

Einleitung

Seit den 50er Jahren (Alan Turing, Stephen C. Kleene, Noah Chomsky, ...) werden im Zusammenhang mit Fragestellungen aus der Informatik und der Grammatik formale Sprachen untersucht.

Eine solche Sprachklasse bilden die *regulären Sprachen*. Das sind Sprachen, die durch sogenannte *deterministische endliche Automaten* (*deterministic finite automaton*, *DFA*) erzeugt werden.

Definition

Ein *deterministischer endlicher Automat* (*deterministic finite automaton*, *DFA*) ist ein 5-Tupel $(\Sigma, Q, q_0, F, \delta)$, mit:

- Ein endliches Eingabealphabet Σ . Bei einer natürlichen Sprache sind das die Wörter; es können aber auch Buchstaben oder andere Symbole sein.
- Eine endliche Menge von Zuständen Q . Meist werden sie durchnummeriert. Zum Beispiel: $Q = \{q_0, q_1, q_2, \dots, q_n\}$
- Einen Startzustand $q_0 \in Q$
- Eine Menge $F \subset Q$ von Endzuständen, die eine Eingabe „akzeptieren“
- Eine Tabelle $\delta(q, \sigma)$ mit erlaubten Zustandsübergängen. Sie beschreibt, in welchen Zustand q_j man vorrückt, wenn sich der Automat im Zustand q_i befindet und das Eingabesymbol σ liest.

Sofern es für ein bestimmtes Eingabesymbol jeweils nur eine Möglichkeit einer Zustandsänderung gibt, wird der Automat *deterministisch* genannt, sonst *nichtdeterministisch*.

Beispiel 1 („Sheep-Talk“)

- Eingabealphabet: $\Sigma = \{b, ä, !\}$.
- Zustandsmenge: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Startzustand: q_0
- Endzustand: $F = \{q_3\}$ (akzeptierender Zustand)
- Übergangstabelle δ :

q	Eingabe		
	b	ä	!
0	1	4	4
1	4	2	4
2	4	2	3
3	4	4	4
4	4	4	4

Ein endlicher Automat kann unterschiedlich verwendet werden. Eine erste Anwendung ist die Kontrolle, ob ein Folge von Eingabesymbolen (Buchstaben, Worte, ...) zu dieser Sprache gehört; das heisst vom Automaten *akzeptiert* wird.

Werden die folgenden Zeichenketten vom Sheep-Talk-Automaten akzeptiert?

- „bäää!“
- „äää!“
- „bä“

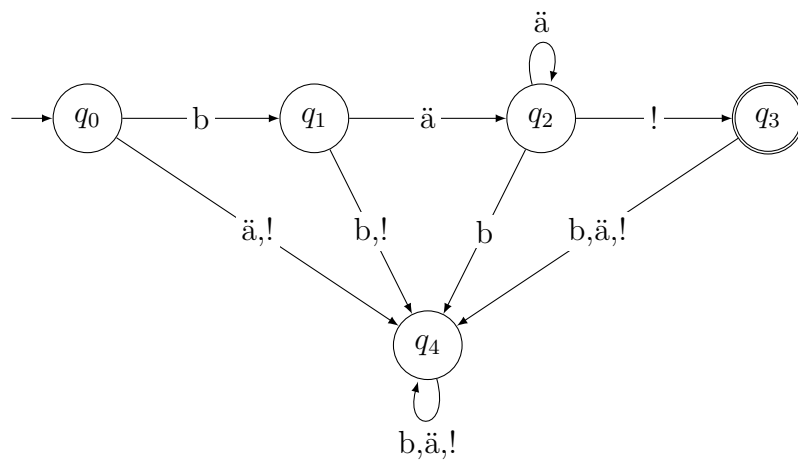
Graphische Darstellung

Wenn ein endlicher Automat nicht allzu komplex ist, lässt er sich grafisch darstellen.

Die Zustände werden durch Kreise mit der jeweiligen Bezeichnung im Innern dargestellt. Die Übergänge werden durch Pfeile symbolisiert, wobei die Beschriftung des Pfeils aus dem zugehörigen Symbol des Eingabealphabets besteht.

Ein Pfeil ohne Beschriftung zeigt auf den Startzustand.

Endzustände sind von einer doppelten Kreislinie umgeben.



Umgekehrt lässt sich mit Hilfe des endlichen Automaten die zugehörige (von ihm akzeptierte) Sprache erzeugen, indem man alle erlaubten Kombinationsmöglichkeiten ausschöpft, bis man den einzigen akzeptierenden Zustand erreicht. In unserem Fall sind das die Wörter *bä!*, *bää!*, *bäää!*, usw.

Man beachte, dass die Länge der so generierten Wörter theoretisch unbegrenzt ist.

Beispiel 3 (Primitive Formensprache)

Durch die Symbole „U“ (unten), „L“ (links), „R“ (rechts) wird von einer Startposition jeweils eine Einheit in die entsprechende Richtung gegangen und ein Streckensegment angehängt. Die Zustände und Übergänge sind so gewählt, dass die Zeichenoperationen sinnvoll sind.

- Eingabealphabet: $\Sigma = \{L, R, U\}$
- Zustandsmenge: $Q = \{q_0, q_1, q_2\}$
- Startzustand: q_0
- Endzustand: $F = \{q_0, q_1, q_2\}$ (akzeptierende Zustände)
- Übergangstabelle δ :

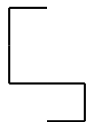
q	Eingabe		
	U	L	R
0	0	1	2
1	0	1	3
2	0	3	2
3	3	3	3

Hier einige mögliche Wörter dieser Sprache und ihre graphische Entsprechung.

- „ULUR“:



- „LUURRUL“:



Das letzte Beispiel könnte auch mit anderen Bauelementen (Würfeln, Stäben, ...) und in höheren Dimensionen modelliert werden.

Aufgabe

Stelle den Automaten graphisch dar.