Formale Sprachen

1 Def. Grammatik

$G = (V, \Sigma, \rightarrow, S) \qquad V \cap \Sigma = \emptyset, \rightarrow \subset (V \cup \Sigma)^+ \times (V \cup \Sigma)^*,\ S \in V,\ \ V, \Sigma, \rightarrow$ endlich

V: Menge der Variablen, Σ: Terminalalphabet, S: Satzsymbol

$\rightarrow$: Regel–/Produktionsrelation (auch Bez. P).

2 Chomsky–Hierarchie

Typ 0 (Phrasenstrukturg.): keine Einschränkungen

Typ 1 (kontextsensitiv): Für alle Regeln $w \rightarrow w'$ gilt: $|w| \leq |w'|$

Typ 2 (kontextfrei): Für alle Regeln $w \rightarrow w'$ gilt: $w \in V$

Typ 3 (regulär): Für alle Regeln $w \rightarrow w'$ gilt: $w \in V$ und $w' \in \Sigma \cup \Sigma V$

ε–Sonderregelung: Ist bei Typ 1, 2, 3–Grammatiken $\varepsilon \in L$ erwünscht, so ist zusätzlich die Regel $S \rightarrow \varepsilon$ zugelassen (dann darf S aber auf keiner rechten Seite einer Produktionsregel vorkommen!). Bei kontextfreien Grammatiken sind auch Regeln $A \rightarrow \varepsilon\ \forall A \in V$ zugelassen, da man solche Grammatiken äquivalent in ε–freie umformen kann.

3 Reguläre Sprachen

Reguläre Sprachen werden äquivalent durch Typ–3–Grammatiken, deterministische endliche Automaten (DFA), nichtdeterministische endliche Automaten (NFA) und reguläre Ausdrücke beschrieben.

3.1 DFA

$M = (Z, \Sigma, \delta, z_0, E) \qquad Z \cap \Sigma = \emptyset, z_0 \in Z, E \subseteq Z,\ \delta : Z \times \Sigma \rightarrow Z\ ,\ Z, \Sigma, \delta$ endlich

Z: Menge der Zustände, δ: Überführungsfunktion, z_0: Startzustand, E: Menge der Endzustände

Erweiterung von δ zu $\hat{\delta} : Z \times \Sigma^* \rightarrow Z$ (Erkennen von Wörtern):

$$\hat{\delta}(z, \varepsilon) = z \qquad \hat{\delta}(z, ax) = \hat{\delta}(\delta(z, a), x)$$

Akzeptierte Sprache: $T(M) = \{x \in \Sigma^* \mid \hat{\delta}(z_0, x) \in E\}$

3.2 NFA

$M = (Z, \Sigma, \delta, S, E) \qquad Z \cap \Sigma = \emptyset, S \subseteq Z, E \subseteq Z,\ \delta : Z \times \Sigma \rightarrow \mathcal{P}(Z)$

S: Menge der Startzustände

Verallgemeinerung von δ zu $\hat{\delta} : \mathcal{P}(Z) \times \Sigma^* \rightarrow \mathcal{P}(Z)$:

$$\hat{\delta}(Z', \varepsilon) = Z'\ \ \forall Z' \subseteq Z \qquad \hat{\delta}(Z', ax) = \bigcup_{z \in Z'} \hat{\delta}(\delta(z, a), x)$$

Akzeptierte Sprache: $T(M) = \{x \in \Sigma * \mid \hat{\delta}(S, x) \cap E \neq \emptyset\}$

3.3 Reguläre Ausdrücke

Die Menge $\mathcal{R}_\Sigma$ der regulären Ausdrücke über einem Alphabet Σ ist wie folgt definiert:

$$\emptyset \in \mathcal{R}_\Sigma \qquad \varepsilon \in \mathcal{R}_\Sigma \qquad (\forall a \in \Sigma)\; a \in \mathcal{R}_\Sigma \qquad \alpha, \beta \in \mathcal{R}_\Sigma \;\rightarrow\; \alpha\beta, (\alpha|\beta), (\alpha)^* \;\in \mathcal{R}_\Sigma$$

Semantik $L(\gamma)$ eines regulären Ausdrucks γ:

$$L(\emptyset) = \emptyset \qquad L(\varepsilon) = \{\varepsilon\} \qquad L(a) = \{a\}$$

$$L(\alpha\beta) = L(\alpha)L(\beta) \qquad L((\alpha|\beta)) = L(\alpha) \cup L(\beta) \qquad L((\alpha)^*) = L(\alpha)^*$$

3.4 Umwandlung reg. Grammatik → NFA

geg: reguläre Grammatik $G = (V, \Sigma, P, S)$

$\Rightarrow$ NFA $M = (V \cup \{X\}, \Sigma, \delta, \{S\}, E)$, $X \notin V$ mit

$$\begin{array}{rcl} E & = & \begin{cases} \{S, X\}, & S \rightarrow \varepsilon \in P \\ \{X\}, & S \rightarrow \varepsilon \notin P \end{cases} \\ \delta(A, a) & \ni & B \text{ falls } A \rightarrow aB \in P \\ \delta(A, a) & \ni & X \text{ falls } A \rightarrow a \in P \end{array}$$

3.5 Umwandlung NFA→ DFA

geg: NFA $M = (Z, \Sigma, \delta, S, E)$

$\Rightarrow$ DFA $M' = (\mathcal{P}(Z), \Sigma, \delta', S, E')$ (Zustandsnamen sind jetzt Teilmengen von Z!) mit

$$\delta'(X, a) = \bigcup_{x \in X} \delta(x, a) \qquad E' = \{X \subseteq Z \mid X \cap E \neq \emptyset\}$$

3.6 Umwandlung DFA→ reg. Grammatik

geg: DFA $M = (Z, \Sigma, \delta, z_0, E)$

$\Rightarrow$ Grammatik $G = (Z, \Sigma, P, z_0)$ mit

$$\begin{array}{rcl} z_0 \rightarrow \varepsilon \;\in P & \leftrightarrow & \varepsilon \in T(M) \\ z_1 \rightarrow az_2 \;\in P & \leftrightarrow & \delta(z_1, a) = z_2 \\ z_1 \rightarrow a \;\in P & \leftrightarrow & \delta(z_1, a) = z_E \;\wedge\; z_E \in E \end{array}$$

3.7 Pumping Lemma

$$(L \text{ regulär}) \rightarrow (\exists n \in \mathbb{N})(\forall z \in L, |z| \geq n)(\exists u, v, w) \left[\, uvw = z \quad \wedge \quad |v| \geq 1 \quad \wedge \quad |uv| \leq n \quad \wedge \quad (*) \,\right]$$

$(*) \qquad (\forall i \in \mathbb{N}_0) \quad uv^iw \in L$

3.8 Abschlußeigenschaften

Die regulären Sprachen sind abgeschlossen unter Vereinigung, Schnitt, Komplement, Produkt und Stern.

4 Kontextfreie Sprachen

Kontextfreie Sprachen werden äquivalent durch Typ–2–Grammatiken, EBNF–Terme und (nichtdeterministische) Kellerautomaten (PDA) beschrieben.

4.1 Grammatik–Normalformen

(große Buchstaben: Variable, kleine Buchstaben: Terminalsymbole)

Chomsky–Normalform: Alle Regeln haben die Form $A \to BC$ oder $A \to a$.

Greibach–Normalform: Alle Regeln haben die Form $A \to aB_1B_2 \dots B_k$, $k \geq 0$.

4.2 EBNF (Extended Backus–Naur–Form)

$$\begin{aligned}
\{A \to \beta_1, A \to \beta_2, \dots, A \to \beta_n\} \quad &\Leftrightarrow \quad A ::= \beta_1 \mid \beta_2 \mid \dots \mid \beta_n \\
\{A \to \alpha\gamma, A \to \alpha\beta\gamma\} \quad &\Leftrightarrow \quad A ::= \alpha[\beta]\gamma \\
\{A \to \alpha\gamma, A \to \alpha B\gamma, B \to \beta, B \to \beta B\} \quad &\Leftrightarrow \quad A ::= \alpha\{\beta\}\gamma
\end{aligned}$$

4.3 PDA

$M = (Z, \Sigma, \Gamma, \delta, z_0, \#) \qquad \delta : Z \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_e(Z \times \Gamma^*)$, $\# \in \Gamma$

Γ: Kelleralphabet, #: unterstes Kellerzeichen

Eine **Konfiguration** k eines PDA ist ein Tripel $k \in K$ mit $K := Z \times \Sigma^* \times \Gamma^*$.

Definition einer Relation $\vdash \subseteq K \times K$:

$$(z, a_1 \dots a_n, A_1 \dots A_m) \vdash \begin{cases} (z', a_2 \dots a_n, B_1 \dots B_k A_2 \dots A_m) & \text{falls } \delta(z, a_1, A_1) \ni (z', B_1 \dots B_k) \\ (z', a_1 \dots a_n, B_1 \dots B_k A_2 \dots A_m) & \text{falls } \delta(z, \varepsilon, A_1) \ni (z', B_1 \dots B_k) \end{cases}$$

Akzeptiere Sprache: $N(M) = \{x \in \Sigma^* \mid (z_0, x, \#) \vdash^* (z, \varepsilon, \varepsilon)\}$

4.4 Pumping Lemma

$$(L \text{ kontextfrei}) \to (\exists n \in \mathbb{N})(\forall z \in L, |z| \geq n)(\exists u, v, w, x, y) \left[uvwxy = z \wedge |vx| \geq 1 \wedge |vwx| \leq n \wedge (*) \right]$$

$(*) \qquad (\forall i \in \mathbb{N}_0) \quad uv^iwx^iy \in L$

4.5 Abschlußeigenschaften

Die kontextfreien Sprachen sind abgeschlossen unter Vereinigung, Produkt und Stern.

5 Kontextsensitive Sprachen

Kontextsensitive Sprachen werden äquivalent durch Typ–1–Grammatiken und linear beschränkte Turingmaschinen (LBA) beschrieben.

5.1 LBA

Definition der Turing–Maschine im nächsten Abschnitt!

"Verdopplung" des Eingabealphabets der TM zu $\Sigma' = \Sigma \cup \{\hat{a} \mid a \in \Sigma\}$. Die "eigentliche" Eingabe $a_1a_2 \dots a_{n-1}a_n$ wird auf dem Band repräsentiert durch $a_1a_2 \dots a_{n-1}\hat{a_n}$, d. h., das letzte Zeichen wird markiert.

Eine nichtdeterministische Turingmaschine (Def. siehe nächster Abschnitt) heißt **linear beschränkt**, wenn für alle $a_1 a_2 \ldots a_n \in \Sigma^+$ und alle Konfigurationen $\alpha z \beta$ mit $z_0 a_1 a_2 \ldots a_{n-1} \hat{a_n} \vdash^* \alpha z \beta$ gilt: $|\alpha\beta| = n$.

Ob deterministische LBA (DLBA) gleichmächtig zu LBA sind, ist noch nicht bekannt.

Akzeptierte Sprache:

$T(M) = \{a_1 a_2 \ldots a_{n-1} a_n \in \Sigma^* \mid z_0 a_1 a_2 \ldots a_{n-1} \hat{a_n} \vdash^* \alpha z \beta \; ; \; \alpha, \beta \in \Gamma^*, z \in E\}$

5.2 Abschlußeigenschaften

Die kontextsensitiven Sprachen sind abgeschlossen unter Schnitt, Vereinigung, Komplement, Produkt und Stern.

6 Typ–0–Sprachen

Typ–0–Sprachen werden äquivalent durch Typ–0–Grammatiken und Turingmaschinen (TM) beschrieben.

6.1 TM

$M = (Z, \Sigma, \Gamma, \delta, z_0, \Box, E)$ $\quad \Gamma \supset \Sigma, \Box \in \Gamma, E \subseteq Z, \delta : Z \times \Gamma \to Z \times \Gamma \times \{L, R, N\}$ (deterministisch), bzw. $\delta : Z \times \Gamma \to \mathcal{P}(Z \times \Gamma \times \{L, R, N\})$ (nichtdeterministisch)

Γ: Arbeitsalphabet, $\Box$: "Blank"

Eine **Konfiguration** einer Turingmaschine ist ein Wort $k \in K$ mit $K := \Gamma^* Z \Gamma^*$.

Definition einer "Schritt"–Relation $\vdash \subseteq K \times K$:

$$a_1 \ldots a_m z b_1 \ldots b_n \vdash \begin{cases} a_1 \ldots a_m z' c b_2 \ldots b_n & \text{falls } \delta(z, b_1) = (z', c, N), m \geq 0, n \geq 1 \\ a_1 \ldots a_m c z' b_2 \ldots b_n & \text{falls } \delta(z, b_1) = (z', c, R), m \geq 0, n \geq 2 \\ a_1 \ldots a_{m-1} z' a_m c b_2 \ldots b_n & \text{falls } \delta(z, b_1) = (z', c, L), m \geq 1, n \geq 1 \end{cases}$$

Sonderfälle an den "Rändern":

$$\begin{aligned} a_1 \ldots a_m z b_1 &\vdash a_1 \ldots a_m c z' \Box \quad \text{falls} \quad \delta(z, b_1) = (z', c, R) \\ z b_1 \ldots b_n &\vdash z' \Box c b_2 \ldots b_n \quad \text{falls} \quad \delta(z, b_1) = (z', c, L) \end{aligned}$$

Akzeptierte Sprache: $T(M) = \{x \in \Sigma^* \mid z_0 x \vdash^* \alpha z \beta \; ; \; \alpha, \beta \in \Gamma^*, z \in E\}$

6.2 Abschlußeigenschaften

Die Typ–0–Sprachen sind abgeschlossen unter Schnitt, Vereinigung, Produkt und Stern.