1 Chomsky Hierarchie

1.1. Definition:

(i) Eine *Grammatik* $G = (V, \Sigma, P, S)$ ist ein 4-Tupel mit *Grammatik*

V: endliche Menge von *Variablen* *Variable*

Σ: endliche Menge von *Terminalsymbolen* *Terminalsymbol*

S: *Startsymbol* *Startsymbol*

P: $\subseteq ((V \cup \Sigma)^* \setminus \Sigma^*) \times (V \cup \Sigma)^*$ Menge der *Produktionen*. *Produktion*

Schreibweise: $\alpha \to \beta$ für $(\alpha, \beta) \in P$

(ii) Seien $G = (V, \Sigma, P, S)$ Grammatik, $u, v \in (V \cup \Sigma)^*$.
$u \to v :\Leftrightarrow \exists \alpha, \beta, x, y \in (V \cup \Sigma)^* : u = \alpha x \beta \wedge v = \alpha y \beta \wedge (x, y) \in P$

(iii) $\overset{*}{\to}$ ist der reflexive transitive Abschluss von $\to$

(iv) von der Grammatik G *erzeugte Sprache*: $L(G) := \{w \in \Sigma^*; S \overset{*}{\to} w\}$ *erzeugte Sprache*

1.2. Definition (eingeschränkte Grammatiken): Sei $G = (V, \Sigma, P, S)$ Grammatik.

(i) G *rechtslinear* $:\Leftrightarrow \forall (\alpha \to \beta) \in P : \alpha \in V \wedge (\exists u \in \Sigma, B \in V : \beta = uB \vee \beta = u)$ *rechtslinear*

(ii) G *linear* $:\Leftrightarrow \forall (\alpha \to \beta) \in P : \alpha \in V \wedge (\exists u, v \in \Sigma, B \in V : \beta = uBv \vee \beta = u)$ *linear*

(iii) G *kontextfrei* $:\Leftrightarrow \forall (\alpha \to \beta) \in P : \alpha \in V \wedge \beta \in (V \cup \Sigma)^*$ *kontextfrei*

(iv) G *kontextsensitiv* $:\Leftrightarrow \begin{cases} \forall (\alpha \to \beta) \in P : & \exists u, v, x \in (V \cup \Sigma)^*, A \in V : \alpha = uAv \\ & \wedge \beta = uxv \wedge |x| \geq 1 \end{cases}$ *kontextsensitiv*

(v) G *beschränkt* $:\Leftrightarrow \forall (\alpha \to \beta) \in P : |\alpha| \leq |\beta|$ *beschränkt*

(vi) G heißt allgemein, wenn keine Einschränkungen gelten

1.3. Lemma: Zu jeder beschränkten Grammatik G gibt es eine kontextsensitive Grammatik G' mit $L(G) = L(G')$.

Beweis: Sei $G = (V, \Sigma, P, S)$ eine beschränkte Grammatik. Konstruktion von $G' = (V', \Sigma, P', S)$ kontextsensitiv mit $L(G) = L(G')$. Sei $(\alpha \to \beta) \in P$. Betrachte folgende Fälle:

(i) $|\alpha| = 1$: Regel ist kontextsensitiv und wird in P' aufgenommen

(ii) $|\alpha| = 2$: $X_1X_2 \to Y_1Y_2 \ldots V_r$ mit $r \geq 2$, $X_1, X_2, Y_1, \ldots, Y_r \in V \cup \Sigma$, Einführung von zwei neuen Variablen: $H_1, H_2 \in V' \setminus V$ und Regeln $X_1X_2 \to H_1X_2$, $H_1X_2 \to H_1H_2Y_3 \ldots Y_r$, $r \geq m$, $H_1H_2 \to Y_1H_2$, $Y_1H_2 \to Y_1Y_2$

(iii) $|\alpha| = m > 2$: $X_1 \dots X_m \to Y_1 \dots Y_r$, Einführung von m neuen Variablen $H_1, \dots, H_m \in V' \setminus V$ und Regeln $X_1 \dots X_m \to H_1 X_2 \dots X_m$, $H_1 X_2 \dots X_m \to H_1 H_2 X_3 \dots X_m$, ..., $H_1 \dots H_{m-1} X_m \to H_1 \dots H_m Y_{m+1} \dots Y_r$, $H_1 \dots H_m \to Y_1 H_2 \dots H_m$, ..., $Y_1 \to Y_{m-1} H_m \to H_1 \dots H_m$ (insgesamt $2m$ Regeln).

□

1.4. Definition (Sprachklassen):

reguläre Sprachen

lineare Sprachen

kontextfreie Sprachen

kontextsensitive Sprachen

allgemeine Sprachen

(i) $\mathcal{L}_3 = \mathrm{REG} = \{L(G) | G \text{ rechtslinear}\}$ Klasse der *regulären Sprachen*

(ii) $\mathrm{LIN} = \{L(G) | G \text{ linear}\}$ Klasse der *linearen Sprachen*

(iii) $\mathcal{L}_2 = \mathrm{CFL} = \{L(G) | G \text{ kontextfrei}\}$ Klasse der *kontextfreien Sprachen*

(iv) $\mathcal{L}_1 = \mathrm{CSL} = \{L(G) | G \text{ kontextsensitiv}\}$ Klasse der *kontextsensitiven Sprachen*

(v) $\mathcal{L}_0 = \{L(G) | G \text{ allgeine Grammatik}\}$ Klasse der *allgemeinen Sprachen*

1.5. Bemerkung (Beziehungen zwischen Sprachklassen): Es gilt:

$$\mathcal{L}_3 \subsetneq \mathrm{LIN} \subsetneq \mathcal{L}_2 \subsetneq \mathcal{L}_1 \subsetneq \mathcal{L}_0$$

Beweis: Teilmengenbeziehung klar, Ungleichheit:

(i) $\mathcal{L}_3 \neq \mathrm{LIN}$: $\{a^n b^n | n \in \mathbb{N}\} \in \mathcal{L}_3 \setminus \mathrm{LIN}$

(ii) $\mathrm{LIN} \neq \mathcal{L}_2$: $\{a^n b^n c^m d^m | m, n \in \mathbb{N}\} \in \mathcal{L}_2 \setminus \mathrm{LIN}$

(iii) $\mathcal{L}_2 \neq \mathcal{L}_1$: $\{a^n b^n c^n | n \in \mathbb{N}\} \in \mathcal{L}_1 \setminus \mathcal{L}_2$

(iv) $\mathcal{L}_1 \neq \mathcal{L}_0$: Beweis s.u. (Abschlußeigenschaft)

□

1.6. Definition: Seien X, Y Alphabete und $h : X \to Y^*$ eine Funktion.

Homomorphismus

(i) Die eindeutige Erweiterung $h : X^* \to Y^* : h(a_1 \dots a_n) := h(a_1) \dots h(a_n)$ heißt *Homomorphismus.*

(ii) Sei $L \subseteq X^*$. $h(L) := \{h(v) | v \in L\}$

(iii) Sei $L \subseteq Y^*$. $h^{-1}(L) := \{w \in X^* | h(w) \in L\}$

(iv) Sei $\mathcal{L}$ eine Menge von Sprachen. $\mathrm{HOM}(\mathcal{L}) := \{h(L) | L \in \mathcal{L}, h \text{ Homomorphismus}\}$ und $\mathrm{HOM}^{-1}(\mathcal{L}) := \{h^{-1}(L) | L \in \mathcal{L}, h \text{ Homomorphismus}\}$

1.7. Definition: Eine Sprachklasse $\mathcal{L}$ heißt *abgeschlossen unter Homomorphismus* $:\Leftrightarrow$ $\mathrm{HOM}(\mathcal{L}) = \mathcal{L}$

abgeschlossen unter Homomorphismus

1.8. Satz:

(i) $\mathcal{L}_3, \mathcal{L}_2, \mathcal{L}_0$ sind abgeschlossen unter Homomorphismus

(ii) $\mathrm{HOM}(\mathcal{L}_1) = \mathcal{L}_0$

(iii) $\mathcal{L}_3, \mathcal{L}_2, \mathcal{L}_1, \mathcal{L}_0$ sind abgeschlossen unter inversem Homomorphismus

Beweis: (i) Sei $G = (V, \Sigma, P, S)$ Grammatik vom Typ $i \in \{0, 2, 4\}$, $h : \Sigma^* \to \Gamma^*$ Homomorphismus. Sei $L = L(G) \in \mathcal{L}_i$.

Beh.: $h(L) \in \mathcal{L}_i$

Beweis: Konstruktion von G' mit $L(G') = L$ und G' ist Grammatik vom Typ i: Erweiterung von h auf $(\Sigma \cup V)^*$:

$$h' : \Sigma \cup V \to \Gamma^* \cup V : x \mapsto \begin{cases} h(x) & \text{falls } x \in \Sigma \\ x & \text{falls } x \in V \end{cases}$$

h' wird auf $(\Sigma \cup V)^*$ erweitert: Für $a_1, \dots, a_n \in \Sigma \cup V$: $h'(a_1 \dots a_n) := h'(a_1) \dots h'(a_n)$.

$G' := (V, \Gamma, P', S)$ mit $P' = \{(h(\alpha), h(\beta) \big| (\alpha, \beta) \in P\}$.

Beh.: $S \underset{G}{\overset{*}{\to}} \alpha \Leftrightarrow S \underset{G'}{\overset{*}{\to}} h(\alpha)$

Beweis: Induktion nach Zahl der Ableitungsschritte r.

IA: $r = 1$: $S \underset{G}{\overset{*}{\to}} \alpha \Leftrightarrow S \underset{G}{\to} \in P \Leftrightarrow S \underset{G'}{\to} h(\alpha) \in P' \Leftrightarrow S \underset{G'}{\overset{*}{\to}} \alpha$

IS: $r \leadsto r+1$: Die Behauptung gelte für r Ableitungsschritte. Sei $S \underset{G}{\overset{*}{\to}} \alpha_1\alpha_2\alpha_3 \to \alpha_1\beta\alpha_3$ mit $(\alpha_2 \to \beta) \in P$ eine beliebige Ableitung mit $r+1$ Schritten. Nach IV gilt $S \underset{G'}{\overset{*}{\to}} h(\alpha_1\alpha_2\alpha_3) = h(\alpha_1)h(\alpha_2)h(\alpha_3) \underset{G'}{\to} h(\alpha_1)h(\beta)h(\alpha_3) = h(\alpha_1\beta\alpha_3)$, da $h(\alpha_2) \to h(\beta) \in P'$. Analog folgt „$\Leftarrow$“.

Durch Anwendung von h' auf die Produktionen bleiben die Eigenschaften kontextsenitiv bzw. rechtslinear erhalten.

(ii) $\mathrm{HOM}(\mathcal{L}_1) \subseteq \mathcal{L}_0$ (da $\mathcal{L}_0$ abgeschlossen unter HOM), z.z. $\mathcal{L}_0 \subseteq \mathrm{HOM}(\mathcal{L}_1)$. Sei $L \in \mathcal{L}_0$. Konstruktion von $L' \in \mathcal{L}_1$ und Homomorphismus h mit $h(L') = L$. Sei $G = (V, \Sigma, P, S)$ mit $L(G) = L$ und $\forall(\alpha \to \beta) \in P : \alpha \in V^+$ (dies kann leicht durch zusätzliche Variablen erreicht werden).

Sei $G' = (V \cup \{D\}, \Sigma \cup \{d\}, P', S)$ mit

$$\begin{aligned} P' = \ & \{(u, v) \in P \big| |u| \le |v|\} \\ & \cup \{(u, vD^{|u|-|v|}) \big| (u, v) \in P, |u| > |v|\} \\ & \cup \{(AD, DA), (DA, AD) \big| A \in V \cup \Sigma\} \cup \{(D, d)\} \end{aligned} .$$

Dann gilt: $S \underset{G}{\overset{*}{\to}} \alpha \Leftrightarrow \exists r \in \mathbb{N} : S \underset{G'}{\overset{*}{\to}} \alpha D^r$.

$\Rightarrow L' = L(G') \subseteq \{d^{r_0}u_1 d^{r_1} u_2 \ldots d^{r_{n-1}} u_n d^{r_n} \big| u = u_1 u_2 \ldots u_n \in L, r_0, \ldots, r_n \in \mathbb{N}_0\}$.

Mit $h(a) := \begin{cases} a & \text{falls } a \in \Sigma \\ \varepsilon & \text{falls } a = d \end{cases}$ gilt $h(L') = L$.

(iii) **Beh.:** $\mathcal{L}_0$, $\mathcal{L}_1$ abgeschlossen gegen h^{-1}

Beweis: Sei $L \in \mathcal{L}_0$ (oder $\in \mathcal{L}_1$). Sei M die TM (oder der LBA), der L akzeptiert. Sei h ein Homomorphismus. Sei M' die Turingmaschine, die aus der Eingabe x zuerst $h(x)$ berechnet und dann M simuliert. Dann akzeptiert M' $h(L)$. Falls M ein LBA, so ist auch M' ein LBA, denn die Anwendung von h verlängert die Eingabe maximal um den konstanten Faktor $c = \max\{|h(a)| \big| a \in \Sigma\}$.

Beh.: $\mathcal{L}_2$, $\mathcal{L}_3$ abgeschlossen gegen h^{-1}

Beweis: Sei M ein endlicher Automat (oder ein 1-PDA). M' liest Schritt für Schritt ein Zeichen a und simuliert M mit der Eingabe $h(a)$.

□

1.9. Satz: Zu jedem $L \in \mathcal{L}_0$ gibt es

(i) $L_1, L_2 \in \mathcal{L}_2$ und Homomorphismus h mit $h(L_1 \cap L_2) = L$

(ii) L_1 mit $\overline{L_1} \in \mathcal{L}_2$ und Homomorphismus h mit $h(L_1) = L$

Beweis: (i) Sei $L \in \mathcal{L}_0$, sei $M = (Q, \Sigma, \Gamma, \delta, q_o, F)$ TM mit $L = L(M)$. Definiere:

$$L_{KF} := \Big\{\overline{w}¢K_1¢\ldots¢K_r \Big| \begin{array}{l} r \in \mathbb{N} \text{ ungerade}, \forall i \in \{1, \ldots, r-1\} : K_i \in \Gamma^* Q \Gamma^*, \\ K_1 = \mathrm{rev}(q_0 w), K_r \in \Gamma^* F \Gamma^*, \\ \forall i \text{ gerade} : K_i \underset{M}{\to} \mathrm{rev}(K_{i+1}), \\ \forall i \text{ ungerade} : \mathrm{rev}(K_i) \underset{M}{\to} K_{i+1} \end{array}\Big\}$$

Weiter: $\overline{\Sigma} := \{\overline{a} \big| a \in \Sigma\}$ und $h : \Sigma \cup \overline{\Sigma} \cup \Gamma \cup \{¢\} \to \Sigma^* : x \mapsto \begin{cases} a & \text{falls } x = \overline{a} \\ \varepsilon & \text{sonst} \end{cases}$.

Dann gilt: $h(L_{KF}) = L(M)$.

Beh.: L_{KF} ist Schnitt zweier kontextfreier Sprachen.

Beweis: Sei L_1 und L_2 folgende Sprachen:

$$\begin{aligned} L_1 &= \{\overline{w}¢\,\mathrm{rev}(q_0 w)¢K_2¢\ldots¢K_{r-1}¢K_r \big| K_i \in \Gamma^* Q \Gamma^*, K_r \in \Gamma^* F \Gamma^*, \\ &\qquad K_{2i} \to \mathrm{rev}(K_{2i+1})\} \\ L_2 &= \{\overline{w}¢K_1¢\ldots¢K_r \big| K_i \in \Gamma^* Q \Gamma^*, K_r \in \Gamma^* F \Gamma^*, \mathrm{rev}(K_{2i-1}) \to K_{2i}\} \end{aligned}$$

Dann gilt: $L = L_1 \cap L_2$ und L_1 und L_2 sind kontextfrei.

(ii) **Beh.:** $\overline{L_{KF}(M)} \in \mathcal{L}_2$

Beweis: $\overline{L_{KF}(M)}$ wird als Vereinigung einer regulären Sprache $\overline{R}$ und einer kontextfreien Sprache L' dargestellt. In $\overline{R}$ sind alle Worte, die ein falsches Format haben, in L' sind Worte, bei denen mindestens ein Übergang falsch ist.

Sei $Y := \Gamma \cup Q \cup \Sigma \cup \overline{\Sigma} \cup \{\cent\}$.

$$R := \{u \in Y \big| u = \overline{w}\cent K_1 \cent \ldots \cent K_r, r \geq 3, K_i \in \Gamma^* Q \Gamma^*, K_r \in \Gamma^* F \Gamma^*\}$$
$$L' := \{\alpha \cent u_1 \cent u_2 \cent \beta \big| \alpha, \beta \in Y^*, u1, u2 \in \Gamma^* Q \Gamma^*, u_1 \underset{M}{\not\vdash} \mathrm{rev}(u_2)\}$$

Dann gilt $R \in \mathcal{L}_3$ und somit auch $\overline{R} \in \mathcal{L}_3$. Weiter gilt $L' \in \mathcal{L}_2$ (Übungsaufgabe) und $\overline{L_{KF}(M)} = \overline{R} \cup L'$.

□

1.10. Lemma: $\mathcal{L}_2$ ist nicht abgeschlossen gegen Komplementbildung.

Beweis: Annahme: $\mathcal{L}_2$ abgeschlossen gegenüber Komplement. Sei $L \in \mathcal{L}_0$. Dann existiert nach Satz 1.9 L_1 mit $\overline{L_1} \in \mathcal{L}_2$ und Homomorphismus h mit $h(L_1) = L$. Also folgt $L_1 \in \mathcal{L}_2 \Rightarrow L = h(L_1) \in \mathcal{L}_2 \Rightarrow \mathcal{L}_0 \subseteq \mathcal{L}_2$ Widerspruch.

□

1.11. Bemerkung (Abschlußeigenschaften):

	h	h^{-1}	nichtlösch.h	$\cup$	$\cap$	Komplement	Kleene+
$\mathcal{L}_3$	+	+	+	+	+	+	+
$\mathcal{L}_2$	+	+	+	+	−	−	+
$\mathcal{L}_1$	−	+	+	+	+	$+^*$	+
$\mathcal{L}_0$	+	+	+	+	+	−	+

1.12. Definition (Turingmaschinen):

(i) Eine *Turingmaschine* (TM) M ist ein 6-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, F)$ mit *Turingmaschine*

Q: Endliche Menge von *Zuständen* *Zustände*

Σ: Endliche Menge, *Alphabet*

Γ: Endliche Menge, $\Sigma \subset \Gamma$, , $B \in \Gamma \setminus \Sigma$ *Blanksymbol* *Bandalphabet*

δ: Abbildung *Übergangsrelation*, bei *deterministischen* TMs $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, N, R\}$, bei *nichtdeterministischen* TMs $\delta : Q \times \Gamma \to \mathrm{Pot}(Q \times \Gamma \times \{L, N, R\})$. *Blanksymbol* *Übergangsrelation*

$q_0 \in Q$: Startzustand *deterministisch*

$F \subseteq Q$: Akzeptierende Zustände *nichtdeterministisch*

*Satz von Immerman, Szelepcsényi, s. Seite 8 *linear beschränkter Automat*

(ii) Ein *linear beschränkter Automat* (LBA) ist eine TM, die den durch BwB definierten Bandabschnitt bei Eingabe w nicht verläßt.

Endlicher Automat

(iii) Ein *Endlicher Automat* (EA, auch FA*) ist eine Turingmaschine, die die Eingabe nur einmal von links nach rechts lesen kann. $M = (Q, \Sigma, \delta, q_0, F)$ mit $\delta : Q \times \Sigma \to Q$.

Kellerautomat
Kelleralphabet
Kellerendesymbol

(iv) Ein *Kellerautomat* (KA, auch PDA†) ist ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ mit Γ endliches *Kelleralphabet*, $Z_0 \in \Gamma$ *Kellerendesymbol*, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \subseteq Q \to \mathrm{Pot}(Q \times \Gamma^*)$

Ein Kellerautomat heißt *deterministisch*, falls

(1) $\delta(q, \varepsilon, c) \neq \emptyset \Rightarrow \forall a \in \Sigma : \delta(q, a, c) = \emptyset$ und

(2) $\forall q \in Q, c \in \Gamma, a \in \Sigma \cup \{\varepsilon\} : |\delta(q, a, c)| \leq 1$.

Weitere Kellerautomaten: Ein- bzw. Zwei-Wege (nicht) det. Kellerautomaten mit k Köpfen, Sprachklassen: 1DPDA(k), 2DPDA(k), 1NPDA(k), 2NPDA(k).

Zählmaschine

(v) Eine *Zählmaschine* ist ein Kellerautomat mit nur einem Kellersymbol (außer dem Kellerendesymbol), Sprachklassen: 1DC, 1NC, 2DC, 2NC

1.13. Satz (Zusammenhang Sprachklassen und Automaten): Sei $L \subset \Sigma^*$ eine Sprache. Dann gilt:

$$L \in \mathcal{L}_1 \;\Leftrightarrow\; \exists G \text{ beschränkt mit } L(G) = L \;\Leftrightarrow\; \exists \text{NLBA } M \text{ mit } L = L(M)$$

Beweis: Zu zeigen bleibt die zweite Äquivalenz.

"⇒": Seien $G = (V, \Sigma, P, S)$ beschränkte Grammatik und $S = \varphi_1 \to \cdots \to \varphi_r = w \in \Sigma^*$ eine Ableitung. Da G beschränkt ist, gilt $\forall i \in \{1, \ldots, r-1\} : |\varphi_i| \leq |\varphi_{i+1}|$, insbesondere $\forall i \in \{1, \ldots, r-1\} : |\varphi_i| \leq |w|$, d.h. die Zwischenergebnisse jeder Ableitung in G sind nie länger als das abgeleitete Wort.

Konstruktion eines 2-Band nichtdet. linear beschränkten Automatens M:

(i) Simululiere auf dem zweiten Band eine beliebige Ableitung, ohne dabei die ersten $n = |w|$ Zellen zu verlassen.

(ii) Vergleiche, ob auf beiden Bändern das gleiche Wort steht.

"⇐": Sei $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ ein NLBA mit einem Band. O.B.d.A. ersetze M die beiden Randzeichen a_1 und a_n der Eingabe durch $\overline{a_1}$ bzw. $\overline{a_1}$ und verlasse den so markierten Bereich niemals.

Wir konstruieren eine beschränkte Grammatik $G = (\Sigma, V, P, S)$, deren Variablen teilweise Paare (vergl. 2-Band-TMs) sind. Im ersten Element des Paares bleibt die Eingabe stehen, im zweiten Element wird M simuliert. Dies ergibt die folgenden Produktionen:

*finite automaton
†push down automaton

Startphase: $\forall a \in \Sigma : S \to \binom{a}{q_0\overline{a}}A,\ A \to \binom{a}{a}A,\ A \to \binom{a}{\overline{a}}$
Simulation: Simulation eines Schrittes von M:

$$\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}\begin{pmatrix} a_2 \\ qb_2 \end{pmatrix}\begin{pmatrix} a_3 \\ b_3 \end{pmatrix} \underset{G}{\to} \begin{cases} \binom{a_1}{b_1}\binom{a_2}{b'}\binom{a_3}{q'b_3} & \text{falls } (q',b',R) \in \delta(q,b_2) \\ \binom{a_1}{b_1}\binom{a_2}{q'b'}\binom{a_3}{b_3} & \text{falls } (q',b',N) \in \delta(q,b_2) \\ \binom{a_1}{q'b_1}\binom{a_2}{b'}\binom{a_3}{b_3} & \text{falls } (q',b',L) \in \delta(q,b_2) \end{cases}$$

Akzeptierphase: Löschen der ersten Spur, falls ein Endzustand erreicht ist:
$\forall a \in \Sigma, b \in \Gamma : \binom{a}{b} \to a$ und $\forall q \in F, a \in \Sigma, b \in \Gamma : \binom{a}{qb} \to a$

□

1.14. Bemerkung (Zusammenhang Sprachklassen und Automaten):

(i) Die regulären Sprachen werden von deterministischen und nichtdeterministischen EAs akzeptiert, die kontextfreien Sprachen von den nd PDAs, die deterministisch kontextfreien Sprachen von den det. PDAs, die kontextsensitiven Sprachen von den nd. LBAs (ob auch von den det. LBAs, ist offen), die rekursiv aufzählbaren Sprachen ($\mathcal{L}_0$) von den TMs.

(ii) Es gelten folgende Zusammenhänge:

(1) $\mathcal{L}_3 = \text{1DFA}(1) = \text{1NFA}(1) = \text{2NFA}(1) = \text{2DFA}(1)$

(2) $\{a^n b^n c^n | n \in \mathbb{N}\} \in \text{1DFA}(2)$

(3) $\text{HOM}(\text{1DFA}(2)) = \mathcal{L}_0$

(4) $\bigcup\limits_k \text{2DFA}(k) = \text{DSPACE}(\log n),\ \bigcup\limits_k \text{2NFA}(k) = \text{NSPACE}(\log n)$

(5) $\text{1NPDA}(1) = \mathcal{L}_2 = \text{CFL},\ \text{1DPDA}(1) = \text{DCFL}$

(6) $\bigcup\limits_k \text{2DPDA}(k) = \bigcup\limits_k \text{2NPDA}(k) = \text{P},\ \text{2DPDA}(k) \subseteq \text{TIME}(n^k)$

1.15. Definition:

(i) Eine Funktion $s : \mathbb{N} \to \mathbb{N}$ heißt *bandkonstruierbar*, wenn es eine $s(n)$-platzbeschränkte Turingmaschine gibt, die bei Eingabe w die Binärdarstellung von $s(|w|)$ berechnet. *bandkonstruierbar*

(ii) $\text{DSPACE}(s(n)) := \{L | \exists\ s(n)\text{-bandbeschränkte det. TM } M \text{ mit } L = L(M)\}$, analog $\text{NSPACE}(s(n))$ für nichtdet. TMs

Algorithmus 1 Wortproblem für co(NSPACE($s(n)$))

Eingabe: N: Anzahl erreichbarer Konfigurationen
Ausgabe: w: true falls keine akzeptierende Berechnung gefunden

$z = 0$, w =true, K =erste Konfiguration
repeat
  rate Konfigurationsfolge $q_0 w \to k_1 \to \cdots \to k_r$ mit $|k_i| \leq s(n)$
  if $k_r = K$ **then**
    $z = z + 1$ {richtig geraten}
    if $K \in \Gamma^* F \Gamma^*$ **then**
      w =false
  K =Nachfolger(K)
until nicht w oder K ist keine gültige Konfiguration mehr
if $z < N$ **then**
  w =false {nicht alle erreichbaren Konfigurationen richtig geraten}

1.16. Satz (Satz von Immerman und Szelepcsényi): Für bandkonstruierbare Funktionen $s : \mathbb{N} \to \mathbb{N}$ gilt: NSPACE($s(n)$) = co(NSPACE($s(n)$))

Speziell: $\mathcal{L}_1$ = NSPACE(n) ist abgeschlossen unter Komplement.

Beweis: (vergl. [Weg99], Seite 143ff) Sei $s(n)$ bandkonstruierbar, $L \in$ NSPACE($s(n)$). Sei M die TM, die L mit Platz $s(n)$ akzeptiert. Dann existiert $c > 0$ mit M akzeptiert jedes $w \in L$ in $2^{cs(|w|)}$ Schritten. Wir konstruieren eine TM M', die $\overline{L}$ akzeptiert, indem sie alle möglichen Berechnungen von M der Länge maximal $2^{cs(|w|)}$ simuliert und akzeptiert, wenn keine Berechnung von M akzepierend war. Eine deterministische TM würde hierzu wahrscheinlich zu viel Platz benötigen, allerdings gelingt die Konstruktion mit Nichtdeterminismus.

Zuerst berechnen wir durch iterative Anwendung von Algorithmus 2 die Anzahl erreichbarer Konfigurationen von M bei Eingabe w und Platzbeschränkung $s(|w|)$ mit t Schritten für $t = 1, 2, \ldots$ bis $N_t = N_{t+1}$ gilt.

Anschließend überprüfen wir, ob keine der erreichbaren Konfigurationen akzeptierend ist (Algorithmus 1). Dazu raten wir zu jeder Konfiguration eine Konfigurationenfolge und benutzen die Anzahl erreichbarer Konfigurationen, um zu überprüfen, ob wir immer richtig geraten haben.

□

1.17. Bemerkung: Es gilt: $\mathcal{L}_0$ = HOM(1DFA(2)).

Beweis: Sei $L \in \mathcal{L}_0$ mit TM M und $L = L(M)$.

Algorithmus 2 Berechung von N_{t+1}

Eingabe: t, N_t: Anzahl erreichbarer Konfigurationen mit t Schritten
Ausgabe: w: true falls $y = N_{t+1}$ gilt, y: N_t+1 wenn w=true

$y = 0$, w =true, K =erste Konfiguration
repeat
    $z = 0$, f =false, L =erste Konfiguration
    repeat
        rate Konfigurationsfolge $q_0w \to k_1 \to \cdots \to k_r$ mit $r \le t$, $|k_i| \le s(n)$
        if $k_r = L$ **then**
            $z = z + 1$ {richtig geraten}
            if $K = L$ oder $L \to K$ **then**
                f =true
        L =Nachfolger(L)
    until f oder L ist keine gültige Konfiguration mehr
    if f **then**
        $y = y + 1$
    else
        if $z < N_t$ **then**
            w =false
    K =Nachfolger(K)
until not w or K ist keine gültige Konfiguration mehr

$$\begin{aligned}\tilde{L}_{KF} := \Big\{ \overline{w}¢K_1¢\ldots¢K_r \big| r \in \mathbb{N}, K_1 = (q_0w), K_r \in \Gamma^*F\Gamma^*, \\ \forall i \in \{1,\ldots,r-1\} : K_i \in \Gamma^*Q\Gamma^*, K_i \underset{M}{\to} K_{i+}1 \Big\}\end{aligned}$$

Dann gilt: $\tilde{L}_{KF} \in$ 1DFA(2) und mit h wie im Beweis zu Satz 1.9 folgt durch $L = h(\tilde{L}_{KF})$ die Behauptung. □

2 Kontextfreie Sprachen

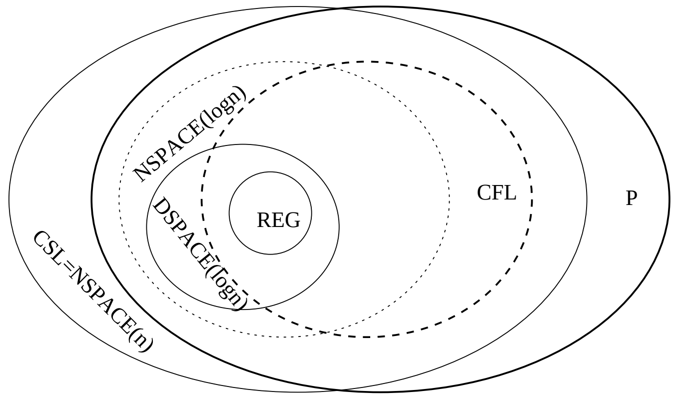
2.1. Definition: Sei $G = (V\Sigma, P, S)$ eine Grammatik.

(i) G heißt *reduziert* $:\Leftrightarrow\ \forall A \in V : \exists \alpha\beta \in (\Sigma \cup V)^*, \gamma \in \Sigma^* : S \overset{*}{\to} \alpha A \beta \wedge A \overset{*}{\to} \gamma$ *reduziert*

(ii) G ist in *Chomsky-Normalform*, falls für alle Regeln $(A \to \beta) \in P$ gilt: $A \in V, \exists B, C \in V : \beta = BC \vee \beta \in \Sigma$. *Chomsky-Normalform*

2.2. Satz: Zu jeder kontextfreien Sprache $L \in$ CFL existiert eine reduzierte Grammatik G in Chomsky-Normalform mit $L(G) = L \setminus \{\varepsilon\}$.

Abbildung 1 Übersicht über die Sprachklassen



Beweis: s. [HU90], Seite 99f. oder [Weg99], Seite 151ff und 161 □

2.3. Folgerung: Für jede kontextfreie Grammatik G ist entscheidbar, ob $L(G) = \emptyset$ gilt.

2.4. Satz: Folgende Probleme sind nicht entscheidbar:

(i) G beschränkt, $L(G) = \emptyset$?

(ii) G_1, G_2 kontextfrei, $L(G_1 \cap G_2) = \emptyset$?

(iii) $G = (V, \Sigma, P, S)$ kontextfrei, $L(G) = \Sigma^*$?

(iv) M 2DFA(2), $L(M) = \emptyset$?

Beweis: Beh.: Für eine allgemeine Grammatik G ist $L(G) = \emptyset$ unentscheidbar

Beweis: Reduktion auf das Halteproblem (Übungsaufgabe)

(i) **Annahme:** $L(G) = \emptyset$ entscheidbar. Sei $L \in \mathcal{L}_0$. Nach Satz 1.9 existieren $L_1 \in \mathcal{L}_1$ und h mit $h(L_1) = L$. Sei G die beschränkte Grammatik mit $L(G) = L_1$. Dann gilt: $L = \emptyset \Leftrightarrow L_1 = \emptyset \Leftrightarrow L(G) = \emptyset$. Damit wäre L_0 entscheidbar, dies ist ein Widerspruch.

(ii) **Annahme:** Für G_1, G_2 kontextfrei ist $L(G_1) \cap L(G_2) = \emptyset$ entscheidbar. Sei $L \in \mathcal{L}_0$. Nach Satz 1.9 existieren $L_1, L_2 \in \mathcal{L}_2 : h(L_1 \cap L_2) = L$. Dann gilt: $L = \emptyset \Leftrightarrow h(L_1 \cap L_2) = \emptyset \Leftrightarrow L_1 \cap L_2 = \emptyset$. Damit wäre L_0 entscheidbar, dies ist ein Widerspruch.

(iii) s. [Weg99], Seite 170

□

2.5. Definition (Parikh-Abbildung): Sei $\Sigma = \{a_1, \ldots, a_r\}$ mit $r = |\Sigma|$, sei Anz: $\Sigma \times \Sigma^* \to \mathbb{N}$ die Abbildung, die jedem Buchstaben-Wort-Paar die Anzahl der Vorkommen des Buchstabens in dem Wort zuordnet.

Die *Parikh-Abbildung* ist wie folgt definiert: *Parikh-Abbildung*

$$\Psi : \Sigma^* \to \mathbb{N}_0^{|\Sigma|} : u \mapsto (\mathrm{Anz}(a_1, u), \ldots, \mathrm{Anz}(a_r, u))$$

2.6. Definition: Sei $S \subseteq \mathbb{N}^r$ eine Menge.

S heißt *linear* $:\Leftrightarrow\ \exists u_0, \ldots, u_p \in \mathbb{N}^p : S = \left\{ u_0 + \sum_{i=1}^{p} x_i u_i \middle| x_i \in \mathbb{N}_0 \right\}$ *linear*

S heißt *semilinear* $:\Leftrightarrow \exists n \in \mathbb{N}, S_1, \ldots, S_n$ linear, $S = \bigcup_{i=1}^{n} S_i$ *semilinear*

2.7. Satz (Satz von Parikh): Sei $L \in$ CFL eine kontextfreie Sprache.

Dann ist $\Psi(L) := \{\Psi(u) | u \in L\}$ semilinear.

Beweis: Sei $L \in \mathcal{L}_2$ und G eine reduzierte kontextsensitive Grammatik mit $L = L(G)$. G habe keine ε-Regeln. Für alle $U \subseteq V$ sei $L_U := \{v \in L \big| \exists$ Ableitung $S \xrightarrow{*} v$ in der nur Variablen aus U auftreten$\}$. Dann gilt

$$L = \bigcup_{U \subseteq V, S \in U} L_U$$

Wir zeigen im folgenden, dass $\forall U \subseteq V : L_U$ ist semilinear gilt. Im folgenden sei $U \subseteq V$ mit $S \in U$ fest.

Zuerst definieren wir zwei Arten von guten Ableitungen, diese spielen bei der Linearität die Rolle von Start- und Richtungsvektoren.

(i) Eine Ableitung $S \xrightarrow{*} v$ mit $v \in \Sigma^*$ heißt *gut*, falls für den zugehörige Ableitungsbaum gilt:

  (1) Wurzel ist mit S markiert, als Markierungen der inneren Knoten treten alle Variablen aus U auf, und

  (2) Auf jedem von der Wurzel startenden Weg tritt jede Variable höchstens $r = |U|$-mal auf.

  $\Rightarrow$ Ableitungsbäume von guten Ableitungen haben Tiefe maximal r^2. Sei $H = \{v \in \Sigma^* \big| \exists S \xrightarrow{*} v \text{ gut}\}$. Dann gilt $|H| < \infty$.

(ii) Eine Ableitung $A \xrightarrow{*} uAv$ mit $u, v \in \Sigma^*$, $A \in U$ heißt *gut*, falls gilt:

(1) Im Ableitungsbaum treten nur Variablen aus U auf, und

(2) auf jedem an der Wurzel startenden Weg tritt jede Variable höchstens zweimal auf.

Sei $H_A = \{uv | \exists A \xrightarrow{*} uAv \text{ gut mit } uv \neq \varepsilon\}$. Dann gilt $\forall A \in U : |H_A| < \infty$.

Seien $S_1 := \Psi(H) =: \{a_1, \ldots, a_\lambda\}$, $S_2 = \Psi(\bigcup_{A \in U} H_A) =: \{v_1, \ldots, v_r\}$ und

$$S := \bigcup_{i=1}^{\lambda} \left\{ a_i + \sum_{i=1}^{r} x_i v_i \middle| x_i \in \mathbb{N}_0 \right\}$$

Beh.: $\Psi(L_U) = S$

Beweis: „$\subseteq$“: Sei $s \in S \Rightarrow s = u + \sum_{i=1}^{r} x_i v_i$ mit $u \in S_1$, $v_i \in S_2$, $x_i \in \mathbb{N}_0$. Beweis durch Induktion nach $x = \sum_{i=1}^{r} x_i$.

IA: $x = 0 \Rightarrow \forall i \in \{1, \ldots, r\} : x_i = 0 \Rightarrow s = u \in S_1 = \Psi(H) \subseteq \Psi(L_U)$

IS: Die Aussage gelte für $x \geq 0$. Sei $\sum_{i=1}^{r} x_i = x + 1 \Rightarrow \exists j : x_j > 0$. Sei $s' = u + \sum_{i=1}^{r} y_i$ mit $y_i = \begin{cases} x_i & i \neq j \\ x_i - 1 & i = j \end{cases}$

Dann gilt $x = \sum_{i=1}^{r} y_i$ und somit nach IV $s' \in \Psi(L_U) \Rightarrow \exists w \in L_U : s' = \Psi(w)$.

Es gibt $A \in U : v_j \in \Psi(H_A)$ (nach Definition S_2). Da in L_U nur Wörter sind, für die es eine Ableitung gibt, die alle Variablen aus U benutzt, existiert eine Ableitung für w mit $S \xrightarrow{*} w_1 A w_3$, $A \xrightarrow{*} w_2$ und $w = w_1 w_2 w_3$. Weiter existiert $w' \in H_A$ mit $v_j = \Psi(w')$, d.h. $A \xrightarrow{*} w_1' A w_2'$ mit $w' = w_1' w_2'$.

Zusammen folgt $S \xrightarrow{*} w_1 A w_3 \xrightarrow{*} w_1 w_1' A w_2' w_3 \xrightarrow{*} w_1 w_1' w_2 w_2' w_3 \in L_U$ und $\Psi(w_1 w_1' w_2 w_2' w_3) = \Psi(w_1 w_2 w_3) + \Psi(w_1' w_2') = \Psi(w) + \Psi(w') = s' + v_j = s \Rightarrow s \in \Psi(L_U)$.

„$\supseteq$“: Sei $w \in L_U$. Beweis von $\Psi(w) \in S$ durch Induktion nach $n = |w|$:

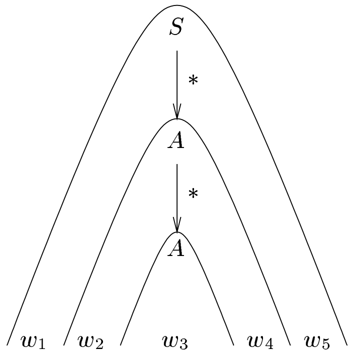
IA: $n = |w| = 1 \Rightarrow w \in H \Rightarrow \Psi(w) \in S_1 \subseteq S$

IS: $n \rightsquigarrow n + 1$: $|w| = n + 1$, $\exists$ Ableitung $S \xrightarrow{*} w$ in der genau Variablen aus U auftreten.

Falls diese Ableitung gut ist, gilt $w \in H \Rightarrow \Psi(w) \in S_1 \subseteq S$.

Sei diese Ableitung nicht gut, d.h. $\exists$ Variable A, die auf einem Weg des Ableitungsbaumes mindestens $(r + 1)$-mal auftritt.

Idee: Wir suchen nun eine Ableitung der folgenden Form:



(i) in $S \xrightarrow{*} w_1Aw_5 \xrightarrow{*} w_1w_3w_5$ treten alle Variablen aus U auf

(ii) $w = w_1w_2w_3w_4w_5$, $|w_2w_4| > 0$

(iii) $A \xrightarrow{*} w_2Aw_4$ ist gut

Dann gilt: $\Psi(w) = \Psi(w_1w_3w_5) + \Psi(w_2w_4)$ mit $\Psi(w_2w_4) \in S_2$, $\Psi(w_1w_3w_5) \in S$ nach IV

Seien $\alpha_0, \ldots, \alpha_r$ Knoten mit Variablen A auf einem Weg von der Wurzel zu einem Blatt. Sei τ_i der maximale Teilbaum des Ableitungsbaumes, der α_i enthält, aber nicht die Kinder von α_i. Seien K_i die Menge aller Variablen, die in Baum τ_i auftreten.
Dann gilt: $\forall i \in \{0, \ldots, r-1\} : K_i \subseteq K_{i+1} \subseteq U$ und $|K_0| \geq 1$
$\Rightarrow \exists j : K_j = K_{j+1}$
$\Rightarrow$ Mit den Bezeichnungen der Skizze (das obere A steht an Knoten α_j, das untere A an Knoten α_{j+1}) gilt: In der Ableitung $S \xrightarrow{*} w_1Aw_5 \xrightarrow{*} w_1w_3w_5$ treten genau die Variablen aus S auf, d.h. $w_1w_3w_5 \in L_U$ und damit $\Psi(w_1w_3w_5) \in S$ nach IV.
Falls $A \xrightarrow{*} w_2Aw_4$ gut ist, gilt $w_2w_4 \in H_A$ und damit $\exists l : \Psi(w_2w_4) = v_l \Rightarrow \Psi(w) = \Psi(w_1w_3w_5) + v_j \in S$ und es folgt die Behauptung für $n+1$.
Sei $A \xrightarrow{*} w_2Aw_4$ nicht gut $\Rightarrow \exists$ Variable B, die auf einem Weg mindestens dreimal auftritt.
Betrachte nun den Ableitungsbaum mit den Knoten β_i, an denen die B stehen und iteriere diese Analyse mit β_i statt α_i. Da die untersuchten Teilbäume bei jeder Iteration echt kleiner werden, existiert eine Zerlegung mit einer guten Ableitung.

$\square$

2.8. Bemerkung: Die Umkehrung des Satzes von Parikh gilt i.a. nicht.

Beweis: $L = \{a^nb^nc^n\}$ ist nicht kontextfrei (Beweis z.B. mit Pumping-Lemma). Aber $\Psi(L) = \{(n,n,n) \mid n \in \mathbb{N}\} = \{(1,1,1) + x(1,1,1) \mid x \in \mathbb{N}_0\}$ ist linear und damit auch semilinear. $\square$

2.9. Folgerung: Sei $L \in \mathcal{L}_2$ mit $L \subseteq \{a\}^*$. Dann gilt $L \in \mathcal{L}_3$.

Beweis: Sei $L \subseteq \{a\}^*$. Dann gilt $\Psi(L) \sim L$. Sei $\Psi(L) = \{u_0 + \sum_{i+1}^{r} x_i u_i | x_i \in \mathbb{N}_0\}$. Die Frage „Gibt es für gegebenes $z \in \mathbb{N}$ x_i mit $z = u_0 + \sum_{i+1}^{r} x_i u_i$?" kann von einem endlichen Automaten entschieden werden ($u_0, \ldots, u_r$ werden im endlichen Zustand gespeichert). □

2.10. Satz: Sei $L \subseteq \Sigma^*$. Dann gilt:

(i) $\exists G$ kontextfreie Grammatik mit $L = L(G) \Leftrightarrow \exists$ NPDA M mit $L(M) = L$ $\Leftrightarrow \exists$ NPDA M mit $N^*(M) = L$

(ii) Zu jedem det. KA M gibt es einen det. KA M' mit $N(M) = L(M')$

Beweis: (i) (1) Sei $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ gegeben. Konstruktion von einem PDA $M' = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', \{t\})$ mit $L(M') = N(M)$. Sei $Q' = Q \cup \{t, q_0'\}$, $\Gamma' = \Gamma \cup \{Z_0'\}$

$$\delta'(q,a,Z) = \begin{cases} \delta(q,a,Z) & \text{falls } Z \neq Z_0 \\ & \text{oder } \delta(q,a,Z) \cap Q \times \varepsilon = \emptyset \\ \delta(q,a,Z) \cup (t, Z_0) & \text{sonst} \end{cases}$$

(2) Sei $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ gegeben. Konstruktion von einem PDA $M' = (Q', \Sigma, \Gamma', \delta', q_0', Z_0, \emptyset)$ mit $N(M') = L(M)$. Immer wenn M in einen akzeptierenden Zustand gelangt, rät M', ob er den Keller leert.
$Q' = Q \cup \{t\}$,

$$\delta'(q,a,Z) = \begin{cases} \delta(q,a,Z) & \text{falls } \delta(q,a,Z) \cap F \times \Gamma^* = \emptyset \\ \delta(q,a,Z) \cup \{(t,Z)\} & \text{sonst} \end{cases}$$

$\forall Z \in \Gamma : \delta'(t, \varepsilon, Z) = (t, \varepsilon)$

(3) Sei M gegeben. Konstruktion einer kontextfreien Grammatik G mit $L(G) = N(M)$.
Idee: Um mit leerem Keller zu akzeptieren, ist wichtig, welcher Zustand p erreicht wird und welche Eingabe x gelesen wird, um ein bestimmtes Symbol vom Keller zu löschen. Es soll folgende Äquivalenz gelten:

$$(q, Z, p) \xrightarrow[G]{*} x \Leftrightarrow (q, x, Z) \xrightarrow[M]{*} (p, \varepsilon, \varepsilon) \qquad (1)$$

Konstruktion der Produktionen von G:

*Akzeptieren mit leerem Keller

(a) $\forall q \in F : S \to (q_0, Z_0, q)$

(b) $\forall q \in Q, a \in \Sigma \cup \{\varepsilon\}, Z \in \Gamma : \forall (q', \varepsilon) \in \delta(q, a, Z) : (q, Z, q') \to a$

(c) $\forall q \in Q, a \in \Sigma \cup \{\varepsilon\}, Z \in \Gamma : \forall (q', B_1 \ldots B_r) \in \delta(q, a, Z) : \forall q_1, \ldots, q_r \in Q : q_r(q, Z, q_r) \to a(q', B_1, q_1)(q_1, B_2, q_2) \ldots (q_{r-1}, B_r, q_r)$

Beh.: Diese Produktionen implizieren die Äquivalenz (1).

Beweis: „$\Leftarrow$“: Es gelte $(q, x, Z) \xrightarrow[M]{k} (p, \varepsilon, \varepsilon)$. Beweis von $(q, Z, p) \xrightarrow[G]{*} x$ mit Induktion nach k:

IA: $k = 1 \Rightarrow x \in \Sigma \cup \{\varepsilon\} \Rightarrow (p, \varepsilon) \in \delta(q, x, Z) \Rightarrow (q, Z, p) \xrightarrow[G]{} x$.

IS: $k - 1 \rightsquigarrow k$, $k > 1$: Betrachte den ersten Berechnungsschritt von M: $(q_1, B_1 \ldots B_r) \in \delta(q, a, Z)$ mit $a \in \Sigma \cup \{\varepsilon\}$ und $\exists x_1, \ldots, x_r$ und $q_i \in Q : x = a x_1 \ldots x_r, \forall i \in \{1, \ldots, r+1\} : (q_i, x_i, B_i) \xrightarrow[M]{*} (q_{i+1}, \varepsilon)$ mit $q_{r+1} = p$, da M die im ersten Schritt auf den Keller geschriebenen Symbole irgendwann löschen muss.
Nach IV gilt $(q_i, B_i, q_{i+1}) \xrightarrow[G]{*} x_i$

$\Rightarrow (q, Z, p) \xrightarrow[G]{} a(q_1, B_1, q_2)(q_2, B_2, q_3) \ldots (q_r, B_r, p) \xrightarrow[G]{*} a x_1 \ldots x_r = x$.

(4) Sei G kontextfreie Grammatik. Beweis der Existenz eines PDAs mit $L(M) = L(G)$ s. [HU90], Seite 123f.

(ii) analog dem nichtdeterministischen Fall mit

$$\delta'(q, a, Z) = \begin{cases} \delta(q, a, Z) & \text{falls } Z \neq Z_0 \text{oder } \delta(q, a, Z) \cap Q \times \varepsilon = \emptyset \\ \delta(t, Z_0) & \text{sonst} \end{cases}$$

□

2.11. Definition:

(i) Eine Grammatik heißt *eindeutig*, falls es für jedes $w \in L(G)$ genau eine Linksableitung gibt. Sonst heißt G *mehrdeutig*. *eindeutig* *mehrdeutig*

(ii) Eine Sprache L heißt *eindeutig*, falls es eine eindeutige Grammatik G mit $L(G) = L$ gibt, sonst heißt L *mehrdeutig*.

2.12. Satz:

(i) DCFL ist abgeschlossen unter Komplementbildung.

(ii) DCFL ist abgeschlossen unter inversen Homomorphismen und unter Schnittbildung mit regulären Sprachen.

(iii) DCFL ist nicht abgeschlossen unter Homomorphismus, Vereinigung, Schnitt, Konkatenation und Kleenschem Abschluss.

Beweis: (i) (vergl. [HU90], Seite 257ff) Im Gegensatz zu endlichen Automaten kann man bei Kellerautomaten nicht einfach die Menge der akzeptierenden Zustände ändern, um das Komplement einer Sprache zu akzeptieren, da es mehrere Gründe gibt, warum ein Kellerautomat ein Wort nicht akzeptiert. Ein DKA M akzeptiert ein Wort w nicht, wenn einer der folgenden Fälle auftritt:

(1) M bewegt sich niemals über einen Punkt der Eingabe hinweg, weil ein Übergang nicht definiert ist oder M eine unendliche ε-Bewegung macht.

(2) M macht am Ende des Wortes eine unendliche ε-Bewegung, in der kein akzeptierender Zustand erreicht wird.

Zuerst werden wir einen DKA M konstruieren, der immer die gesamte Eingabe liest und keine endlosen ε-Bewegungen macht. Anschließend modifizieren wir M so, daß die Wahl zwischen ε-Bewegung und Lesen eines Zeichens nur vom Zustand und nicht mehr vom Kellerinhalt abhängt.

Definition: M ist in *Normalform*, falls $\forall a \in \Sigma \cup \{\varepsilon\}, q \in Q, Z \in \Gamma$ mit $\delta(q, a, Z) = (q', \lambda)$ gilt:

(1) $|\lambda| \leq 2$

(2) $|\lambda| = 1 \Rightarrow \lambda = Z$

(Dies bedeutet, daß M entweder ein Kellersymbol löscht (pop), den Keller unverändert lässt, oder ein neues Zeichen auf den Keller schreibt (push).)

Beh.: $\forall$DKA M : $\exists$DKA M' in Normalform mit $L(M) = L(M')$

Beweis: Sei $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ ein DKA. Sei $M' = (Q', \Sigma, \Gamma, \delta', q_0, Z_0, F)$ mit $Q' = Q \cup \{$neue Zustände durch Definition der Regeln, s.u.$\}$.
M' simuliert M mit folgenden Regeln:

(1) $\forall q \in Q, a \in \Sigma \cup \{\varepsilon\}, Z \in \Gamma : \delta(q, a, Z) = (q', A_1 \dots A_n)$ wird simuliert durch $\delta'(q, a, Z) = (q^{n-1}, A_{n-1}A_n), \dots, \delta'(q^2, \varepsilon, A_2) = (q', A_1A_2)$ mit neuen Zuständen $q^2, \dots, q^{n-1}$.

(2) $\forall q \in Q, a \in \Sigma \cup \{\varepsilon\}, Z \in \Gamma : \delta(q, a, Z) = (q', A), A \neq Z$ wird simuliert durch $\delta'(q, a, Z) = (\hat{q}, BA)$ und $\delta'(\hat{q}, \varepsilon, B) = (q', \varepsilon)$ mit neuem Zustand $\hat{q}$ und beliebigem $B \in V$.

Lemma: Sei M ein DKA. Dann existiert ein DKA M' mit $L(M) = L(M')$ und M' liest jede Eingabe komplett.

Beweis: Sei M ein DKA in Normalform. Sei $M' = (Q', \Sigma, \Gamma, \delta', q_0', Z_0', F')$ mit $Q' = Q \cup \{q_0', d, f\}$, $\Gamma' = \Gamma \cup \{Z_0'\}$, $F' = F \cup \{f\}$ mit folgenden Regeln:

(1) $\delta'(q_0', \varepsilon, Z_0') = (q_0, Z_0X_0)$ (damit Löschen des alten Kellerendesymbols erkannt werden kann)