Formale Systeme, Automaten und Prozesse Zusammenfassung

Justin Korte

Februar 2023

1 Deterministische Endliche Automaten

1.1 Alphabet

- Ein **Alphabet** ist eine nichtleere, endliche Menge, deren Elemente als Zeichen bezeichnet werden.
- Ein **Wort** über einem Alphabet ist eine endliche Folge/ein Tupel von Zeichen aus dem Alphabet.
- Eine **formale Sprache** über einem Alphabet ist eine (Teil-) Menge von Wörtern über diesem Alphabet.

Für die Notation werden für Alphabete eine Variation von Σ, Γ verwendet, für die Zeichen eine Variation von $a, b, c, ...$ und für die Zeichen eine Variation von $u, v, w, , ..$ und für Sprachen eine Variation von $K, L, M, ...$

1.2 Wörter

- Σ^* bezeichnet die Menge aller Wörter über dem Alphabet Σ.
- ε bezeichnet hierbei das **leere Wort**.
- $|w|$ bezeichnet die Länge des Wortes w.
- $|w|_a$ bezeichnet die Häufigkeit des Zeichens a im Wort w.
- uv bezeichnet die Verkettung von u und v.

1.3 Teilwörter

Sind u, v Wörter. Dann ist u

- ein Präfix von v, falls es ein Wort w gibt mit $v = uw$
- ein Infix von v, falls es zwei Wörter w, w' gibt mit $v = wuw'$
- ein Suffix von v, falls es ein Wort w gibt mit $v = wu$

1.4 Deterministischer endlicher Automat

Ein deterministischer endlicher Automat (DFA oder DEA) ist ein 5-Tupel der Form:

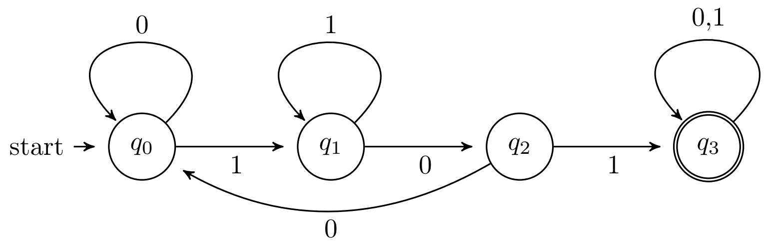
$$(Q, \Sigma, \delta, q_0, F)$$

- Q ist eine endliche Menge, die als **Zustände** bezeichnet werden.
- Σ ist ein Alphabet, das **Eingabealphabet**
- $\delta : Q \times \Sigma \to Q$ eine Abbildung, die **Transitionsfunktion**
- $q_0 \in Q$ der **Anfangszustand**
- $F \subseteq Q$ die Menge der **Endzustände**

1.5 Graphische Darstellung eines DFA's

In der graphischen Darstellung eines Automatens verwendet man

- Knoten als Darstellungen von Zuständen
- eine gerichtete mit a beschriftete Kante von Knoten q zu Knoten r falls $\delta(q, a) = r$
- einen Pfeil zur Darstellung des Anfangzustandes
- einen umkreisten Zustand als Darstellung eines Endzustands



1.6 Lauf

Die Berechnung eines DFA auf einem Eingabewort bezeichnet man als **Lauf**.
Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA. Dann ist ein Lauf von A eine Folge $(r_0, a_1, r_1, a_2, ..., r_{n-1}, a_n, r_n)$ mit $n \geq 0$ und $r_0, ..., r_n \in Q$ sowie $a_1, ..., a_n \in \Sigma$, sodass $r_0 = q_0$ und $\delta(r_{i-1}, a_i) = r_i, \quad 1 \leq i \leq n$ gilt. Dann ist $(r_0, a_1, r_1, a_2, ..., r_{n-1}, a_n, r_n)$ bzw. $(r_0, r_1, r_2, ..., r_{n-1}, r_n)$ ein Lauf von A auf dem Wort $w = a_1...a_n$.

Dabei kann der DFA ein Wort **akzeptieren**, falls $r_n \in F$,
ansonsten **verwirft** der DFA das Wort.

1.7 DFA-Erkennbarkeit

Die von einem DFA A erkannte Sprache ist

$$L(A) := \{w \in \Sigma^* \mid A \text{ akzeptiert } w\}$$

Eine Sprache L heißt **DFA-erkennbar**, wenn es einen DFA A gibt, sodass $L = L(A)$

2 Operationen auf Sprachen

2.1 Mengenoperationen

Sei K, L, M und Varianten Sprachen über dem Alphabet Σ. Da Sprachen Mengen bilden, können auch **Vereinigung**, **Durchschnitt** und **Differenz** gebildet werden. Zusätzlich definieren wir für eine Sprache L das Komplement $\overline{L}$.

$$\begin{aligned}
K \cup L &:= \{\, x \in K \cup L \mid x \in K \lor x \in L \,\} \\
K \cap L &:= \{\, x \in K \cap L \mid x \in K \land x \in L \,\} \\
K \setminus L &:= \{\, x \in K \setminus L \mid x \in K \land x \notin L \,\} \\
\overline{L} &:= \Sigma^* \setminus L.
\end{aligned}$$

2.2 Verkettung von Sprachen

Seien $K, L \subseteq \Sigma^*$ Sprachen. Dann is die Verkettung von K und L definiert als

$$KL := \{\, uv \mid u \in K,\ v \in L \,\} \subseteq \Sigma^*$$

2.2.1 Rechenregeln von Sprachverkettungen

Für alle Sprachen K, L, M gilt (Beweise Kap. 1)

- Assoziativgesetz
  - ➤ $(KL)M = K(LM)$
  - ➤ $L\{\varepsilon\} = \{\varepsilon\}L = L$
  - ➤ $L\emptyset = \emptyset L = \emptyset$
- Distributivgesetz
  - ➤ $K(L \cup M) = KL \cup KM$
  - ➤ $(K \cup L)M = KM \cup LM$

2.2.2 Potenzen von Sprachen

Sei $L \subseteq \Sigma^*$ eine Sprache. Dann sind die Potenzen $L^n, \quad n \in \mathbb{N}$ wiefolgt induktiv definiert:

$$\begin{aligned} L^0 &:= \{\varepsilon\}, \\ L^{n+1} &:= L^N L && \text{(Für alle } n \in \mathbb{N}) \end{aligned}$$

2.2.3 Iteration von Sprachen

Die Iteration (auch **Kleene-Stern** einer Sprache L ist die Sprache

$$L^* := \bigcup_{n \in \mathbb{N}} L^n$$

Da $L^0 := \{\varepsilon\}$, gilt : $\varepsilon \in L^0 \subseteq L^*$

2.2.4 Rechenregeln auf Iterationen

Für alle $L \subseteq \Sigma^*$ gilt:

- $L^* L^* = L^*$
- $(L^*)^* = L^*$
- $L^* = \{\varepsilon\} \cup LL^* = \{\varepsilon\} \cup L^* L$
- $\emptyset^* = \{\varepsilon\}$