Formale Sprachen

Formale Grundlagen (WIN)

2008S, F. Binder

Franz Binder
Institut für Algebra
Johannes Kepler Universität Linz

Vorlesung im 2008S

http://www.algebra.uni-linz.ac.at/Students/Win/fg

Inhalt

Wörter

- Das Alphabet Σ sei eine endliche Menge von Symbolen.
- Die Listen über Σ heißen Wörter.
- Für $\alpha, \beta, \gamma \in \Sigma$ und $u, v, w \, \epsilon \, \Sigma^*$ schreiben wir: uv, αu, $v\beta$, $u^0 = \epsilon$, $u^{n+1} = u^n u$
- und es gilt $(uv)w = u(vw)$ und $u\epsilon = u = \epsilon u$ sowie $|\epsilon| = 0$ und $|uv| = |u| + |v|$.

Theorem

Sei $\mathcal{M}$ ein beliebiges Monoid. Dann läßt sich jede Abbildung $f : \Sigma \to \mathcal{M}$ zu genau einem Homomorphismus $f^* : \Sigma^* \to \mathcal{M}$ fortsetzen.

Sprachen

Definition

Eine formale Sprache über Σ ist eine Teilmenge von Σ^*.

Bemerkung

Sind L_1, L_2 Sprachen, so auch $L_1 \cap L_2$, $L_1 \cup L_2$, $L_1 \setminus L_2$.

Definition

Für $L, L_1, L_2 \subseteq \Sigma^*$ definieren wir weiters

$$L_1 L_2 = \{ uv \mid u \in L_1,\ v \in L_2 \} \quad \text{(Verkettung)} \quad (1)$$

$$L^0 = \{ \epsilon \} \quad \text{(einelementig)} \quad (2)$$

$$L^{n+1} = L L^n \quad \text{(für } n \colon \mathbb{N}\text{)} \quad (3)$$

$$L^* = \bigcup_{k \colon \mathbb{N}} L^k \quad \text{(Iteration)} \quad (4)$$

Endliche Sprachen

Bemerkung

Triviale Sprache: $\{\alpha\}, \{\epsilon\}, \varnothing$.

Bemerkung

Die endlichen Sprachen lassen sich somit auch als jene Klasse von Sprachen charakterisieren, welche gerade groß genug ist, daß sie

- die oben erwähnten trivialen Sprachen enthält,
- gegenüber Verkettung abgeschlossen ist, und
- gegenüber Vereinigung abgeschlossen ist.

Zusätzlich ist die Klasse aller endlichen Sprachen gegenüber Durchschnitt und Differenz abgeschlossen, nicht aber gegenüber Komplement und Iteration.

Reguläre Ausdrücke

Definition

Die kleinste Klasse von formalen Sprachen, welche alle endlichen Sprachen umfaßt und abgeschlossen ist gegenüber Verkettung, Vereinigung und Iteration, heißt die Klasse der regulären Sprachen.

Definition

Sei Σ ein Alphabet. Dann ist ϵ und jedes $\alpha \in \Sigma$ ein regulärer Ausdruck, und wenn r, s reguläre Ausdrücke sind, dann auch (rs), $(r|s)$ und r^*, entsprechend den definierenden Konstruktionen für reguläre Sprachen.

Definition

Jedem regulären Ausdruck r wird gemäß der folgenden induktiven Definition eine reguläre Sprache $L(r)$ zugeordnet:

$$L(\epsilon) = \{\, \epsilon \,\} \tag{5}$$

$$L(\alpha) = \{\, \alpha \,\} \tag{6}$$

$$L(rs) = L(r)\, L(s) \tag{7}$$

$$L(r|s) = L(r) \cup L(s) \tag{8}$$

$$L(r^*) = L(r)^* \tag{9}$$

Regulare Ausdrücke in Programmen

Bemerkung

Darüberhinaus verwenden die gängigen Programme zur Verarbeitung von regulären Ausdrücken aus praktischen Gründen diverse Abkürzungen, z.B.

$$
\begin{aligned}
r? &= (r|\epsilon) \\
r^+ &= rr^* \\
[xyz] &= x|y|z \\
. &= \alpha|\beta|\gamma|\ldots|\zeta, \text{ falls } \Sigma = \{\alpha, \beta, \gamma, \ldots, \zeta\} \\
[c{-}u] &= c|d|\ldots|t|u \\
[\hat{}xyz] &= l_1|l_2|l_3|\ldots|l_n, \text{ falls } \Sigma \setminus \{x, y, z\} = \{l_1, \ldots, l_n\}
\end{aligned}
$$

und viele mehr. Details dazu findet man z.B. in `http://en.wikipedia.org/wiki/Regular_expression`.
Einen guten Vergleich der Syntaxregeln für reguläre Ausdrücke in verschiedenen Programmiersprachen bietet:
`http://www.greenend.org.uk/rjk/2002/06/regexp.html`

Zustandsüberführungsrelation

Definition

Eine Funktion vom Typ $\delta : \Sigma \to (Q \to Q \to \mathbb{P})$ heißt eine Zustandsüberführungsrelation.

Bemerkung

- $\delta(\alpha)$ schreiben wir als δ_α.
- Jedes δ_α ist eine Relation und wird oft durch einen Digraphen dargestellt.
- Um alle δ_α in einem einzigen Graphen darzustellen, kennzeichnen wir die Kanten von δ_α mit α.
- Eine mit α bezeichnete Kante von i nach j bedeutet dann, daß die Eingabe α einen Übergang vom Zustand i ind den Zustand j erlaubt.
- Der Zustand j heißt dann auch ein emphFolgezustand von i bei der Eingbe α.;
- $\delta_\alpha(i)$ bezeichnet die Menge aller Folgezustände von i bei der Eingabe α.
- Für $I \subset Q$ definieren wir $\delta_\alpha(I) := \bigcup_{i \in I} \delta_\alpha(i)$.

Fortsetzeung, Sprache

Bemerkung

$\delta : \Sigma \to (Q \to Q \to \mathbb{P})$ läßt sich eindeutig zu einem Homomorphismus $\delta^* : \Sigma^* \to (Q \to Q \to \mathbb{P})$ fortsetzen.

Definition

Für $i, j \epsilon Q$ definieren wir

$$L_\delta(i,j) := \{ u \epsilon \Sigma^* \mid i \xrightarrow{\delta_\alpha} j \}$$

.

Bemerkung

- $L_\delta(i,j)$ ist eine Sprache über Σ.
- Wird schreiben $L(i,j)$ wenn δ aus dem Zusammenhang klar ist.
- Für $I \subseteq Q$ und $J \subseteq Q$ setzen definieren wir $L(I,J) = \bigcup i \epsilon I \bigcup j \epsilon J L(i,j)$.

Automat

Definition

Ein nicht-deterministischer, endlicher Automat besteht aus

- einer endlichen Menge Σ, dem Eingabealphabet;
- einer endlichen Menge Q von Zuständen (states);
- einer entscheidbaren Zustandsüberführungsrelation $\delta \colon \Sigma \to (Q \to Q \to \mathbb{B})$;
- einer Teilmenge $I \subseteq Q$ von Initialzuständen;
- einer Teilmenge $F \subseteq Q$ von Finalzuständen.

Diese Konstruktion wird dann auch mit dem Quintupel $(\Sigma, Q, \delta, I, F)$ bezeichnet.

Definition

Die Sprache $L(\mathscr{A})$ eines Automaten $\mathscr{A} = (\Sigma, Q, \delta, I, F)$, besteht aus allen Wörtern, die einen Übergang von einem Initialzustand in einen Finalzustand erlauben, d.h. $L(\mathscr{A}) := L(I, F)$. Man sagt auch, der Automat erkennt oder akzeptiert die Sprache $L(\mathscr{A})$, bzw. jedes Wort in dieser Sprache.

Grammatik

Definition

Eine Grammatik $\mathcal{G} = (T, N, S, \mathcal{P})$ besteht aus

- einem Alphabet Σ von Terminalzeichen;
- einem Alphabet N von Nicht-Terminalzeichen;
- einem Startsymbol $S \in N$;
- einer Menge von Produktionsregeln der Form

$$l \Rightarrow r,$$

mit $l \in (N \cup \Sigma)^* \setminus \Sigma^*$ und $r \in (N \cup \Sigma)^*$.

Definition

Sei $\mathcal{G} = (\Sigma, N, S, \mathcal{P})$ eine Grammatik, und $u, w \in (N \cup \Sigma)^*$. Dann ist w aus u ableitbar (Schreibweise: $u \twoheadrightarrow w$) wenn es Zerlegungen $u = ale$ und $w = are$ gibt, sodaß $l \Rightarrow r$ eine Produktionsregel der Grammatik ist.

Rechtslineare Sprachen

Definition

Eine Grammatik $(\mathcal{G} = (\Sigma, N, S, \mathcal{P})$ heißt rechtslinear, falls alle Produktionsregel die Form

$$A \Rightarrow x \qquad \text{oder} \qquad A \Rightarrow xB$$

haben, mit $A, B \in N$, $x \in (N \cup \Sigma)^*$.

Theorem

Eine Sprache läßt sich genau dann durch eine rechtslineare Grammatik definieren, wenn sie regulär ist.

Reguläre Sprachen

Theorem

Eine Sprache ist genau dann regulär wenn sie

- *durch einen regulären Ausdruck beschrieben werden kann;*
- *durch einen deterministischen regulären Automaten erkannt wird;*
- *durch eine nicht-deterministischen regulären Automaten erkannt wird;*
- *durch eine rechtslineare Grammatik beschrieben werden kann;*
- *durch eine linkslineare Grammatiken beschrieben werden kann.*

Kontexfreie und kontextsensitiv

Definition

Eine Grammatik $\mathcal{G} = (\Sigma, N, S, \mathcal{P})$ heißt kontextfrei, falls alle Produktionsregel die Form

$$A \Rightarrow r$$

haben, mit $A \in N$, $r \in (N \cup \Sigma)^*$. Eine Sprache heißt kontextfrei wenn sie durch eine kontextfreie Grammatik definiert werden kann.

Definition

Eine Grammatik $\mathcal{G} = (\Sigma, N, S, \mathcal{P})$ heißt kontextsensitiv, falls alle Produktionsregel die Form

$$uAv \Rightarrow urv$$

haben, mit $A \in N$, $r, u, v \in (N \cup \Sigma)^*$, $r \neq \epsilon$. Eine Sprache heißt kontextsensitiv wenn sie durch eine kontextsensitive Grammatik definiert werden kann.

Definition

Eine Sprache heißt kontextfrei (bzw. kontextsensitiv) wenn sie durch

Bemerkung

Jede kontextfreie Sprache ist kontextsensitiv.