

# Theoretische Informatik

Marvin Baeumer 2023-12-06 10:20

---

## Begrifflichkeiten

### Alphabet

Ein Alphabet ist eine endliche, nicht leere Menge  $\Sigma$  von Zeichen. Zum Beispiel für die Sprache Schulnoten:  $\{1, 2, 3, 4, 5, 6, +, -\}$

### Wort

Ein Wort ist eine endliche Menge an Zeichen aus  $\Sigma$ . Zum Beispiel für die Sprache Schulnoten wären  $\{123, 2+, 454-\}$  dies wären zulässige Worte der Sprache Schulnote.

### Leeres Wort $\epsilon$

Das leere Wort  $\epsilon$  enthält jede Sprache und dient als neutraler Multiplikator.  $\epsilon \cdot 1- = 1-$  für das Beispiel Schulnoten.

### Alle Möglichkeiten $\Sigma^*$

alle denkbaren Aneinanderreihungen der Zeichen aus der Sprache, heißt  $\Sigma^*$ , sind unendlich lang. Eine Formale Sprache ist nun die Teilmenge von  $\Sigma^*$ . Beispiel Schulnote:  $\{1+, 1, 1-, 2+, 2, 2-, 3+, 3, 3-, 4+, 4, 4-, 5+, 5, 5-, 6\} \subset \Sigma^*$

### Sprache

Eine Sprache ist eine Teilmenge der möglichen Worte über einem Alphabet  $\Sigma$

# Grammatik

## Definition

Eine Grammatik  $G = (N, \Sigma, S, P)$  ist ein 4-Tupel bestehend aus:

- der Menge  $N$  an Nichtterminalsymbolen
- der Menge  $\Sigma$  an Terminalsymbolen
- einem Startsymbol  $S$
- einer Menge  $P$  an Produktionsregeln Dabei gilt  $N \cap \Sigma = \emptyset$ , das heißt ein Nichtterminalsymbol kann nicht gleichzeitig ein Terminalsymbol sein. Außerdem gilt  $S \in N$ , das heißt, ein Startsymbol muss ein Nichtterminalsymbol sein. Die durch eine Grammatik  $G$  beschriebene formale Sprache ist die Sprache, die genau aus den Worten über  $\Sigma$  besteht, die man mithilfe der Produktionsregeln  $P$  aus dem Startsymbol  $S$  ableiten kann.

## Bilden von Produktionsregeln - Schulnoten

Produktionsregeln werden gebildet, indem man seine Nichtterminalsymbole, die Terminalsymbole und den Startpunkt definiert. Somit ergibt sich die Form  $\sigma B$ .

### Beispiel

$$\begin{array}{ll} N & \rightarrow \{S\} \\ \Sigma & \rightarrow \{+, -, \epsilon\} \\ S & \rightarrow \{S\} \\ P & \rightarrow \{ \\ & S \rightarrow 1T, 2T, 3T, 4T, 5T, 6T \\ & T \rightarrow +, -, \epsilon \\ & \} \end{array}$$

Man kann nun bei  $S$  starten und muss dann  $T$  aufrufen.  $T$  ist dann die Abbruchbedingung. Es lassen sich nun alle Schulnoten ableiten.

## Definition einer regulären Grammatik

Eine Grammatik  $G = (N, \Sigma, S, P)$  nennt man dann regulär, wenn  $P$  so aussieht

$$\begin{array}{ll} A & \rightarrow \sigma B \text{ oder} \\ A & \rightarrow \sigma \end{array}$$

Hier gilt  $A$  und  $B$  sind Nichtterminalsymbole und  $\sigma \in \Sigma$  ein Terminalsymbol ist. Dabei ist auch erlaubt, dass  $A = B$  gilt. Beachte das  $\sigma \neq \epsilon$ , da das leere Wort  $\epsilon$  kein Zeichen des Alphabets sein kann. Eine Sprache, die von einer regulären Grammatik erzeugt werden kann, nennt man reguläre Sprache.

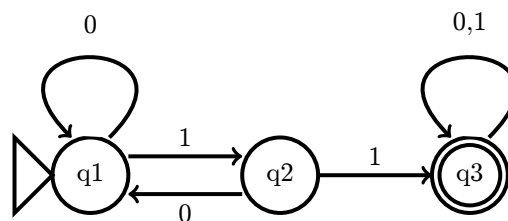
# Endliche Automaten

## Deterministischer endlicher Automat (DEA)

Ein deterministischer endlicher Automat  $A = \{Q, \Sigma, \delta, q_0, F\}$  besteht aus:

- einer endlichen Menge an  $Q$  von Zuständen
- einem endlichen Eingabealphabet  $\Sigma$
- einer Übergangsfunktion  $\delta$ , die jedem Paar  $(q, \sigma)$  bestehend aus einem Zustand  $q \in Q$  und einem Zeichen  $\sigma \in \Sigma$  einen eindeutigen Folgezustand  $\delta(q, \sigma) = q'$  zuordnet
- einem Startzustand  $q_0 \in Q$
- einer Menge an  $F \subseteq Q$  von Endzuständen

### Beispiel



Hier gilt

- $Q = \{q1, q2, q3\}$
- $q_0 = q1$
- $\Sigma = \{0, 1\}$
- $F = q3$  **Übergangsfunktion**

$$\delta(q1, 0) = q1$$

$$\delta(q2, 0) = q1$$

$$\delta(q3, 0) = q3$$

$$\delta(q1, 1) = q2$$

$$\delta(q2, 1) = q3$$

$$\delta(q3, 1) = q3$$

oder

| $\delta$ | 0    | 1    |
|----------|------|------|
| $q1$     | $q1$ | $q2$ |
| $q2$     | $q1$ | $q3$ |
| $q3$     | $q3$ | $q3$ |

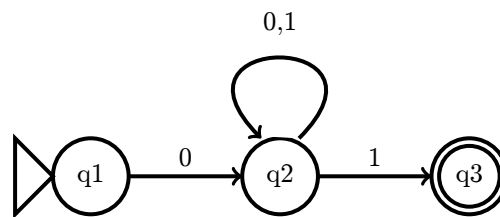
Bei der Übergangsfunktion ist es wichtig, jeden Zustand zu beachten und zu prüfen, von welchem Zustand es wohin geht.

# Nichtdeterministischer endlicher Automat (NEA)

Ein nichtdeterministischer endlicher Automat  $A = \{Q, \Sigma, \delta, q_0, F\}$  besteht aus:

- einer endlichen Menge an  $Q$  von Zuständen
- einem endlichen Eingabealphabet  $\Sigma$
- einer Übergangsfunktion  $\delta$ , die jedem Paar  $(q, \sigma)$  bestehend aus einem Zustand  $q \in Q$  und einem Zeichen  $\sigma \in \Sigma$  einen nicht eindeutigen Folgezustand  $\delta(q, \sigma) = q'$  zuordnet
- einem Startzustand  $q_0 \in Q$
- einer Menge an  $F \subseteq Q$  von Endzuständen

## Beispiel



### Hier gilt

- $Q = \{q1, q2, q3\}$
- $\Sigma = \{1, 0\}$
- $q_0 = q1$
- $F = q3$

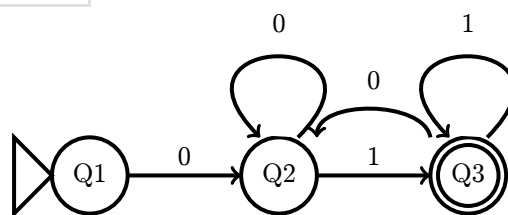
## Übergangsfunktion/Umwandeln

Man kann einen NEA in einen DEA umwandeln mithilfe der Potenzmengenkonstruktion, dafür benötigt man das Verständnis der Übergangsfunktion.

| NEA $\rightarrow$ DEA | 0        | 1            |
|-----------------------|----------|--------------|
| $Q1 = \{q1\}$         | $\{q2\}$ | $F$          |
| $Q2 = \{q2\}$         | $\{q2\}$ | $\{q2, q3\}$ |
| $Q3 = \{q2, q3\}$     | $\{q2\}$ | $\{q2, q3\}$ |
| $F$                   | $F$      | $F$          |

Man geht wie bei einer Übergangsfunktion zum nächsten Zustand mit der jeweiligen Eingabe. Für jeden neuen Zustand erstellt man dann eine Hilfsvariable, die später in der DEA-Tabelle wichtig wird. Zum Beispiel sehen wir das in der ersten Zeile der Tabelle:  $Q1$  ist eine Variable für  $q1$ . Von  $q1$  aus gehen wir mit einer 0 nach  $q2$ . Dies ist nun ein neuer Zustand, also müssen wir eine neue Hilfsvariable anlegen. Dieser Prozess geht solange, bis keine neuen Kombinationen auftreten.  $F$  steht immer für den Fehlerzustand, das heißt, aus diesem Zustand kommt man nicht wieder heraus. Er löst aus, wenn man eine nicht weiterführende Eingabe trifft. Zum Beispiel von  $q1$  mit einer 1 kommen wir in den Fehlerzustand.

| DEA Tabelle       | 0    | 1    |
|-------------------|------|------|
| $Q1 = \{q1\}$     | $Q2$ | $F$  |
| $Q2 = \{q2\}$     | $Q2$ | $Q3$ |
| $Q3 = \{q2, q3\}$ | $Q2$ | $Q3$ |
| $F$               | $F$  | $F$  |



# Dinge zum Wissen

## Automat → Grammatik

Zu jedem endlichen Automaten  $A$  (DEA & NEA) gibt es eine reguläre Grammatik, die exakt die von  $A$  erkannte formale Sprache erzeugt. Diese Grammatik lässt sich konkret aus  $A$  konstruieren. Insbesondere ist damit jede formale Sprache, die von einem endlichen Automaten erkannt wird, eine reguläre Sprache.

## Grammatik → Automat

Zu jeder Grammatik  $G$  gibt es auch einen endlichen Automaten, der exakt die von  $G$  erzeugte formale Sprache akzeptiert. Dieser Automat lässt sich konkret aus  $G$  konstruieren. Damit gibt es für jede reguläre Grammatik ein Verfahren, um festzustellen, ob ein beliebiges gegebenes Wort zu dieser Grammatik gehört.

## Längenangaben

$|1 + 3 - -| = 5$  das Wort steht in  $||$ , hinter den Betragsstrichen steht die Länge des Wortes, die Länge sind alle Zeichen.

## Aneinanderhängung:

$$\begin{aligned} |v| &= 1 + 3 \\ |w| &= - - \\ |vw| &= |v| + |w| = 1 + 3 - - \end{aligned}$$

## Tupel

Ein Tupel kommt aus der Vektormathematik und beschreibt eine Erfahrung auf Basis von 5 Sinnesmodalitäten, also VAKOG. Beschreibt einfach  $X$  Dinge

## DEA & NEA

Ein DEA steht für deterministischer Automat, der Weg ist eindeutig. Ein NEA hingegen ist ein nichtdeterministischer Automat, der Weg ist nicht eindeutig.

# Syntaxdiagramme

Mit Syntaxdiagrammen kann man auch formale Sprachen beschreiben!

Syntaxdiagramme bestehen aus Terminalsymbolen, Nichtterminalsymbolen und Verbindungspfeilen. Terminalsymbole sind Symbole des Alphabets der Sprache, die in Diagrammen durch abgerundete Rahmen zu erkennen sind. Nichtterminalsymbole sind Hilfssymbole, die in Diagrammen durch rechteckige Rahmen zu erkennen sind.

Nichtterminalsymbole stehen jeweils für eigene Diagramme. Somit kann man mit einem Syntaxdiagramm eine gültiges Wort bilden.

## Beispiel Schulnoten

