Definition

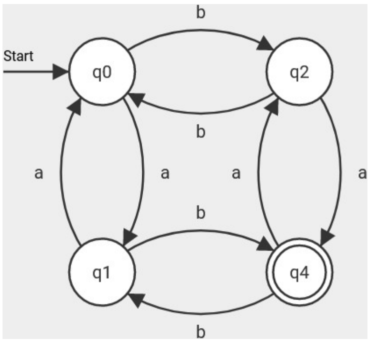
Ein deterministischer endlicher Automat (DEA) ist ein Tupel[1] $A = (\Sigma, S, s_0, F, R)$
mit den folgenden Bestandteilen:

Symbol	*Erläuterung*	*Beispiel*
Σ	Menge von Zeichen, das „Alphabet“	$\Sigma = \{ a, b \}$
S	Menge der Zustände	$S = \{ q_0, q_1, q_2, q_3 \}$
$s_0 \in S$	der Startzustand, ein Element von S d.h. ein Zustand wird als Startzustand gewählt	$s_0 = q_0$
$F \subseteq S$	Menge der Endzustände, eine Teilmenge von S d.h. es kann mehrere Endzustände geben	$F = \{ q_3 \}$
$R: S \times \Sigma \rightarrow S$	Übergangsfunktion d.h. die Definition der Zustandsübergänge	$R(q_0, a) = q_1$ $R(q_0, b) = q_2$, usw.

Übergangsfunktion

Die Übergangsfunktion R bestimmt, bei welchem Zeichen der Automat von welchem Zustand in welchen Folgezustand übergeht. Es gibt mehrere Möglichkeiten für die Darstellung von R:

<table>
<tr><th>Darstellung als Graph</th><th>Darstellung als Funktion</th><th>Darstellung als Tabelle</th></tr>
<tr><td>



</td><td>

$R(q_0, a) = q_1$
$R(q_0, b) = q_2$
$R(q_1, a) = q_0$
$R(q_1, b) = q_3$
$R(q_2, a) = q_3$
$R(q_2, b) = q_0$
$R(q_3, a) = q_2$
$R(q_3, b) = q_1$

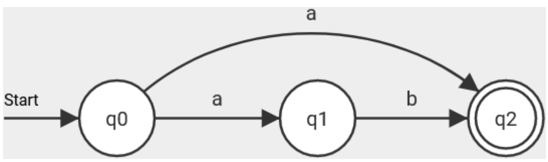
</td><td>

<table>
<tr><td></td><td>a</td><td>b</td></tr>
<tr><td>q_0</td><td>q_1</td><td>q_2</td></tr>
<tr><td>q_1</td><td>q_0</td><td>q_3</td></tr>
<tr><td>q_2</td><td>q_3</td><td>q_0</td></tr>
<tr><td>q_3</td><td>q_2</td><td>q_1</td></tr>
</table>

</td></tr>
</table>

Determinismus

„Deterministisch“ bedeutet hier, dass immer eindeutig festgelegt ist, in welchen Folgezustand der Automat bei einem bestimmten Zeichen geht. Es darf also nicht mehrere Wahlmöglichkeiten für den Folgezustand geben. Der folgende Beispielgraph ist **nicht** deterministisch, weil in Zustand q_0 nicht festgelegt ist, ob man mit dem Zeichen a in den Folgezustand q_1 oder q_2 gelangt.



1 Ein „Tupel“ in der Mathematik ist ein Objekt, das aus mehreren Bestandteilen besteht. Dabei ist (im Gegensatz zur Menge) die Reihenfolge der Bestandteile bedeutsam. Im Gegensatz zum Vektor können die Bestandteile verschiedenartig sein – wie hier, im Beispiel Mengen, Symbole, Funktionen etc.

Akzeptieren und Verwerfen

Ein **Wort** in der Informatik ist eine Kette von Zeichen über einem Alphabet. Ein Wort kann also ein Wort einer natürlichen Sprache sein, wie z.B. „Hund“. Ein Wort kann jedoch auch etwas abstraktes wie sein wie „abaabb“ oder auch ein ganzer Text, z.B. ein Java-Programm.

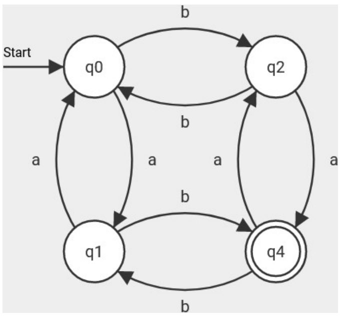
Ein endlicher Automat „liest“ ein Eingabewort und prüft, ob es bestimmten Regeln entspricht.
Der Automat beginnt beim ersten Zeichen und liest nacheinander alle Zeichen des Wortes.
Er befindet sich zu Anfang immer in seinem Startzustand. Mit jedem Zeichen, das er liest, ändert er seinen Zustand entsprechend der Übergangsfunktion.

Es gibt zwei mögliche Ergebnisse dieses Prozesses:

1. Der Automat „**akzeptiert**“ das Eingabewort.
   Dazu muss er das Wort **bis zum Ende** gelesen haben, und mit dem letzten Zeichen des Wortes in einen **Endzustand** gelangt sein.

2. Der Automat „**verwirft**“ das Eingabewort.
   Entweder hat er das Wort bis zum Ende gelesen, befindet sich aber nicht in einem Endzustand. Oder er kann das Wort nicht bis zum Ende lesen, weil es an einer Stelle keinen Übergang für das aktuelle Zeichen gibt.

Beispiel:



Der Automat akzeptiert das Eingabewort aaabaa.
Begründung: Er liest das Wort mit der folgenden **Zustandsfolge**:
q_0 / a → q_1 / a → q_0 / a → q_1 / b → q_3 / a → q_2 / a → q_3
Mit dem letzten Übergang befindet er sich im Endzustand q_3.

Der Automat verwirft das Eingabewort abba.
Begründung: Er liest das Wort zwar bis zum Ende:
q_0 / a → q_1 / b → q_3 / b → q_1 / a → q_0
Mit dem letzten Übergang befindet er sich jedoch in q_0, was kein Endzustand ist.

Formale Sprachen

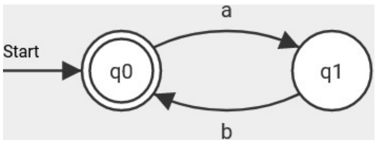
Eine „formale Sprache“ ist eine Menge von Worten im Sinn der Informatik.
Eine formale Sprache ist z.B. die Menge aller korrekten Java-Programme.
Auch die Menge aller Worte, die aus beliebig vielen a's bestehen, ist eine formale Sprache – die zwar wenig Sinn hat, aber als Beispiel für die Funktionsweise von Automaten nützlich sein kann.

Da ein endlicher Automaten in der Regel mehrere Worte akzeptiert, definiert er eine formale Sprache: Die „Sprache eines endl. Automaten“ ist die Menge aller Wörter, die der Automat akzeptiert. Wenn der Automat mit A bezeichnet wird, ist L(A) die Sprache des Automaten.

Um formale Sprachen zu beschreiben verwenden wir einige abkürzende Schreibweisen:

- Σ^* ist die Menge aller möglichen Worte über dem Alphabet Σ, inklusive dem leeren Wort.
- ε ist das „leere Wort“, also das Wort ohne Buchstaben.
- Der Exponent in a^5 bedeutet: 5x a hintereinander (aaaaa).
  Entsprechend bedeutet a^n „beliebig viele a's hintereinander“
  (n steht in der Regel für eine beliebige natürliche Zahl).
- Mit der folgenden Schreibweise können beliebige formale Sprachen beschrieben werden:
  $L(A) = \{\ w \in \Sigma^* \mid w$ erfüllt diese und jene Bedingung $\}$
  Das liest sich so:
  Die Sprache des Automaten ist die Menge aller Wörter w über dem Alphabet Σ, für die diese und jene Bedingung gilt.

Beispiel:



Der Automat akzeptiert die Wörter ε, ab, abab, ababab, …

Damit ist
$L(A) = \{\ w \in \Sigma^* \mid w$ enthält beliebig oft hintereinander ab $\}$
$= \{\ (ab)^n \mid n \in N_0\ \}$

Aufgabe 1

Entwickle je einen DEA, der die gegebene Sprache erkennt.
Gib den Automat vollständig als Tupel an, die Übergangsfunktion sowohl als Graph wie als Tabelle.
Das Alphabet ist jeweils $\Sigma = \{\ a, b\ \}$.

a) $L(A) = \{\ a^n b^m \mid n, m \in N_0\ \}$

b) $L(A) = \{\ w \in \Sigma^* \mid w$ enthält eine gerade Anzahl a's und eine gerade Anzahl b's $\}$ (0 sei gerade)

c) Für deinen Automaten aus Aufgabe b) gib jeweils die Zustandsfolge für die Eingabeworte aabaa und ababbb an und begründe so, ob der Automat diese Worte akzeptiert oder nicht.

Aufgabe 2

Gib jeweils (nur) den Graphen zu einem DEA an, die die gegebene Sprache akzeptiert.

a) $L(A) = \{\ w \in \Sigma^* \mid w$ enthält „irgendwo“ die Kombination abba $\}$

b) $L(A) = \{\ w \in \Sigma^* \mid w$ enthält nirgends die Kombination abba $\}$

c) $L(A) = \{\ w \in \Sigma^* \mid w$ enthält mindestens 3 a's $\}$

d) $L(A) = \{\ w \in \Sigma^* \mid w$ enthält mindestens 3 a's **oder** mindestens 2 b's (beides auch zulässig) $\}$

e) $L(A) = \{\ w \in \Sigma^* \mid w$ enthält mindestens 3 a's **und** mindestens 2 b's $\}$

f) $L(A) = \{\ a^n b^n \mid n \in N, n < 10\ \}$ (d.h. die gleiche Anzahl a's und b's, bis zu 9 Stück)

g) $L(A) = \{\ a^n b^n \mid n \in N_0\ \}$ (d.h. die gleiche Anzahl a's und b's, beliebig viele)