Formale Sprachen und Automaten

Robin Rausch

22. Oktober 2022

Inhaltsverzeichnis

1 Grundlagen

1.1 Alphabet

Ein Alphabet Σ ist eine nicht-leere Menge von Symbolen(Zeichen, Buchstaben). Beispiel: $\Sigma_{ab} = a, b$

1.2 Wort

Ein Wort w über dem Alphabet Σ(Sigma) ist eine endliche Folge von Symbolen aus Σ. Das Wort $w = abaabab$ wurde beispielsweise aus dem Alphabet Σ_{ab} gebildet.
Die Länge eines Wortes kann durch Betragsstriche angegeben werden. Beispiel: $|w| = 7$
Ebenso kann man die Anzahl bestimmter Symbole in einem Wort bestimmen: $|w|_b = 3$
Ein einzelnes Zeichen kann durch eckige Klammern angegeben werden: $w[2] = b$
Wörter können bliebig konkateniert werden(hintereinanderschreiben ohne abstand): $w_1 w_2 = abbabaab$ mit $w_1 = abba$ und $w_2 = baab$.
Wörter dürfen auch potenziert werden: $w^3 = abaabababaabababaabab = www$
Das leere Wort lautet ϵ.

1.3 Formale Sprachen

Eine formale Sprache L über einem Alphabet Σ ist eine Menge von Wörtern aus $\Sigma^* : L \subseteq \Sigma^*$. Eine Sprache kann sowohl endlich als auch unendlich sein.
Beispiel: $L_1 = \{w \in \Sigma_{bin}^* | \; |w| \geqslant 2 \wedge w[|w| - 1] = 1\}$ ist die Menge aller Binärwörter, an deren vorletzter Stelle 1 steht.
Das Produkt zweier formaler Sprachen: $L_1 \cdot L_2 = \{abac, abcb, bcac, bccb\}$ mit $L_1 = \{ab, bc\}$ und $L_2 = \{ac, cb\}$.
Sprachen können ebenfalls potenziert werden: $L^2 = \{ab, ba\} \cdot \{ab, ba\} = \{abab, abba, baab, baba\}$

1.4 Kleene Stern

Für ein Alphabet Σ und eine formale Sprache $L \subseteq \Sigma^*$ ist der Operator Kleene Stern wie folgt definiert: $L^* = \bigcup\limits_{n \in \mathbb{N}} L^n$.

Beispiel: Sei $L_1 = \{ab, ba\}$, dann $L^* = \{\epsilon, ab, ba, abab, abba, baab, baba, ababab, ...\}$.

2 Reguläre Sprachen und endliche Ausdrücke

2.1 Reguläre Ausdrücke

Ein regulärer Ausdruck über Σ beschreibt eine formale Sprache.
Die Menge aller regulären Ausdrücke über Σ ist eine formale Sprache.

Beispiel: Sprache aller Wörter über Σ_{abc}, die nur aus genau zwei Symbolen bestehen:
Ausdruck: $r_1 = (a + b + c)(a + b + c)$
Sprache: $\mathcal{L}(r_1) = \{w \in \Sigma_{abc}^* \mid |w| = 2\}$

Operatoren:

$$
\begin{array}{rcll}
r_1 + r_2 & \equiv & r_2 + r_1 & \text{Kommutativität von } + \\
(r_1 + r_2) + r_3 & \equiv & r_1 + (r_2 + r_3) & \text{Assoziativität von } + \\
(r_1 r_2) r_3 & \equiv & r_1 (r_2 r_3) & \text{Assoziativität von } \cdot \\
\emptyset r \equiv r \emptyset & \equiv & \emptyset & \text{Absorbierendes Element für } \cdot \\
\varepsilon r \equiv r \varepsilon & \equiv & r & \text{Neutrales Element für } \cdot \\
\emptyset + r & \equiv & r & \text{Neutrales Element für } + \\
(r_1 + r_2) r_3 & \equiv & r_1 r_3 + r_2 r_3 & \text{Distributivität links} \\
r_1 (r_2 + r_3) & \equiv & r_1 r_2 + r_1 r_3 & \text{Distributivität rechts} \\
r + r & \equiv & r & \text{Idempotenz von } + \\
(r^*)^* & \equiv & r^* & \text{Idempotenz von } ^* \\
\emptyset^* & \equiv & \varepsilon & \\
\varepsilon^* & \equiv & \varepsilon & \\
\varepsilon + r^* r & \equiv & r^* & \\
(\varepsilon + r)^* & \equiv & r^* & \\
r^* r & \equiv & r r^* &
\end{array}
$$

Nicht alle Operatoren sind für alle Typen zulässig:

	Vereinigung	Konkatenation	Potenz	Kleene-Stern
Wörter	✗	$w_1 \cdot w_2$	w^n	✗
Sprachen	$L_1 \cup L_2$	$L_1 \cdot L_2$	L^n	L^*
Reguläre Ausdrücke	$r_1 + r_2$	$r_1 \cdot r_2$	✗	r^*

2.2 Endliche Automaten

Endliche Automaten sind eine andere Darstellung einer regulären Sprache. Endliche Ausdrücke lassen sich in Reguläre Ausdrücke umformen. Genauso auch anders herum.
Endliche Automaten lassen sich sowohl deterministisch als auch nicht-deterministisch darstellen.

2.2.1 Deterministische endliche Automaten(DEA)

2.2.2 Nicht-deterministische endliche Automaten(NEA)

2.2.3 Endliche Automaten und reguläre Ausdrücke

2.2.4 Minimierung

2.3 Nicht-reguläre Sprachen und das Pumping-Lemma

2.4 Eigenschaften regulärer Sprachen

3 Chomsky Grammatiken und kontextfreie Sprachen

4 Turing Maschine

5 Entscheidbarkeit

6 Berechenbarkeit

7 Komplexität