<https://www.info-bw.de/> -

Permanent link:  
<https://www.info-bw.de/faecher:informatik:oberstufe:automaten:lepro:jflap01:start?rev=1675321826>

Last update: **02.02.2023 07:10**

