Automatentheorie und Formale Sprachen

Vorlesungsskript

Dietmar Wätjen

Institut für Theoretische Informatik
Technische Universität Braunschweig

1999

Inhaltsverzeichnis

Kapitel 11

Gesteuerte Ersetzung

In diesem Kapitel soll vor allem darauf eingegangen werden, wie man die erzeugende Kapazität von kontextfreien Grammatiken erhöhen kann, ohne auf die Kontextfreiheit zu verzichten. Mit Hilfe einer geeigneten Steuerung der Anwendung kontextfreier Produktionen können nicht-kontextfreie Sprachen erzeugt werden.

11.1 Matrix-Grammatiken

Definition 11.1 (a) $G = (V_N, V_T, X_0, M)$ heißt *Matrix-Grammatik*, wenn folgendes gilt:

(1) V_N und V_T sind disjunkte Alphabete (*Nichtterminal- und Terminalalphabet*),

(2) $X_0 \in V_N$ (das *Anfangssymbol*),

(3) M ist eine endliche Menge von endlichen nichtleeren Folgen von geordneten Paaren (P, Q) mit $P \in V^*V_NV^*$, $Q \in V^*$ und $V = V_N \cup V_T$. Die Paare (P, Q) werden auch mit $P \to Q$ bezeichnet und heißen *Produktionen*. Die Folgen werden in der Form $m = [P_1 \to Q_1, \ldots, P_r \to Q_r]$ mit $r \in \mathbb{N}$ geschrieben. Sie werden *Matrizen* genannt.

(b) Es sei $G = (V_N, V_T, X_0, M)$ eine Matrix-Grammatik und F die Menge der in den Matrizen auftretenden Produktionen. Die Matrix-Grammatik G ist *vom Typ* i $(i = 0, 1, 2, 3)$, *monoton, linear, ε-frei, kontextfrei, kontextsensitiv*, wenn die Grammatik (V_N, V_T, X_0, F) die entsprechende Eigenschaft besitzt. □

Definition 11.2 Es sei G eine Matrix-Grammatik.

(a) Es seien $P, Q \in V^*$. Für (P, Q) ist die Relation $P \Longrightarrow_G Q$ oder kurz $P \Longrightarrow Q$ erfüllt, wenn eine Zahl $r \in \mathbb{N}$ und Wörter

$$\alpha_1, \ldots, \alpha_{r+1}, P_1, \ldots, P_r, Q_1, \ldots, Q_r, R_1, \ldots, R_r, R^1, \ldots, R^r \in V^*$$

existieren mit der folgenden Eigenschaft:

(1) $\alpha_1 = P$, $\alpha_{r+1} = Q$,

(2) $m = [P_1 \to Q_1, \ldots, P_r \to Q_r]$ ist eine Matrix von G,

(3) $\alpha_i = R_i P_i R^i$ und $\alpha_{i+1} = R_i Q_i R^i$ für alle $i = 1, \ldots, r$.

(b) $\Longrightarrow^*$ ist die reflexive transitive Hülle von $\Longrightarrow$.

(c) $L(G) = \{w \in V_T^* \mid X_0 \Longrightarrow^* w\}$ heißt die *von G erzeugte Sprache.* $\square$

Definition 11.3 (a) Mit $\mathcal{M}$ bezeichnen wir die Familie der von allen ε-freien kontextfreien Matrix-Grammatiken erzeugten Sprachen.

(b) Mit $\mathcal{M}^\varepsilon$ bezeichnen wir die Familie der von allen kontextfreien Matrix-Grammatiken erzeugten Sprachen. $\square$

Nur für kontextfreie Matrix-Grammatiken wurde hier eine Bezeichnung der entsprechenden Sprachfamilien eingeführt. Der Grund dafür ist, daß die Erzeugungskapazität von Typ-i-Grammatiken für $i \neq 2$ gleich der von Typ-i-Matrix-Grammatiken ist (siehe Satz 11.12).

Beispiel 11.1 Es sei

$$G = (\{X_0, X_1\}, \{a, b, c\}, X_0, \{[X_0 \to X_0 X_1], [X_0 \to aX_0 b, X_1 \to cX_1], [X_0 \to ab, X_1 \to c]\})$$

eine Matrix-Grammatik. Die erste Matrix muß in der Ableitung eines Terminalwortes genau einmal auftreten, und zwar zu Beginn. Würde sie wiederholt angewendet, so würden sich mehr Vorkommen von X_1 als von X_0 in einem Wort befinden, so daß schließlich nicht alle Nichtterminalsymbole entfernt werden könnten. Es gilt $L(G) = \{a^n b^n c^n \mid n \geq 1\}$. Für $n = 1$ oder $n = 2$ erhalten wir zum Beispiel die Ableitungen

$$X_0 \Longrightarrow X_0 X_1 \Longrightarrow abc \text{ bzw. } X_0 \Longrightarrow X_0 X_1 \Longrightarrow aX_0 bcX_1 \Longrightarrow aabbcc. \quad \square$$

Beispiel 11.2 Es sei

$$\begin{aligned} G \;=\; & (\{X_0, X_1\}, \{a, b\}, X_0, \{[X_0 \to X_0 X_1], [X_0 \to aX_0, X_1 \to aX_1], \\ & [X_0 \to bX_0, X_1 \to bX_1], [X_0 \to a, X_1 \to a], [X_0 \to b, X_1 \to b]\}) \end{aligned}$$

eine Matrix-Grammatik. Die erste Matrix muß in jeder Ableitung genau einmal auftreten. Dann steht X_0 für die linke und X_1 für die rechte Seite eines zu erzeugenden Wortes. Im Laufe der Anwendung einer Matrix ergeben sich für X_0 und X_1 entweder jeweils ein neues Symbol a oder jeweils ein neues Symbol b. Wegen der zu Beginn auftretenden Matrix $[X_0 \to X_0 X_1]$ ergibt sich damit $L(G) = \{ww \mid w \in \{a, b\}^+\}$. $\square$

Die Sprachen aus Beispiel 11.1 und Beispiel 11.2 werden von ε-freien kontextfreien Matrix-Grammatiken erzeugt. Sie sind jedoch nicht kontextfrei. Für die Sprache aus Beispiel 11.1 wurde dies in Beispiel 5.7 gezeigt. Andererseits kann jede Grammatik als eine Matrix-Grammatik aufgefaßt werden, indem jede einzelne Produktion als eine Matrix verstanden wird. Folglich gilt

$$\mathfrak{L}_2 \subset \mathcal{M}^\varepsilon \text{ und } \mathfrak{L}_2^{-\varepsilon} \subset \mathcal{M},$$

wobei $\mathfrak{L}_2^{-\varepsilon}$ die Familie der ε-freien kontextfreien Sprachen ist.

Außer der in Definition 11.2 angegebenen Erzeugung von Sprachen durch Matrix-Grammatiken ist eine andere Vorgehensweise denkbar. Wir nehmen an, daß auf das Wort $P = abX_1b$ die Matrix $m = [X_1 \to abX_1, X_2 \to b, X_1 \to bX_2]$ angewendet werden soll. Die Anwendung der ersten Produktion stellt kein Problem dar und liefert

$$ababX_1b.$$

Die zweite Produktion kann wegen des Fehlens von X_2 nicht im Sinn von Definition 11.2 angewendet werden. Dies wird jedoch als eine „Anwendung" betrachtet, bei der das Wort nicht verändert wird. Die dritte Produktion liefert dann

$$ababbX_2b.$$

Die Anwendung der zweiten Produktion erfolgt hier im sogenannten *Vorkommensprüfungssinn*, d.h., es wird getestet, ob die linke Seite der Produktion ein Teilwort des betrachteten Wortes ist. Ist dies der Fall, wird sie im normalen Sinn angewendet, anderenfalls wird die Ableitung mit der nächsten Produktion fortgesetzt (wie es im obigen Beispiel der Fall ist). Diese Ersetzung im Vorkommensprüfungssinn kann man für einige Produktionen zulassen, für andere dagegen nicht erlauben. Es sei F die Menge des Vorkommens aller Produktionen in allen Matrizen. Dann werde eine Teilmenge $F_1 \subset F$ ausgezeichnet, für die die Anwendung im Vorkommensprüfungssinn erlaubt sein soll. Bei der Anwendung von Produktionen aus $F - F_1$ soll dagegen stets eine echte Ersetzung stattfinden. Wir bemerken, daß es möglich ist, daß eine Produktion $P_1 \to P_2$ aus einer Matrix m in F_1 liegt, dieselbe Produktion aus einer anderen Matrix m' dagegen nicht. Diese Überlegungen werden in der folgenden Definition formalisiert.

Definition 11.4 Es sei $G = (V_N, V_T, X_0, M)$ eine Matrix-Grammatik, F die Gesamtheit der Vorkommen aller Produktionen in den Matrizen von M und $F_1 \subset F$.

(a) Die von G und F_1 abhängige binäre Relation $\Longrightarrow_{VP}$ auf V^* ist wie folgt definiert: Für $P, Q \in V^*$ gilt $P \Longrightarrow_{VP} Q$, wenn eine Zahl $r \in \mathbb{N}$ und Wörter

$$\alpha_1, \ldots, \alpha_{r+1}, P_1, \ldots, P_r, Q_1, \ldots, Q_r, R_1, \ldots, R_r, R^1, \ldots, R^r \in V^*$$

existieren mit folgenden Eigenschaften:

(1) $\alpha_1 = P$, $\alpha_{r+1} = Q$,

(2) $m = [P_1 \to Q_1, \ldots, P_r \to Q_r]$ ist eine Matrix von G,

(3) für alle $i = 1, \ldots, r$ gilt entweder

(α) $\alpha_i = R_iP_iR^i$ und $\alpha_{i+1} = R_iQ_iR^i$ oder

(β) das Vorkommen der Produktion $P_i \to Q_i$ liegt in F_1, P_i ist kein Teilwort von α_i, und es gilt $\alpha_i = \alpha_{i+1}$.

(b) $\Longrightarrow^*_{VP}$ ist die reflexive transitive Hülle von $\Longrightarrow_{VP}$.

(c) $L_{VP}(G, F_1) = \{w \in V_T^* \mid X_0 \Longrightarrow^*_{VP} w\}$ heißt die *von G mit Vorkommensprüfung für die Produktionen in F_1 erzeugte Sprache.* □

Für eine Matrixgrammatik G folgt unmittelbar aus Definition 11.2 und Definition 11.4 die Gleichung

$$L(G) = L_{VP}(G, \emptyset).$$

Definition 11.5 (a) Mit $\mathcal{M}_{VP}$ bezeichnen wir die Familie der von allen ε-freien kontextfreien Matrix-Grammatiken mit Vorkommensprüfung erzeugten Sprachen.

(b) Mit $\mathcal{M}_{VP}^{\varepsilon}$ bezeichnen wir die Familie der von allen kontextfreien Matrix-Grammatiken mit Vorkommensprüfung erzeugten Sprachen. □

Aufgrund der Definitionen folgt sofort

$$\mathcal{M} \subset \mathcal{M}^{\varepsilon} \subset \mathcal{M}_{VP}^{\varepsilon} \quad \text{und} \quad \mathcal{M} \subset \mathcal{M}_{VP} \subset \mathcal{M}_{VP}^{\varepsilon}.$$

Die Beziehung zwischen $\mathcal{M}^{\varepsilon}$ und $\mathcal{M}_{VP}$ ist nicht bekannt.

Beispiel 11.3 Es sei $G = (\{X, Y, Z, U, A\}, \{a\}, X, M)$ eine ε-freie kontextfreie Matrix- Grammatik mit

$$\begin{aligned} M = \{&m_1 = [Y \to U, A \to U, X \to ZZ], \\ &m_2 = [X \to U, Z \to Y], \\ &m_3 = [Z \to U, Y \to X], \\ &m_4 = [Y \to U, Z \to U, X \to A], \\ &m_5 = [X \to U, A \to a]\}, \end{aligned}$$

und F_1 enthalte alle Vorkommen der Produktionen $X \to U$, $Y \to U$, $Z \to U$ und $A \to U$. Das Nichtterminalzeichen U erscheint nie auf der linken Seite einer Produktion, so daß es, einmal eingeführt, nicht entfernt werden kann. Die wesentliche Produktion in jeder Matrix ist die letzte. Die anderen Produktionen dienen dazu, bei einem gerade betrachteten Wort festzustellen, ob es gewisse Nichtterminalzeichen enthält, bei deren Anwesenheit die Anwendung der entsprechenden Matrix dazu führt, daß daraus resultierende weitere Ableitungen niemals zu einem Terminalwort führen können. Im folgenden werden nur diejenigen Ableitungen unter Angabe der zugehörigen Matrizen skizziert, die kein U enthalten und somit ein Wort der Sprache $L_{VP}(G, F_1)$ liefern.

$$X \begin{cases} \overset{m_4}{\Longrightarrow} A \\ \overset{m_1}{\Longrightarrow} ZZ \overset{m_2}{\Longrightarrow} \left\{ \begin{array}{c} YZ \\ ZY \end{array} \right\} \overset{m_2}{\Longrightarrow} YY \overset{m_3}{\Longrightarrow} \left\{ \begin{array}{c} XY \\ YX \end{array} \right\} \overset{m_3}{\Longrightarrow} XX \begin{cases} \overset{m_1}{\Longrightarrow} \left\{ \begin{array}{c} ZZX \\ XZZ \end{array} \right\} \overset{m_1}{\Longrightarrow} ZZZZ \\ \overset{m_4}{\Longrightarrow} \left\{ \begin{array}{c} AX \\ XA \end{array} \right\} \overset{m_4}{\Longrightarrow} AA. \end{cases} \end{cases}$$

Mit XX beginnt eine weitere Iteration der möglichen Ableitungen, jedoch wird dabei jede Matrix doppelt so oft angewendet wie zu Beginn. Allgemein liefert X^{2^n} für $n \geq 0$ entweder das Wort A^{2^n} oder (in der gegebenen Reihenfolge) die Wörter $Z^{2^{n+1}}$, $Y^{2^{n+1}}$, $X^{2^{n+1}}$. Wegen m_5 folgt daher

$$L_{VP}(G, F_1) = \{a^{2^n} \mid n \geq 0\} \in \mathcal{M}_{VP}. \quad \square$$

11.2 Zeitvariable Grammatiken

Eine zeitvariable Grammatik ist dadurch gekennzeichnet, daß bei einem Ableitungsschritt nicht die ganze Produktionenmenge zur Verfügung steht, sondern nur jeweils eine Teilmenge davon. Es hängt von der Zeit, also von der Anzahl der Schritte seit Beginn der Ableitung ab, ob eine gewisse Produktion benutzt werden darf oder nicht.

Definition 11.6 (a) (G, φ) heißt *zeitvariable Grammatik vom Typ* i, $i = 0, 1, 2, 3$, wenn $G = (V_N, V_T, X_0, F)$ eine Typ-i-Grammatik und $\varphi : \mathbb{N} \to \mathfrak{P}(F)$ eine Abbildung ist.

(b) Für $P, Q \in V^*$ und $j_1, j_2 \in \mathbb{N}$ gilt die Relation $(P, j_1) \Longrightarrow (Q, j_2)$, wenn
   (1) $j_2 = j_1 + 1$ erfüllt ist und
   (2) Wörter $R_1, R_2, P', Q' \in V^*$ existieren mit $P = R_1 P' R_2$, $Q = R_1 Q' R_2$ und $(P', Q') \in \varphi(j_1)$.

(c) $\Longrightarrow^*$ ist die reflexive transitive Hülle von $\Longrightarrow$.

(d) $L(G, \varphi) = \{w \in V_T^* \mid (X_0, 1) \Longrightarrow^* (w, j),\ j \in \mathbb{N}\}$ heißt die *von* (G, φ) *erzeugte Sprache*. Eine Sprache L heißt *zeitvariabel vom Typ* i, $i = 0, 1, 2, 3$, wenn eine zeitvariable Grammatik (G, φ) vom Typ i mit $L = L(G, \varphi)$ existiert. □

Eine zeitvariable Sprache ist nicht unbedingt rekursiv aufzählbar, da es keinen Algorithmus geben muß, um die Werte von φ zu bestimmen. Schon zeitvariable Grammatiken vom Typ 3 erzeugen Sprachen, die nicht rekursiv-aufzählbar sind.

Satz 11.1 Jede Sprache ist zeitvariabel vom Typ 3.

Beweis: Wir betrachten das Alphabet $V_T = \{a_1, \ldots, a_n\}$ und eine Sprache $L \subset V_T^*$. Bei Wahl von $\varphi(\mathbb{N}) = \{\emptyset\}$ erzeugt jede zeitvariable Grammatik die Sprache $L = \emptyset$. Die leere Menge ist also zeitvariabel vom Typ 3. Im weiteren sei $L \neq \emptyset$. Da V_T^* abzählbar ist, können wir eine unendliche Folge $w_1, w_2, \ldots$ betrachten, die alle Wörter von L, ggf. mit Wiederholungen, enthält. Diese Folge ist im allgemeinen nicht effektiv konstruierbar. Weiter gelte $u(i) = |w_i|$ für alle $i \in \mathbb{N}$. Wir betrachten die rechtslineare Grammatik $G = (\{X_0, X\}, V_T, X_0, F)$ mit

$$F = \{X_0 \to X_0, X_0 \to X, X \to \varepsilon\} \cup \{X \to a_i X \mid 1 \leq i \leq n\}.$$

Dazu geben wir mit Hilfe der unenendlichen Folge der Wörter w_i eine Abbildung $\varphi : \mathbb{N} \to \mathfrak{P}(F)$ in folgender Weise an.

Für $i \in \mathbb{N}$ bestehe $\varphi(i)$ aus genau zwei Produktionen, und zwar aus der Produktion $X_0 \to X_0$ und einer weiteren Produktion, die wie folgt bestimmt wird. Es sei $w_k = c_{k,1} \ldots c_{k,u(k)}$ für $k \in \mathbb{N}$. Wir setzen $s_0 = 0$ und $s_k = |w_1 \ldots w_k| + 2$. Für alle $k \in \mathbb{N}$ wird

$$\begin{aligned} (X_0, X) &\in \varphi(s_{k-1} + 1), \\ (X, c_{k,j} X) &\in \varphi(s_{k-1} + j + 1) \quad \text{für alle } 1 \leq j \leq u(k), \\ (X, \varepsilon) &\in \varphi(s_{k-1} + u(k) + 2) \end{aligned}$$

gesetzt. Damit ist $\varphi(i)$ für alle $i \in \mathbb{N}$ bestimmt.

Die Produktion $X_0 \to X_0$ läßt sich beliebig oft anwenden, die Anwendung von $X_0 \to X$ ist jedoch für irgendein s_{k-1} nur im $(s_{k-1}+1)$-ten Schritt möglich. In diesem Fall wird das Wort $w_k = c_{k,1} \dots c_{k,u(k)} \in L$ erzeugt. Speziell für $s_0 = 0$ ergibt sich w_1. Andere Ableitungen, die zu Wörtern über V_T führen, sind nicht möglich. Somit folgt $L = L(G, \varphi)$. $\square$

Da φ eine beliebige Funktion $\varphi : \mathbb{N} \to \mathfrak{P}(F)$ ist, muß sie nicht berechenbar sein. Man kann natürlich die Berechenbarkeit fordern, indem wir annehmen, daß φ durch eine Turingmaschine berechnet wird. Im folgenden werden wir die viel strengere Forderung der Periodizität der Abbildung φ stellen.

Definition 11.7 (a) Eine zeitvariable Grammatik (G, φ) heißt *periodisch zeitvariabel*, wenn φ *periodisch* ist, d.h. ein $k \in \mathbb{N}$ existiert, so daß für alle $j \in \mathbb{N}$

$$\varphi(j + k) = \varphi(j)$$

gilt.

(b) Mit $\mathcal{T}^{\varepsilon}$ (bzw. $\mathcal{T}$) bezeichnen wir die Familie der Sprachen $L(G, \varphi)$, die durch periodisch zeitvariable kontextfreie (bzw. ε-freie kontextfreie) Grammatiken erzeugt werden. Die Sprachen in $\mathcal{T}^{\varepsilon}$ (bzw. $\mathcal{T}$) heißen *periodisch zeitvariable (ε-freie) kontextfreie Sprachen.* $\square$

Beispiel 11.4 Eine periodisch zeitvariable ε-freie kontextfreie Grammatik $(G, \varphi) = (\{X_0, X, Y, Z, U\}, \{a, b, c\}, X_0, F)$ mit der Periode $k = 3$ werde durch

$$\begin{aligned}
\varphi(1) &= \{X_0 \to XYZ, Z \to cZ, Z \to c\}, \\
\varphi(2) &= \{X \to aX, X \to a, U \to b\}, \\
\varphi(3) &= \{Y \to bY, Y \to U\}
\end{aligned}$$

gegeben. Gestartet wird mit $X_0 \to XYZ$. Man sieht, daß X, Y bzw. Z die Terminalsymbole a, b bzw. c liefern. Um ein Terminalwort zu erzeugen, müssen die Produktionen

$$X \to a,\ Y \to U,\ Z \to c \text{ und } U \to b$$

erst zum Schluß, und dann in genau dieser Reihenfolge, angewendet werden. Es ergibt sich die Sprache

$$L(G, \varphi) = \{a^n b^n c^n \mid n \geq 1\}.$$

U wird dabei eingeführt, um z.B. Wörter $a^{n+1} b^n c^n$ $(n \in \mathbb{N})$ zu verhindern. $\square$

Beispiel 11.5 Wir betrachten die periodisch zeitvariable kontextfreie Grammatik $(G, \varphi) = (\{X_0, X_1, X_2, Y_1, Y_2\}, \{a, b\}, X_0, F)$ mit der Periode $k = 4$ und

$$\begin{aligned}
\varphi(1) &= \{X_0 \to X_1 Y_1, X_1 \to X_1, Y_2 \to Y_2\}, \\
\varphi(2) &= \{X_1 \to aX_1, X_1 \to bX_2, X_1 \to \varepsilon, X_2 \to aX_1, X_2 \to bX_2, X_2 \to \varepsilon\}, \\
\varphi(3) &= \{Y_1 \to aY_1, Y_1 \to bY_2, Y_1 \to \varepsilon, Y_2 \to aY_1, Y_2 \to bY_2, Y_2 \to \varepsilon\}, \\
\varphi(4) &= \{X_2 \to X_2, Y_1 \to Y_1\}.
\end{aligned}$$

Gestartet wird mit $X_0 \to X_1 Y_1$. X_1 und X_2 gehören zur vorderen Hälfte eines erzeugten Wortes, Y_1 und Y_2 zur hinteren. Wird durch eine Produktion das Symbol a (bzw. b) erzeugt, so wird gleichzeitig der Index 1 (bzw. 2) dem zusammen mit a (bzw. b) erzeugten Nichtendsymbol zugeordnet. Wenn in einer Ableitung für ein $j \geq 0$ nach dem $(3+4j)$-ten Schritt, d.h. nach Anwendung einer Produktion aus $\varphi(3)$, die Indizes 1 und 2 vorkommen, so bedeutet dies, daß in den beiden Schritten zuvor genau einmal das Symbol a und genau einmal das Symbol b erzeugt wurde. Dann kommen jedoch genau die Nichtterminalsymbole X_1, Y_2 bzw. X_2, Y_1 in einem Wort vor, so daß man nicht „erfolgreich" weiterkommt bei einer Anwendung einer Produktion aus $\varphi(4)$ bzw. aus $\varphi(1)$. Um ein Terminalwort zu erzeugen, müssen folglich in der vorderen und in der hinteren Hälfte des Wortes durch $\varphi(2)$ und $\varphi(3)$ dieselben Terminalsymbole erzeugt werden. Somit gilt

$$L(G, \varphi) = \{ww \mid w \in \{a, b\}^*\}. \quad \square$$

Die Beispiele 11.4 und 11.5 liefern sofort die Gültigkeit von

$$\mathfrak{L}_2^{-\varepsilon} \subsetneq \mathcal{T} \text{ und } \mathfrak{L}_2 \subsetneq \mathcal{T}^{\varepsilon}.$$

Definition 11.8 Es sei (G, φ) eine zeitvariable Grammatik mit $G = (V_N, V_T, X_0, F)$ und $F_1 \subset F$.

(a) Für $P, Q \in V^*$ und $j_1, j_2 \in \mathbb{N}$ gilt die Relation $(P, j_1) \Longrightarrow_{VP} (Q, j_2)$, wenn
   (1) entweder $(P, j_1) \Longrightarrow (Q, j_2)$ erfüllt ist
   (2) oder $j_2 = j_1 + 1$, $P = Q$ gilt und ein Paar $(P', Q') \in F_1 \cap \varphi(j_1)$ existiert, wobei P' kein Teilwort von P ist.

(b) $\Longrightarrow_{VP}^*$ ist die reflexive transitive Hülle von $\Longrightarrow_{VP}$.

(c) $L_{VP}(G, \varphi, F_1) = \{w \in V_T^* \mid (X_0, 1) \Longrightarrow_{VP}^* (w, j),\ j \in \mathbb{N}\}$ heißt die *von* (G, φ) *mit Vorkommensprüfung für Produktionen in* F_1 *erzeugte Sprache.* Mit $\mathcal{T}_{VP}^{\varepsilon}$ (bzw. $\mathcal{T}_{VP}$) bezeichnen wir die Familie der Sprachen $L_{VP}(G, \varphi, F_1)$, die durch periodisch zeitvariable (ε-freie) kontextfreie Grammatiken (G, φ) mit Vorkommensprüfung für Produktionen in F_1 erzeugt werden. Die Sprachen in $\mathcal{T}_{VP}^{\varepsilon}$ (bzw. $\mathcal{T}_{VP}$) heißen *periodisch zeitvariable kontextfreie* bzw. *periodisch zeitvariable ε-freie kontextfreie Sprachen mit Vorkommensprüfung.* $\square$

Aus der Definition ergibt sich

$$\mathcal{T} \subset \mathcal{T}^{\varepsilon} \subset \mathcal{T}_{VP}^{\varepsilon} \quad \text{und} \quad \mathcal{T} \subset \mathcal{T}_{VP} \subset \mathcal{T}_{VP}^{\varepsilon}.$$

Da sich zeigen läßt (siehe Satz 11.13), daß jede Matrix-Grammatik in eine äquivalente Matrix-Grammatik umgeformt werden kann, deren Matrizen genau zwei Produktionen enthalten, bedeuten die Voraussetzungen des folgenden Satzes keine Einschränkung.

Satz 11.2 Es sei L eine Sprache, die von einer kontextfreien Matrix-Grammatik mit genau zwei Produktionen in jeder Matrix erzeugt wird. Dann gilt $L \in \mathcal{T}^{\varepsilon}$.

Beweis: Die Sprache L werde von der Matrix-Grammatik $G = (V_N, V_T, X_0, M)$ mit

$$M = \{[X_1^i \to P_1^i, X_2^i \to P_2^i] \mid X_1^i, X_2^i \in V_N, P_1^i, P_2^i \in V^*, i = 1, \ldots, u\}$$

erzeugt. Wir konstruieren eine periodisch zeitvariable kontextfreie Grammatik (G', φ) mit

$$L = L(G) = L(G', \varphi).$$

Es werde

$$V_N' = V_N \cup \{Y_j^i \mid 1 \leq i \leq u \text{ und } j = 1, 2\}, \quad V_T' = V_T \text{ und } X_0' = X_0$$

gesetzt, die Produktionenmenge sei durch

$$F' = \{X_j^i \to P_j^i Y_j^i, Y_j^i \to Y_j^i, Y_j^i \to \varepsilon \mid 1 \leq i \leq u \text{ und } j = 1, 2\}$$

(Y_j^i markiert das Vorkommen von $X_j^i \to P_j^i$ in der i-ten Matrix) und die periodische Funktion φ mit der Periode $4 + u$ durch das folgende Schema gegeben:

$$\begin{array}{ll}
\varphi(j) = \{X_j^i \to P_j^i Y_j^i \mid 1 \leq i \leq u\}, & \text{für } j = 1, 2, \\
\varphi(v) = \{Y_1^i \to Y_1^i \mid 1 \leq i \leq u,\ i \neq h\} \cup \{Y_2^h \to Y_2^h\} & \text{für } v = 2 + h,\ h = 1, \ldots, u, \\
\varphi(v) = \{Y_j^i \to \varepsilon \mid 1 \leq i \leq u\} & \text{für } v = 2 + u + j,\ j = 1, 2.
\end{array}$$

Wir stellen zunächst fest, daß jede Produktionenmenge $\varphi(v)$, $v = 2 + h$, $h = 1, \ldots, u$, für jedes $i = 1, \ldots, u$ genau eine der Produktionen (Y_1^i, Y_1^i) oder (Y_2^i, Y_2^i) enthält. Wird ein Ableitungsschritt von G gemäß der i-ten Matrix simuliert, indem zunächst aus $\varphi(1)$ die Produktion $X_1^i \to P_1^i Y_1^i$ und dann aus $\varphi(2)$ die Produktion $X_2^i \to P_2^i Y_2^i$ angewendet wird, so gehen daher die nächsten u Ableitungsschritte ohne Änderung des erzeugten Wortes durch, bis dann durch $\varphi(2 + u + 1)$ und $\varphi(2 + u + 2)$ die Matrixmarkierer Y_1^i und Y_2^i entfernt werden. Offenbar gilt damit $L(G) \subset L(G', \varphi)$. Wird dagegen in $\varphi(1)$ die Produktion $X_1^i \to P_1^i Y_1^i$ und anschließend in $\varphi(2)$ die Produktion $X_2^{i'} \to P_2^{i'} Y_2^{i'}$ mit $i \neq i'$ gewählt, so ist keine Produktion aus $\varphi(2+i)$ anwendbar, da $(Y_1^i, Y_1^i), (Y_2^{i'}, Y_2^{i'}) \notin \varphi(2 + i)$ gilt. Es folgt $L(G) = L(G', \varphi)$. $\square$

Dieser Satz kann mit einigen Änderungen auf den Fall der Vorkommensprüfung übertragen werden (siehe [23], Theorem V.4.2, Seite 158), ebenso auch auf den ε-freien Fall.

11.3 Programmierte Grammatiken

In einer programmierten Grammatik sind jeder Produktion f sogenannte „goto"-Felder zugeordnet, und zwar eine Teilmenge $\sigma(f)$ der Produktionenmenge als „Erfolgsfeld" und eine Teilmenge $\mu(f)$ als „Mißerfolgsfeld". Dabei ist es nötig, daß die Produktionen der Grammatik mit Marken versehen werden.

Es sei S ein Alphabet. Eine endliche Menge Lab(S) wird als *Menge von Marken von* S bezeichnet, wenn sie für jedes $s \in S$ mindestens eine Marke enthält, wobei

die Mengen der Marken für $s_1, s_2 \in S$, $s_1 \neq s_2$, disjunkt sind. Das bedeutet, daß jede surjektive Abbildung $S' \to S$ eine Menge von Marken von S liefert. Wir machen darauf aufmerksam, daß für eine Menge S die Menge $\mathrm{Lab}(S)$ nicht eindeutig bestimmt ist. Mit $\mathrm{Lab}(F)$ ist in der folgenden Definition jede feste Menge von Marken von F gemeint.

Definition 11.9 (G, σ, μ) heißt *programmierte Grammatik vom Typ* i, $i = 0, 1, 2, 3$, wenn

(a) $G = (V_N, V_T, X_0, F)$ eine Typ-i-Grammatik ist und

(b) $\sigma, \mu : \mathrm{Lab}(F) \to \mathfrak{P}(\mathrm{Lab}(F))$ Abbildungen sind. Für $f \in \mathrm{Lab}(F)$ werde $\sigma(f)$ als *Erfolgsfeld* und $\mu(f)$ als *Mißerfolgsfeld* von f bezeichnet. □

Definition 11.10 Es sei (G, σ, μ) mit $G = (V_N, V_T, X_0, F)$ eine programmierte Grammatik.

(a) Für $P, Q \in V^*$ und $f_1, f_2 \in \mathrm{Lab}(F)$ gilt die Relation

$$(P, f_1) \Longrightarrow (Q, f_2),$$

wenn Wörter P_1, P_2, P', Q' existieren mit

(1) $P = P_1 P' P_2$ und $Q = P_1 Q' P_2$,

(2) die Produktion $f_1 : P' \to Q'$ in F liegt und

(3) $f_2 \in \sigma(f_1)$ gilt.

(b) $\Longrightarrow^*$ ist die reflexive transitive Hülle von $\Longrightarrow$.

(c) $L(G, \sigma) = \{w \in V_T^* \mid (X_0, f) \Longrightarrow^* (w, f')$ für $f, f' \in \mathrm{Lab}(F)\}$ heißt die *von* (G, σ, μ) *erzeugte Sprache.*

(d) Für $P, Q \in V^*$ und $f_1, f_2 \in \mathrm{Lab}(F)$ gilt die Relation

$$(P, f_1) \Longrightarrow_{VP} (Q, f_2),$$

wenn entweder $(P, f_1) \Longrightarrow (Q, f_2)$ gilt oder Wörter P', Q' existieren mit

(1) $P = Q$,

(2) die Produktion $f_1 : P' \to Q'$ in F liegt,

(3) P' kein Teilwort von P ist und

(4) $f_2 \in \mu(f_1)$ gilt.

(e) $\Longrightarrow_{VP}^*$ ist die reflexive transitive Hülle von $\Longrightarrow_{VP}$.

(f) $L_{VP}(G, \sigma, \mu) = \{w \in V_T^* \mid (X_0, f) \Longrightarrow_{VP}^* (w, f')$ für $f, f' \in \mathrm{Lab}(F)\}$ heißt die *von* (G, σ, μ) *mit Vorkommensprüfung erzeugte Sprache.*

(g) Mit $\mathcal{P}^\varepsilon$ (bzw. $\mathcal{P}$) bezeichnen wir die Familie der Sprachen, die durch programmierte (ε-freie) kontextfreie Grammatiken erzeugt werden, mit $\mathcal{P}_{VP}^\varepsilon$ (bzw. $\mathcal{P}_{VP}$) die Familie der Sprachen, die durch programmierte (ε-freie) kontextfreie Grammatiken mit Vorkommensprüfung erzeugt werden. □

Aus den Definitionen ergibt sich sofort

$$\mathcal{P} \subset \mathcal{P}^\varepsilon \subset \mathcal{P}_{VP}^\varepsilon \text{ und } \mathcal{P} \subset \mathcal{P}_{VP} \subset \mathcal{P}_{VP}^\varepsilon.$$

Satz 11.3 Es gilt $\mathfrak{L}_2^{-\varepsilon} \subsetneqq \mathcal{P}$ und $\mathfrak{L}_2 \subsetneqq \mathcal{P}^{\varepsilon}$.

Beweis: Es sei G eine ε-freie Typ-2-Grammatik. Mit $\sigma(f) = \mathrm{Lab}(F)$ und $\mu(f) = \emptyset$ für alle $f \in \mathrm{Lab}(F)$ folgt $\mathfrak{L}_2^{-\varepsilon} \subset \mathcal{P}$ und $\mathfrak{L}_2 \subset \mathcal{P}^{\varepsilon}$.

Weiter sei (G, σ, μ) eine ε-freie kontextfreie programmierte Grammatik mit

	σ	μ
$f_1 : X_0 \to XYZ$	$\{f_2, f_5\}$	$\emptyset$
$f_2 : X \to aX$	$\{f_3\}$	$\emptyset$
$f_3 : Y \to bY$	$\{f_4\}$	$\emptyset$
$f_4 : Z \to aZ$	$\{f_2, f_5\}$	$\emptyset$
$f_5 : X \to a$	$\{f_6\}$	$\emptyset$
$f_6 : Y \to b$	$\{f_7\}$	$\emptyset$
$f_7 : Z \to a$	$\{f_1\}$	$\emptyset$.

Die Produktion mit der Marke f_1 muß als erstes angewendet werden. Durch iterierte Anwendung der Produktionen f_2, f_3 und f_4 in dieser Reihenfolge und abschließende Anwendung von f_5, f_6 und f_7 erhält man ein Terminalwort. Es folgt $L(G, \sigma) = L_{VP}(G, \sigma, \mu) = \{a^n b^n a^n \mid n \geq 1\}$. $L(G, \sigma)$ ist nach dem Lemma von Bar-Hillel nicht kontextfrei. Daraus folgen die echten Inklusionen. □

Beispiel 11.6 Es sei (G, σ, μ) eine ε-freie kontextfreie programmierte Grammatik mit

	σ	μ
$f_1 : X_0 \to XY$	$\{f_2, f_3, f_6, f_7\}$	$\emptyset$
$f_2 : X \to aX$	$\{f_4\}$	$\emptyset$
$f_3 : X \to bX$	$\{f_5\}$	$\emptyset$
$f_4 : Y \to aY$	$\{f_2, f_3, f_6, f_7\}$	$\emptyset$
$f_5 : Y \to bY$	$\{f_2, f_3, f_6, f_7\}$	$\emptyset$
$f_6 : X \to a$	$\{f_8\}$	$\emptyset$
$f_7 : X \to b$	$\{f_9\}$	$\emptyset$
$f_8 : Y \to a$	$\{f_1\}$	$\emptyset$
$f_9 : Y \to b$	$\{f_1\}$	$\emptyset$.

Die Produktion mit der Marke f_1 führt zu XY. Wählt man als nächstes die Produktion f_2, so muß anschließend f_4 angewendet werden, und es wird $aXaY$ erzeugt. Entsprechend erhält man mit f_3 gefolgt von f_5 das Wort $bXbY$. Diese Schritte können miteinander verwoben iteriert werden und liefern beliebige Wörter $vXvY$ mit $v \in \{a, b\}^*$. Mit f_6 gefolgt von f_8 bzw. f_7 gefolgt von f_9 können so beliebige Wörter der Sprache erzeugt werden. Somit ist $L(G, \sigma) = L_{VP}(G, \sigma, \mu) = \{ww \mid w \in \{a, b\}^+\}$. □

Beispiel 11.7 Zum Abschluß soll ein Beispiel einer ε-freien kontextfreien programmierten Grammatik (G, σ, μ) mit Vorkommensprüfung angegeben werden, nämlich

	σ	μ
$f_1 : X_0 \to ZZ$	$\{f_1\}$	$\{f_2, f_3\}$
$f_2 : Z \to X_0$	$\{f_2\}$	$\{f_1\}$
$f_3 : Z \to a$	$\{f_3\}$	$\emptyset$.

Mögliche Ableitungen gemäß (G, σ, μ) werden durch die folgende Skizze gegeben.

$$(X_0, f_1) \Longrightarrow (ZZ, f_1) \begin{cases} \Longrightarrow (ZZ, f_3) \Longrightarrow \left\{ \begin{array}{c} (aZ, f_3) \\ (Za, f_3) \end{array} \right\} \Longrightarrow (aa, f_3) \\ \Longrightarrow (ZZ, f_2) \Longrightarrow \left\{ \begin{array}{c} (X_0Z, f_2) \\ (ZX_0, f_2) \end{array} \right\} \Longrightarrow (X_0X_0, f_2) \Longrightarrow (X_0X_0, f_1) \ldots \end{cases}$$

Von (X_0X_0, f_1) ausgehend kann das Verfahren iteriert werden, wobei sich in jedem Iteratiosschritt eine Verdoppelung der Ableitungsschritte in bezug auf die vorhergehende Iteration ergibt. Folglich gilt $L_{VP}(G, \sigma, \mu) = \{a^{2^n} \mid n \geq 1\}$. $\square$

Satz 11.4 Es gilt $\mathfrak{L}_0 = \mathcal{P}^{\varepsilon}_{VP}$.

Beweis: Wegen der Churchschen These gilt $\mathcal{P}^{\varepsilon}_{VP} \subset \mathfrak{L}_0$. Ist nämlich eine kontextfreie programmierte Grammatik (G, σ, μ) mit Vorkommensprüfung vorgegeben, so erhält man einen Algorithmus zur Auflistung der Wörter der erzeugten Sprache L wie folgt (siehe auch Beweis von Satz 7.2). Man betrachtet der Reihe nach für $k = 1, 2, 3, \ldots$ alle Ableitungen gemäß (G, σ, μ) der Länge k. Erzeugt eine solche Ableitung ein Terminalwort, so wird es in die Liste der Wörter von L aufgenommen. L ist also rekursivaufzählbar und daher nach der Churchschen These vom Typ 0. Alternativ läßt sich auch eine Typ-0-Grammatik für L konstruieren, was wir hier jedoch nicht durchführen wollen.

Es bleibt $\mathfrak{L}_0 \subset \mathcal{P}^{\varepsilon}_{VP}$ zu zeigen. Es sei $L = L(G) \in \mathfrak{L}_0$ eine beliebige Sprache mit $G = (V_N, V_T, X_0, F)$. Wir müssen eine programmierte kontextfreie Grammatik $G_p = (G', \sigma, \mu)$ angeben, die L erzeugt.

Es sei $m \geq 3$ definiert durch $m - 1 = \mathrm{card}(V) = \mathrm{card}(V_N \cup V_T)$. Anstelle von Wörtern über V werden ihre Zahlenkodierungen betrachtet, d.h., es wird eine Bijektion

$$g : V \to \{1, \ldots, m-1\}$$

gewählt und zu einer Bijektion

$$g : V^* \to (\mathbb{N}_0 - \{km \mid k \in \mathbb{N}\})$$

durch

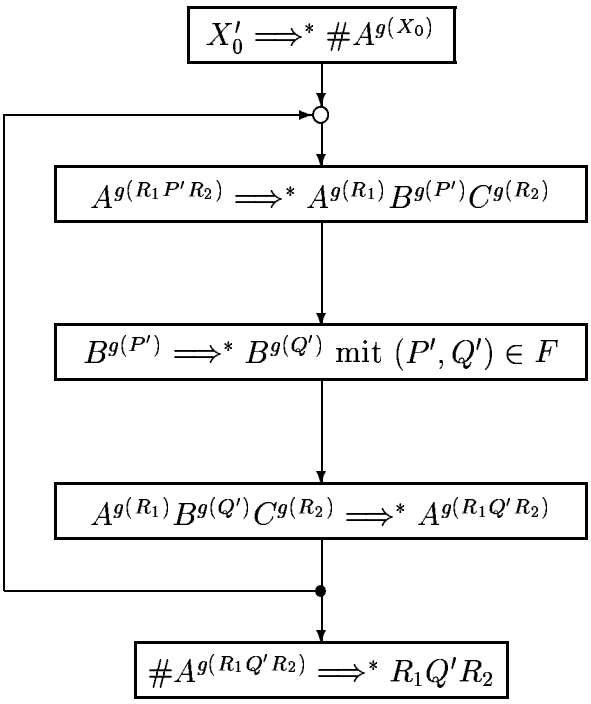
$$\begin{aligned} g(\varepsilon) &= 0, \\ g(a_1 \ldots a_n) &= g(a_1)m^{n-1} + \ldots + g(a_{n-1})m + g(a_n) \end{aligned}$$

für alle $n \geq 1$ und $a_i \in V$, $1 \leq i \leq n$, erweitert. Man beachte, daß $\{km \mid k \in \mathbb{N}\}$ nicht im Bildbereich von g vorkommt, da 0 in der Menge $\{1, \ldots, m-1\}$ fehlt.

Wir geben zunächst eine Grobskizze des Beweises an. Es sei $w \in L(G)$. Dann existiert eine Ableitung

$$X_0 \Longrightarrow W_0 \Longrightarrow W_1 \Longrightarrow \ldots \Longrightarrow W_r = w$$

gemäß G. Die simulierende Ableitung gemäß G_p arbeitet statt mit Wörtern mit deren Kodierungen nach folgendem Schema:



A, B und C sind neue Symbole. Nach dem Eintritt in die Schleife wird in nichtdeterministischer Weise eine Aufspaltung in drei Teilwörter vorgenommen. Dieser Nichtdeterminismus wird dadurch erreicht, daß die „goto"-Felder der programmierten Grammatik mehr als eine Marke enthalten. Nach jeder Iteration hat man außerdem die Wahl zwischen Dekodierung und erneuter Aufspaltung.

Die Produktionen von G_p werden in Unterprogramme unterteilt, deren Zusammenhang durch Flußdiagramme beschrieben wird. Dabei werden folgende Bezeichnungen eingeführt:

(α) $\rightarrow f$ für ein $f \in \mathrm{Lab}(F)$ eines Unterprogramms gibt die *Eintrittsstelle* in das Unterprogramm an.

(β) Das Wort „exit" in einem „goto"-Feld bedeutet, daß ins nächste Unterprogramm gemäß dem entsprechenden Flußdiagramm zu springen ist.

(γ) $P \Longrightarrow^*: Q$ bedeutet, daß das zugehörige Unterprogramm aus einem Wort mit der Anzahl der Vorkommen der Symbole wie in P ein Wort mit der Anzahl der Vorkommen der Symbole wie in Q erzeugt, wobei Zeichen, die nicht in P enthalten sind, nicht verändert werden. Die Ordnung der Symbole ist bei dieser Betrachtungsweise also irrelevant.

Weiter werde für $i \in \mathbb{N}_0$ mit $\left[\frac{i}{m}\right]$ der ganzzahlige Anteil von $\frac{i}{m}$ und mit $\mathrm{rem}(\frac{i}{m})$ der Divisionsrest bezeichnet, d.h., es gilt

$$\frac{i}{m} = \left[\frac{i}{m}\right] + \mathrm{rem}\left(\frac{i}{m}\right) \cdot \frac{1}{m} \qquad \left(\Longleftrightarrow i = \left[\frac{i}{m}\right] \cdot m + \mathrm{rem}\left(\frac{i}{m}\right)\right).$$

Wir definieren nun das Unterprogramm DIV(A, B). Anstelle von A und B können dabei auch andere Nichtendsymbole verwendet werden.

			σ	μ	
$\to$	d_1	$: A \to A'$	$\{d_2\}$	$\{d_{2m}\}$	
	d_i	$: A \to \varepsilon$	$\{d_{i+1}\}$	$\{d_{m+i-1}\}$	$, i = 2, \ldots, m-1$
	d_m	$: A \to \varepsilon$	$\{d_1\}$	$\{d_{2m-1}\}$	
	d_{m+i}	$: A' \to B^i$	$\{d_{2m}\}$	$\emptyset$	$, i = 2, \ldots, m-1$
	d_{2m}	$: A' \to A$	$\{d_{2m}\}$	exit	

DIV(A, B) erfüllt für alle $i \in \mathbb{N}_0$ die Relation $A^i \Longrightarrow^*: A^{\left[\frac{i}{m}\right]} B^{\mathrm{rem}\left(\frac{i}{m}\right)}$. Dies sehen wir wie folgt ein. Durch d_1 wird ein Symbol A in A' überführt. Falls möglich, werden in den nächsten $m-1$ Schritten $m-1$ Symbole A durch die Produktionen d_2 bis d_m gelöscht, insgesamt also m Symbole A in ein einziges A' überführt. Dieses Verfahren wird wiederholt, bis im letzten Durchgang kein A mehr vorhanden ist. Sind zu Beginn dieser letzten Iteration $i = \mathrm{rem}\left(\frac{i}{m}\right) \in \{1, \ldots, m-1\}$ Vorkommen von A vorhanden, so ist d_{i+1} nicht anwendbar. Das entsprechende Mißerfolgsfeld führt zur Anwendung von $d_{m+i} : A' \to B^i$. Anschließend werden durch d_{2m} die A' in Symbole A umgewandelt. Gibt es dagegen keinen Rest, ist also $i = 0$, so führt das Mißerfolgsfeld direkt zu d_{2m}. Ist kein A in dem vorgelegten Wort enthalten, so kommt man gleich zu „exit“.

Weiter definieren wir das Unterprogramm MULT(A, B, C):

			σ	μ
$\to$	e_1	$: A \to A'$	$\{e_2\}$	$\{e_4\}$
	e_2	$: B \to B'C$	$\{e_2\}$	$\{e_3\}$
	e_3	$: B' \to B$	$\{e_3\}$	$\{e_1\}$
	e_4	$: A' \to A$	$\{e_4\}$	exit

MULT(A, B, C) erfüllt für alle $i, j \in \mathbb{N}_0$ die Relation $A^i B^j \Longrightarrow^*: A^i B^j C^{ij}$. Durch e_1 wird ein A in ein A' überführt. Dann werden durch wiederholte Anwendung von e_2 alle Vorkommen von B in jeweils ein Vorkommen von $B'C$ verwandelt. Die B' werden anschließend durch e_3 in Symbole B zurückverwandelt. Mit e_1 beginnt dann die Wiederholung der vorhergehenden Schritte, bis schließlich nach i Iterationen i Symbole A', j Symbole B und $i \cdot j$ Vorkommen von C vorhanden sind. Durch e_4 werden abschließend die Vorkommen von A' in Symbole A zurückverwandelt.

Sehr einfach ist das Unterprogramm ER(A), das durch

			σ	μ
$\to$	f_1	$: A \to \varepsilon$	$\{f_1\}$	exit

gegeben ist und alle Vorkommen von A löscht. Das Unterprogramm CH(A) wird durch

			σ	μ
$\to$	f_2	$: A \to A$	exit	exit

definiert. Das nächste auszuführende Unterprogramm hängt davon ab, ob A in dem betrachteten Wort vorkommt oder nicht. Das Unterprogramm $G_1(A, B)$ mit

$$\begin{array}{cccc} & & \sigma & \mu \\ \rightarrow & f_3 : A \rightarrow AB & \text{exit} & \text{exit} \end{array}$$

erfüllt für alle $i \in \mathbb{N}$ die Relation $A^i \Longrightarrow^*: A^iB$. Für das Unterprogramm $G_2(A, B, C)$ mit

$$\begin{array}{cccc} & & \sigma & \mu \\ \rightarrow & f_4 : A \rightarrow AC & \{f_5\} & \text{exit} \\ & f_5 : A \rightarrow A' & \{f_5\} & \{f_6\} \\ & f_6 : A' \rightarrow AB & \{f_6\} & \text{exit} \end{array}$$

gilt $A^i \Longrightarrow^*: A^iB^iC$ für alle $i \in \mathbb{N}$. Das Unterprogramm $G_3(A)$ mit

$$\begin{array}{cccc} & & \sigma & \mu \\ \rightarrow & f_7 : A \rightarrow A' & \{f_7\} & \{f_8\} \\ & f_8 : A' \rightarrow A^m & \{f_8\} & \text{exit} \end{array}$$

bewirkt für alle $i \in \mathbb{N}_0$ die Relation $A^i \Longrightarrow^*: A^{mi}$. Weiter ersetzt das Unterprogramm $G_4(A, B)$ mit

$$\begin{array}{cccc} & & \sigma & \mu \\ \rightarrow & f_9 : A \rightarrow B & \{f_9\} & \text{exit} \end{array}$$

alle Symbole A durch B. Schließlich erfüllt G_5 mit

$$\begin{array}{cccc} & & \sigma & \mu \\ \rightarrow & f_{10} : X_0' \rightarrow \#A^{g(X_0)} & \text{exit} & \emptyset \end{array}$$

die Relation $X_0' \Longrightarrow^* \#A^{g(X_0)}$.

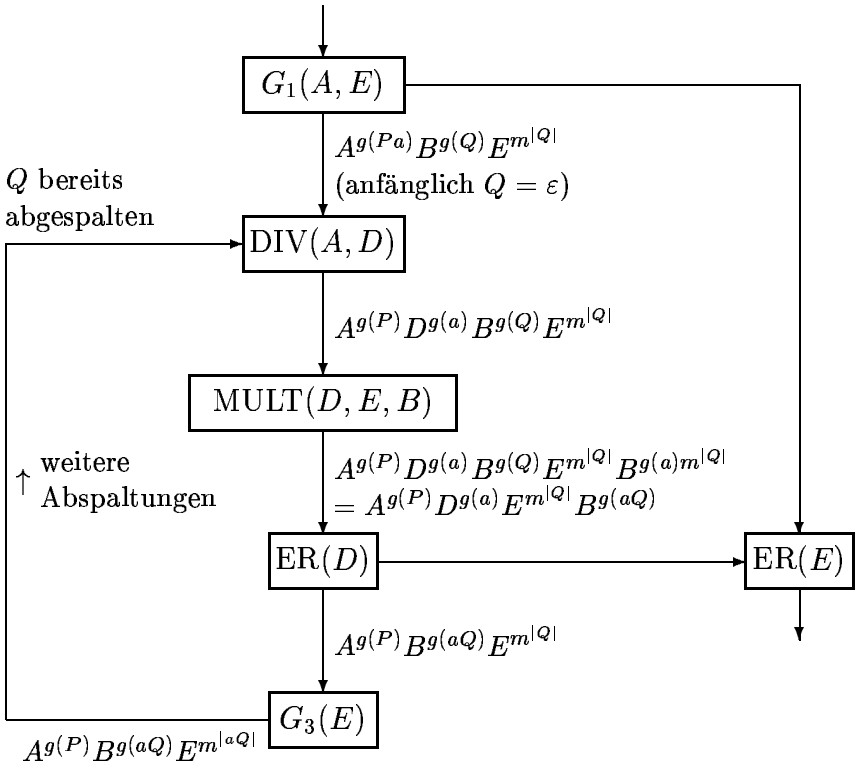
Wir kommen zur Definition des Unterprogramms $G_6(A, B)$. Nach Anwendung von $G_6(A, B)$ soll für alle $P, Q \in V^*$ mit $PQ \neq \varepsilon$

$$A^{g(PQ)} \Longrightarrow^*: A^{g(P)}B^{g(Q)}$$

gelten. Dabei beachte man, daß aufgrund der Definition von g für alle $P, Q \in V^*$ die Beziehung

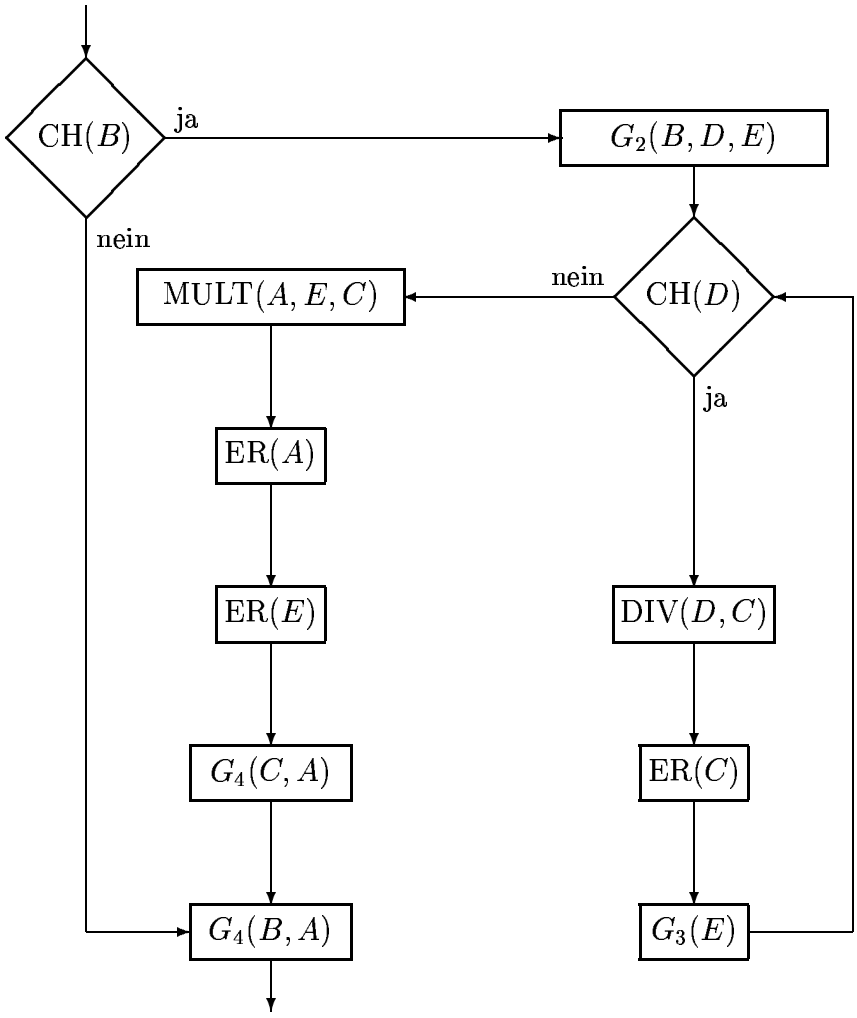
$$g(PQ) = g(P)m^{|Q|} + g(Q)$$

erfüllt ist. $G_6(A, B)$ ist wie folgt definiert:



Auf ein Wort $A^{g(R)}$ mit $R \neq \varepsilon$ wird zunächst $G_1(A,E)$ angewandt und liefert $A^{g(R)}E$. Es kann nun E gelöscht werden, so daß keine Abspaltung erfolgt. Anderenfalls tritt man in die Schleife ein. Setzen wir $R = Pa$ und $Q = \varepsilon$, so erfolgt die Abspaltung, wie sie oben im Flußdiagramm angegeben ist. Durch die aufeinanderfolgende Anwendung von $\mathrm{ER}(D)$ und $\mathrm{ER}(E)$ wird der Abspaltungsprozeß nichtdeterministisch beendet.

Das Unterprogramm $G_7(A,B)$ soll in umgekehrter Weise wie $G_6(A,B)$ wirken. Es soll Wörter zusammenfügen, also $A^{g(P)}B^{g(Q)} \Longrightarrow^*: A^{g(PQ)}$ für alle Wörter $P,Q \in V^*$ liefern. Wir setzen



Falls keine Symbole B vorhanden sind, ist $Q = \varepsilon$, und G_7 wird verlassen, ohne den Exponenten von A geändert zu haben. Anderenfalls gilt $Q = P'a$. Wir erhalten zunächst

$$A^{g(P)}B^{g(Q)} \Longrightarrow^*_{:G_2(B,D,E)} A^{g(P)}B^{g(Q)}D^{g(Q)}E.$$

Die Schleife liefert im ersten Umlauf

$$D^{g(Q)}E \Longrightarrow^*_{:\mathrm{DIV}(D,C)} D^{g(P')}C^{g(a)}E \Longrightarrow^*_{:\mathrm{ER}(C)} D^{g(P')}E \Longrightarrow^*_{:G_3(E)} D^{g(P')}E^m.$$

Offenbar werden innerhalb von $|Q|$ Iterationen alle Symbole D gelöscht und gleichzeitig das Wort $E^{m^{|Q|}}$ erzeugt. Anschließend ergibt sich

$$\begin{aligned} &A^{g(P)}B^{g(Q)}E^{m^{|Q|}} \Longrightarrow^*_{:\mathrm{MULT}(A,E,C)} A^{g(P)}B^{g(Q)}E^{m^{|Q|}}C^{g(P)m^{|Q|}} \Longrightarrow^*_{:\mathrm{ER}(A),\mathrm{ER}(E)} \\ &B^{g(Q)}C^{g(P)m^{|Q|}} \Longrightarrow^*_{:G_4(C,A)} B^{g(Q)}A^{g(P)m^{|Q|}} \Longrightarrow^*_{:G_4(B,A)} A^{g(Q)+g(P)m^{|Q|}} = A^{g(PQ)}. \end{aligned}$$

Das Unterprogramm G_8 soll $\#A^{g(P)}$ mit $P \in V^*$ im normalen Sinn der Relation $\Longrightarrow^*$ in das Wort P überführen. G_8 dient also zur Dekodierung. G_8 wird wie folgt gegeben:

		σ	μ	
die ersten $2m-1$ Produktionen von DIV(A,B)				
d_{2m}	$: A' \to A$	$\{d_{2m}\}$	$\{h_1^1\}$	
h_1^1	$: B \to \varepsilon$	$\{h_2^1\}$	$\{h_1^5\}$	
h_i^1	$: B \to \varepsilon$	$\{h_{i+1}^1\}$	$\{h_{i-1}^2\}$	$, i = 2, \ldots, m-1$
h_m^1	$: B \to \varepsilon$	$\emptyset$	$\{h_{m-1}^2\}$	
h_i^2	$: A \to A$	$\{h_i^3\}$	$\{h_i^4\}$	$, i = 1, \ldots, m-1$
h_i^3	$: \# \to \#g^{-1}(i)$	$\{d_1\}$	$\emptyset$	$, i = 1, \ldots, m-1$
h_i^4	$: \# \to g^{-1}(i)$	$\emptyset$	$\emptyset$	$, i = 1, \ldots, m-1$
h_1^5	$: \# \to \varepsilon$	$\emptyset$	$\emptyset$	

Ist ursprünglich $P = \varepsilon$ und wird somit mit dem Wort $\#$ in G_8 eingesprungen, so erzeugt die Produktionenfolge $d_1, d_{2m}, h_1^1, h_1^5$ das Wort ε. Dies ist auch der einzige Fall, in dem h_1^1 erfolglos ist und folglich h_1^5 angewendet wird. Anderenfalls ist $P \neq \varepsilon$, und wir können annehmen, daß wir mit einem Wort $\#QA^{g(Ra)}$ mit $P = RaQ$, $a \in V$ und $R, Q \in V^*$, bei der Produktion d_1 beginnen. Beim Eintritt in G_8 ist $Q = \varepsilon$. Allgemein erhalten wir

$$\#QA^{g(Ra)} \Longrightarrow^*_{:\mathrm{DIV}(A,B)} \#QA^{g(R)}B^{g(a)} \Longrightarrow^*_{:h_1^1} \ldots$$
$$\ldots \Longrightarrow^*_{:h_{g(a)}^1} \#QA^{g(R)} \Longrightarrow^*_{:h_{g(a)+1}^1} \#QA^{g(R)} \Longrightarrow_{h_{g(a)}^2} \#QA^{g(R)}.$$

Für $g(R) = 0$ endet die Dekodierung mit $\#Q \Longrightarrow_{h_{g(a)}^4} aQ$. Für $g(R) \neq 0$ wird durch $\#QA^{g(R)} \Longrightarrow_{h_{g(a)}^3} \#aQA^{g(R)}$ das Symbol a von $g(Ra)$ dekodiert und ein erneuter Schleifendurchlauf begonnen, und zwar mit dem Wort $R = R_1 b$ für ein $b \in V$.

Das Unterprogramm G_9 nimmt die Ersetzung $B^{g(P)} \Longrightarrow^*: B^{g(Q)}$ für eine beliebige Produktion $P \to Q$ aus F vor. Es sei

$$u = \max\{g(P) \mid (P, Q) \in F\},$$

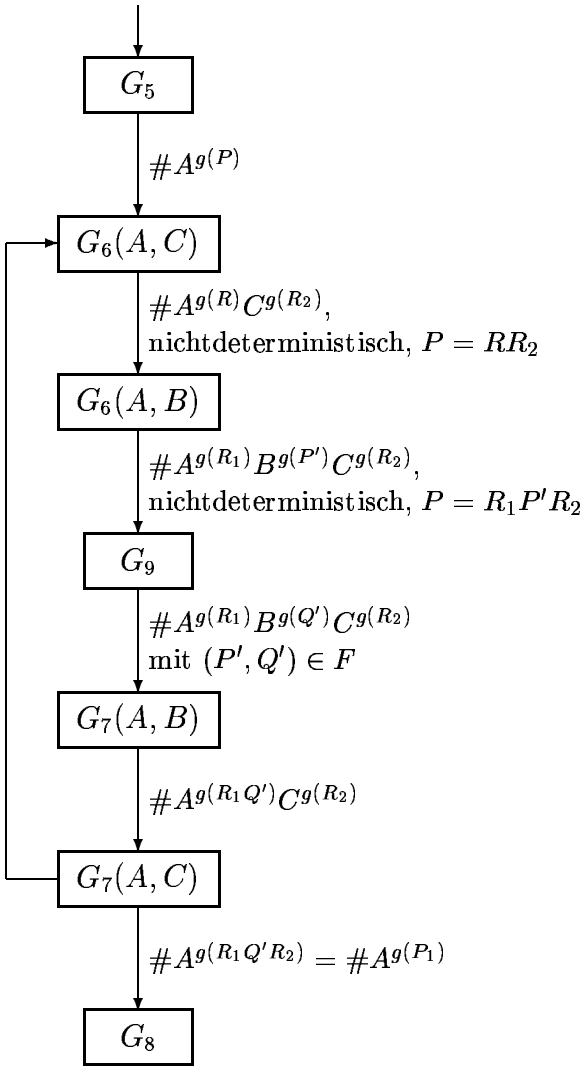
und die Produktionen in F seien mit den Marken $t_1^1, \ldots, t_v^1$ bezeichnet. Die rechte Seite der Produktion mit der Marke t_j^1, $j = 1, \ldots, v$, sei durch das Wort Q_j gegeben, und die (ggf. leere) Menge von Marken der Produktionen aus F, deren linke Seite gleich $g^{-1}(i)$, $i = 1, \ldots, u$, ist, heiße $T(i)$. Dann definieren wir G_9 durch

			σ	μ	
	t_j^1	$: D \to B^{g(Q_j)}$	exit	$\emptyset$	$, j = 1, \ldots, v$
$\to$	t_1^2	$: B \to D$	$\{t_2^2\}$	$\emptyset$	
	t_i^2	$: B \to \varepsilon$	$\{t_{i+1}^2\}$	$T(i-1)$	$, i = 2, \ldots, u$
	t_{u+1}^2	$: B \to \varepsilon$	$\emptyset$	$T(u)$.	

Da die linken Seiten aller Produktionen in F nichtleer sind, ist die Anwendung von t_1^2 immer erfolgreich. Anschließend löschen die t_i^2 die Symbole B, wobei ihre Anzahl

j durch das Mißerfolgsfeld $T(j)$ der Produktion t^2_{j+1} gemerkt wird. Dann wird durch $t^1_j \in T(j)$ eine der Produktionen $P_j \to Q_j$ mit $g(P_j) = j$ simuliert.

Schließlich kann die gesuchte programmierte kontextfreie Grammatik G_p angegeben werden. Durch das Flußdiagramm sind implizit die Nichtendsymbole und die Marken definiert. X'_0 ist das Anfangssymbol.



Begonnen wird mit dem Schritt $X'_0 \Longrightarrow_{G_5} \#A^{g(P)}$. Allgemein tritt man in die Schleife mit $\#A^{g(P)}$ ein, wobei $X_0 \Longrightarrow^*_G P$ gilt. Danach werden die Ersetzungen so durchgeführt, wie es im Flußdiagramm angegeben ist. Wir erhalten also $X'_0 \Longrightarrow^*: \#A^{g(P_1)}$, wobei $X_0 \Longrightarrow^*_G P_1$ erfüllt ist. Bei einer „erfolgreichen" Ableitung wird im Falle $P_1 \in V^*_T$ der Ableitungsprozess durch $\#A^{g(P_1)} \Longrightarrow^*_{G_8}: P_1$ beendet, für $P_1 \notin V^*_T$ wird die Schleife erneut durchlaufen.

Offensichtlich erhalten wir $L(G) = L_{VP}(G_p)$. $\square$

Satz 11.5 Für alle $L \in \mathfrak{L}_0$ existieren ein Homomorphismus h und eine Sprache $L_1 \in \mathcal{P}_{VP}$ mit $h(L_1) = L$.

Beweis: Es sei $L \in \mathfrak{L}_0$ eine Sprache über dem Alphabet V_T. Dann existiert nach Satz 11.4 eine kontextfreie programmierte Grammatik (G, σ, μ) mit $L = L_{VP}(G, \sigma, \mu)$.

Aus (G, σ, μ) konstruieren wir eine ε-freie kontextfreie programmierte Grammatik (G_1, σ, μ), indem wir ein neues Terminalzeichen c einführen und Produktionen $X \to \varepsilon$ in G durch $X \to c$ in G_1 ersetzen. Mit der Definition eines Homomorphismus $h : (V_T \cup \{c\})^* \to V_T^*$ mit

$$h(c) = \varepsilon \text{ und } h(a) = a \text{ für alle } a \in V_T$$

folgt offensichtlich $L = h(L_{VP}(G_1, \sigma, \mu))$. $\square$

Satz 11.6 Es gilt $\mathcal{M}^\varepsilon \subset \mathcal{P}^\varepsilon$ und $\mathcal{M}_{VP}^\varepsilon \subset \mathcal{P}_{VP}^\varepsilon$.

Beweis: Es sei $G = (V_N, V_T, X_0, M)$ eine kontextfreie Matrix-Grammatik, F die Menge aller Produktionen von G entsprechend ihrem jeweiligen Auftreten in den Matrizen von M und $F_1 \subset F$. Dann genügt es zu zeigen, daß eine kontextfreie programmierte Grammatik (G_1, σ, μ) existiert mit $L_{VP}(G, F_1) = L_{VP}(G_1, \sigma, \mu)$, wobei der Wertebereich von μ leer sein muß, falls F_1 leer ist. Wir setzen

$$V_1 = \{X_0', Y\} \text{ und } F_2 = \{X_0' \to X_0 Y, Y \to \varepsilon\},$$

wobei die Produktionen von F_2 durch sich selbst markiert seien. Die Produktionen von M seien so markiert, daß jedes Auftreten einer Produktion in einer Matrix eine eigene Marke erhält. Wir definieren die Grammatik $G_1 = (V_N \cup V_1, V_T, X_0', F \cup F_2)$, wobei $\mathrm{Lab}(F \cup F_2)$ durch die eben angegebenen Marken bestimmt sei. Mit $H \subset \mathrm{Lab}(F \cup F_2)$ bezeichnen wir die Menge der Marken der jeweils ersten Produktion der Matrizen von G. Es sei

$$[f_1 : X_1 \to P_1, \ldots, f_n : X_n \to P_n]$$

eine beliebige Matrix von M, wobei die zugehörigen Marken bereits mit notiert sind. Dann werden die Abbildungen σ und μ wie folgt definiert.

$$j < n: \quad \sigma(f_j) = \{f_{j+1}\},$$
$$\mu(f_j) = \begin{cases} \{f_{j+1}\}, & \text{falls } (X_j, P_j) \in F_1, \\ \emptyset & \text{sonst,} \end{cases}$$

$$j = n: \quad \sigma(f_n) = H \cup F_2,$$
$$\mu(f_n) = \begin{cases} H \cup F_2, & \text{falls } (X_n, P_n) \in F_1 \\ \emptyset & \text{sonst,} \end{cases}$$

$$f \in F_2: \quad \sigma(f) = \begin{cases} H, & \text{falls } f = (X_0', X_0 Y) \\ \{f\}, & \text{falls } f = (Y, \varepsilon) \end{cases}$$
$$\mu(f) = \emptyset.$$

Offenbar ist der Wertebereich von μ leer, falls $F_1 = \emptyset$ gilt.

Der erste Ableitungsschritt wird durch die Produktion $X_0' \to X_0 Y$ gegeben. Dann wird die Ableitung mit der ersten Produktion einer Matrix fortgesetzt. Das Symbol Y sorgt dafür, daß der vollständige Durchlauf einer Matrix simuliert wird, bevor gegebenenfalls ein Terminalwort erzeugt werden kann. Nach Anwendung der letzten Produktion einer Matrix kann wieder die erste Produktion einer anderen Matrix oder $Y \to \varepsilon$ angewendet werden. Im letzteren Fall wird Y gelöscht, man erhält ein Wort P mit $X_0 \Longrightarrow_G^* P$, und wegen $\sigma(Y \to \varepsilon) = Y \to \varepsilon$ und $\mu(Y \to \varepsilon) = \emptyset$ ist die Ableitung zu Ende. Jedes Wort P mit $X_0 \Longrightarrow_G^* P$ kann so erzeugt werden. Wir erhalten somit $L_{VP}(G, F_1) = L_{VP}(G_1, \sigma, \mu)$. □

Der Beweis für $\mathcal{M} \subset \mathcal{P}$ und $\mathcal{M}_{VP} \subset \mathcal{P}_{VP}$ verläuft sehr ähnlich. Er kann in [23], Theorem V.5.3, Seite 171, nachgelesen werden.

11.4 Gesteuerte Sprachen

Bei gesteuerten Sprachen werden die zugehörigen Ableitungen durch Sprachen über dem Alphabet der Marken der Produktionen der zugehörigen Grammatik gesteuert.

Definition 11.11 (a) Es sei $G = (V_N, V_T, X_0, F)$ eine Grammatik, $\mathrm{Lab}(F)$ eine Menge von Marken für F, $F_1 \subset \mathrm{Lab}(F)$, D eine Ableitung in G und $U \in (\mathrm{Lab}(F))^*$. U heißt *Steuerwort von* D, wenn eine der folgenden Bedingungen gilt.

(1) Es existieren Wörter P, Q, P', Q', R_1, R_2, so daß D die Ableitung $P \Longrightarrow Q$ ist, $P = R_1 P' R_2$ und $Q = R_1 Q' R_2$ gelten und die Produktion $P' \to Q'$ aus F mit $U = f \in \mathrm{Lab}(F)$ markiert ist.

(2) Es existieren Wörter P, P', Q', so daß D die aus P allein bestehende Ableitung ist und entweder $U = \varepsilon$ gilt oder anderenfalls $U = f \in F_1$ ist, wobei $P' \to Q'$ mit f markiert ist, P' jedoch kein Teilwort von P ist.

(3) Es existieren Wörter P, Q, R, U_1, U_2, so daß D die Ableitung $P \Longrightarrow^* Q \Longrightarrow^* R$ ist und $U = U_1 U_2$ gilt, wobei U_1 ein Steuerwort von $P \Longrightarrow^* Q$ und U_2 ein Steuerwort von $Q \Longrightarrow^* R$ ist.

(b) Für $C \subset (\mathrm{Lab}(F))^*$ heißt

$$L_{VP}(G, C, F_1) = \{w \in V_T^* \mid D : X_0 \Longrightarrow_{VP}^* w, U \in C \text{ ist ein Steuerwort von } D\}$$

die *von G mit Steuersprache C und mit Vorkommensprüfung für Produktionen in F_1 erzeugte Sprache.*

(c) Für $C \subset (\mathrm{Lab}(F))^*$ und $F_1 = \emptyset$ heißt

$$L(G, C) = \{w \in V_T^* \mid D : X_0 \Longrightarrow^* w, U \in C \text{ ist ein Steuerwort von } D\}$$

die *von G mit Steuersprache C erzeugte Sprache.* □

Definition 11.12 (a) Eine Sprache L heißt *Sprache vom Typ* $(i, j, 1)$, wenn $L = L_{VP}(G, C, F_1)$ gilt, wobei G eine Typ-i-Grammatik und C eine Typ-j-Sprache ist.

(b) Eine Sprache L heißt *Sprache vom Typ* $(i,j,0)$, wenn $L = L(G,C)$ gilt, wobei G eine Typ-i-Grammatik G und C eine Typ-j-Sprache ist.

(c) Die *Familien der Sprachen vom Typ* (i,j,k), $i,j = 0,1,2,3$, $k = 0,1$, werden mit $\mathfrak{L}(i,j,k)$ bezeichnet.
ε-freie kontextfreie Grammatiken bzw. Sprachen werden kurz *Typ(2 − ε)-Grammatiken* bzw. *-Sprachen* genannt. In diesem Sinn gibt es auch *Sprachen vom Typ* $(2-\varepsilon,j,k)$. □

Beispiel 11.8 Dieses Beispiel soll den Begriff des Steuerwortes verdeutlichen. Es sei $G = (\{X,Y\},\{a,b\},X,F)$ eine Grammatik. F und $\mathrm{Lab}(F)$ seien durch

$$f_1 : X \to X, \quad f_2 : X \to X, \quad f_3 : X \to aY, \quad f_4 : Y \to ab, \quad f_5 : X \to Y.$$

gegeben. Für $F_1 = \{f_2, f_5\}$ soll die Vorkommensprüfung durchgeführt werden. Steuerwörter für Ableitungen $X \Longrightarrow^* aab$ sind dann

$$f_3f_4, \quad f_3f_4f_2, \quad f_1f_3f_2f_4, \quad f_1^3f_3f_2^5f_5f_4f_2^7.$$

Keine Steuerwörter solcher Ableitungen sind hingegen

$$f_3f_4^2, \quad f_1f_5f_4f_3, \quad f_3^2f_4. \quad \square$$

Beispiel 11.9 Es sei $G = (\{X_0,X,Y,Z\},\{a,b,c\},X_0,F)$ eine Grammatik. F und $\mathrm{Lab}(F)$ und damit auch G seien durch

$$\begin{array}{llll} f_1 : X_0 \to XYZ, & f_2 : X \to aX, & f_3 : Y \to bY, & f_4 : Z \to cZ, \\ f_5 : X \to a, & f_6 : Y \to b, & f_7 : Z \to c & \end{array}$$

gegeben. Weiter sei $F_1 = \emptyset$, und die Sprache C werde durch den regulären Ausdruck $f_1(f_2f_3f_4)^*f_5f_6f_7$ bestimmt. Offensichtlich gilt

$$L(G,C) = \{a^nb^nc^n \mid n \geq 1\} \in \mathfrak{L}(2-\varepsilon,3,0). \quad \square$$

Beispiel 11.10 Die Grammatik $G = (\{X_0,Y,Z\},\{a\},X_0,F)$ werde durch

$$f_1 : X_0 \to ZZ, f_2 : Z \to X_0, f_3 : X_0 \to a, f_4 : X_0 \to Y, f_5 : Z \to Y$$

definiert. Weiter gelte $F_1 = \{f_4, f_5\}$ und $C = (f_1^*f_4f_2^*f_5)^*f_3^*$. Im „Erfolgsfall“ führen f_4 und f_5 das Symbol Y ein, das auf keiner rechten Seite einer Produktion auftritt und daher nicht mehr entfernt werden kann. Somit dürfen beide Produktionen nur im Vorkommensprüfungssinn angewendet werden, um eventuell später ein Terminalwort erzeugen zu können. f_4 sorgt dafür, daß zunächst durch f_1 jedes X_0 durch ZZ ersetzt wird. Wegen f_5 müssen anschließend durch f_2 alle Z wieder in Symbole X_0 zurückverwandelt werden. Die iterierte Anwendung von $f_1^*f_4f_2^*f_5$ liefert offenbar $X_0^{2^n}$ für alle $n \in \mathbb{N}_0$. Durch f_3^* werden die Symbole X_0 jeweils durch a ersetzt. Wir erhalten somit

$$L_{VP}(G,C,F_1) = \{a^{2^n} \mid n \geq 0\} \in \mathfrak{L}(2-\varepsilon,3,1). \quad \square$$

Definition 11.13 Wir setzen

$$\mathcal{R} = \mathfrak{L}(2-\varepsilon, 3, 0),\ \mathcal{R}^{\varepsilon} = \mathfrak{L}(2, 3, 0),\ \mathcal{R}_{VP} = \mathfrak{L}(2-\varepsilon, 3, 1) \text{ und } \mathcal{R}_{VP}^{\varepsilon} = \mathfrak{L}(2, 3, 1). \quad \square$$

Satz 11.7 Eine Sprache L werde von einer Matrix-, periodisch zeitvariablen oder programmierten Grammatik vom Typ i ($i = 3, 2-\varepsilon, 2, 1, 0$) (mit Vorkommensprüfung) erzeugt. Dann folgt

$$L \in \mathfrak{L}(i, 3, 0) \quad (L \in \mathfrak{L}(i, 3, 1)).$$

Beweis: Es sei zunächst $G = (V_N, V_T, X_0, M)$ eine Matrix-Grammatik. Dabei sei F die Menge der Vorkommen von Produktionen in M und $F_1 \subset F$. Jedem Vorkommen einer Produktion in M sei eine eigene Marke zugeordnet, und $\mathrm{Lab}(F)$ sei die Menge dieser Marken. F und $\mathrm{Lab}(F)$ sind einander bijektiv zugeordnet. $F_1' \subset \mathrm{Lab}(F)$ enthalte alle Marken der Produktionen in F_1. Für jede Matrix

$$m_i = [f_1^i : P_1^i \to Q_1^i, \ldots, f_{n_i}^i : P_{n_i}^i \to Q_{n_i}^i] \in M,\ i = 1, \ldots, u,$$

wird ein Steuerwort

$$s_i = f_1^i \ldots f_{n_i}^i \in (\mathrm{Lab}(F))^*$$

definiert. Bei u Matrizen sind $s_1, \ldots, s_u$ alle Wörter dieser Art. Wir definieren die reguläre Steuersprache

$$C = (s_1 \cup \ldots \cup s_u)^*$$

und

$$G_1 = (V_N, V_T, X_0, F).$$

Ersichtlich ist

$$L_{VP}(G, F_1) = L_{VP}(G_1, C, F_1').$$

Da $F_1' = \emptyset$ aus $F_1 = \emptyset$ folgt, ist die Behauptung für Matrix-Grammatiken bewiesen.

Weiter sei (G, φ) eine periodisch zeitvariable Grammatik mit $G = (V_N, V_T, X_0, F)$, der Periode k und $F_1 \subset F$. Wir betrachten $\varphi(j) \subset F$ für $j = 1, \ldots, k$. Der Menge der Vorkommen der Produktionen in allen $\varphi(j)$ werde bijektiv eine Menge $\mathrm{Lab}(F)$ von zugehörigen Marken zugeordnet.

Es sei $F_1' \subset \mathrm{Lab}(F)$ die Menge der Marken der Produktionen in F_1. Mit K_j für $j = 1, \ldots, k$ bezeichnen wir die Menge der Marken der Produktionen in $\varphi(j)$. Wir definieren die reguläre Steuersprache

$$C = (K_1 \ldots K_k)^*(\{\varepsilon\} \cup K_1 \cup K_1K_2 \cup \ldots \cup K_1K_2 \ldots K_{k-1}).$$

Der hintere Teil der Steuersprache ist erforderlich, da eine Ableitung gemäß (G, φ) nach Anwendung einer Produktion aus einem beliebigen $\varphi(j)$, $j = 1, \ldots, k$, zu einem Terminalwort führen kann, die Periode also nicht vollständig ausgeführt werden muß. Wir erkennen, daß

$$L_{VP}(G, \varphi, F_1) = L_{VP}(G_1, C, F_1')$$

gilt, wobei wieder F_1' leer ist, falls F_1 leer ist.

Schließlich sei (G, σ, μ) eine programmierte Grammatik mit $G = (V_N, V_T, X_0, F)$, $\mathrm{Lab}(F) = \{f_1, \ldots, f_n\}$ und den Abbildungen $\sigma, \mu : \mathrm{Lab}(F) \to \mathfrak{P}(\mathrm{Lab}(F))$. Es gelte $f_j : P_j \to Q_j$ für $j = 1, \ldots, n$. Zunächst wird eine äquivalente programmierte Grammatik (G_1, σ_1, μ_1) mit

$$G_1 = (V_N \cup \{Y\}, V_T, X_0, \{f_j : P_j \to Q_j \mid j = 1, \ldots, n\} \\ \cup \{g_j : P_j \to \alpha_j(Y) \mid j = 1, \ldots, n\})$$

definiert, wobei

$$\alpha_j(Y) = \begin{cases} \beta Y \gamma, & \text{falls } G \text{ vom Typ 1 und } P_j = \beta X \gamma, Q_j = \beta \delta \gamma \\ Y & \text{sonst} \end{cases}$$

gilt. Mit $S = \{f_1, \ldots, f_n, g_1, \ldots, g_n\}$ bezeichnen wir die Menge der Marken von G_1. Wir definieren $\sigma_1, \mu_1 : S \to \mathfrak{P}(S)$ durch

$$\begin{aligned} &\mu_1(f_j) = \sigma_1(g_j) = \emptyset, \\ &\sigma_1(f_j) = \sigma(f_j) \cup \{g_u \mid f_u \in \sigma(f_j)\} \text{ und} \\ &\mu_1(g_j) = \mu(f_j) \cup \{g_u \mid f_u \in \mu(f_j)\} \end{aligned}$$

für $j = 1, \ldots, n$. Wir zeigen

$$L_{VP}(G, \sigma, \mu) = L_{VP}(G_1, \sigma_1, \mu_1).$$

Wir werden uns zunächst überlegen, daß die erfolgreiche Anwendung einer Produktion mit der Marke f_j gemäß (G, σ, μ) äquivalent ist zu der erfolgreichen Anwendung derselben Produktion mit derselben Marke f_j gemäß (G_1, σ_1, μ_1). Außerdem ist die Anwendung einer Produktion mit der Marke f_j gemäß (G, σ, μ) im Vorkommensprüfungssinn zur Anwendung der Produktion mit der Marke g_j gemäß (G_1, σ_1, μ_1) im Vorkommensprüfungssinn äquivalent. Es sei nämlich P das nach einer erfolgreichen Anwendung einer Produktion mit der Marke f_u gemäß (G_1, σ_1, μ_1) erzeugte Wort. Dann wird mit einer Produktion mit einer Marke $h \in \sigma_1(f_u)$ fortgefahren.

(α) Für $h = f_j \in \sigma(f_u)$ gilt: Falls P_j ein Teilwort von P ist, dann kann die Produktion mit der Marke f_j wie in (G, σ, μ) erfolgreich angewendet werden. Die nächste Produktion wird dann gemäß $\sigma_1(f_j)$ gewählt. Ist P_j jedoch kein Teilwort von P, so stoppt die Ableitung wegen $\mu_1(f_j) = \emptyset$.

(β) Für $h = g_j$ mit $f_j \in \sigma(f_u)$ gilt: Falls P_j ein Teilwort von P ist, dann wird das Nichtterminalzeichen Y eingeführt, und die Ableitung stoppt wegen $\sigma_1(g_j) = \emptyset$. Anderenfalls erfolgt eine Anwendung von g_j im Vorkommensprüfungssinn, und die Ableitung wird mit $f_{j'}$ oder $g_{j'}$ aus $\mu_1(g_j)$ für $f_{j'} \in \mu(f_j)$ fortgesetzt und dadurch die Anwendung von f_j gemäß (G, σ, μ) im Vorkommensprüfungssinn simuliert.

Dieselben Überlegungen gelten nach einer Anwendung von g_u im Vorkommensprüfungssinn. In (α) und (β) muß nur $\sigma(f_u)$ durch $\mu(f_u)$ ersetzt werden.

Als ein Beispiel betrachten wir die Ableitung mit dem Steuerwort $f_u f_j f_{j'}$ gemäß (G, σ, μ) mit $f_j \in \sigma(f_u)$, $f_{j'} \in \mu(f_j)$. Diese wird durch die Ableitung mit dem Steuerwort $f_u g_j f_{j'}$ mit $g_j \in \sigma_1(f_u)$, $f_{j'} \in \mu(f_j) \subset \mu_1(g_j)$ in (G_1, σ_1, μ_1) simuliert, falls die Produktion mit der Marke $f_{j'}$ in (G, σ, μ) erfolgreich angewendet werden kann (siehe auch Beispiel 11.11) . Anderenfalls wird die Ableitung mit dem Steuerwort $f_u g_j g_{j'}$ mit $g_{j'} \in \mu_1(g_j)$ in (G_1, σ_1, μ_1) durchgeführt.

Die Grammatiken G und G_1 sind offenbar von demselben Typ i. Falls der Wertebereich von μ leer ist, gilt dies auch für den Wertebereich von μ_1. Aufgrund der vorstehenden Überlegungen sind die beiden programmierten Grammatiken äquivalent.

Weiter wird eine binäre Relation R auf $S = \{f_1, \ldots, f_n, g_1, \ldots, g_n\}$ durch

$$\alpha R \beta \Longleftrightarrow \beta \in \sigma_1(\alpha) \cup \mu_1(\alpha)$$

definiert und die Steuersprache C über S durch

$$C = \{a_1 \ldots a_m \mid m \geq 2, a_i \in S \text{ für } i = 1, \ldots, m, a_j R a_{j+1} \text{ für } j = 1, \ldots, m-1\} \cup S$$

gegeben. Nach Satz 10.8 ist C regulär. Wir haben zuvor gesehen, daß die Produktionen $g_1, \ldots, g_n$ in der programmierten Grammatik G_1 nur im Vorkommensprüfungssinn angewendet werden dürfen, wenn sie nicht das Symbol Y einführen und die Ableitung beenden sollen. Aufgrund der Definition der Relation R ist klar, daß sich dasselbe Verhalten auch mit der Steuersprache C ergibt und somit

$$L_{VP}(G_1, \sigma_1, \mu_1) = L_{VP}(G_1, C, \{g_1, \ldots, g_n\})$$

gilt, wobei $\{g_1, \ldots, g_n\}$ durch $\emptyset$ zu ersetzen ist, falls der Wertebereich von μ und somit der von μ_1 leer ist. Insgesamt folgt also

$$L_{VP}(G, \sigma, \mu) = L_{VP}(G_1, C, \{g_1, \ldots, g_n\}). \quad \square$$

Beispiel 11.11 Es sei (G, σ, μ) mit

	σ	μ
$f_1 : X_0 \to ZZ$	$\{f_1\}$	$\{f_2, f_3\}$
$f_2 : Z \to X_0$	$\{f_2\}$	$\{f_1\}$
$f_3 : Z \to a$	$\{f_3\}$	$\emptyset$

die ε-freie kontextfreie programmierte Grammatik aus Beispiel 11.7. Gemäß dem Beweis von Satz 11.7 ergibt sich die folgende äquivalente Grammatik (G_1, σ_1, μ_1) :

	σ_1	μ_1
$f_1 : X_0 \to ZZ$	$\{f_1, g_1\}$	$\emptyset$
$f_2 : Z \to X_0$	$\{f_2, g_2\}$	$\emptyset$
$f_3 : Z \to a$	$\{f_3, g_3\}$	$\emptyset$
$g_1 : X_0 \to Y$	$\emptyset$	$\{f_2, f_3, g_2, g_3\}$
$g_2 : Z \to Y$	$\emptyset$	$\{f_1, g_1\}$
$g_3 : Z \to Y$	$\emptyset$	$\emptyset$.

Um zum Beispiel a^2 bzw. a^4 abzuleiten, werden die folgenden Steuerwörter verwendet:

	für a^2	für a^4
in (G, σ, μ)	$f_1 \overset{VP}{f_1} f_3 f_3$	$f_1 \overset{VP}{f_1} f_2 f_2 \overset{VP}{f_2} f_1 f_1 \overset{VP}{f_1} f_3 f_3 f_3 f_3$
in (G_1, σ_1, μ_1)	$f_1 \overset{VP}{g_1} f_3 f_3$	$f_1 \overset{VP}{g_1} f_2 f_2 \overset{VP}{g_2} f_1 f_1 \overset{VP}{g_1} f_3 f_3 f_3 f_3$. $\square$

Satz 11.8 Für $i, j, j_1 \in \{0,1,2,3\}$ und $k \in \{0,1\}$ gilt:
(a) $\mathfrak{L}(i,j,0) \subset \mathfrak{L}(i,j,1)$,
(b) $\mathfrak{L}_i \subset \mathfrak{L}(i,j,k)$,
(c) $\mathfrak{L}(i,j,k) \subset \mathfrak{L}(i,j_1,k)$ für $j \geq j_1$ und
(d) $\mathfrak{L}_j \subset \mathfrak{L}(i,j,k)$ bzw. $\mathfrak{L}_j^{-\varepsilon} \subset \mathfrak{L}(2-\varepsilon,j,k)$, wobei $\mathfrak{L}_j^{-\varepsilon}$ die Familie der ε-freien Sprachen aus $\mathfrak{L}_j$ bezeichne.

Beweis: (a) Nach Definition 11.11 gilt $L(G,C) = L_{VP}(G,C,\emptyset)$.
(b) Offensichtlich gilt $L(G) = L_{VP}(G,(\mathrm{Lab}(F))^*,\emptyset)$. Dabei ist $(\mathrm{Lab}(F))^* \in \mathfrak{L}_3 \subset \mathfrak{L}_2 \subset \mathfrak{L}_1 \subset \mathfrak{L}_0$.
(c) Für $j \geq j_1$ gilt $C \in \mathfrak{L}_j \subset \mathfrak{L}_{j_1}$, woraus die Aussage (c) folgt.
(d) Es sei $C \in \mathfrak{L}_j$ eine Sprache über $\{f_1,\ldots,f_n\}$ und G eine Grammatik mit den markierten Produktionen

$$\begin{aligned} &f_0 : X_0' \to X_0, \\ &g_0 : X_0' \to \varepsilon, \\ &f_u : X_0 \to X_0 f_u \quad \text{für } u = 1,\ldots,n, \\ &g_u : X_0 \to f_u \quad \text{für } u = 1,\ldots,n \end{aligned}$$

und dem Anfangssymbol X_0'. G ist von allen Typen 0, 1, 2 und 3. Nach Abschnitt 9.4 gilt auch $sp(C) \in \mathfrak{L}_j$. Mit der ε-freien regulären Substitution $\sigma(f_u) = \{f_u, g_u\}$ für $u = 1,\ldots,n$ definieren wir die Sprache

$$C' = \sigma(sp(C)) \cap (\{f_1,\ldots,f_n\}^*\{g_1,\ldots,g_n\} \cup \{\varepsilon\}),$$

die nach Satz 10.13 ein Element von $\mathfrak{L}_j$ ist. Aufgrund von Satz 10.1 folgt $\{f_0,g_0\}C' \in \mathfrak{L}_j$. Offensichtlich erhalten wir $C = L(G,\{f_0,g_0\}C') \in \mathfrak{L}(i,j,0)$. Eine ε-freie Typ-2-Grammatik G_2 ergibt sich aus G durch Weglassen von f_0 und g_0, und aus $C \in \mathfrak{L}_j^{-\varepsilon}$ folgt dann $C = L(G_2,C') \in \mathfrak{L}(2-\varepsilon,j,k)$. $\square$

Nur für $\mathfrak{L}_0 \subset \mathfrak{L}(1,0,k)$ und $\mathfrak{L}_j^{-\varepsilon} \subset \mathfrak{L}(2-\varepsilon,j,k)$ ist eine wie oben angegebene Grammatik erforderlich. Anderenfalls reicht für $C \in \mathfrak{L}_j$ auch eine Typ-3-Grammatik G mit den Produktionen

$$\begin{aligned} &f_u : X_0 \to X_0 f_u \text{ für } u = 1,\ldots,n \text{ und} \\ &f_{n+1} : X_0 \to \varepsilon. \end{aligned}$$

Dann folgt $C = L(G, sp(C) f_{n+1}) \in \mathfrak{L}(3,j,0)$.

Satz 11.9 Es gilt $\mathfrak{L}(0,3,1) = \mathfrak{L}_0$.

Beweis: Nach Satz 11.8 gilt $\mathfrak{L}_0 \subset \mathfrak{L}(0,3,1)$. Da jedoch alle $\mathfrak{L}(i,j,k)$ rekursiv aufzählbar sind, folgt mit der Churchschen These auch die umgekehrte Inklusion. $\square$

Satz 11.10 Es gilt $\mathfrak{L}(1,3,1) = \mathfrak{L}_1$.

Beweis: Nach Satz 11.8 gilt $\mathfrak{L}_1 \subset \mathfrak{L}(1,3,1)$, weshalb es reicht, die umgekehrte Inklusion zu zeigen. Es sei also $L = L_{VP}(G, C, F_1) \in \mathfrak{L}(1,3,1)$, wobei $G = (V_N, V_T, X_0, F)$ eine Typ-1-Grammatik ist, C regulär ist und $F_1 \subset \mathrm{Lab}(F)$ gilt. Für die Produktionen mit Marken aus F_1 wird die Ersetzung im Vorkommensprüfungssinn durchgeführt. Im ersten Schritt des Beweises wird die Vorkommensprüfung und im zweiten Schritt die Steuersprache entfernt.

Falls $f : X_0 \to \varepsilon$ in F enthalten ist, so wird f in Ableitungen von nichtleeren Wörtern nur im Vorkommensprüfungssinn angewendet. Folglich kann f durch die Produktion $f' : X_0 \to Y$ mit einem neuen Nichtterminalzeichen Y ersetzt werden, ohne dadurch die Menge der erzeugten nichttleeren Wörter von L zu beeinflussen. Auf diese Weise wird also $L - \varepsilon$ erzeugt. Unabhängig davon, ob nun $\varepsilon \in L$ gilt oder nicht, ist L kontextsensitiv, wenn auch $L - \varepsilon$ kontextsensitiv ist. Folglich können wir ohne Beschränkung der Allgemeinheit $(X_0, \varepsilon) \notin F$ annehmen. Zusätzlich bemerken wir, daß man durch Betrachtung von C herausfinden kann, ob $\varepsilon \in L$ gilt, da dann ein Wort $g_1 \ldots g_k f g_{k+1} \ldots g_n$ mit $g_1, \ldots, g_n \in F_1$, $k \geq 0$ und $n \geq k$ in C enthalten sein muß.

Zur Entfernung der Vorkommensprüfung werden eine monotone Grammatik G_1 und eine reguläre Kontrollsprache C_1 mit

$$L(G_1, C_1) = \{bwb \mid w \in L_{VP}(G, C, F_1)\}$$

definiert, wobei b ein neues Terminalzeichen ist. Wir setzen $G_1 = (V'_N, V_T \cup \{b\}, Y_0, F')$ mit

$$V'_N = V_N \cup \{Y_0, B\} \cup \{[\alpha, f] \mid \alpha \in V_N \cup V_T, f \in F_1\}.$$

Die Produktionenmenge F' enthalte

(a) alle Produktionen aus F, die wie zuvor markiert sind,
(b) die beiden Produktionen $g_1 : Y_0 \to BX_0B$ und $g_2 : B \to b$ und
(c) für jedes $f \in F_1$ die drei unten definierten Produktionenmengen A_f, E_f und B_f, wobei alle Produktionen aus A_f, E_f und B_f als ihre eigenen Marken aufgefaßt werden. Damit ist auch $\mathrm{Lab}(F_1)$ für G_1 vollständig angegeben.

Für eine beliebige Marke $f \in F_1$ mit der zugehörigen Produktion $f : P \to Q$ besteht die Menge A_f aus den Produktionen

$$\begin{array}{lcll} [\alpha, f]\beta & \to & \alpha[\beta, f], & |P| = 1,\ \alpha, \beta \in V_N \cup V_T, \alpha \neq P, \\ [\alpha, f]\beta P' & \to & \alpha[\beta, f]P', & |P| > 1,\ \alpha, \beta \in (V_N \cup V_T),\ P' \in (V_N \cup V_T)^*\{\varepsilon, B\}, \\ & & & |\alpha\beta P'| \leq |P|,\ \alpha\beta P' \text{ kein Präfix von } P, \end{array}$$

die Menge E_f aus den Produktionen

$$[\alpha, f]B \to \alpha B, \quad \alpha \in V_N \cup V_T, \alpha \neq P,$$

und schließlich die Menge B_f aus den Produktionen

$$B\alpha \to B[\alpha, f], \quad \alpha \in V_N \cup V_T.$$

Damit ist die Grammatik G_1 definiert. Die neu eingeführten Produktionen sind monoton. Da G vom Typ 1 ist, ist G_1 monoton. Wir definieren eine reguläre Substitution σ auf $\mathrm{Lab}(F)$ durch

$$\sigma(f) = \begin{cases} f, & \text{falls } f \in \mathrm{Lab}(F) - F_1, \\ f \cup B_f A_f^* E_f, & \text{falls } f \in F_1, \end{cases}$$

und es werde $C_1 = g_1\sigma(C)g_2^2$ gesetzt. Dann ist C_1 offenbar regulär, und es gilt

$$L(G_1, C_1) = \{bwb \mid w \in L_{VP}(G, C, F_1)\},$$

was wir wie folgt einsehen.

Zunächst wird durch g_1 das Wort BX_0B mit den beiden Begrenzungszeichen B abgeleitet. Es sei nun allgemein ein Wort BRB mit $X_0 \Longrightarrow_{VP}^* R$ gemäß G gegeben. Soll eine Produktion $f \in \mathrm{Lab}(F) - F_1$ simuliert werden, so wird sie wie in G abgearbeitet und liefert, abgesehen von den Begrenzungszeichen, dasselbe Wort wie in G. Eine Produktion $f : P \to Q$ in F_1, die in G im normalen Sinn angewendet werden kann, kann auf dieselbe Weise auch gemäß G_1 angewendet werden. Ein Versuch, für sie eine Vorkommensprüfungsanwendung durch $B_f A_f^* E_f$ zu simulieren, scheitert, da für $P = x_1 \ldots x_r$, $r \geq 1$ und $x_\rho \in V_N \cup V_T$ für $\rho = 1, \ldots, r$, nach anfänglicher Anwendung von B_f (Einführung eines Nichtterminalzeichens $[\alpha, f]$) und einigen eventuellen Anwendungen von A_f (Verschiebung von $[-, f]$ nach rechts) ein Wort $B \ldots [x_1, f]x_2 \ldots x_r \ldots B$ entsteht, für das weder eine weitere Anwendung von A_f noch eine von E_f möglich ist. Ist dagegen P kein Teilwort von R, so rückt nach einigen Rechtsbewegungen $[-, f]$ ganz an den rechten Rand B heran. Mit einer Produktion aus E_f wird die Klammerung entfernt, und man erhält dasselbe Wort wie vor Anwendung von B_f. In Beispiel 11.12 betrachten wir dazu einige mögliche Fälle. Insgesamt wird so genau ein Schritt $R \Longrightarrow_{VP} R'$ simuliert. Am Ende werden durch g_2 die Zeichen B durch die Terminalsymbole b ersetzt.

Zur Entfernung der Steuersprache wird eine monotone Grammatik G_2 mit

$$L(G_2) = \{bbwb \mid w \in L_{VP}(G, C, F_1)\}.$$

angegeben. Wenn die Konstruktion einer solchen monotonen Grammatik durchgeführt ist, wird eine 4-lineare Löschung h auf $V_T \cup \{b\}$ bzgl. $L(G_2)$ gemäß

$$h(b) = \varepsilon, \quad h(a) = a \text{ für alle } a \in V_T$$

definiert. Da $|w| \leq 4 \cdot |h(w)|$ für alle $w \in L(G_2)$ gilt, handelt es sich tatsächlich um eine 4-lineare Löschung. Satz 9.6 liefert dann

$$h(L(G_2)) = L_{VP}(G, C, F_1) = L \in \mathfrak{L}_1.$$

Es werde $G_2 = (V_N' \cup S \cup \{Z_0\}, V_T \cup \{b\}, Z_0, F_2)$ gesetzt, wobei S und F_2 noch anzugeben sind. Wir betrachten die Steuersprache $C_1 = g_1\sigma(C)g_2^2$ von G_1. Da C_1 regulär ist, wird C_1 von einem endlichen erkennenden Automaten $(S, V_1, \delta, s_0, S_1)$ akzeptiert, wobei V_1 die Menge der Marken von G_1 ist. Wir definieren

$$\begin{aligned} F_2 = &\{sP \to \delta(s,f)Q \mid s \in S, f : P \to Q \text{ Produktion von } G_1\} \\ &\cup \{s\alpha \to \alpha s, \alpha s \to s\alpha \mid s \in S, \alpha \in V_N' \cup V_T\} \\ &\cup \{sb \to bb \mid s \in S_1\} \cup \{Z_0 \to s_0 Y_0\}. \end{aligned}$$

Da G_1 monoton ist, ist es auch G_2. Die Produktion $Z_0 \to s_0 Y_0$ wird zu Beginn angewendet. Die Produktionen der ersten Menge der Vereinigung sorgen dafür, daß nur Produktionen aus G_1 entsprechend dem Kontrollwort aus G_1 angewendet werden können. Um an einer passenden Stelle die Ableitung wie in G_1 fortsetzen zu können, sind die Produktionen der zweiten Menge der Vereinigung nötig. Die Produktionen $sb \to bb$ mit $s \in S_1$ löschen das Zustandssymbol, wobei zu beachten ist, daß zuerst das rechte B durch b ersetzt wird und dann das Zustandssymbol vor dem linken B stehen muß, bevor dieses durch b ersetzt und schließlich eine Produktion $sb \to bb$ angewendet wird. Offenbar erhalten wir

$$L(G_2) = \{bbwb \mid w \in L_{VP}(G, C, F_1)\}. \quad \square$$

Beispiel 11.12 Wir wollen die Konstruktion im Beweis von Satz 11.10 verdeutlichen. Es sei $V_T = \{a, c, d, e, f\}$. Wir betrachten eine Produktion $f : cdd \to Q$ von G mit $Q \in V^*$, wobei $f \in F_1$ gilt, die gemäß G_1 zu bearbeiten ist. Es sei cdd ein Teilwort des vorgelegten Wortes $R = BaccddeB$. Eine Anwendung von f auf R im üblichen Sinn ist ohne weiteres möglich, bei einer Anwendung im Vorkommensprüfungssinn erhalten wir

$$BaccddeB \Longrightarrow_{B_f} B[a,f]ccddeB \Longrightarrow_{A_f} Ba[c,f]cddeB \Longrightarrow_{A_f} Bac[c,f]ddeB.$$

An dieser Stelle stoppt der Ableitungsprozeß, da weder Produktionen aus A_f noch aus E_f anwendbar sind. Für $f \in F_1$ mit $f : e \to Q$ und demselben Wort R erhält man zunächst die entsprechende Ableitung und dann weiter

$$Bac[c,f]ddeB \Longrightarrow_{A_f} Bacc[d,f]deB \Longrightarrow_{A_f} Baccd[d,f]eB \Longrightarrow_{A_f} Baccdd[e,f]B.$$

Danach stoppt die Ableitung. Gehen wir jedoch von einer Produktion $f : cad \to Q$ mit $f \in F_1$ aus, so können wir noch einen Schritt

$$Baccdd[e,f]B \Longrightarrow_{E_f} BaccddeB$$

durchführen, und die Ersetzung im Vorkommensprüfungssinn ist erfolgreich simuliert. Eine Produktion $f : ddee \to Q$ wird im Vorkommensprüfungssinn durch

$$BaccddeB \Longrightarrow_{B_f} B[a,f]ccddeB \Longrightarrow_{A_f}^* Baccdd[e,f]B \Longrightarrow_{E_f} BaccddeB$$

simuliert. Im mittleren Ableitungsteil muß dabei die Produktion $[d,f]deB \to d[d,f]eB$ verwendet werden. $\square$

Satz 11.11 Es gilt $\mathfrak{L}(3,3,1) = \mathfrak{L}_3$.

Beweis: Wegen Satz 11.8 gilt $\mathfrak{L}_3 \subset \mathfrak{L}(3,3,1)$, so daß nur noch die umgekehrte Inklusion gezeigt werden muß. Es sei also $L = L_{VP}(G, C, F_1) \in \mathfrak{L}(3,3,1)$ mit der links-linearen Grammatik $G = (V_N, V_T, X_0, F)$, der regulären Sprache C und der Menge $F_1 \subset \mathrm{Lab}(F)$ von Marken gegeben. Wir konstruieren einen endlichen Übersetzungsautomaten U mit $L = sp(U(C))$. Nach Satz 10.9(b) und Satz 4.2 folgt dann $L \in \mathfrak{L}_3$.

Die Zustände von U werden durch die Menge $V_N \cup \{e\}$ gegeben, dabei sei $\{e\}$ die Endzustandsmenge und X_0 der Anfangszustand. Die Eingabemenge wird durch $\mathrm{Lab}(F)$ und die Ausgabemenge durch V_T definiert. Die Produktionenmenge F_U besteht aus folgenden Produktionen:

(1) Für jede Produktion

$$f : X \to Yw \quad \text{mit } X, Y \in V_N, w \in V_T^*, f \in \mathrm{Lab}(F)$$

aus F enthält F_U die Produktion $Xf \to sp(w)Y$ und, falls $f \in F_1$ gilt, zusätzlich alle Produktionen $Zf \to Z$ mit $Z \in (V_N \cup \{e\}) - \{X\}$.

(2) Für jede Produktion

$$g : X \to w \quad \text{mit } X \in V_N, w \in V_T^*, g \in \mathrm{Lab}(F)$$

aus F enthält F_U die Produktion $Xg \to sp(w)e$ und, falls $g \in F_1$ gilt, zusätzlich alle Produktionen $Zg \to Z$ mit $Z \in (V_N \cup \{e\}) - \{X\}$.

Die Ersetzung einer Produktion von G im normalen Sinn wird durch die Produktionen

$$Xf \to sp(w)Y \text{ oder } Xg \to sp(w)e$$

und im Vorkommensprüfungssinn durch

$$Zf \to Z \text{ oder } Zg \to Z$$

simuliert. Somit folgt $sp(U(C)) = L_{VP}(G, C, F_1)$. □

Beispiel 11.13 Wir wollen die Konstruktion aus Satz 11.11 verdeutlichen. Es seien

$$f_1 : X_0 \to Y_1 w_1, \; f_2 : Y_2 \to Y_3 w_2, \; g : Y_1 \to w_3$$

mit $X_0, Y_1, Y_2, Y_3 \in V_N$ und $w_1, w_2, w_3 \in V_T^*$ Produktionen aus F, wobei $f_2 \in F_1$ gelte. Mit $f_1 f_2 g \in C$ wird das Wort $w_3 w_1 \in L(G, C, F_1)$ erzeugt. Wegen

$$X_0 f_1 f_2 g \Longrightarrow sp(w_1) Y_1 f_2 g \Longrightarrow_{VP} sp(w_1) Y_1 g \Longrightarrow sp(w_1) sp(w_3) e$$

übersetzt U das Steuerwort $f_1 f_2 g$ in das Wort $sp(w_1)sp(w_3) = sp(w_3 w_1)$. □

Aus den Sätzen 11.7, 11.8(b), 11.9, 11.10 und 11.11 ergibt sich unmittelbar der folgende Satz.

Satz 11.12 Für $i = 0, 1, 3$ ist $\mathfrak{L}_i$ gleich der Familie der von Matrix-Grammatiken (periodisch zeitvariablen Grammatiken, programmierten Grammatiken) vom Typ i mit oder ohne Vorkommensprüfung erzeugten Sprachen. □

Dieser Satz zeigt, warum wir uns in den Abschnitten 11.1, 11.2 und 11.3 auf Grammatiken vom Typ 2 beschränkt haben.

Satz 11.13 Jede Sprache der Familie $\mathfrak{L}(2, 3, 0) = \mathcal{R}^\varepsilon$ wird von einer kontextfreien Matrix-Grammatik erzeugt, deren Matrizen jeweils genau zwei Produktionen enthalten.

Beweis: Es sei $L = L(G_1, C) \in \mathcal{R}^\varepsilon$, wobei $G_1 = (V_N, V_T, X_0, F)$ eine kontextfreie Grammatik und $C \subset (\mathrm{Lab}(F))^*$ regulär ist. C werde von einem endlichen erkennenden Automaten $E = (S, \mathrm{Lab}(F), \delta, s_0, S_1)$ akzeptiert. Es sei $\mathrm{Lab}(F) = \{f_1, \ldots, f_n\}$ mit $f_j : X_j \to P_j$ für $j = 1, \ldots, n$. Dann wird L von der kontextfreien Matrix-Grammatik

$$G = (V_N \cup S, V_T, Y_0, M)$$

erzeugt, wobei M die Matrizen

$$\begin{array}{ll} [Y_0 \to Y_0, Y_0 \to X_0 s_0], & \\ [X_j \to P_j, s \to \delta(s, f_j)] & \text{für } s \in S, 1 \le j \le n, \text{ und} \\ [X_j \to P_j, s \to \varepsilon] & \text{für } s \in S, 1 \le j \le n, \delta(s, f_j) \in S_1, \end{array}$$

besitzt. Durch die erste Matrix wird zunächst das Anfangssymbol Y_0 durch $X_0 s_0$ ersetzt. In den anderen Matrizen führt die erste Produktion jeweils die eigentliche Ersetzung gemäß einer Produktion f durch. In der zweiten Komponente wird der endliche erkennende Automat simuliert, der dieses f als Eingabesymbol erhält. Nach jeder Anwendung einer solchen Matrix gilt somit für den Zustand s, der das letzte Symbol des gerade betrachteten Wortes ist, die Gleichung $s = \delta(s_0, f_{i_1} \ldots f_{i_k})$ mit einem $k \in \mathbb{N}$, falls, ausgehend von $X_0 s_0$, die den Produktionen f_{i_κ}, $\kappa = 1, \ldots, k$, zugeordneten Matrizen in dieser Reihenfolge angewendet wurden. Ist s ein Endzustand, so gilt $f_{i_1} \ldots f_{i_k} \in C$. Dann hätte s auch durch f_{i_k} gelöscht werden können. In diesem Fall ist die Ableitung beendet, es ergibt sich ein Wort über $(V_N \cup V_T)^*$, das zu $L(G_1)$ gehört, falls es terminal ist. Jedes Wort aus $L(G_1)$ läßt sich so gemäß G erzeugen. Wir schließen $L = L(G)$. □

Satz 11.14 Es gilt $\mathcal{M}^\varepsilon = \mathcal{T}^\varepsilon = \mathcal{P}^\varepsilon = \mathcal{R}^\varepsilon$.

Beweis: Es sei $\mathcal{M}_2^\varepsilon$ die Familie der Sprachen aus $\mathcal{M}^\varepsilon$, die von einer Matrix-Grammatik erzeugt werden, deren Matrizen aus genau zwei Produktionen bestehen. Dann gelten die Beziehungen

$$\mathcal{M}^\varepsilon \subset \mathcal{P}^\varepsilon \subset \mathcal{R}^\varepsilon \subset \mathcal{M}_2^\varepsilon \subset \mathcal{M}^\varepsilon \text{ und } \mathcal{T}^\varepsilon \subset \mathcal{R}^\varepsilon \subset \mathcal{M}_2^\varepsilon \subset \mathcal{T}^\varepsilon.$$

Für die erste Inklusionskette werden die Sätze 11.6, 11.7 und 11.13 in dieser Reihenfolge benutzt, die letzte Inklusion ist trivial. Für die zweite Inklusionskette sind es die Sätze 11.7, 11.13 und 11.2. □

Satz 11.15 Jede Sprache der Familie $\mathcal{M}^{\varepsilon}$ wird von einer kontextfreien Matrix-Grammatik erzeugt, deren Matrizen jeweils genau zwei Produktionen enthalten.

Beweis: Dies folgt unmittelbar aus der ersten Inklusionskette des Beweises von Satz 11.14. □

Der folgende Satz wird entsprechend Satz 11.14 bewiesen. Da wir jedoch die zugehörigen Inklusionen nicht alle gezeigt haben, fehlt auch hier der Beweis.

Satz 11.16 Es gilt

$$\begin{aligned}\mathcal{T} &= \mathcal{M} = \mathcal{P} = \mathcal{R},\\ \mathcal{T}_{VP} &= \mathcal{M}_{VP} = \mathcal{P}_{VP} = \mathcal{R}_{VP},\\ \mathcal{T}^{\varepsilon}_{VP} &= \mathcal{M}^{\varepsilon}_{VP} = \mathcal{P}^{\varepsilon}_{VP} = \mathcal{R}^{\varepsilon}_{VP} = \mathfrak{L}_0. \quad \square\end{aligned}$$

Nicht bekannt ist, ob $\mathcal{R} = \mathcal{R}_{VP}$ oder $\mathcal{R} \subsetneqq \mathcal{R}_{VP}$ gilt. Des weiteren ist die Lage von $\mathcal{R}^{\varepsilon}$ bzgl. $\mathfrak{L}_1$ und $\mathcal{R}_{VP}$ unbekannt. Außerdem erhalten wir die folgenden Beziehungen (siehe [23], S. 181):

$\mathfrak{L} = \mathfrak{L}(i, j, 0)$

	j			
i	0	1	2	3
0	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$
1	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L}_1 \subset \mathfrak{L} \subsetneqq \mathfrak{L}(rek)$	$\mathfrak{L} = \mathfrak{L}_1$
2	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L}_2 \subsetneqq \mathfrak{L}$	$\mathfrak{L}_2 \subsetneqq \mathfrak{L}$
$2-\varepsilon$	$\mathfrak{L} = \mathfrak{L}_0^{-\varepsilon}$	$\mathfrak{L} = \mathfrak{L}_0^{-\varepsilon}$	$\mathfrak{L}_2^{-\varepsilon} \subsetneqq \mathfrak{L} \subsetneqq \mathfrak{L}^{-\varepsilon}(rek)$	$\mathfrak{L}_2^{-\varepsilon} \subsetneqq \mathfrak{L} \subsetneqq \mathfrak{L}_1^{-\varepsilon}$
3	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_2$	$\mathfrak{L} = \mathfrak{L}_3$

$\mathfrak{L} = \mathfrak{L}(i, j, 1)$

	j			
i	0	1	2	3
0	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$
1	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L}_1 \subset \mathfrak{L} \subsetneqq \mathfrak{L}(rek)$	$\mathfrak{L} = \mathfrak{L}_1$
2	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$
$2-\varepsilon$	$\mathfrak{L} = \mathfrak{L}_0^{-\varepsilon}$	$\mathfrak{L} = \mathfrak{L}_0^{-\varepsilon}$	$\mathfrak{L}_2^{-\varepsilon} \subsetneqq \mathfrak{L} \subsetneqq \mathfrak{L}^{-\varepsilon}(rek)$	$\mathfrak{L}_2^{-\varepsilon} \subsetneqq \mathfrak{L} \subsetneqq \mathfrak{L}_1^{-\varepsilon}$
3	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_0$	$\mathfrak{L} = \mathfrak{L}_2$	$\mathfrak{L} = \mathfrak{L}_3$

11.5 Geordnete Grammatiken

In diesem Abschnitt betrachten wir eine letzte Einschränkung für die Anwendung von Produktionen, die durch die Einführung einer partiellen Ordnung auf der Produktionenmenge gegeben wird.

Definition 11.14 (a) $(G,>)$ heißt *geordnete Grammatik vom Typ* i, $i = 3, 2, 2-\varepsilon, 1, 0$, wenn $G = (V_N, V_T, X_0, F)$ eine Typ-i-Grammatik und $>$ eine partielle Ordnung auf F ist.

(b) Für $P, Q \in (V_N \cup V_T)^*$ gilt die Relation $P \Longrightarrow Q$, wenn Wörter $R_1, R_2, P', Q' \in (V_N \cup V_T)^*$ und eine Produktion $f \in F$ existieren mit $P = R_1 P' R_2$, $Q = R_1 Q' R_2$ und $f : P' \to Q'$, wobei für alle $f' \in F$ mit $f' > f$, $f' \neq f$, die linke Seite von f' kein Teilwort von P ist.

(c) $\Longrightarrow^*$ ist die reflexive transitive Hülle von $\Longrightarrow$.

(d) $L(G,>) = \{w \in V_T^* \mid X_0 \Longrightarrow^* w\}$ heißt die *von* $(G,>)$ *erzeugte Sprache.* □

Wählt man für $>$ die leere Relation, so folgt unmittelbar, daß jede Sprache $L \in \mathfrak{L}_i$ von einer geordneten Typ-i-Grammatik erzeugt wird.

Beispiel 11.14 Wir betrachten eine Typ-$(2-\varepsilon)$-Grammatik G, deren Produktionen wie folgt gegeben sind:

$$
\begin{array}{lll}
1: X_0 \to XYZ & 2: X \to aX_1 & 3: X \to X_2 \\
4: X \to U_1 & 5: Y \to bY_1 & 6: Y \to Y_2 \\
7: Y \to U_1 & 8: Z \to cZ_1 & 9: Z \to Z_2 \\
10: Z \to U_1 & 11: X_1 \to X & 12: X_1 \to U_1 \\
13: X_1 \to U_2 & 14: Y_1 \to Y & 15: Y_1 \to U_1 \\
16: Y_1 \to U_2 & 17: Z_1 \to Z & 18: Z_1 \to U_1 \\
19: X_2 \to a & 20: X_2 \to U_1 & 21: Y_2 \to b \\
22: Y_2 \to U_1 & 23: Z_2 \to c. &
\end{array}
$$

X_0 sei das Anfangssymbol. Für die Relation $>$ gelte

$$
\begin{array}{llll}
18 > 2, 3; & 4 > 20 > 5; & 4 > 12 > 6; & 13 > 14; \\
7 > 22 > 8; & 7 > 15 > 9; & 16 > 17; & 10 > 11, 19, 21.
\end{array}
$$

Dann gilt $L(G,>) = \{a^n b^n c^n \mid n \geq 1\}$. Dies ergibt sich wie folgt. Jede Ableitung beginnt mit der Produktion 1, es wird also das Wort XYZ abgeleitet. Allgemein nehmen wir an, daß bereits ein Wort $a^j X b^j Y c^j Z$ für $j \geq 0$ abgeleitet ist. Aufgrund der linken Seiten der Produktionen sind höchstens die Produktionen 2 bis 10 anwendbar. Die Ordnungsrelation schließt die Produktionen 5, 6, 8 und 9 aus. Die Produktionen 4, 7 und 10 führen das Fehlschlagsymbol U_1 ein, das auf keiner linken Seite einer Produktion vorkommt und daher nicht mehr entfernt werden kann. Um im folgenden eventuell ein Terminalwort abzuleiten, kann man also höchstens die Produktionen 2 oder 3 verwenden. Wird 2 angewendet, dann ist die Ableitung

$$
\begin{array}{lll}
a^j X b^j Y c^j Z & \overset{2}{\Longrightarrow} a^{j+1} X_1 b^j Y c^j Z & \overset{5}{\Longrightarrow} a^{j+1} X_1 b^{j+1} Y_1 c^j Z \\
& \overset{8}{\Longrightarrow} a^{j+1} X_1 b^{j+1} Y_1 c^{j+1} Z_1 & \overset{11}{\Longrightarrow} a^{j+1} X b^{j+1} Y_1 c^{j+1} Z_1 \\
& \overset{14}{\Longrightarrow} a^{j+1} X b^{j+1} Y c^{j+1} Z_1 & \overset{17}{\Longrightarrow} a^{j+1} X b^{j+1} Y c^{j+1} Z
\end{array}
$$

entsprechend den zuvor durchgeführten Überlegungen eindeutig bestimmt. In diesem Fall entsteht also wieder ein Wort der ursprünglichen Form, wobei die Anzahl der Symbole a, b und c um jeweils 1 erhöht wurde. Beginnt man dagegen mit der Produktion 3, erhalten wir die Ableitung

$$\begin{aligned} a^j X b^j Y c^j Z &\overset{3}{\Longrightarrow} a^j X_2 b^j Y c^j Z \quad \overset{6}{\Longrightarrow} a^j X_2 b^j Y_2 c^j Z \\ &\overset{9}{\Longrightarrow} a^j X_2 b^j Y_2 c^j Z_2 \quad \overset{19,21,23}{\Longrightarrow}{}^* \; a^{j+1} b^{j+1} c^{j+1}, \end{aligned}$$

wobei die Reihenfolge der Anwendung der Produktionen 19, 21 und 23 beliebig ist. Dies ist der einzige Fall, in dem ein Terminalwort entsteht. Offenbar erhalten wir die angegebene Sprache.

Die Produktionen mit den Fehlschlagsymbolen U_1 und U_2 werden eingeführt, um zu gewissen Zeitpunkten die Anwendung „kleinerer“ Produktionen mit derselben linken Seite zu verhindern. □

Falls gefordert wird, daß $>$ eine totale (d.h. lineare) Ordnung auf F ist, wird die Erzeugungsmächtigkeit von $(G,>)$ stark eingeschränkt. So ist $\{a,b\}$ z.B. von keiner geordneten Typ-1-Grammatik $(G,>)$ mit einer totalen Ordnung $>$ erzeugbar.

Satz 11.17 Für jede von einer geordneten Typ-i-Grammatik erzeugte Sprache L, $i = 3, 2-\varepsilon, 2, 1, 0$, gilt $L \in \mathfrak{L}(i, 3, 1)$.

Beweis: Es sei $(G,>)$ eine geordnete Typ-i-Grammatik mit $G = (V_N, V_T, X_0, F)$. Die Menge F bestehe aus den n Produktionen $f_j : P_j \to Q_j$, $j = 1, \ldots, n$. Ein neues Symbol $Y \notin V_N \cup V_T$ werde eingeführt und damit eine Produktionenmenge

$$F' = \{g_j : P_j \to \alpha_j(Y) \mid 1 \leq j \leq n\}$$

definiert, wobei

$$\alpha_j(Y) = \begin{cases} \beta Y \gamma, & \text{falls } G \text{ vom Typ 1, } P_j = \beta X \gamma, Q_j = \beta \delta \gamma \\ Y & \text{sonst} \end{cases}$$

wie im Beweis von Satz 11.7 gegeben sei. Dann ist die Grammatik $G_1 = (V_N \cup \{Y\}, V_T, X_0, F \cup F')$ offensichtlich vom Typ i.

Es sei $\mathrm{Lab}(F \cup F') = \{f_j, g_j \mid j = 1, \ldots, n\}$. Wir setzen $F_1 = \{g_j \mid j = 1, \ldots, n\}$. Dann definieren wir eine Kontrollsprache C über $\mathrm{Lab}(F \cup F')$. Für jedes j, $j = 1, \ldots, n$, sei $s_j \in (\mathrm{Lab}(F \cup F'))^*$ ein Wort, dessen letztes Zeichen gleich f_j ist und deren Zeichen davor in beliebiger Reihenfolge die Marken g_k mit $f_k > f_j$, $f_k \neq f_j$, sind. Falls keine Marke $f_k > f_j$, $f_k \neq f_j$, existiert, dann wird $s_j = f_j$ gesetzt. Die Sprache C wird durch den regulären Ausdruck

$$C = (s_1 \cup \ldots \cup s_n)^*$$

definiert. Es folgt $L(G,>) = L_{VP}(G_1, C, F_1)$. □

Satz 11.18 Es sei L eine Sprache und $i = 0, 1$ oder 3.

(a) L ist genau dann vom Typ i, wenn L von einer geordneten Typ-i-Grammatik erzeugt wird.
(b) Wird L von einer geordneten Typ-$(2-\varepsilon)$-Grammatik erzeugt, dann ist L kontextsensitiv.

Beweis: (a) ergibt sich aus der Folgerung zu Definition 11.14, aus Satz 11.17 und aus den Sätzen 11.9, 11.10 und 11.11.

Für (b) gilt nach Satz 11.17 $L \in \mathfrak{L}(2-\varepsilon,3,1)$. Die Inklusion $\mathfrak{L}(2-\varepsilon,3,1) \subset \mathfrak{L}(1,3,1) = \mathfrak{L}_1$ ist trivial, wobei die Gleichheit wegen Satz 11.10 erfüllt ist. Es folgt $L \in \mathfrak{L}_1$. □

Aus Beispiel 11.14 folgt, daß $\mathfrak{L}_2^{-\varepsilon}$ echt in der Familie $\mathfrak{L}$ der Sprachen enthalten ist, die von geordneten Typ-$(2-\varepsilon)$-Grammatiken erzeugt werden. Nach der Tabelle auf Seite 182 gilt $\mathcal{R}_{VP} = \mathfrak{L}(2-\varepsilon,3,1) \subsetneqq \mathfrak{L}_1^{-\varepsilon}$, so daß sich insgesamt

$$\mathfrak{L}_2^{-\varepsilon} \subsetneqq \mathfrak{L} \subsetneqq \mathfrak{L}_1^{-\varepsilon}$$

ergibt.

Zum Abschluß wollen wir eine Verallgemeinerung von geordneten Grammatiken einführen. Dazu sehen wir uns noch einmal die geordneten Grammatiken an. Es sei f eine Produktion der geordneten Grammatik $(G,>)$ und $P_1,\ldots,P_k$ die linken Seiten der Produktionen f_j mit $f_j > f$, $f_j \neq f$, für $j = 1,\ldots,k$. f ist genau dann auf ein Wort Q anwendbar, wenn die linke Seite von f ein Teilwort von Q ist und Q selbst keines der P_j als Teilwort enthält, d.h. ein Wort der regulären Sprache

$$V^* - \bigcup_{j=1}^{k} V^* P_j V^*$$

mit $V = V_N \cup V_T$ ist. Dies führt zur folgenden Verallgemeinerung geordneter Grammatiken: Für jede Produktion f einer Grammatik G wird eine reguläre Sprache $\rho(f)$ so festgelegt, daß f nur auf Wörter in $\rho(f)$ anwendbar ist. Dies wird durch die folgende Definition genauer angegeben.

Definition 11.15 (a) (G,ρ) heißt *Typ-i-Grammatik mit regulären Einschränkungen*, $i = 3,2,2-\varepsilon,1,0$, wenn $G = (V_N, V_T, X_0, F)$ eine Typ-i-Grammatik und ρ eine Abbildung von F in die Menge der regulären Sprachen über $V = V_N \cup V_T$ ist.
(b) Für $P,Q \in V^*$ gilt die Relation $P \Longrightarrow Q$, wenn Wörter $R_1,R_2,P',Q' \in V^*$ und eine Produktion $(P',Q') \in F$ existieren mit $P = R_1P'R_2$, $Q = R_1Q'R_2$ und $P \in \rho(P' \to Q')$.
(c) $\Longrightarrow^*$ ist die reflexive transitive Hülle von $\Longrightarrow$.
(d) $L(G,\rho) = \{w \in V_T^* \mid X_0 \Longrightarrow^* w\}$ heißt die *von* (G,ρ) *erzeugte Sprache.* □

Satz 11.19 (a) $\mathfrak{L}_1$ ist gleich der Familie von Sprachen, die von Typ-1-Grammatiken mit regulären Einschränkungen erzeugt werden.
(b) $\mathfrak{L}_1^{-\varepsilon}$ ist gleich der Familie der Sprachen, die von Typ-$(2-\varepsilon)$-Grammatiken mit regulären Einschränkungen erzeugt werden.

Beweis: (a) Falls L kontextsensitiv ist, ist die Aussage trivial. Anderenfalls sei (G, ρ) mit $G = (V_N, V_T, X_0, F)$ eine Typ-1-Grammatik mit regulären Einschränkungen. Die eventuell vorhandene Produktion $X_0 \to \varepsilon$ dient nur dazu, im Fall $X_0 \in \rho(X_0 \to \varepsilon)$ das leere Wort zu erzeugen. Ohne Beschränkung der Allgemeinheit können wir daher annehmen, daß $X_0 \to \varepsilon$ nicht in G enthalten ist. Die Produktionen von F seien durch

$$f_1 : P_1 \to Q_1, \ldots, f_n : P_n \to Q_n$$

für ein $n \in \mathbb{N}$ gegeben. Für jedes $f_j \in F$, $j = 1, \ldots, n$, wird $\rho(f_j)$ von einem endlichen erkennenden Automaten

$$A_j = (S^j, V, \delta^j, s_0^j, S_1^j), \quad j = 1, \ldots, n,$$

akzeptiert, wobei wir annehmen können, daß die S^j paarweise disjunkt sind. Wir betrachten die Grammatik

$$G_1 = (V_N \cup \bigcup_{j=1}^{n} S^j \cup \{Y_j \mid 1 \le j \le n\} \cup \{Z_0, Y, B\}, V_T, Z_0, F_1)$$

mit

$$\begin{aligned}
F_1 = {} & \{Z_0 \to BYX_0B, B \to \varepsilon, Y \to \varepsilon\} && (1)\\
& \cup \{\alpha Y \to Y\alpha \mid \alpha \in V\} && (2)\\
& \cup \{BY \to Bs_0^j \mid 1 \le j \le n\} && (3)\\
& \cup \{s^j\alpha \to \alpha\delta^j(s^j, \alpha) \mid \alpha \in V, 1 \le j \le n, s^j \in S^j\} && (4)\\
& \cup \{s_1^jB \to Y_jB \mid 1 \le j \le n, s_1^j \in S_1^j\} && (5)\\
& \cup \{\alpha Y_j \to Y_j\alpha \mid \alpha \in V, 1 \le j \le n\} && (6)\\
& \cup \{Y_jP_j \to YQ_j \mid 1 \le j \le n\} && (7).
\end{aligned}$$

Zunächst werden mit Hilfe der Produktion $Z_0 \to BYX_0B$ aus (1) die Begrenzungszeichen B und der allgemeine Prüfer Y eingeführt. Neben dem linken B kann mit einer Produktion aus (3) das Symbol Y in den Anfangszustand s_0^j des Automaten A_j überführt werden. Dann wird unter Benutzung der Produktionen aus (4) getestet, ob das Wort zwischen den Begrenzungszeichen zu $\rho(f_j)$ gehört. Ist dies der Fall, wird mit einer Produktion aus (5) der Produktionsmarkierer Y_j eingeführt. Die Anwendung von f_j ist also möglich. Zunächst wird mit Hilfe der Produktionen aus (6) der Produktionsmarkierer vor die linke Seite P_j der Produktion f_j gebracht. Dann erfolgt mit $Y_jP_j \to YQ_j$ aus (7) die Simulierung der Produktion. Entweder ist nun ein Terminalwort gefunden worden, das wir nach Anwendung der Produktionen $B \to \varepsilon$ und $Y \to \varepsilon$ aus (1) erhalten, oder der Prüfer Y wird durch Produktionen aus (2) neben das linke B gebracht, so daß ein weiterer Simulationsschritt beginnen kann.
Da G vom Typ 1 ist, ist G monoton. Für den Platzbedarf eines Wortes $w \in L(G)$ gilt also $PB_G(w) = |w|$. Nach der obigen Konstruktion erhalten wir daher $PB_{G_1}(w) \le 4 \cdot |w|$. Nach Satz 9.3 folgt $L(G_1) \in \mathfrak{L}_1$.

(b) Wegen Teil (a) ist für (b) nur noch zu zeigen, daß jede Sprache $L \in \mathfrak{L}_1^{-\varepsilon}$ von einer Typ-$(2-\varepsilon)$-Grammatik mit regulären Einschränkungen erzeugt werden kann. Es sei L eine ε-freie Sprache, die von einer Typ-1-Grammatik $G = (V_N, V_T, X_0, F)$ erzeugt wird. Alle Produktionen von F sind von der Form

$$f : P^f X^f Q^f \to P^f R^f Q^f, \quad X^f \in V_N, \quad R^f \neq \varepsilon.$$

Für jedes $f \in F$ definieren wir ein neues Zeichen Y^f und Produktionen $u_f : X^f \to Y^f$ und $v_f : Y^f \to R^f$. Dann wird L von der Typ-$(2-\varepsilon)$-Grammatik (G_1, ρ) mit

$$G_1 = (V_N \cup \{Y^f \mid f \in F\}, V_T, X_0, \{u_f \mid f \in F\} \cup \{v_f \mid f \in F\})$$

und den regulären Einschränkungen

$$\rho(g) = V^* \cup \bigcup_{f \in F} (V^* P^f Y^f Q^f V^*) \text{ für alle Produktionen } g \text{ aus } G_1$$

erzeugt. □

Satz 11.19(a) bleibt richtig, wenn die Typ-1-Grammatik durch eine monotone Grammatik ersetzt wird.

Kapitel 12

Mehrdeutigkeit von kontextfreien Sprachen

Mehrdeutigkeit von kontextfreien Grammatiken und Sprachen wurde bereits in Abschnitt 5.1 behandelt. In Beispiel 5.3 wurden ohne Beweis zwei mehrdeutige Sprachen angegeben. Für eine von ihnen wollen wir in diesem Kapitel den Beweis der Mehrdeutigkeit führen.

Zunächst schränken wir den Begriff der Mehrdeutigkeit auf Teilfamilien der Familie der kontextfreien Sprachen ein. Wir werden sehen, daß wir leicht Sprachen erhalten, die in bezug auf gewisse Einschränkungen mehrdeutig sind.

Definition 12.1 Eine lineare Grammatik $G = (V_N, V_T, X_0, F)$ heißt *minimal linear*, wenn

(a) $V_N = \{X_0\}$ ist und

(b) für jede Produktion $X \to P$ aus F mit $P \in V_T^*$ die Beziehung $P = a \in V_T$ gilt, wobei a kein Teilwort einer rechten Seite einer anderen Produktion aus F ist. $\square$

Beispiel 12.1 $G = (\{X\}, \{a, b\}, X, \{X \to aXa, X \to b\})$ ist offensichtlich eine minimal lineare Grammatik und erzeugt die Sprache $\{a^i b a^i \mid i \geq 0\}$. $\square$

Satz 12.1 Es existiert eine minimal lineare Sprache L mit folgenden Eigenschaften:

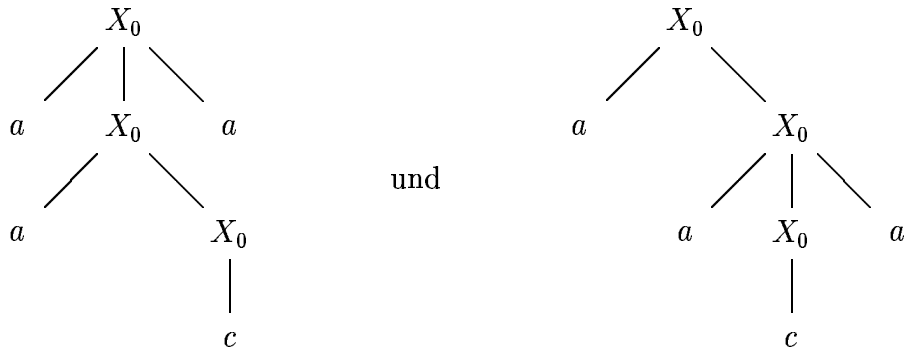
(a) Jede L erzeugende, minimal lineare Grammatik ist mehrdeutig.

(b) L wird von einer eindeutigen linearen Grammatik erzeugt.

Beweis: Die Sprache $L = \{a^m c a^n \mid m \geq n \geq 0\}$ wird von der minimal linearen Grammatik

$$G = (\{X_0\}, \{a, c\}, X_0, \{X_0 \to aX_0a, X_0 \to aX_0, X_0 \to c\})$$

erzeugt. G ist mehrdeutig, da für das Wort $a^2ca \in L$ zwei Ableitungsbäume gemäß G existieren, nämlich



Allgemein kann gezeigt werden, daß $a^m c a^n$ für jedes $m \geq n \geq 0$ genau $\binom{m}{n}$ Linksableitungen besitzt. L erfüllt die Aussage (b), da L von der eindeutigen linearen Grammatik

$$G_1 = (\{X_0, X_1\}, \{a, c\}, X_0, \{X_0 \to aX_0a, X_0 \to aX_1, X_1 \to aX_1, X_0 \to c, X_1 \to c\})$$

erzeugt wird. Um zu zeigen, daß L auch die Aussage (a) erfüllt, betrachten wir eine beliebige L erzeugende minimal lineare Grammatik G_2. Ihre Produktionen müssen von einer der beiden Formen

$$X_0 \to a^i X_0 a^j, \ i \geq j \geq 0, \ \text{oder} \ \ X_0 \to c$$

sein. Da ac und aca zu L gehören, sind $X_0 \to aX_0$ und $X_0 \to aX_0a$ Produktionen von G_2. Das heißt, daß alle Produktionen von G auch in G_2 enthalten sind. Daher besitzt jedes Wort aus L mindestens ebenso viele Linksableitungen gemäß G_2 wie gemäß G. Daher ist G_2 ebenfalls mehrdeutig. □

Definition 12.2 Es sei $G = (V_N, V_T, X_0, F)$ eine kontextfreie Grammatik und

$$U(G) = \{X \in V_N \mid X \Longrightarrow_G^* vXw, \ v, w \in V_T^*, \ vw \neq \varepsilon\}.$$

G heißt *selbstzyklisch*, wenn die folgenden Eigenschaften erfüllt sind:

(a) Für jedes $X \in V_N$ existiert ein Wort $w \in V_T^*$ mit $X \Longrightarrow^* w$.
(b) Für jedes $X \in V_N - \{X_0\}$ existieren Wörter $P, Q \in (V_N \cup V_T)^*$ mit $X_0 \Longrightarrow^* PXQ$.
(c) Entweder gilt $X_0 \in U(G)$, oder X_0 tritt in jeder Ableitung für ein beliebiges Wort $w \in L(G)$ genau einmal auf.
(d) $V_N - \{X_0\} \subset U(G)$. □

Satz 12.2 Für jede eindeutige kontextfreie Grammatik $G = (V_N, V_T, X_0, F)$ mit $L(G) \neq \emptyset$ existiert eine äquivalente eindeutige selbstzyklische Grammatik G'.

Beweis: Ohne Beschränkung der Allgemeinheit erfülle G die Bedingungen (a) und (b) aus Definition 12.2. Anderenfalls werden alle Nichtterminalzeichen aus G, für die (a) oder (b) nicht gilt, sowie alle Produktionen, die eines dieser Nichtterminalzeichen enthalten, aus G entfernt. Die so erzeugte Grammatik G' erzeugt offensichtlich dieselbe

Sprache wie G, und wegen $L(G) \neq \emptyset$ werden nicht alle Symbole aus V_N entfernt. Falls G' mehrdeutig wäre, so müßte auch G mehrdeutig sein, was jedoch der Voraussetzung widerspricht.

Wir nehmen nun an, daß G die Bedingung (c) nicht erfüllt. Wegen (a) gilt $X_0 \Longrightarrow^+ X_0$. Für jedes $w \in L(G)$ existieren dann zwei verschiedene Linksableitungen

$$X_0 \Longrightarrow^* w \text{ und } X_0 \Longrightarrow^+ X_0 \Longrightarrow^* w,$$

was wieder der vorausgesetzten Eindeutigkeit von G widerspricht. Also wird die Bedingung (c) von G erfüllt.

Wir konstruieren eine eindeutige Grammatik $G' = (V'_N, V_T, X_0, F')$, die alle Bedingungen (a) bis (d) erfüllt. Es sei

$$V_N = \{X_0, X_1, \ldots, X_k\}, \quad k \geq 0.$$

Falls $k = 0$ gilt, sind bei Wahl von $G' = G$ trivialerweise alle Bedingungen erfüllt. Anderenfalls wird eine Folge äquivalenter Grammatiken

$$G_0 = G, G_1, \ldots, G_k = G'$$

konstruiert. Für ein festes j, $1 \leq j \leq k$, sei

$$G_{j-1} = (V_N^{j-1}, V_T, X_0, F^{j-1})$$

bereits so definiert, daß G_{j-1} eindeutig ist und (a) bis (c) erfüllt sind. Dies ist bereits für $G_0 = G$ gezeigt worden. Falls $X_j \in U(G)$ gilt, setzen wir $G_j = G_{j-1}$. Anderenfalls seien

$$(*) \qquad X_j \to P_1, \ldots, X_j \to P_r$$

alle Produktionen in F^{j-1} mit linker Seite X_j, wobei $r \geq 1$ wegen (a) gilt. Dabei ist X_j kein Teilwort von $P_1 \ldots P_r$. Dies sehen wir wie folgt ein.

(1) Falls ein i, $1 \leq i \leq r$, existiert mit $X_j = P_i$, so erhalten wir wegen (b), der Produktion $X_j \to X_j$ und (a) die Ableitung

$$X_0 \Longrightarrow^* PX_jQ \Longrightarrow PX_jQ \Longrightarrow^* w \in V_T^*,$$

was der Eindeutigkeit von G_{j-1} widerspricht.

(2) Falls ein i, $1 \leq i \leq r$, und Wörter $P, Q \in (V_N \cup V_T)^*$ existieren mit $PQ \neq \varepsilon$ und $PX_jQ = P_i$, so folgt mit Hilfe von (a)

$$X_j \Longrightarrow P_i = PX_jQ \Longrightarrow^* uX_jw \text{ mit } u, w \in V_T^*.$$

Dabei gilt $uw \neq \varepsilon$, da sonst $X_j \Longrightarrow^+ X_j$ im Widerspruch zur Eindeutigkeit von G_{j-1} gelten würde. Es folgt also $X_j \in U(G_{j-1})$, was der Annahme $X_j \notin U(G_{j-1})$ widerspricht.

Zur Definition von G_j setzen wir im betrachteten Fall $X_j \notin U(G_{j-1})$ zunächst $V_N^j = V_N^{j-1} - \{X_j\}$. Weiter entsteht die Produktionenmenge F^j aus F^{j-1}, indem die Produktionen $(*)$ entfernt werden und X_j in den übrigen Produktionen durch jedes der Wörter $P_1, \ldots, P_r$ ersetzt wird. Diese Wörter enthalten, wie eben gezeigt wurde, nicht das Symbol X_j. Offenbar gilt $L(G_j) = L(G_{j-1})$. Da aus der Mehrdeutigkeit von G_j auch die von G_{j-1} folgen würde, ist G_j eindeutig. Die Eigenschaften (a) bis (c) werden bei dieser Konstruktion von G_{j-1} auf G_j übertragen.

Insgesamt sehen wir, daß für $j = 1, \ldots, k$ das Alphabet V_N^j das Zeichen $X_i \in V_N$, $1 \leq i \leq j$, genau dann enthält, wenn $X_i \in U(G) = U(G_j)$ gilt. Damit erfüllt speziell $G_k = G'$ alle Bedingungen (a) bis (d). $\square$

Satz 12.3 Die Sprache

$$L = \{a^m b^n c^n \mid m, n \geq 1\} \cup \{a^m b^m c^n \mid m, n \geq 1\}$$

ist echt mehrdeutig.

Beweis: Wir nehmen an, daß L eindeutig ist. Nach Satz 12.2 wird L somit von einer eindeutigen selbstzyklischen Grammatik $G = (V_N, V_T, X_0, F)$ erzeugt. Für G weisen wir zunächst drei Eigenschaften (I), (II) und (III) nach.

(I) Jedes $X \in V_N$ mit $X \neq X_0$ ist von genau einem der drei folgenden Typen.

Typ 1*:* Es existieren $m \in \mathbb{N}$ und $v, w \in V_T^*$, so daß $X \Longrightarrow^* vXw$ mit entweder $vw = a^m$ oder $vw = c^m$ gilt.

Typ 2*:* Es existiert $m \in \mathbb{N}$, so daß $X \Longrightarrow^* a^m X b^m$ gilt.

Typ 3*:* Es existiert $m \in \mathbb{N}$, so daß $X \Longrightarrow^* b^m X c^m$ gilt.

Andere Ableitungen der Form $X \Longrightarrow^* vXw$ mit $v, w \in V_T^*$ existieren nicht.

Wir zeigen zunächst, daß jede Ableitung $X \Longrightarrow^* vXw$ mit $v, w \in V_T^*$ von einer der Formen in den drei Typen ist. Damit ist auch jedes X von mindestens einem der drei Typen. Da G selbstzyklisch ist, existieren Wörter $v_1, w_1 \in V_T^*$ mit $X \Longrightarrow^* v_1 X w_1$. Durch Iteration dieser Ableitung ergibt sich $X \Longrightarrow^* v_1^2 X w_1^2$. Da die Zeichen a, b und c nur in alphabetischer Reihenfolge in den Wörtern aus L auftreten, sind v_1 und w_1 jeweils Wörter aus a^*, b^* oder c^*. Im folgenden zeigen wir, daß v_1 und w_1 nur so gewählt werden können, wie es den Typen 1, 2 oder 3 entspricht. Wegen Definition 12.2(b) und (c) folgt die Existenz einer Ableitung

$$X_0 \Longrightarrow^* R_1 X R_2 \Longrightarrow^* a^r b^s c^t$$

mit $r, s, t \in \mathbb{N}$, wobei $r = s$ oder $s = t$ gilt. Ausgehend von dem Zeichen X dieser Ableitung, kann mehrfach die Ableitung $X \Longrightarrow^* v_1 X w_1$ eingeschoben werden. Für jedes $q \in \mathbb{N}_0$ erhält man so eine Ableitung

$$(*) \qquad X_0 \Longrightarrow^* R_1 X R_2 \Longrightarrow^* R_1 v_1^q X w_1^q R_2 \Longrightarrow^* a^{r+qr'} b^{s+qs'} c^{t+qt'}$$

mit geeigneten $r', s', t' \in \mathbb{N}_0$, die durch v_1 und w_1 bestimmt sind. Wir führen vier Fallunterscheidungen durch.

(1) Es sei $v_1 = a^m$ mit $m > 0$.
Ist $w_1 = c^n$ mit $n > 0$, so können wir wegen $(*)$ das Wort $a^{r+qm}b^s c^{t+qn} \in L$ für alle $q \in \mathbb{N}_0$ ableiten. Für ein q mit $r + qm > s$ und $t + qn > s$ steht dies im Widerspruch zur Gestalt der Wörter von L. Ist $w_1 = b^n$ mit $m \neq n \geq 1$, so erhalten wir $a^{r+qm}b^{s+qn}c^t \in L$ für alle $q \in \mathbb{N}_0$ und damit erneut einen Widerspruch. Es gilt also $w_1 = b^m$ oder $w_1 = a^n$ für ein $n > 0$, d.h., X ist vom Typ 1 oder 2.

(2) Es sei $v_1 = b^m$ mit $m > 0$.
Für $w_1 = a^n$ mit $n > 0$ ergibt sich ein Widerspruch zur Reihenfolge der Symbole a, b und c in den Wörtern von L. Für $w_1 = b^n$ mit $n \geq 0$ können wir mit $(*)$ Wörter $a^r b^{s+q(m+n)} c^t \in L$ für alle $q \geq 0$ ableiten. Bei entsprechend großem q ergibt sich sofort ein Widerspruch. Für $w_1 = c^n$ mit $n \neq m$ ergibt sich schließlich $a^r b^{s+qm} c^{t+qn} \in L$ für alle $q \in \mathbb{N}_0$ und damit ebenfalls ein Widerspruch. Es bleibt nur der Fall $w_1 = c^m$ übrig, d.h., X ist vom Typ 3.

(3) Es sei $v_1 = c^m$ mit $m > 0$.
Wegen der Reihenfolge der Symbole a, b und c in den Wörtern von L folgt $w_1 = c^n$ mit $n \geq 0$, d.h., X ist vom Typ 1.

(4) Es sei $v_1 = \varepsilon$.
Wegen $v_1 w_1 \neq \varepsilon$ ist w_1 eine positive Potenz eines der Zeichen a, b oder c. Falls $w_1 = b^n$ mit $n > 0$ gilt, folgt $a^r b^{s+qn} c^t \in L$ für jedes $q \in \mathbb{N}_0$ und somit ein Widerspruch. Also ist X vom Typ 1.

Damit ist gezeigt, daß X von mindestens einem der Typen 1, 2 oder 3 ist. Wir beweisen, daß X von genau einem dieser Typen ist. Es sei X vom Typ 1 und 2. Dann gilt

$$X \Longrightarrow^* vXw \text{ und } X \Longrightarrow^* a^n X b^n$$

mit $vw = x^m$, $x = a$ oder $x = c$ und $m, n \in \mathbb{N}$. Die Kombination beider Ableitungen liefert

$$(**) \qquad X \Longrightarrow^* a^n x^{m_1} X x^{m_2} b^n \text{ mit } m_1 + m_2 = m.$$

Da diese Ableitung iteriert werden kann und wegen der Gestalt von L jedes der Wörter $a^n x^{m_1}$ und $x^{m_2} b^n$ eine Potenz eines der Zeichen a, b, oder c ist, folgt $x = a$ und $m_2 = 0$. Wegen $v_1 = a^n x^{m_1} = a^{n+m_1}$ kann das Symbol X in der Ableitung $(**)$ in bezug auf diese Ableitung nur vom Typ 1 oder Typ 2 sein. Daher folgt $w_1 = b^{n+m_1}$ oder $w_1 = a^\nu$ für $\nu \geq 0$, wegen $m_1 > 0$ in jedem Fall also $w_1 \neq b^n = x^{m_2} b^n$ und somit ein Widerspruch. Folglich ist X nicht gleichzeitig vom Typ 1 und 2.
Analog kann gefolgert werden, daß X nicht vom Typ 1 und 3 ist. Schließlich sei X vom Typ 2 und 3. Dann ergibt sich

$$X \Longrightarrow^* a^m b^n X c^n b^m \text{ mit } m, n \geq 1.$$

Dies ist ein Widerspruch zur Gestalt von L.

(II) X_0 kommt in jeder Ableitung gemäß G eines jeden Wortes aus L genau einmal vor.

Wir nehmen das Gegenteil an. Definition 12.2(c) liefert

$$X_0 \Longrightarrow^* vX_0w, \quad vw \neq \varepsilon, \quad v, w \in V_T^*.$$

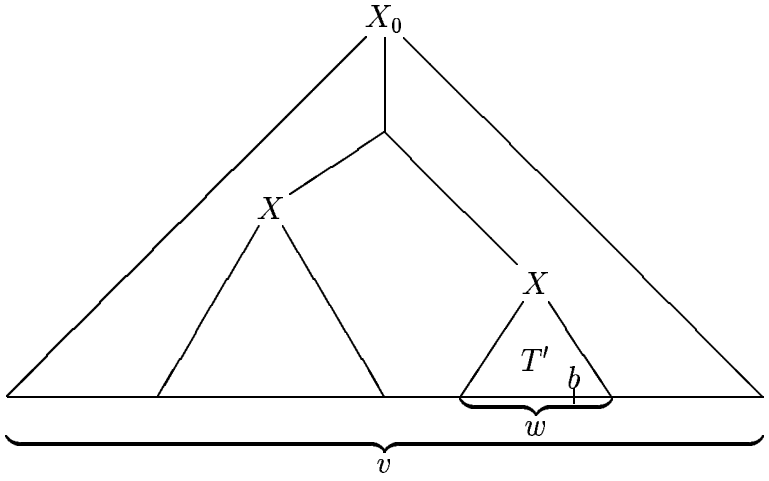
Somit sind die Wörter v und w Potenzen eines der Zeichen a, b oder c. Da alle Wörter in L mit a beginnen und mit c enden, existieren nur die folgenden drei Möglichkeiten:

$$\begin{array}{llll} (1) & v = a^m, & m \geq 1, & w = \varepsilon, \\ (2) & v = \varepsilon, & & w = c^m, \quad m \geq 1, \\ (3) & v = a^m, & m \geq 1, & w = c^n, \quad n \geq 1. \end{array}$$

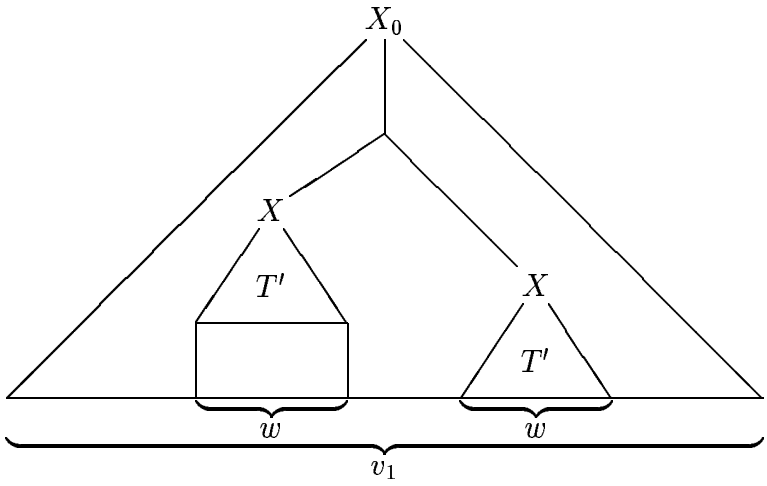
Die Ableitung $X_0 \Longrightarrow^* vX_0w$ läßt sich für jedes Wort $u \in L$ durch $vX_0w \Longrightarrow^* vuw \in L$ fortsetzen. Alle drei Fälle führen aufgrund der Gestalt von L zu einem Widerspruch. Im Fall (1) folgt wegen $abc^2 \in L$ auch $a^{m+1}bc^2 \in L$, im Fall (2) wissen wir, daß wegen $a^2bc \in L$ auch $a^2bc^{m+1} \in L$ gilt, und im Fall (3) erhalten wir wegen $abc \in L$ das Wort $a^{m+1}bc^{n+1} \in L$.

(III) Es existiert eine Zahl $\nu \in \mathbb{N}$, so daß jedes Wort $v \in L$, das eine Ableitung ohne Nichtterminalzeichen vom Typ 2 und Typ 3 besitzt, weniger als ν Vorkommen des Zeichen b enthält.

Zum Beweis von (III) definieren wir zunächst u_1 als die Anzahl der Nichtterminalzeichen vom Typ 1 und u_2 als die maximale Anzahl der auf der rechten Seite einer Produktion auftretenden Symbole. Wegen (II) kommt X_0 nur als Startzeichen in jeder Ableitung vor und kann direkt höchstens u_2 Vorkommen von b erzeugen. Wir müssen in den betrachteten Ableitungen solche Nichtterminalzeichen betrachten, die Symbole b erzeugen. Wir nehmen zunächst an, daß in einer Ableitung, die keine Nichtterminalzeichen vom Typ 2 und Typ 3 enthält, das Zeichen X vom Typ 1 in zwei verschiedenen Pfaden des entsprechenden Ableitungsbaums auftritt und mindestens eines der X (direkt oder indirekt) ein Vorkommen von b erzeugt. Dies wird durch die folgende Skizze verdeutlicht.



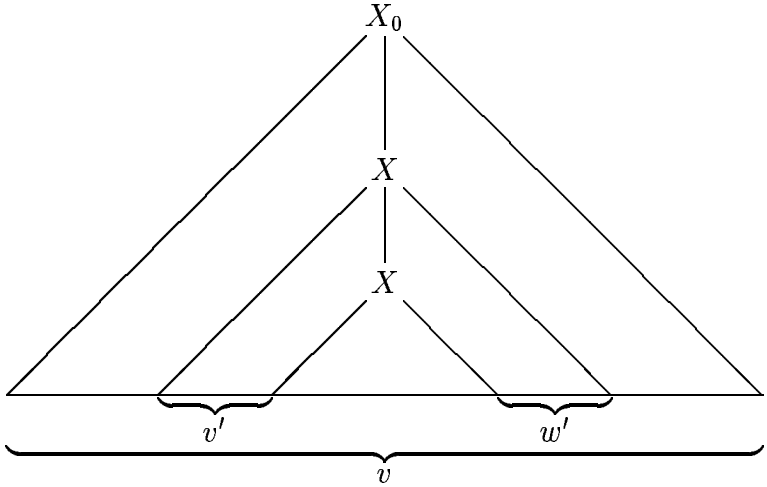
Zur Ableitung $X \Longrightarrow^* w$, bei der b in w vorkommt, gehöre ein entsprechender Teilbaum T'. Der Ableitungsbaum wird nun so abgeändert, daß bei beiden X der Teilbaum T' eingesetzt wird, was im allgemeinen eine Änderung des erzeugten Wortes von v in v_1 bedeutet.



Da X vom Typ 1 ist, existiert eine Ableitung $X \Longrightarrow^* v'Xw'$ mit $v'w' = a^m$ oder $v'w' = c^m$ und $m \in \mathbb{N}$. Wenn der Teilbaum T' durch einen Ableitungsbaum der Ableitung

$$X \Longrightarrow^* v'Xw' \Longrightarrow^* v'ww'$$

ersetzt wird, entsteht ein Wort $v_2 \in L$, das ein Teilwort der Form aw_1bw_2a bzw. cw_1bw_2c enthält, was jedoch in beiden Fällen der Gestalt von L widerspricht.
Als nächstes nehmen wir an, daß ein Zeichen X vom Typ 1 zweimal auf demselben Pfad vorkommt. Dann besitzt der Ableitungsbaum von v die Gestalt



mit $v'w' = a^m$ oder $v'w' = c^m$ für ein $m \in \mathbb{N}$. Folglich können höchstens aus dem X, das am weitesten von X_0 entfernt ist, Symbole b erzeugt werden, ohne einen Weg über andere X dafür zu benutzen. Insgesamt können daher auch höchstens aus X_0 und aus je einem Vorkommen eines jeden Zeichens vom Typ 1 Vorkommen von b erzeugt werden. Bei Wahl von $\nu = (u_1 + 1) \cdot u_2$ ist dann die Aussage (III)

erfüllt.

Für jedes Zeichen X vom Typ 2 oder 3 legen wir eine natürliche Zahl $m(X)$ gemäß der Definition dieser beiden Typen fest. Es sei u ein gemeinsames Vielfaches aller $m(X)$, und wir wählen ein $p > \nu$, das ein Vielfaches von u ist. Wir betrachten das Wort $r_1 = a^p b^p c^{2p} \in L$. Falls ein X vom Typ 3 und Wörter R' und R'' existieren, so daß

$$X_0 \Longrightarrow^* R'XR'' \Longrightarrow^* r_1$$

gilt, dann können wir wegen $m(X) \mid p$ die Ableitung

$$X_0 \Longrightarrow^* R'XR'' \Longrightarrow^* R'b^pXc^pR'' \Longrightarrow^* a^p b^{2p} c^{3p} \notin L$$

bilden. Wegen dieses Widerspruchs tritt in keiner Ableitung von r_1 ein Zeichen vom Typ 3 auf. Wegen $p > \nu$ und der Aussage (III) folgt, daß r_1 für geeignete Wörter R' und R'' und ein X vom Typ 2 durch eine Ableitung

$$X_0 \Longrightarrow^* R'XR'' \Longrightarrow^* r_1$$

erzeugt wird. Da p ein Vielfaches von $m(X)$ ist, erhalten wir für dieses X vom Typ 2 eine Ableitung $X \Longrightarrow^* a^pXb^p$. Wegen (I) ist das Symbol X nicht vom Typ 3. In der Ableitung $X \Longrightarrow^* a^pXb^p$ kommt auch kein weiteres Zeichen vom Typ 3 vor, da anderenfalls das Symbol c in a^pXb^p enthalten wäre. Somit befindet sich in der Ableitung

$$\text{(a)} \qquad X_0 \Longrightarrow^* R'XR'' \Longrightarrow^* R'a^pXb^pR'' \Longrightarrow^* a^{2p}b^{2p}c^{2p}$$

das Zeichen X vom Typ 2, jedoch kein Zeichen vom Typ 3. Analog erhalten wir mit $r_2 = a^{2p}b^pc^p \in L$ eine Ableitung

$$\text{(b)} \qquad X_0 \Longrightarrow^* R_3X_1R_4 \Longrightarrow^* R_3b^pX_1c^pR_4 \Longrightarrow^* a^{2p}b^{2p}c^{2p},$$

die ein $X_1 \in V_N$ vom Typ 3, jedoch kein Zeichen vom Typ 2 enthält. Wir schließen, daß das Wort $a^{2p}b^{2p}c^{2p}$ zwei verschiedene Linksableitungen (a) und (b) besitzt. Somit ist G mehrdeutig, was ein Widerspruch zu unserer ursprünglichen Annahme ist. □

Kapitel 13

Lindenmayersysteme

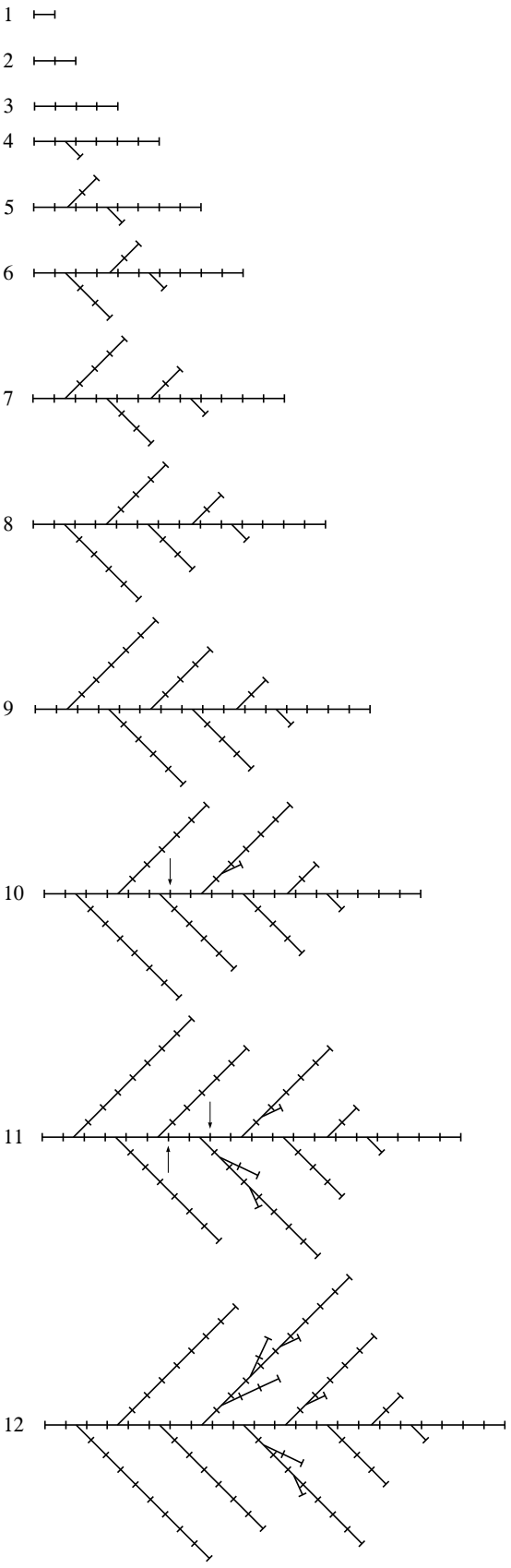
Lindenmayersysteme wurden von *Aristid Lindenmayer* in den 60-er Jahren eingeführt. Lindenmayer, der Biologe war, hat diese Systeme zunächst zur Beschreibung biologischer Phänomene benutzt. Sehr bald wurden *Lindenmayer* und andere auf den Zusammenhang mit der Theorie der formalen Sprachen aufmerksam. Die Theorie entwickelte sich in der Folgezeit immer mehr von ihren biologischen Ursprüngen fort und wurde ein umfangreiches eigenständiges Gebiet innerhalb der Theorie der formalen Sprachen. Die wichtigsten Aspekte dieser Theorie wollen wir in diesem Kapitel ausführlich darstellen.

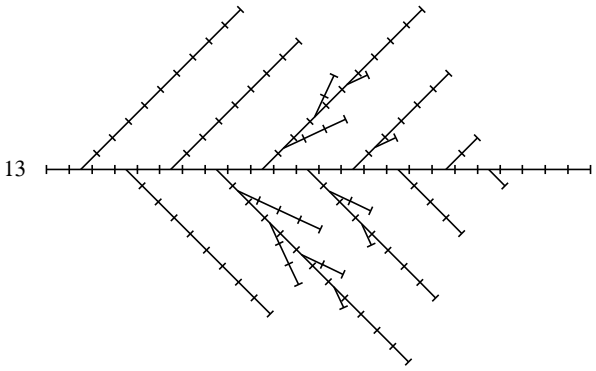
Zur Ergänzung verweisen wir auf die Bücher von *Herman* und *Rozenberg* [12] sowie *Rozenberg* und *Salomaa* [22]. Schöne graphische Interpretationen von Lindenmayersystemen finden sich bei *Prusinkiewicz* und *Hanan* [19] bzw. *Prusinkiewicz* und *Lindenmayer* [20].

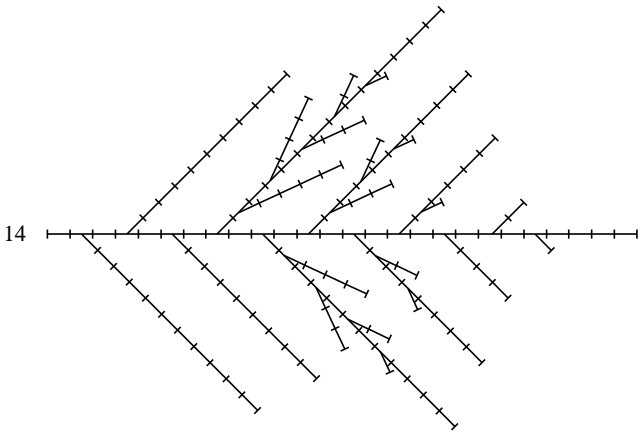
13.1 0L-Systeme

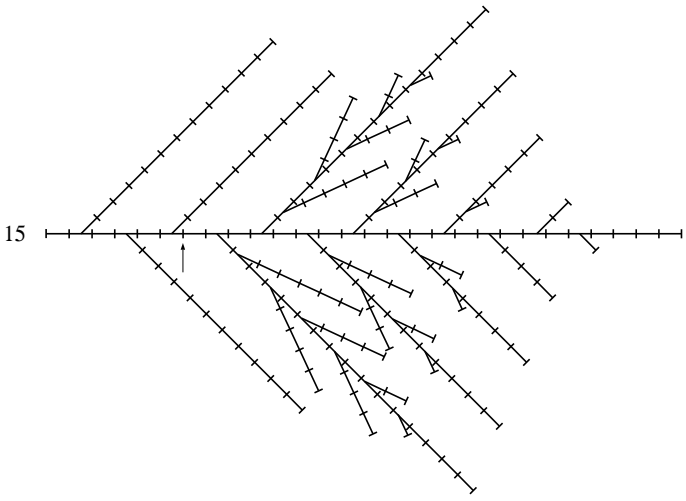
In vielen Wachstumsprozessen lebender Organismen wie z.B. bei Pflanzen bemerkt man häufig ein regelmäßiges wiederholtes Auftreten von gewissen mehrzelligen Strukturen. In den einfachsten Fällen wird dieselbe Struktur regelmäßig längs einer Achse wiederholt, wie z.B. Blätter längs eines Stammes. In komplizierteren Fällen wird die ganze Struktur eines Entwicklungsstadiums wiederholt, und zwar als Teil des Organismus zu einem späteren Zeitpunkt. So gibt es zusammengesetzte Organismen, etwa bei Blättern oder anderen zusammengesetzten Verzweigungsstrukturen. Im Fall der Blätter können einige Lappen (oder Blättchen), die Teil des Blattes zu einem bestimmten Zeitpunkt sind, denselben Umriß wie das ganze Blatt zu einem vorhergehenden Zeitpunkt haben. Dies kann man bei Pflanzen erkennen, bei denen einige Blätter ihre Entwicklung früher als andere beenden: die kleineren Blätter sind identisch zu Teilen der größeren Blätter.

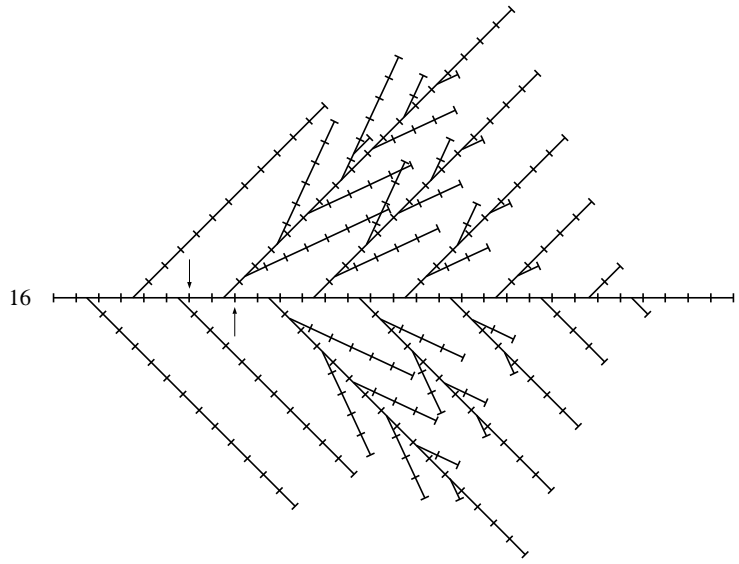
Um solche biologischen Phänomene zu beschreiben, hat *Lindenmayer* die nach ihm benannten Lindenmayersysteme eingeführt. Speziell konnte er die Entwicklungsstufen gewisser Faserpflanzen (Algen) mit Hilfe sogenannter 0L-Systeme kennzeichnen. Als Beispiel betrachten wir die ersten 16 Stadien einer solchen Pflanze:


1
2
3
4
5
6
7
8
9
10
11
12

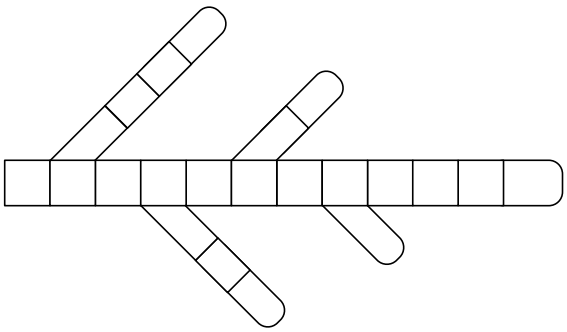

13


14


15



Eine realistischere Darstellung von z.B. Stadium 7, die auch die Ausdehnung der Zellen berücksichtigt, ist durch das folgende Bild gegeben:



Wir wollen die Entwicklung beschreiben, indem wir alle Stadien beschreiben. Vom Stadium 6 an zerfällt der Organismus in zwei Teile, und zwar zum einen in die ersten sechs Zellen (von links), den *basalen* Teil. Dabei trägt jede zweite Zelle eine nicht verzweigende Faser, die sich jeweils linear von Schritt zu Schritt um eine Zelle verlängert. Die Länge dieser Fasern im Stadium 6 sind (von links nach rechts) 3, 2 und 1.

Der Rest des Organismus ist der *apikale* Teil. Im Stadium 6 sind dies vier Zellen ohne Zweige. Bei jedem weiteren Schritt kommt es zu einer Wiederholung des vorhergehenden apikalen Teils mit zwei neuen Zellen unmittelbar rechts vom basalen Teil. Die zweite dieser Zellen trägt einen Zweig, der zum gesamten Organismus sechs Stadien zuvor identisch ist. Diese beiden Zellen sind im Stadium 16 durch die Pfeile eingeschlossen, wobei der apikale Teil des Stadiums 15 beim eingezeichneten Pfeil beginnt.

Wenn die ersten sechs Stadien bekannt sind, dann zeigt diese Beschreibung, wie sich der Organismus entwickelt. Da es jedoch zu Ungenauigkeiten der Sprache kommen kann, ist eine formale Beschreibung vorzuziehen. Jede Zelle wird durch das Symbol c dargestellt, wobei die verschiedenen Zustände einer Zelle zunächst nicht unterschieden

werden. Der Beginn eines Zweiges wird durch „(“, sein Ende durch „)“ symbolisiert. Zur Unterscheidung der Seiten des Zweiges können gegebenenfalls verschiedene Arten von Klammern verwendet werden. Das obige Beispiel kann jetzt durch eine Folge von Wörtern über $\{c\}$ dargestellt werden:

1	c
2	cc
3	$cccc$
4	$cc(c)cccc$
5	$cc(cc)cc(c)cccc$
6	$cc(ccc)cc(cc)cc(c)cccc$
7	$cc(cccc)cc(ccc)cc(cc)cc(c)cccc$
⋮	⋮

Die Entwicklung einzelner Zellen kann auf diese Weise nicht ausgedrückt werden. Es werden daher die Symbole $0, \ldots, 9$ eingeführt, wodurch Zellen in zehn verschiedenen Zuständen beschrieben werden können. Für jeden Zustand wird wie folgt eine Entwicklungsregel (Produktion) eingeführt:

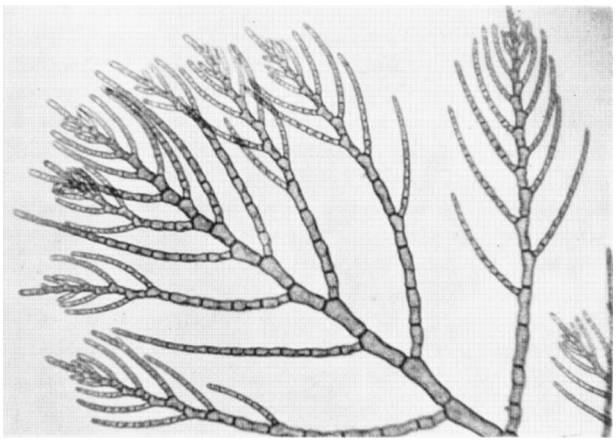
$0 \to 10$	(Zelle im Zustand 0 teilt sich in zwei Zellen im Zustand 1 und 0)
$1 \to 32$	
$2 \to 3(4)$	(Übergang in eine Zelle im Zustand 3 mit einem Zweig aus einer Zelle im Zustand 4)
$3 \to 3$	
$4 \to 56$	
$5 \to 37$	
$6 \to 58$	
$7 \to 3(9)$	
$8 \to 50$	
$9 \to 39$	
$(\to ($	
$) \to)$	

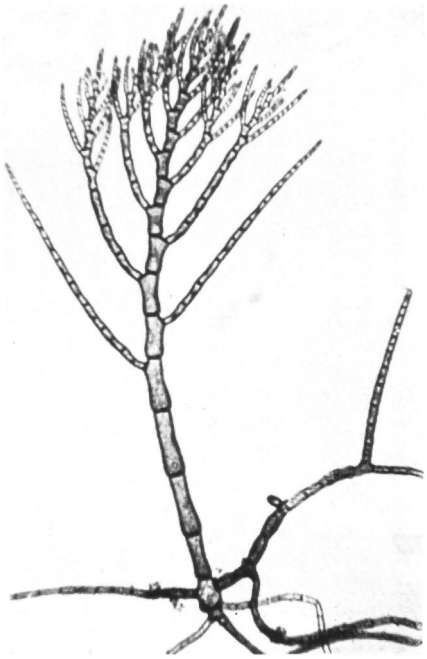
Wir nehmen an, daß zu Beginn der Entwicklung eine Zelle im Zustand 4 vorliegt. Entwickeln sich alle Zellen unabhängig voneinander, jedoch in synchronisierter Weise, dann ergibt sich die folgende Darstellung der weiteren Entwicklung:

1	4
2	56
3	3758
4	33(9)3750
5	33(39)33(9)3710
6	33(339)33(39)33(9)3210
7	33(3339)33(339)33(39)33(4)3210
⋮	⋮

Bei Ersetzung der Ziffern durch das Symbol c erhalten wir offenbar die zuvor angegebene Darstellung.

Die folgenden Bilder zeigen, daß dieses Beispiel sehr gut eine gewisse Sorte von Algen beschreibt.





Junger Ableger (Regenerat) von Callithamonion roseum Harvey, aus: Eva Konrad-Hawkins: Developmental Studies on Regenerates of Callithamonion roseum Harvey, in K. Höfler, K.R. Porter (Hrgs): Protoplasma, S.48 u. 45, Bd.58, Springer, Wien 1964.

Vor ihrer formalen Definition wollen wir anhand des Beispiels schon einige Begriffe für Lindenmayersysteme einführen. So ist durch $\{0,1,\ldots,9,),(\}$ das *Alphabet* des Systems gegeben, die Entwicklungsregeln bestimmen die *Produktionen*. 4 ist das *Axiom* des speziellen Systems, mit dem die Entwicklung ihren Anfang nimmt. Da keine Wechselwirkung zwischen den Zellen besteht, wird das System 0L-*System* genannt (bei Ersetzung eines Symbols wird kein Kontext, also 0-Kontext, berücksichtigt). Unser System ist auch *deterministisch*, da keine Wahl bei der Anwendung der Produktionen besteht. Außerdem ist es *propagierend* (fortpflanzend), da keine Zelle gelöscht wird.

Definition 13.1 Ein geordnetes Tripel $G = (\Sigma, h, \omega)$ heißt 0L-*System*, wenn folgende Eigenschaften gelten:

(a) Σ ist ein Alphabet,

(b) $h : \Sigma \to \mathfrak{P}(\Sigma^*)$ ist eine nichtleere, endliche Substitution,

(c) und es gilt $\omega \in \Sigma^*$ (das *Axiom* von G).

$L(G) = \bigcup_{i=0}^{\infty} h^i(\omega)$ heißt die *von G erzeugte* 0L-*Sprache*. □

Beispiel 13.1 Das 0L-System $G = (\{a\}, h, a^2)$ mit $h(a) = \{a, a^2\}$ erzeugt die Sprache $L(G) = \{a^n \mid n \geq 2\}$. □

Beispiel 13.2 Das 0L-System

$$G = (\{a, b\}, h, ab) \text{ mit } h(a) = \{(ab)^2\} \text{ und } h(b) = \{\varepsilon\}$$

erzeugt die Sprache $L(G) = \{(ab)^{2^n} \mid n \geq 0\}$. Wir erhalten also mit kontextfreien Produktionen eine nicht kontextfreie Sprache. □

Anstelle von $w \in h(a)$ verwenden wir oft die *Produktionsschreibweise* $a \to_h w$, wobei, falls keine Mißverständnisse auftreten, h auch weggelassen werden darf. Wir sprechen aber auch von der Produktion $w \in h(a)$.

Definition 13.2 Es sei $G = (\Sigma, h, \omega)$ ein 0L-System, $w \in \Sigma^*$ und $i \in \mathbb{N}$. Wir definieren

$$L^i(G, w) = \{v \in \Sigma^* \mid v \in h^i(w)\} \quad \text{und} \quad L^0(G, w) = \{w\}. \quad \square$$

Unmittelbar folgt $L(G) = \bigcup_{i=0}^{\infty} L^i(G, \omega)$. Wird in Definition 13.2 speziell $w = \omega$ gewählt, so wird zur Abkürzung die Schreibweise $L^i(G) = L^i(G, \omega)$ verwendet.

Definition 13.3 (a) Es sei G ein 0L-System. w_1 leitet w_2 *direkt* (in G) ab (in Zeichen $w_1 \Longrightarrow_G w_2$), wenn $w_2 \in L^1(G, w_1)$ gilt.

(b) w_1 leitet w_2 *in n Schritten* (in G) ab (in Zeichen $w_1 \Longrightarrow_G^n w_2$), wenn $w_2 \in L^n(G, w_1)$ gilt.

(c) w_1 leitet w_2 *echt* (in G) ab (in Zeichen $w_1 \Longrightarrow_G^+ w_2$), wenn $w_2 \in L^n(G, w_1)$ für ein $n \in \mathbb{N}$ gilt.

(d) w_1 leitet w_2 (in G) ab (in Zeichen $w_1 \Longrightarrow_G^* w_2$), wenn $w_2 \in L^n(G, w_1)$ für ein $n \in \mathbb{N}_0$ gilt. □

Falls keine Verwechselung möglich ist, wird auch $\Longrightarrow$, $\Longrightarrow^n$, $\Longrightarrow^+$ oder $\Longrightarrow^*$ verwendet. Die von G erzeugte Sprache kann jetzt durch

$$L(G) = \{w \mid \omega \Longrightarrow^* w\}$$

beschrieben werden.

Definition 13.4 Es sei $G = (\Sigma, h, \omega)$ ein 0L-System. G heißt *propagierend* oder P0L-*System*, wenn

$$\varepsilon \notin h(a)$$

für alle $a \in \Sigma$ gilt. G heißt *deterministisch* oder D0L-*System*, falls

$$\text{card}(h(a)) = 1$$

für alle $a \in \Sigma$ gilt. Ist G sowohl propagierend als auch deterministisch, so ist es ein PD0L-*System*. Entsprechend heißt die von G erzeugte Sprache P0L-, D0L- oder PD0L-*Sprache*. □

Bei einem D0L-System $G = (\Sigma, h, \omega)$ kann h als Homomorphismus aufgefaßt werden. Beispiel 13.1 ist ein nichtdeterministisches P0L-System, Beispiel 13.2 ein nichtpropagierendes D0L-System. Eine bekannte nicht kontextfreie Sprache, die durch ein 0L-System erzeugt wird, wird im nächsten Beispiel angegeben.

Beispiel 13.3 Das PD0L-System

$$G = (\{a\}, h, a) \quad \text{mit} \quad h(a) = a^2$$

erzeugt die Sprache

$$L(G) = \{a^{2^n} \mid n \geq 0\}. \quad \square$$

Beispiel 13.4 Das D0L-System

$$G = (\{a, b, c\}, h, a) \quad \text{mit} \quad h(a) = abcc,\ h(b) = bcc,\ h(c) = c$$

erzeugt die Sprache

$$L(G) = \{a, abcc, abccbcccc, abccbccccbcccccc, \ldots\}.$$

Wir vermuten, daß die Länge der Wörter die Quadrate der natürlichen Zahlen sind. In jedem Ableitungsschritt wird zusätzlich zu den vorhandenen Symbolen ein b aus dem einzigen Symbol a des Wortes erzeugt, außerdem liefern jedes a und b jeweils zwei weitere Vorkommen von c. Durch Induktion über $n \in \mathbb{N}$ beweisen wir, daß das n-te Wort $n-1$ Vorkommen von b, ein Vorkommen von a und $n^2 - n$ Vorkommen von c besitzt. Für $n = 1$ ist die Behauptung offenbar erfüllt. Falls die Behauptung für das n-te Wort erfüllt ist, dann enthält das $(n+1)$-te Wort aufgrund des oben beschriebenen Zuwachses $n - 1 + 1 = (n+1) - 1$ Vorkommen von b, eines von a und $n^2 - n + 2(n - 1 + 1) = n^2 + n = (n+1)^2 - (n+1)$ von c. Damit ist auch bewiesen, daß das $(n+1)$-te Wort aus $(n+1)^2$ Symbolen besteht. $\square$

Beispiel 13.5 Das D0L-System

$$G = (\{a, b, c, d\}, h, a) \quad \text{mit} \quad h(a) = abcd^5,\ h(b) = bcd^5,\ h(c) = cd^6,\ h(d) = d$$

erzeugt die Sprache

$$L(G) = \{a, abcd^5, abcd^5bcd^5cd^6d^5, \ldots\}.$$

Man kann zeigen, daß die Länge der Wörter die Kubikzahlen der natürlichen Zahlen darstellen. $\square$

Die Aussagen des folgenden Satzes sind offensichtlich erfüllt.

Satz 13.1 Es sei $G = (\Sigma, h, \omega)$ ein 0L-System.

(1) Für $n \in \mathbb{N}_0$ und $w_1, w_2, v_1, v_2 \in \Sigma^*$ gelte $w_1 \Longrightarrow^n v_1$ und $w_2 \Longrightarrow^n v_2$. Dann folgt $w_1 w_2 \Longrightarrow^n v_1 v_2$.

(2) Für $n \in \mathbb{N}_0$ und $w_1, w_2, z \in \Sigma^*$ gelte $w_1 w_2 \Longrightarrow^n z$. Dann existieren $v_1, v_2 \in \Sigma^*$ mit $v_1 v_2 = z$, $w_1 \Longrightarrow^n v_1$ und $w_2 \Longrightarrow^n v_2$.
(3) Für $m, n \in \mathbb{N}_0$ und $w, v, z \in \Sigma^*$ gelte $w \Longrightarrow^n v$ und $v \Longrightarrow^m z$. Dann folgt $w \Longrightarrow^{n+m} z$. □

Definition 13.5 Mit $\mathfrak{L}(0L)$ bezeichnen wir die Familie der Sprachen, die durch 0L-Systeme erzeugt werden. Entsprechend bezeichnen $\mathfrak{L}(D0L)$, $\mathfrak{L}(P0L)$ bzw. $\mathfrak{L}(PD0L)$ die Familie der Sprachen, die durch D0L-, P0L- bzw. PD0L-Systeme erzeugt werden. □

Satz 13.2 Die Familie $\mathfrak{L}(0L)$ ist eine Anti-AFL.

Beweis: Wir müssen zeigen, daß $\mathfrak{L}(0L)$ unter keiner der AFL-Operationen aus Definition 10.9 abgeschlossen ist.

(a) Wir zeigen, daß $\mathfrak{L}(0L)$ nicht unter Vereinigung abgeschlossen ist. Offenbar gilt $\{a\}, \{a^3\} \in \mathfrak{L}(0L)$. Wäre $\{a\} \cup \{a^3\} = \{a, a^3\} \in \mathfrak{L}(0L)$, so würde ein 0L-System $G = (\{a\}, h, \omega)$ mit $L(G) = \{a, a^3\}$ existieren. Für $\omega \in L(G)$ gibt es nun zwei Möglichkeiten.
   (1) Ist $\omega = a$, so muß $a \Longrightarrow a^3$ und damit $a^3 \in h(a)$ gelten. Daraus folgt jedoch $a^3 \Longrightarrow a^9$, was wegen $a^9 \notin L(G)$ ein Widerspruch ist.
   (2) Ist $\omega = a^3$, so muß $a^3 \Longrightarrow a$ gelten. Dann folgt $\varepsilon, a \in h(a)$. Damit erhalten wir den Widerspruch $a^3 \Longrightarrow a^2 \notin L(G)$.

(b) Der Nichtabschluß von $\mathfrak{L}(0L)$ unter ε-freier Iteration wird anhand der ε-freien Sprache
$$L = \{a^2\} \cup \{b^{2^n} \mid n \geq 2\} \in \mathfrak{L}(0L)$$
gezeigt, die offenbar von dem D0L-System
$$G = (\{a, b\}, h, a^2) \quad \text{mit} \quad h(a) = h(b) = b^2$$
erzeugt wird. Wir nehmen an, daß L^+ eine Sprache aus $\mathfrak{L}(0L)$ ist. Dann existiert ein L^+ erzeugendes 0L-System $G' = (\{a, b\}, h, \omega)$. Da L^+ ε-frei ist, muß $\varepsilon \notin h(a)$ und $\varepsilon \notin h(b)$ gelten. Folglich ist G' ein P0L-System. Die kürzesten Wörter von L^+ sind
$$a^2,\ a^4,\ b^4,\ a^6,\ a^2b^4 \text{ und } b^4a^2.$$
Andere Wörter der Sprache sind länger. Da G' propagierend ist, muß das kürzeste Wort das Axiom sein, also $\omega = a^2$. Wir schließen $b \notin h(a)$, da anderenfalls die Ableitung $\omega \Longrightarrow_{G'} b^2 \notin L^+$ zum Widerspruch führt. Damit trotzdem Wörter erzeugt werden können, die nur aus Symbolen b bestehen, muß es ein $i > 1$ mit einer Produktion $b^i \in h(a)$ geben. Daraus folgt $a \notin h(a)$ und $a^3 \notin h(a)$, da anderenfalls die Ableitungen $a^2 \Longrightarrow_{G'} ab^i \notin L^+$ bzw. $a^2 \Longrightarrow_{G'} a^3b^i \notin L^+$ zum Widerspruch führen. Schließlich nehmen wir $a^2 \in h(a)$ an. Wegen $a^2 \Longrightarrow^*_{G'} b^4 \in L^+$ existiert eine Produktion $b^i \in h(a)$ mit $i = 2$ oder $i = 3$. Dabei kommen $i = 0$ und $i = 1$ wegen der vorhergehenden Überlegungen nicht in Frage, und $i = 4$ ist in dieser Ableitung wegen $\varepsilon \notin h(a)$ nicht möglich. Mit $a^2, b^i \in h(a)$ ergibt sich

die Ableitung $a^2 \Longrightarrow_{G'} a^2b^i \notin L^+$ und damit ein Widerspruch. Zusammenfassend stellen wir fest, daß $\varepsilon, a, a^2, a^3, b \notin h(a)$ und $\varepsilon, b \notin h(b)$ gelten.
Es wird nun gezeigt, daß das Wort $a^2b^4 \in L^+$ nicht gemäß G' erzeugbar ist. Hierzu betrachten wir alle Möglichkeiten, a^2b^4 direkt aus einem anderen Wort der Sprache zu erzeugen. Dafür kommen nur die (oben aufgeführten) Wörter der Länge ≤ 6 in Frage.

(α) Aus $a^2 \Longrightarrow_{G'} a^2b^4$ und dem oben bewiesenen Ausschluß von möglichen Produktionen folgt, daß eine Zahl $i \in \{1, 2\}$ existiert mit $a^2b^i \in h(a)$ und $b^{4-i} \in h(a)$. Dann erhalten wir jedoch auch die Ableitung $a^2 \Longrightarrow_{G'} a^2b^ia^2b^i \notin L^+$.

(β) Die Ableitung $a^4 \Longrightarrow_{G'} a^2b^4$ führt analog zu Fall (α) zum Widerspruch.

(γ) Aus $b^4 \Longrightarrow_{G'} a^2b^4$ folgt, daß das erste, zweite, dritte bzw. vierte Vorkommen von b in b^4 durch aw_1, w_2, w_3 bzw. w_4b durch geeignete Wörter w_1, w_2, w_3 bzw. w_4 mit $|w_2|, |w_3| \geq 1$ ersetzt werden muß. Die entsprechenden Produktionen können auch in einer anderen Reihenfolge verwendet werden, man erhält so eine Ableitung $b^4 \Longrightarrow_{G'} w_2w_3aw_1w_4b = v$ mit einem Wort $v \in L^+$ der Länge 6. Das einzige Wort aus L^+ mit diesen Eigenschaften ist jedoch $v = a^2b^4$. Dies steht im Widerspruch dazu, daß w_2w_3a mit $|w_2w_3| \geq 2$ ein Präfix von v ist.

(δ) Die Ableitungen $a^6 \Longrightarrow_{G'} a^2b^4$ und $b^4a^2 \Longrightarrow_{G'} a^2b^4$ erfordern beide $b \in h(a)$, was jedoch oben bereits ausgeschlossen werden konnte.

Das Wort $a^2b^4 \in L^+$ ist also nicht gemäß G' erzeugbar.

(c) $\mathfrak{L}(0L)$ ist nicht abgeschlossen unter Durchschnitt mit regulären Sprachen, da $\{a^{2^n} \mid n \in \mathbb{N}_0\} \in \mathfrak{L}(0L)$ (siehe Beispiel 13.3) und $\{a, a^4\} \in \mathfrak{L}_3$ gilt, jedoch $\{a^{2^n} \mid n \in \mathbb{N}_0\} \cap \{a, a^4\} = \{a, a^4\}$ keine Sprache aus $\mathfrak{L}(0L)$ ist. Dies kann analog zum Beweis von $\{a, a^3\} \notin \mathfrak{L}(0L)$ in (a) gezeigt werden.

(d) Der Nichtabschluß von $\mathfrak{L}(0L)$ unter ε-freiem Homomorphismus wird anhand des ε-freien Homomorphismus φ auf $\{a\}^*$ mit $\varphi(a) = a^5$ gezeigt. Die Sprache $\{\varepsilon, a, a^2\}$ wird von dem 0L-System $(\{a\}, h', a^2)$ mit $h'(a) = \{\varepsilon, a\}$ erzeugt und gehört daher zu $\mathfrak{L}(0L)$. Falls nun

$$\varphi(\{\varepsilon, a, a^2\}) = \{\varepsilon, a^5, a^{10}\} \in \mathfrak{L}(0L)$$

ist, so wird diese Sprache von dem 0L-System $G = (\{a\}, h, \omega)$ erzeugt. Für ω gibt es nur zwei Möglichkeiten. Ist $\omega = a^5$, so existiert eine Ableitung $a^5 \Longrightarrow_G a^{10}$ und damit ein $i > 1$ mit $a^i \in h(a)$. Daraus folgt jedoch $a^{10} \Longrightarrow_G a^{10i}$, also ein Widerspruch. Ist $\omega = a^{10}$, so existiert die Ableitung $a^{10} \Longrightarrow_G a^5$, woraus, da $a^i \in h(a)$ mit $i > 1$ zum Widerspruch führt, $\varepsilon \in h(a)$ und $a \in h(a)$ geschlossen werden kann. Dann existiert jedoch auch die Ableitung $a^{10} \Longrightarrow_G a$, was ebenfalls ein Widerspruch ist.

(e) Der Nichtabschluß von $\mathfrak{L}(0L)$ unter inversem Homomorphismus kann z.B. anhand des Homomorphismus $h(a) = a^2$ und der 0L-Sprache $\{a\}$ gezeigt werden, da $h^{-1}(\{a\}) = \emptyset \notin \mathfrak{L}(0L)$ gilt. Es existieren jedoch auch weniger triviale Gegenbeispiele.

(f) Zusätzlich zeigen wir, daß $\mathfrak{L}(0L)$ nicht unter Konkatenation abgeschlossen ist. Dies gilt, da sowohl $\{a\}$ als auch $\{\varepsilon, a^2\}$ Sprachen aus $\mathfrak{L}(0L)$ sind, ihre Konka-

tenation $\{a\}\{\varepsilon, a^2\} = \{a, a^3\}$ nach Teil (a) des Beweises jedoch nicht. □

Satz 13.3 $\mathfrak{L}(0L)$ ist nicht unter Substitution, EÜA-Abbildung und inverser EÜA-Abbildung abgeschlossen.

Beweis: Wegen $\{w\} \in \mathfrak{L}(0L)$ für beliebige Wörter w folgt aus dem Abschluß unter Substitution sofort der Abschluß unter Homomorphismus. Dasselbe gilt für den Abschluß unter EÜA-Abbildung, da jeder Homomorphismus $h : V_I^* \to V_O^*$ durch einen endlichen Übersetzungsautomaten mit der Zustandsmenge $S = \{s_0\}$, Endzustandsmenge $S_1 = S$ und der Produktionenmenge $\{s_0 a \to v s_0 \mid h(a) = v, a \in V_I\}$ dargestellt werden kann. Bei Abschluß unter inverser EÜA-Abbildung würde entsprechend der Abschluß unter inversem Homomorphismus folgen. □

Mit $\mathfrak{L}(\text{fin})$ bezeichnen wir die Familie der endlichen Sprachen.

Satz 13.4 In $\mathfrak{L}(\text{fin})$, $\mathfrak{L}_3 - \mathfrak{L}(\text{fin})$, $\mathfrak{L}_2 - \mathfrak{L}_3$ und $\mathfrak{L}_1 - \mathfrak{L}_2$ existieren sowohl Sprachen, die in $\mathfrak{L}(0L)$ liegen, als auch solche Sprachen, die nicht in $\mathfrak{L}(0L)$ liegen.

Beweis: (a) Die Sprachen $\{a\} \in \mathfrak{L}(\text{fin})$ und $\{a\}^* \in \mathfrak{L}_3 - \mathfrak{L}(\text{fin})$ sind offenbar 0L-Sprachen. Die Sprache $\{a^i b a^i \mid i \in \mathbb{N}_0\}$ gehört zu $(\mathfrak{L}_2 - \mathfrak{L}_3) \cap \mathfrak{L}(0L)$, da sie von dem 0L-System $(\{a, b\}, \{a \to a, b \to aba\}, b)$ und von der kontextfreien Grammatik $(\{X_0\}, \{a, b\}, X_0, \{X_0 \to aX_0a, X_0 \to b\})$ erzeugt wird. Andererseits ist sie nach dem Beweis von Satz 3.8 keine Typ-3-Sprache. Die Sprache $\{a^{2^n} \mid n \in \mathbb{N}_0\}$ ist vom Typ 1 (siehe [23], Example 2.4), jedoch nach Satz 5.9 nicht kontextfrei. Das gilt dann auch für die Sprache $\{a^2\} \cup \{b^{2^n} \mid n \in \mathbb{N}, n \geq 2\}$, die nach dem Beweisteil (b) von Satz 13.2 eine 0L-Sprache ist.

(b) Wir geben Sprachen der genannten Familien an, die keine 0L-Sprachen sind. In $\mathfrak{L}(\text{fin})$ ist dies nach dem Beweis von Satz 13.2 die Sprache $L_1 = \{a, a^3\}$.
Offensichtlich gehört $L_2 = \{a, a^3\} \cup \{b^n \mid n \geq 4\}$ zu $\mathfrak{L}_3 - \mathfrak{L}(\text{fin})$. Wir nehmen an, daß L_2 von einem 0L-System $G = (\{a, b\}, h, \omega)$ erzeugt wird. Wäre $a^\mu \in h(b)$ mit $\mu \geq 1$, dann ergäbe sich die Ableitung $b^4 \Longrightarrow_G w$ mit einem Wort w, das mindestens vier Symbole a enthält. Ein solches Wort kommt in L_2 nicht vor, so daß $\omega = a$ oder $\omega = a^3$ gelten muß. Dann existiert die Ableitung $a \Longrightarrow_G a^3$ oder die Ableitung $a^3 \Longrightarrow_G a$. Beide führen wie in dem Beweis von Satz 13.2 zu einem Widerspruch.
Wir betrachten $L_3 = \{a^i b^i \mid i \in \mathbb{N}\} \in \mathfrak{L}_2 - \mathfrak{L}_3$. Es werde L_3 von einem 0L-System $G = (\{a, b\}, h, \omega)$ erzeugt. Dann gilt $h(a) \subset a^*$ und $h(b) \subset b^*$, da sonst Wörter mit einer falschen Reihenfolge der Buchstaben a und b erzeugt werden könnten. Für je zwei Wörter $a^\nu \in h(a)$ und $b^\mu \in h(b)$ mit $\nu, \mu \in \mathbb{N}$ gilt $\nu = \mu$, da anderenfalls Wörter $a^i b^{i'}$ mit $i \neq i'$ abgeleitet werden könnten. Das System ist also deterministisch. Da Wörter mit beliebig großem i zu L_3 gehören, muß $\nu = \mu > 1$ gelten. Somit ist G ein PD0L-System. Folglich ist $\omega = ab$, und es muß die Ableitung $ab \Longrightarrow_G a^2 b^2$ geben. Dann kann nur $\nu = \mu = 2$ gelten. Somit ist $a^3 b^3 \in L_3$ nicht gemäß G erzeugbar.

Nach den Überlegungen aus (a) gilt $L_4 = \{a^3\} \cup \{a^{2^n} \mid n \in \mathbb{N}_0\} \in \mathfrak{L}_1 - \mathfrak{L}_2$. Es werde L_4 von einem 0L-System $G = (\{a\}, h, \omega)$ erzeugt. Dann muß $a^i \in h(a)$ für ein $i > 1$ gelten. Ist weiter $\varepsilon \in h(a)$, so erhalten wir $(a^i)^* \subset L_4$ und somit einen Widerspruch. Folglich ist G ein P0L-System. Das Axiom muß a sein, und um a^2 zu erzeugen, benötigen wir die Produktion $a^2 \in h(a)$. $a^3 \in L_4$ kann nur aus a oder a^2 direkt erzeugt werden. Im ersten Fall erhalten wir $a^3 \in h(a)$ und daher den Widerspruch $a^2 \Longrightarrow_G a^6 \notin L_4$. Im zweiten Fall folgt $a, a^2 \in h(a)$ und somit $L_4 = \{a\}^*$, was der Gestalt der Sprache widerspricht. □

Satz 13.5 (a) Jede Sprache, die von einem 0L-System $G = (\Sigma, h, \omega)$ mit $a \in h(a)$ für alle $a \in \Sigma$ erzeugt wird, ist kontextfrei.

(b) Für jede kontextfreie Sprache L existiert ein 0L-System G und ein Alphabet Δ mit $L = L(G) \cap \Delta^*$.

Beweis: (a) Es sei $G = (\Sigma, h, \omega)$ ein 0L-System mit $a \in h(a)$ für alle $a \in \Sigma$. Dann braucht nur ein Symbol eines vorgelegten Wortes aus $L(G)$ ersetzt zu werden, um ein weiteres Wort aus $L(G)$ zu erhalten. Daher kann durch

$$\begin{aligned} &V_N = \{N_a \mid a \in \Sigma\}, \\ &\omega_N = N_{a_1} \ldots N_{a_n} \quad \text{für } \omega = a_1 \ldots a_n \text{ und} \\ &F_N = \{N_a \to N_{b_1} \ldots N_{b_m} \mid b_1 \ldots b_m \in h(a), m \in \mathbb{N}_0, a \in \Sigma\} \end{aligned}$$

eine kontextfreie Grammatik

$$G_{cf} = (V_N \cup \{X_0\}, \Sigma, X_0, F_N \cup \{X_0 \to \omega_N\} \cup \{N_a \to a \mid a \in \Sigma\})$$

definiert werden, die offenbar zu G äquivalent ist.

(b) Die Sprache L werde von einer kontextfreien Grammatik (V_N, V_T, X_0, F) erzeugt. Mit der Definition des 0L-Systems

$$G = (V_N \cup V_T, F \cup \{a \to a \mid a \in V_N \cup V_T\}, X_0)$$

erhalten wir $L = L(G) \cap V_T^*$. □

Es kann sogar $\mathfrak{L}(0L) \subsetneqq \mathfrak{L}_1$ gezeigt werden. In Satz 13.28 wird ein umfassenderes Ergebnis bewiesen.

Zum Abschluß dieses Paragraphen werden Aussagen über Entscheidbarkeits- und Unentscheidbarkeitsfragen für 0L-Systeme vorgestellt. Dafür dient der folgende Satz als Vorbereitung.

Satz 13.6 Für jedes 0L-System $G = (\Sigma, h, \omega)$ existiert eine natürliche Zahl $k(G)$, so daß jedes nichtleere Wort $w \in L(G)$ eine Ableitung

$$\omega = w_0 \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_\nu = w$$

mit $|w_i| \leq k(G) \cdot |w|$ für alle i, $i = 0, \ldots, \nu$, besitzt.

Beweis: Das Alphabet Σ besitze n Elemente. Es werde

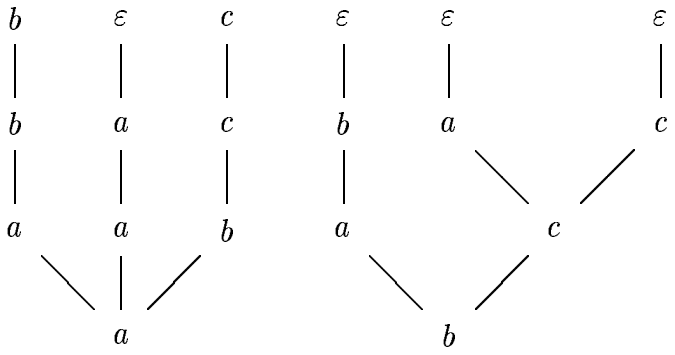
$$\begin{aligned} r &= \max\{|w| \mid w \in h(a), a \in \Sigma\}, \\ s &= \max\{|w| \mid w \in L^i(G), i = 0, 1, \ldots, n\} \text{ und} \\ k(G) &= \max\{s, r^{n+1}\} \end{aligned}$$

gesetzt. Zu jeder Ableitung D von w gemäß G gehört ein Ableitungsgraph, bei dem alle Blätter, die nicht mit ε markiert sind, den gleichen Abstand vom Axiom ω besitzen. Eine Ableitung heißt *reduziert*, falls der zugeordnete Graph keinen Teilbaum T mit den folgenden beiden Eigenschaften enthält:

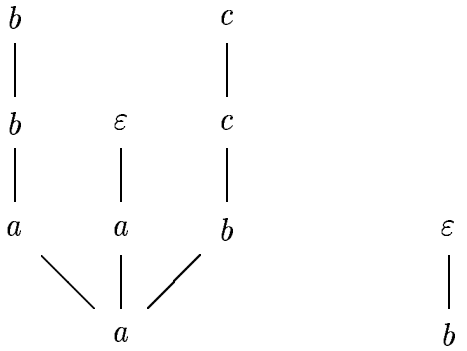
(1) Alle Blätter von T sind mit ε markiert,

(2) T enthält einen Weg mit zwei Knoten, die dieselbe Marke besitzen.

Für jede Ableitung D eines Wortes w gemäß G existiert eine reduzierte Ableitung von w. Dies sehen wir wie folgt ein. Wir betrachten den Ableitungsgraphen von D. Es sei T ein Teilbaum von D mit den Eigenschaften (1) und (2), und a sei die gemäß (2) mehrfach auftretende Marke. Der Teilbaum von T, der von dem Knoten mit der Marke a ausgeht, der dem Axiom ω am nächsten ist, wird durch den Teilbaum von T ersetzt, der von dem Knoten mit der Marke a ausgeht, der von ω am weitesten entfernt ist. Dies machen wir uns an dem folgenden Beispiel klar. Der Ableitungsgraph (mit $\omega = ab$)



wird zu



reduziert. Zur Reduzierung von D muß das Verfahren ggf. wiederholt werden.

Es sei

$$D' \quad \omega = w_0 \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_\nu = w.$$

die reduzierte Ableitung von D. Das Vorkommen eines Buchstaben a in einem Wort w_i mit $i = 0, \ldots, \nu$ nennen wir *unproduktiv*, falls a die Marke der Wurzel eines Teilbaums ist, dessen Blätter alle mit ε markiert sind. Anderenfalls heißt das Vorkommen *produktiv*. Wird ε von einem Symbol in einem Wort der reduzierten Ableitung D' abgeleitet, dann erfolgt dies in höchstens n Schritten, da anderenfalls mindestens ein Buchstabe von Σ auf dem entsprechenden Pfad zu ε mehrfach auftreten würde. Daher hat jeder Teilbaum, der von einem unproduktiven Vorkommen ausgeht, höchstens die Höhe n.

Die Aussage des Satzes wird nun in zwei Fallunterscheidungen bewiesen. Für $i \leq n$ gelten wegen $w_i \in L^i(G)$ und $w \neq \varepsilon$ die Ungleichungen

$$|w_i| \leq s \leq s \cdot |w| \leq k(G) \cdot |w|.$$

Für $i > n$ wird das Wort $w_{i-(n+1)}$ betrachtet. Da w_i in $n+1$ Ableitungsschritten aus $w_{i-(n+1)}$ erzeugt wird, tragen die unproduktiven Vorkommen von $w_{i-(n+1)}$ nicht mehr zu w_i bei, höchstens die produktiven Buchstaben in $w_{i-(n+1)}$ wirken sich auf w_i aus. Mehr als $|w|$ produktive Vorkommen kann es aber in $w_{i-(n+1)}$ nicht geben. Folglich erhalten wir

$$|w_i| \leq r^{n+1} \cdot |w| \leq k(G) \cdot |w|. \quad \square$$

Satz 13.7 Das Wortproblem für 0L-Systeme ist entscheidbar.

Beweis: Falls bereits $\mathfrak{L}(0\mathrm{L}) \subset \mathfrak{L}_1$ gezeigt wäre, so würde schon aus Satz 7.1 das Ergebnis folgen. Wir wollen hier jedoch einen direkten Beweis angeben.

Es sei G ein 0L-System und w ein beliebiges Wort. Für $w = \varepsilon$ kann durch Betrachtung der Produktionen festgestellt werden, ob $\varepsilon \in L(G)$ gilt oder nicht. Ist $w \neq \varepsilon$, dann kann die Zahl $k(G)$ gemäß Satz 13.6 effektiv berechnet werden. Wir definieren für jedes $n \in \mathbb{N}_0$ die Menge

$$K_n(G,w) = \{v \mid v \in L(G), v \text{ besitzt eine Ableitung } D \text{ der Länge } \leq n \text{ mit } \omega = v_0^D \Longrightarrow \ldots \Longrightarrow v_{m_D}^D = v \text{ und } |v_i^D| \leq k(G) \cdot |w|\}.$$

Dann gilt für jedes $n \in \mathbb{N}_0$

$$K_{n+1}(G,w) = K_n(G,w) \cup \{v \mid \bigvee_{v' \in K_n(G,w)} v' \Longrightarrow v, |v| \leq k(G) \cdot |w|\}.$$

Die Mengen $K_n(G,w)$ sind effektiv berechenbar, wir erhalten $K_n(G,w) \subset K_{n+1}(G,w)$ für alle $n \in \mathbb{N}_0$, und es gilt

$$1 = \mathrm{card}(K_0(G,w)) \leq \mathrm{card}(K_1(G,w)) \leq \ldots \leq p,$$

wobei p die Anzahl der Wörter v mit $|v| \leq k(G) \cdot |w|$ ist. Somit existiert ein $t \in \mathbb{N}_0$ mit $K_t(G,w) = K_{t+1}(G,w)$. Dann ergibt sich $K_t(G,w) = K_m(G,w)$ für alle $m \geq t$. Wir erhalten die Äquivalenz

$$w \in L(G) \iff w \in \bigcup_{i=0}^{t} K_i(G,w) = K_t(G,w).$$

Folglich ist in endlich vielen Schritten entscheidbar, ob $w \in L(G)$ gilt oder nicht. $\square$

Satz 13.8 Das Äquivalenzproblem für 0L-Systeme ist unentscheidbar.

Beweis: Der Satz wird bewiesen, indem gezeigt wird, daß aus der Entscheidbarkeit des Äquivalenzproblems für 0L-Systeme die Entscheidbarkeit des Postschen Korrespondenzproblems über einem zweielementigen Alphabet folgt. Dies ist jedoch nach Satz 8.7 unentscheidbar.

Es sei ein beliebiges Postsches Korrespondenzproblem

$$PCP = (\Sigma, n, \alpha, \beta) \text{ mit } \Sigma = \{a, b\},\ \alpha = (\alpha_1, \ldots, \alpha_n) \text{ und } \beta = (\beta_1, \ldots, \beta_n)$$

gegeben. Mit Hilfe eines neuen Alphabetes

$$V = \{S, A, B, C, D, E, F\}$$

wird ein 0L-System $G(PCP) = (V \cup \Sigma, h, \omega)$ mit $\omega = S$ definiert, wobei h durch die Produktionen

$$\begin{array}{ll}
S \to \alpha_i E \widehat{\beta_i} & \text{für } i = 1, \ldots, n, \\
S \to xAx & \text{für } x \in \Sigma, \\
S \to xBy & \text{für } x, y \in \Sigma, x \neq y, \\
A \to xAx & \text{für } x \in \Sigma, \\
A \to xBy & \text{für } x, y \in \Sigma, x \neq y, \\
A \to xC & \text{für } x \in \Sigma, \\
A \to Fx & \text{für } x \in \Sigma, \\
B \to xB & \text{für } x \in \Sigma, \\
B \to Bx & \text{für } x \in \Sigma, \\
B \to D, & \\
C \to xC & \text{für } x \in \Sigma, \\
C \to D, & \\
D \to D, & \\
E \to \alpha_i E \widehat{\beta_i} & \text{für } i = 1, \ldots, n, \\
F \to xF & \text{für } x \in \Sigma, \\
F \to D, & \\
x \to x & \text{für } x \in \Sigma
\end{array}$$

bestimmt ist. Hierbei ist durch $\widehat{w}$ das Spiegelwort eines beliebigen $w \in \Sigma^*$ gekennzeichnet. Außerdem wird ein 0L-System $H(PCP) = (V \cup \Sigma, h_1, \omega)$ definiert, wobei h_1 alle Produktionen von h enthält und zusätzlich noch die Produktion

$$E \to D.$$

Die von $G(PCP)$ und $H(PCP)$ erzeugten Sprachen lauten nun

$$\begin{aligned}
L(G(PCP)) = {} & \{S\} \cup \{wA\widehat{w} \mid w \in \Sigma^+\} \\
& \cup \{wxzBuy\widehat{w} \mid w, u, z \in \Sigma^*,\ x, y \in \Sigma, x \neq y\} \\
& \cup \{wzC\widehat{w} \mid w, z \in \Sigma^+\} \\
& \cup \{wFz\widehat{w} \mid w, z \in \Sigma^+\} \\
& \cup \{wD\widehat{u} \mid w, u \in \Sigma^+, w \neq u\} \\
& \cup \{wE\widehat{u} \mid w = \alpha_{i_1}, \ldots, \alpha_{i_t}, u = \beta_{i_1}, \ldots, \beta_{i_t}, t \geq 1, i_1, \ldots, i_t \in \{1, \ldots, n\}\}
\end{aligned}$$

und

$$L(H(PCP)) = \\ L(G(PCP)) \cup \{wD\widehat{u} \mid w = \alpha_{i_1}, \ldots, \alpha_{i_t}, u = \beta_{i_1}, \ldots, \beta_{i_t}, t \geq 1, i_1, \ldots, i_t \in \{1, \ldots, n\}\}.$$

Die Betrachtung der Sprachen zeigt, daß genau dann $L(G(PCP)) \neq L(H(PCP))$ gilt, wenn das Postsche Korrespondenzproblem PCP eine Lösung besitzt. Aus der Entscheidbarkeit des Äquivalenzproblems für 0L-Systeme würde die Entscheidbarkeit des Postschen Korrespondenzproblems über zweielementigen Alphabeten folgen, ein Widerspruch. □

Es kann gezeigt werden (siehe [22], Theorem III.2.4), daß das Äquivalenzproblem für D0L-Systeme entscheidbar ist.

13.2 E0L- und ET0L-Systeme

Definition 13.6 $G = (\Sigma, H, \omega, \Delta)$ heißt ET0L-*System*, wenn die folgenden Eigenschaften erfüllt sind:

(a) Σ ist ein Alphabet,
(b) H ist eine nichtleere endliche Menge von nichtleeren endlichen Substitutionen auf Σ,
(c) $\omega \in \Sigma^*$ (das *Axiom*) ,
(d) $\Delta \subset \Sigma$ (das *Terminalalphabet* Δ).

$L(G) = \{w \in \Delta^* \mid w = \omega$ oder $w \in h_1(\ldots(h_k(\omega))\ldots)$ mit $h_1, \ldots, h_k \in H, k \in \mathbb{N}\}$ heißt die *von G erzeugte* ET0L-*Sprache*. □

H wird auch als die Menge der *Tafeln* $h \in H$ bezeichnet.

Definition 13.7 (a) $G = (\Sigma, H, \omega)$ heißt T0L-*System*, falls $(\Sigma, H, \omega, \Sigma)$ ein ET0L-System ist.

(b) $G = (\Sigma, h, \omega, \Delta)$ heißt E0L-*System*, falls $(\Sigma, \{h\}, \omega, \Delta)$ ein ET0L-System ist. □

Wie bei den 0L-Systemen werden propagierende und deterministische derartige Systeme definiert, z.B. EPT0L-, ED0L- oder EPDT0L-Systeme. Die entsprechenden Sprachfamilien werden mit $\mathfrak{L}$(ET0L), $\mathfrak{L}$(T0L), $\mathfrak{L}$(E0L), $\mathfrak{L}$(ED0L) usw. bezeichnet. Die Ableitungsrelationen $\Longrightarrow$, $\Longrightarrow^+$ oder $\Longrightarrow^*$ werden wie üblich gegeben.

Als unmittelbare Folgerung der Definitionen erhalten wir den folgenden Satz.

Satz 13.9 Es gelten die Inklusionen

(a) $\mathfrak{L}(0L) \subset \mathfrak{L}(T0L) \subset \mathfrak{L}(ET0L)$ und $\mathfrak{L}(0L) \subset \mathfrak{L}(E0L) \subset \mathfrak{L}(ET0L)$ sowie
(b) $\mathfrak{L}(EPDT0L) \subset \mathfrak{L}(EDT0L) \subset \mathfrak{L}(ET0L)$ und $\mathfrak{L}(EPDT0L) \subset \mathfrak{L}(EPT0L) \subset \mathfrak{L}(ET0L)$. □

Die Aussagen von Satz 13.9(b) gelten natürlich auch für die Familien der 0L-, T0L- oder E0L-Sprachen.

Satz 13.10 Es gilt $\mathfrak{L}(\text{fin}) \subset \mathfrak{L}(\text{E0L})$.

Beweis: Es sei $L = \{w_1, \ldots, w_k\}$ für $k \in \mathbb{N}$ eine endliche Sprache über dem Alphabet Σ. Das E0L-System

$$G = (\Sigma \cup \{S\}, h, S, \Sigma) \text{ mit } h(S) = \{w_1, \ldots, w_k\} \text{ und } h(x) = \{x\} \text{ für alle } x \in \Sigma$$

erzeugt L. Ist $L = \emptyset$, so wird L durch $G = (\{X, a\}, h, X, \{a\})$ mit $h(x) = \{x\}$ für alle $x \in \{X, a\}$ gegeben. $\square$

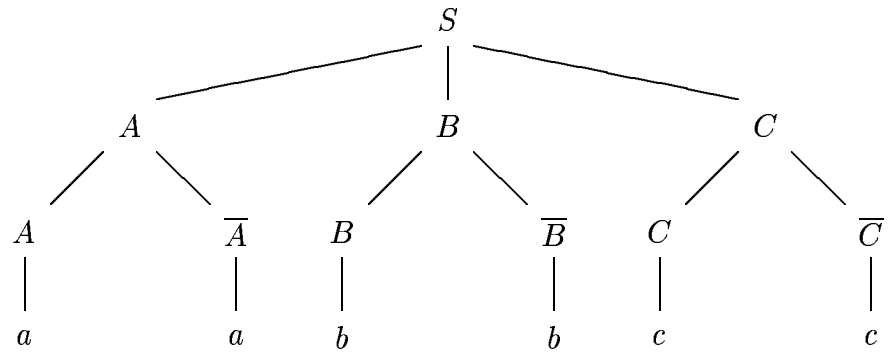
Beispiel 13.6 Es sei $G = (\Sigma, h, S, \Delta)$ ein E0L-System mit

$$\Sigma = \{S, A, B, C, \overline{A}, \overline{B}, \overline{C}, F, a, b, c\} \quad \text{und} \quad \Delta = \{a, b, c\}.$$

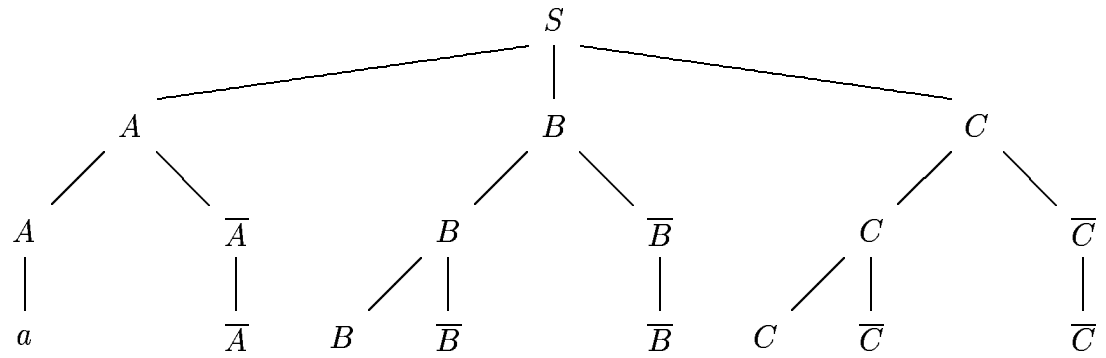
Die Substitution h sei durch die Produktionen

$$\begin{array}{llll} S \to ABC, & & & \\ A \to A\overline{A}, & A \to a\,, & \overline{A} \to \overline{A}, & \overline{A} \to a, \\ B \to B\overline{B}, & B \to b\,, & \overline{B} \to \overline{B}, & \overline{B} \to b, \\ C \to C\overline{C}, & C \to c\,, & \overline{C} \to \overline{C}, & \overline{C} \to c, \\ a \to F\,, & b \to F, & c \to F, & F \to F \end{array}$$

definiert. Ein Ableitungsgraph für $a^2b^2c^2$ ist durch



gegeben. Dagegen kann der Baum



zu keinem Ableitungsgraphen eines beliebigen Terminalworts fortgesetzt werden. Wird ein Terminalsymbol eingeführt, so wird daraus im nächsten und den folgenden Schritten das „Fehlschlagsymbol“ F. Um ein Terminalwort zu erhalten, müssen gemäß G alle Symbole des Wortes „auf einen Schlag“ erzeugt werden. □

Ein ET0L-System, das wie das System aus Beispiel 13.6 alle Symbole eines Terminalwortes auf einen Schlag erzeugt, wird auch synchronisiertes System genannt.

Definition 13.8 Ein ET0L-System $G = (\Sigma, H, \omega, \Delta)$ heißt *synchronisiert*, wenn für alle $a \in \Delta$ und $w \in \Sigma^*$ mit $a \Longrightarrow^+ w$ die Beziehung $w \notin \Delta^*$ folgt. □

Satz 13.11 Es existiert ein Algorithmus, der zu jedem beliebigen E0L-System G' ein äquivalentes synchronisiertes E0L-System G konstruiert. Dabei ist G propagierend, wenn auch G' propagierend ist.

Beweis: Es sei $G' = (\Sigma', h', \omega', \Delta)$ ein E0L-System. Mit Hilfe eines neuen Symbols ω wird zunächst ein äquivalentes E0L-System $G_1 = (\Sigma_1, h_1, \omega, \Delta)$ mit

$$\Sigma_1 = \Sigma' \cup \{\omega\},\ h_1(x) = h'(x) \text{ für alle } x \in \Sigma' \text{ und } h_1(\omega) = \{\omega'\}$$

definiert. Wir setzen $\bar{\Delta} = \{\bar{a} \mid a \in \Delta\}$ und führen ein weiteres Symbol $F \notin \Sigma_1 \cup \bar{\Delta}$ ein. Damit ist das Alphabet

$$\Sigma = \Sigma_1 \cup \bar{\Delta} \cup \{F\}$$

gegeben. Für ein Wort $w = a_1 \ldots a_n \in \Sigma_1^*$, $n \in \mathbb{N}$, werde $\bar{w} = b_1 \ldots b_n \in \Sigma^*$ durch

$$b_i = \begin{cases} \bar{a}_i & \text{für } a_i \in \Delta \\ a_i & \text{sonst} \end{cases}$$

definiert. Speziell gilt $\bar{\varepsilon} = \varepsilon$. Wir konstruieren ein E0L-System $G = (\Sigma, h, \bar{\omega}, \Delta)$, wobei h durch die Produktionenmenge

$$\begin{aligned} &\{a \to \bar{w} \mid a \in \Sigma_1 - \Delta, w \in h_1(a)\} \cup \{a \to w \mid a \in \Sigma_1 - \Delta, w \in h_1(a)\} \\ &\quad \cup \{\bar{a} \to \bar{w} \mid a \in \Delta, w \in h_1(a)\} \cup \{\bar{a} \to w \mid a \in \Delta, w \in h_1(a)\} \\ &\quad \cup \{a \to F \mid a \in \Delta \cup \{F\}\} \end{aligned}$$

bestimmt wird. Offenbar ist G synchronisiert. Die Ableitungen gemäß G_1 werden in G in „gequerter“ Form simuliert, wobei im letzten Schritt die Terminalsymbole eingeführt werden. Es gilt $L(G) = L(G_1) = L(G')$. Wir erkennen sofort, daß G propagierend ist, wenn es auch G' ist. □

In Satz 13.10 hatten wir gesehen, daß im Gegensatz zur Familie $\mathfrak{L}(0L)$ die Familie $\mathfrak{L}(E0L)$ alle endlichen Sprachen enthält. Sie ist auch keine anti-AFL.

Satz 13.12 $\mathfrak{L}(E0L)$ ist abgeschlossen unter Vereinigung, Konkatenation und ε-freier Iteration.

Beweis: Es seien $L_1 = L(G_1)$ und $L_2 = L(G_2)$ Sprachen, die von E0L-Systemen $G_1 = (\Sigma_1, h_1, S_1, \Delta_1)$ und $G_2 = (\Sigma_2, h_2, S_2, \Delta_2)$ erzeugt werden. Nach Satz 13.11 können wir ohne Beschränkung der Allgemeinheit annehmen, daß G_1 und G_2 synchronisiert sind. Der Beweis zeigte auch, daß dabei $S_1 \in \Sigma_1 - \Delta_1$ und $S_2 \in \Sigma_2 - \Delta_2$ gewählt werden kann. Außerdem gelte ohne Beschränkung der Allgemeinheit

$$(\Sigma_1 - \Delta_1) \cap \Sigma_2 = \emptyset \text{ und } (\Sigma_2 - \Delta_2) \cap \Sigma_1 = \emptyset,$$

was zu

$$\Sigma_1 \cap \Sigma_2 = \Delta_1 \cap \Delta_2$$

äquivalent ist. Es seien S, $\widehat{S_1}$ und $\widehat{S_2}$ neue Symbole. Die Abschlußeigenschaften werden wie folgt bewiesen.

(a) Wir konstruieren das E0L-System $G = (\Sigma_1 \cup \Sigma_2 \cup \{S\}, h, S, \Delta_1 \cup \Delta_2)$ mit

$$h(a) = \begin{cases} h_1(a), & \text{falls } a \in \Sigma_1 \\ h_2(a), & \text{falls } a \in \Sigma_2 - \Delta_1 \end{cases} \quad \text{und } h(S) = \{S_1, S_2\}.$$

Da die gegebenen Systeme synchronisiert sind, kann aus einem Wort mit einem Vorkommen eines $a \in \Sigma_1 \cap \Sigma_2 = \Delta_1 \cap \Delta_2$ weder gemäß h_1 noch gemäß h_2 ein Terminalwort erzeugt werden. Folglich erzeugt G die Sprache $L_1 \cup L_2$.

(b) Das E0L-System $G = (\Sigma_1 \cup \Sigma_2 \cup \{S, \widehat{S_1}, \widehat{S_2}\}, h, S, \Delta_1 \cup \Delta_2)$ mit

$$h(a) = \begin{cases} h_1(a), & \text{falls } a \in \Sigma_1 \\ h_2(a), & \text{falls } a \in \Sigma_2 - \Delta_1 \end{cases} \quad \text{sowie}$$

$$h(S) = \{\widehat{S_1}\widehat{S_2}\},\ h(\widehat{S_1}) = \{S_1, \widehat{S_1}\} \text{ und } h(\widehat{S_2}) = \{S_2, \widehat{S_2}\}$$

erzeugt die Sprache $L_1 L_2$. Dabei sorgen die Produktionen $\widehat{S_1} \to \widehat{S_1}$ und $\widehat{S_2} \to \widehat{S_2}$ für die Synchronisation von G.

(c) Das E0L-System $G = (\Sigma_1 \cup \{S\}, h, S, \Delta_1)$ mit

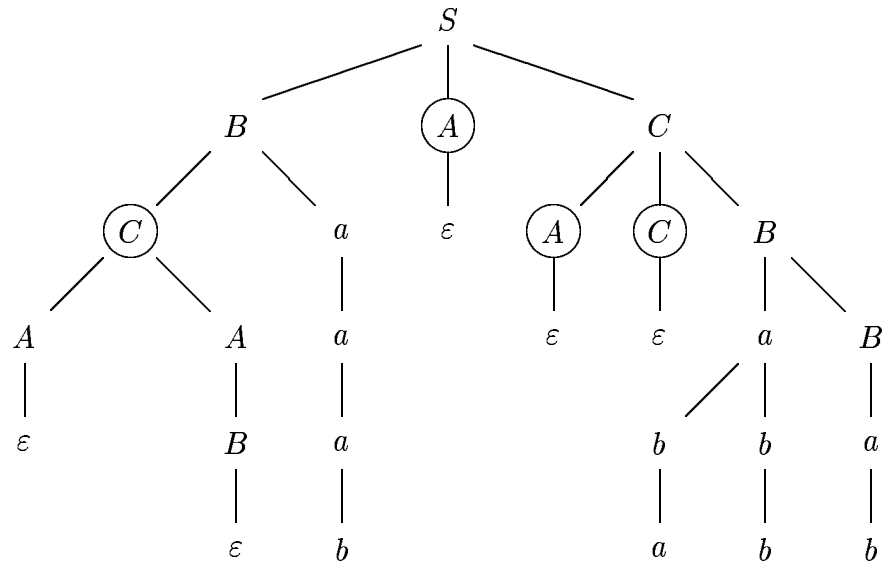
$$h(a) = h_1(a) \text{ für } a \in \Sigma_1 \text{ und } h(S) = \{S, SS, S_1\}$$

erzeugt offenbar die Sprache L_1^+. $\square$

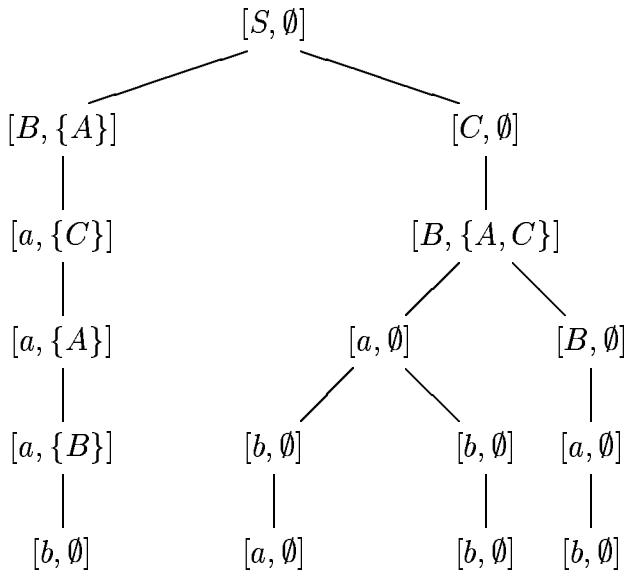
Für ein Wort w über einem beliebigen Alphabet Σ bezeichne $\text{alph}(w)$ im folgenden die Menge aller in w enthaltenen Symbole aus Σ.

Satz 13.13 Es existiert ein Algorithmus, der zu jedem E0L-System H ein EP0L-System G mit $L(G) = L(H) - \{\varepsilon\}$ konstruiert.

Beweis: Es sei $H = (\Sigma, h, \omega, \Delta)$ ein E0L-System. Falls $L(H)$ endlich ist, ist die Aussage nach Satz 13.10 offensichtlich erfüllt. Im folgenden nehmen wir daher an, daß $L(H)$ unendlich ist, wobei wir nach dem Beweis von Satz 13.11 die Gültigkeit von $\omega = S \in \Sigma - \Delta$ voraussetzen können. Zunächst wollen wir die intuitive Beweisidee verdeutlichen. Wir gehen von einem Ableitungsbaum in H für das Wort bab^2 aus.



Dieser Ableitungsbaum soll in G derart simuliert werden, daß die von den eingekreisten Knoten ausgehenden Teilbäume nicht vorkommen, da diese Knoten unproduktive Vorkommen repräsentieren. Die wegzulassenden Teilbäume werden sich zunächst dadurch gemerkt, daß die zugehörigen unproduktiven Vorkommen in den jeweiligen Schritten als zweite Komponente der jeweiligen Marken der produktiven Vorkommen notiert werden. Man merkt sich also die Symbole, die man in das leere Wort ableiten möchte.



Erst in der letzten Stufe sind keine unproduktiven Vorkommen mehr vorhanden. Dann kann bab^2 durch einen weiter unten beschrieben Synchronisationsmechanismus erzeugt werden. Da vorab nicht bekannt ist, ob ein Vorkommen unproduktiv ist, müssen alle

Möglichkeiten solcher Bäume berücksichtigt werden.

Formal wird zu H das EP0L-System $G = (V, g, [S, \emptyset], \Delta)$ konstruiert. Wir setzen

$$V = V_1 \cup \{F\} \cup \Delta \text{ mit } V_1 = \{[a, Z] \mid a \in \Sigma, Z \subset \Sigma\}.$$

Die Substitution g wird wie folgt definiert:

(1) Für alle $a \in \Sigma$ mit $b_1 \dots b_k \in h(a)$, $b_1, \dots, b_k \in \Sigma$, $k \geq 2$, und alle $Z \subset \Sigma$ erhalten wir

$$[b_{i_1}, Z_{i_1}][b_{i_2}, Z_{i_2}] \dots [b_{i_p}, Z_{i_p}] \in g([a, Z]),$$

falls die folgenden Eigenschaften erfüllt sind:
Es gilt $1 \leq i_1 < i_2 < \dots < i_p \leq k$ und

$$\begin{aligned} Z_{i_1} &= \text{alph}(b_1 b_2 \dots b_{i_1 - 1}) \cup \text{alph}(b_{i_1+1} b_{i_1+2} \dots b_{i_2-1}), \\ Z_{i_j} &= \text{alph}(b_{i_j+1} b_{i_j+2} \dots b_{i_{j+1}-1}) \text{ für } 2 \leq j \leq p-1 \text{ sowie} \\ Z_{i_p} &= \text{alph}(b_{i_p+1} b_{i_p+2} \dots b_k) \cup Z' \end{aligned}$$

für $p \geq 2$ und

$$Z_{i_1} = \text{alph}(b_1 b_2 \dots b_{i_1-1}) \cup \text{alph}(b_{i_1+1} b_{i_1+2} \dots b_{i_2-1}) \cup Z'$$

für $p = 1$ mit jeweils

$$Z' \in \text{succ}_H(Z) = \{U \subset \Sigma \mid \bigvee_{w_1, w_2 \in \Sigma^*} \text{alph}(w_1) = Z, \text{alph}(w_2) = U, w_1 \Longrightarrow_H w_2\}.$$

(2) Für alle $a \in \Sigma$ mit $b \in h(a)$ und alle $Z \subset \Sigma$ gilt

$$[b, Z'] \in g([a, Z]),$$

falls $Z' \in \text{succ}_H(Z)$ gilt.

(3) Für alle $a \in \Sigma$ mit $\varepsilon \in h(a)$ und alle $Z \subset \Sigma$ gilt $F \in g([a, Z])$.

(4) Für alle $a \in \Delta$ gilt $a \in g([a, \emptyset])$.

(5) Es gilt $g(F) = \{F\}$ und $g(a) = \{F\}$ für alle $a \in \Delta$.

Durch (4) und (5) wird die Synchronisation erreicht. □

Bis auf das in den Sprachen von $\mathfrak{L}$(E0L) eventuell vorhandene leere Wort ε stimmen $\mathfrak{L}$(E0L) und $\mathfrak{L}$(EP0L) überein. Offensichtlich ist die Konstruktion des Beweises auch für ein gegebenes ET0L-System H richtig, wobei jeder Tafel von H genau eine Tafel von G zugeordnet wird. Somit ist Satz 13.13 auch für ET0L-Systeme erfüllt.

Da jedes E0L-System ein spezielles ET0L-System ist, gilt der folgende Satz auch für E0L-Systeme.

Satz 13.14 Es existiert ein Algorithmus, der für ein beliebiges ET0L-System G entscheidet, ob $\varepsilon \in L(G)$ gilt oder nicht.

Beweis: Es sei $G = (\Sigma, H, S, \Delta)$ ein ET0L-System. Wir können ohne Beschränkung der Allgemeinheit $S \in \Sigma - \Delta$ annehmen. Weiter werde $p = \text{card}(\Sigma)$ und $n = 2^p$ gesetzt. Wir definieren rekursiv eine Folge $A_1, A_2, \dots$ von Mengen von Teilmengen von Σ durch

(1) $A_1 = \{W \mid \bigvee_{h \in H} W = \{a \mid a \to_h \varepsilon\}\}$ und

(2) $A_{i+1} = A_i \cup \{W \mid \bigvee_{h \in H, Z \in A_i} W = \{a \mid \bigvee_{u \in Z^*} a \to_h u\}\}$ für alle $i \geq 1$.

Offenbar sind alle A_i endlich und effektiv konstruierbar. Wir erhalten die folgenden Aussagen:

(a) Für alle $i \geq 1$ gilt $A_i \subset A_{i+1}$. Für alle $j \geq n$ ist $A_j = A_{j+1}$. Dann folgt $A_i \subset A_n$ für alle $i \geq 1$.

(b) Für alle $i \geq 1$ und alle $W \in A_i$ existiert eine Zahl $k \leq i$, so daß es für alle $x \in W^+$ eine Ableitung $x \Longrightarrow^k \varepsilon$ gibt.
Wir beweisen diese Aussage durch Induktion über i. Für $i = 1$ ist die Aussage wegen (1) erfüllt. Wir nehmen an, daß die Aussage für i richtig ist. Für eine Menge $W \in A_{i+1}$ sind wegen (2) zwei Fälle zu unterscheiden.

(α) Für $W \in A_i$ existiert nach Induktionsvoraussetzung eine Zahl k mit $k \leq i < i+1$, so daß es für alle $x \in W^+$ eine Ableitung $x \Longrightarrow^k \varepsilon$ gibt.

(β) Es gelte $W = \{a \mid \bigvee_{u \in Z^*} a \to_h u\}$ mit $h \in H$ und $Z \in A_i$. Für jedes $u \in Z^*$ existiert nach Induktionsannahme eine Ableitung $u \Longrightarrow^k \varepsilon$ mit einer Zahl $k \leq i$. Für alle $x \in W^+$ erhalten wir dann eine Ableitung

$$x = a_1 \ldots a_m \Longrightarrow_h u_1 \ldots u_m \Longrightarrow^k \varepsilon \text{ mit } u_1, \ldots, u_m \in Z^*$$

der Länge $k + 1 \leq i + 1$.

(c) Für alle $x \in \Sigma^+$ mit $x \Longrightarrow^k \varepsilon$ existiert eine Menge $W \in A_k$, so daß jedes Symbol von x ein Element von W ist.
Auch hier erfolgt der Beweis durch Induktion. Für $k = 1$ ist die Aussage wegen (1) erfüllt. Wir nehmen an, daß sie für ein k richtig ist. Es werde eine Ableitung $x \Longrightarrow^{k+1} \varepsilon$ betrachtet. Diese kann mit einem Wort x' als eine Ableitung

$$x \Longrightarrow x' \Longrightarrow^k \varepsilon$$

dargestellt werden. Wegen der Induktionsannahme existiert zur Ableitung $x' \Longrightarrow^k \varepsilon$ eine Menge $W' \in A_k$, so daß jedes Symbol von x' ein Element von W' ist. Wegen (2) ist dann jedes Symbol von x ein Element einer Menge $W \in A_{k+1}$.

(d) Für alle $x \in \Sigma^+$ existiert die Ableitung $x \Longrightarrow^+ \varepsilon$ genau dann, wenn eine Menge $W \in A_n$ existiert, so daß jedes Symbol von x ein Element von W ist.
Gilt $x \Longrightarrow^+ \varepsilon$, so folgt $x \Longrightarrow^k \varepsilon$ für ein $k \geq 1$ und somit wegen (a) und (c) die Behauptung. Wegen (b) gilt umgekehrt, daß eine Zahl $k \leq n$ existiert mit $x \Longrightarrow^k \varepsilon$, d.h., es gilt $x \Longrightarrow^+ \varepsilon$.

Aus der Aussage (d) folgt, daß $\varepsilon \in L(G)$ genau dann erfüllt ist, wenn eine Menge $W \in A_n$ mit $S \in W$ existiert. Da A_n effektiv konstruierbar ist, ist die Frage, ob $\varepsilon \in L(G)$ gilt, entscheidbar. □

Nach Satz 13.13 und Satz 13.14 kann jede E0L-Sprache aus einem EP0L-System (Σ, h, S, Δ) gewonnen werden, wobei ggf. zusätzlich ein Symbol S' und zwei Produktionen $S' \to \varepsilon$ und $S' \to S$ zur Erzeugung des leeren Wortes eingeführt werden müssen.

Mit Satz 13.16 werden wir sehen, daß eine entsprechende Aussage auch für ET0L-Sprachen gilt. Die nächsten beiden Sätze liefern gewisse Normalformeigenschaften für ET0L-Systeme.

Satz 13.15 Es existiert ein Algorithmus, der zu jedem ET0L-System $\bar{G}$ ein äquivalentes ET0L-System $G = (\Sigma, H, \omega, \Delta)$ mit $\text{card}(H) = 2$ konstruiert.

Beweis: Es sei $\bar{G} = (\bar{\Sigma}, \bar{H}, \omega, \Delta)$ ein ET0L-System mit $\bar{H} = \{\bar{h_1}, \ldots, \bar{h_r}\}$. Dazu konstruieren wir ein äquivalentes ET0L-System $G = (\Sigma, H, \omega, \Delta)$ mit zwei Tafeln. Wir setzen

$$\Sigma = \{[a, i] \mid a \in \bar{\Sigma}, 1 \leq i \leq r\} \cup \bar{\Sigma}$$

und $H = \{h_1, h_2\}$ mit

$$\begin{aligned} &h_1(a) = \{[a, 1]\} \text{ für } a \in \bar{\Sigma}, \\ &h_1([a, i]) = \{[a, i+1]\} \text{ für } a \in \bar{\Sigma}, 1 \leq i \leq r-1, \\ &h_1([a, r]) = \{[a, 1]\} \text{ für } a \in \bar{\Sigma} \end{aligned}$$

und

$$\begin{aligned} &h_2(a) = \{a\} \text{ für } a \in \bar{\Sigma} \text{ und} \\ &h_2([a, i]) = \bar{h_i}(a) \text{ für } a \in \bar{\Sigma}, 1 \leq i \leq r. \end{aligned}$$

Ein Ableitungsschritt

$$w_1 \Longrightarrow_{\bar{G}} w_2 \text{ mit } w_1 = a_1 \ldots a_n,\ a_j \in \bar{\Sigma},\ w_2 = u_1 \ldots u_n,\ u_j \in \bar{h_i}(a_j),$$

wird dann gemäß G durch

$$a_1 \ldots a_n \Longrightarrow_{h_1} [a_1, 1] \ldots [a_n, 1] \Longrightarrow_{h_1} \ldots \Longrightarrow_{h_1} [a_1, i] \ldots [a_n, i] \Longrightarrow_{h_2} u_1 \ldots u_n$$

simuliert, womit $L(\bar{G}) \subset L(G)$ bewiesen ist. Da offensichtlich $L(G) \subset L(\bar{G})$ gilt, folgt die Behauptung. □

Es kann gezeigt werden, daß ein solches Ergebnis nicht für T0L-Systeme gilt.

Satz 13.16 Es existiert ein Algorithmus, der zu jedem ET0L-System G_0 ein EPT0L-System $G = (\Sigma, H, \omega, \Delta)$ mit $L(G) = L(G_0) - \{\varepsilon\}$ konstruiert, so daß $\omega \in \Sigma - \Delta$ gilt und ein Symbol $F \in \Sigma - (\Delta \cup \{\omega\})$ sowie Tafeln $h_I, h_T \in H$ (*Fehlschlagsymbol, initiale* und *terminale Tafel*) existieren mit folgenden Eigenschaften:

(a) Es ist $h_I(\omega) \subset (\Sigma - (\Delta \cup \{\omega\}))^+$, und für $a \in \Sigma$, $a \neq \omega$ gilt $h_I(a) = \{a\}$.

(b) Es gilt $h_T(a) \subset \Delta \cup \{F\}$ für $a \in \Sigma - (\Delta \cup \{\omega, F\})$ und $h_T(a) = \{a\}$ für $a \in \Delta \cup \{\omega, F\}$.

(c) Ist $h \in H - \{h_I, h_T\}$, so ist $h(a) \subset (\Sigma - (\Delta \cup \{\omega, F\}))^+$ für $a \in \Sigma - (\Delta \cup \{\omega, F\})$ und $h(a) = \{a\}$ für $a \in \Delta \cup \{\omega, F\}$.

Beweis: Es sei $G_0 = (\Sigma_0, H_0, \omega_0, \Delta)$ ein ET0L-System. Wir nehmen an, daß $L(G_0) \neq \emptyset$ und $L(G_0) \neq \{\varepsilon\}$ gilt, da anderenfalls die Aussage des Satzes trivialer Weise erfüllt ist. Zunächst führen wir die Konstruktion aus dem Beweis von Satz 13.13 für jede Tafel aus

H_0 durch und erhalten dadurch ein EPT0L-System $G_1 = (\Sigma_1, H_1, \omega_1, \Delta)$ mit $L(G_1) = L(G_0) - \{\varepsilon\}$. Im folgenden definieren wir das EPT0L-System $G = (\Sigma, H, \omega, \Delta)$. Wir setzen $\bar{\Delta} = \{\bar{a} \mid a \in \Delta\}$, und mit zwei weiteren neuen Symbolen ω und F ist das Alphabet

$$\Sigma = \Sigma_1 \cup \bar{\Delta} \cup \{\omega, F\}$$

des Systems G bestimmt. Wir geben einen Homomorphismus $\varphi : \Sigma_1^* \to ((\Sigma_1 - \Delta) \cup \bar{\Delta})^*$ durch

$$\varphi(a) = \begin{cases} a, & \text{falls } a \in \Sigma_1 - \Delta \\ \bar{a}, & \text{falls } a \in \Delta \end{cases}$$

an. Die Menge

$$H = \{h_I, h_T\} \cup \{\bar{h} \mid h \in H_1\}$$

der Tafeln wird im folgenden bestimmt. Es werde

$$\begin{aligned} &h_I(\omega) = \{\omega_1\}, \\ &h_I(a) = \{a\} \text{ für } a \in \Sigma_1 \cup \bar{\Delta} \cup \{F\} \text{ und} \end{aligned}$$

$$\begin{aligned} &h_T(\bar{a}) = \{a\} \text{ für } a \in \Delta, \\ &h_T(a) = \{F\} \text{ für } a \in \Sigma - (\bar{\Delta} \cup \Delta \cup \{\omega, F\}) \text{ sowie} \\ &h_T(a) = \{a\} \text{ für } a \in \Delta \cup \{\omega, F\} \end{aligned}$$

gesetzt. Für jedes $h \in H_1$ definieren wir

$$\begin{aligned} &\bar{h}(a) = \{\varphi(\alpha) \mid \alpha \in h(a)\} \text{ für } a \in \Sigma_1 - \Delta, \\ &\bar{h}(\bar{a}) = \{\varphi(\alpha) \mid \alpha \in h(a)\} \text{ für } a \in \Delta \text{ und} \\ &\bar{h}(a) = \{a\} \text{ für } a \in \Delta \cup \{\omega, F\}. \end{aligned}$$

G ist offenbar ein EPT0L-System. Mit der Bezeichnung $\bar{w}_i = \varphi(w_i)$ erhalten wir

$$\omega_1 \Longrightarrow_{h_1} w_1 \Longrightarrow \ldots \Longrightarrow_{h_r} w \in \Delta^+ \text{ gemäß } G_1$$

genau dann, wenn

$$\omega \Longrightarrow_{h_I} \omega_1 \Longrightarrow_{\bar{h}_1} \bar{w}_1 \Longrightarrow \ldots \Longrightarrow_{\bar{h}_r} \bar{w} \Longrightarrow_{h_T} w \in \Delta^+ \text{ gemäß } G$$

gilt. Es folgt $L(G) = L(G_0)$. Offensichtlich sind die Aussagen (a) bis (c) erfüllt. □

Unmittelbar folgt

Satz 13.17 Es sei $G = (\Sigma, H, \omega, \Delta)$ ein EPT0L-System wie in Satz 13.16. Dann gelten die folgenden Eigenschaften:

(a) Es sei $w \in L(G)$. Dann wird in jeder Ableitung D von w gemäß G die initiale Tafel im ersten Schritt und die terminale Tafel im letzten Schritt benutzt, sofern in D nicht die Schritte $\omega \Longrightarrow \omega$ und $w \Longrightarrow w$ benutzt werden. Es existiert eine Ableitung von w gemäß G, bei der die initiale und die terminale Tafel jeweils genau einmal benutzt werden.

(b) Es sei $w \in \Sigma^+$ und $\omega = w_0 \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_n = w$ eine Ableitung von w gemäß G. Falls es existiert, sei $i_0 \in \{0, \ldots, n\}$ das kleinste Element, für das w_{i_0} ein Vorkommen eines Symbols aus $\Delta \cup \{F\}$ enthält. Dann folgt

$$w_{i_0} \in (\Delta \cup \{F\})^+ \text{ und } w_j = w_{i_0} \text{ für alle } j, i_0 \leq j \leq n. \quad \square$$

Satz 13.18 Es existiert ein Algorithmus, der zu jedem ET0L-System G_0 ein EPT0L-System $G = (\Sigma, H, \omega, \Delta)$ mit $L(G) = L(G_0) - \{\varepsilon\}$ mit den folgenden Eigenschaften konstruiert:

(a) $\omega \in \Sigma - \Delta$.

(b) Es existiert ein $F \in \Sigma - (\Delta \cup \{\omega\})$, so daß $h(F) = \{F\}$ und $h(a) = \{F\}$ für alle $a \in \Delta$ und $h \in H$ erfüllt ist.

(c) Für alle $a \in \Sigma$ und $h \in H$ gilt $h(a) \subset \Delta^+$, $h(a) = \{F\}$ oder $h(a) \subset (\Sigma - (\Delta \cup \{\omega, F\}))^+$.

Beweis: Es sei G_0 ein beliebiges ET0L-System. Wie in Satz 13.16 nehmen wir $L(G_0) \neq \emptyset$ und $L(G_0) \neq \{\varepsilon\}$ an. Nach Satz 13.16 existiert ein EPT0L-System $G_1 = (\Sigma_1, H_1, \omega_1, \Delta)$ mit $L(G_1) = L(G_0) - \{\varepsilon\}$, für das die Aussagen von Satz 13.16 gelten. Das EPT0L-System $G = (\Sigma, H, \omega, \Delta)$ wird aus G_1 konstruiert, indem jede Tafel $h \in H_1$ so in eine Tafel $h' \in H$ abgeändert wird, daß jede Produktion $a \in h(a)$ mit $a \in \Delta \cup \{\omega_1\}$ durch $F \in h'(a)$ ersetzt wird. Mit $\omega = \omega_1$ und $\Sigma = \Sigma_1$ folgt wegen der Eigenschaft aus Satz 13.17(b)

$$L(G) = L(G_1) = L(G_0) - \{\varepsilon\}.$$

Die Aussagen (a) bis (c) sind erfüllt. $\square$

Der Satz zeigt, daß zu jedem ET0L-System ein (bis ggf. auf das leere Wort) äquivalentes synchronisiertes ET0L-System existiert.

Satz 13.19 Die Familie $\mathfrak{L}$(ET0L) ist eine volle AFL.

Beweis: Es seien $L_1, L_2 \in \mathfrak{L}$(ET0L). Dann existieren ET0L-Systeme G_1' und G_2' mit $L(G_1') = L_1$ und $L(G_2') = L_2$. Nach Satz 13.16 gibt es EPT0L-Systeme $G_1 = (\Sigma_1, H_1, S_1, \Delta_1)$ und $G_2 = (\Sigma_2, H_2, S_2, \Delta_2)$ mit $L(G_1) = L_1 - \{\varepsilon\}$ und $L(G_2) - \{\varepsilon\}$ und den weiteren in Satz 13.16 genannten Eigenschaften. Insbesondere gilt also $S_1 \in \Sigma_1 - \Delta_1$ und $S_2 \in \Sigma_2 - \Delta_2$. Nach Satz 13.14 ist entscheidbar, ob $\varepsilon \in L(G_i)$, $i = 1, 2$, erfüllt ist. Für $\varepsilon \in L_i$ wird zusätzlich die Produktion $S_i \to \varepsilon$ in jeder Tafel von H_i aufgenommen. Ohne Beschränkung der Allgemeinheit gelte wie in Satz 13.12

$$(\Sigma_1 - \Delta_1) \cap \Sigma_2 = \emptyset \text{ und } (\Sigma_2 - \Delta_2) \cap \Sigma_1 = \emptyset,$$

was zu

$$\Sigma_1 \cap \Sigma_2 = \Delta_1 \cap \Delta_2$$

äquivalent ist. Es sei S ein neues Symbol.

(a) $\mathfrak{L}$(ET0L) ist abgeschlossen unter Vereinigung. Dies zeigen wir, indem wir ein ET0L-System

$$G = (\Sigma, H, S, \Delta_1 \cup \Delta_2)$$

mit $L(G) = L_1 \cup L_2$ konstruieren. Wir setzen

$$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{S\} \text{ und } H = \{h_0\} \cup \{\widehat{h} \mid h \in H_1\} \cup \{\widehat{\widehat{h}} \mid h \in H_2\},$$

wobei die Tafeln h_0, $\widehat{h}$ für $h \in H_1$ und $\widehat{\widehat{h}}$ für $h \in H_2$ durch

$$\begin{aligned} &h_0(a) = \{a\} \text{ für } a \in \Sigma_1 \cup \Sigma_2,\ h_0(S) = \{S_1, S_2\}, \\ &\widehat{h}(a) = h(a) \text{ für } a \in \Sigma_1,\ \widehat{h}(a) = \{a\} \text{ für } a \in \Sigma - \Sigma_1, \\ &\widehat{\widehat{h}}(a) = h(a) \text{ für } a \in \Sigma_2,\ \widehat{\widehat{h}}(a) = \{a\} \text{ für } a \in \Sigma - \Sigma_2 \end{aligned}$$

definiert werden. Da nach Satz 13.16 durch die Produktionen von G_1 und G_2 Terminalzeichen nicht verändert werden und die übrigen Zeichen der beiden Systeme disjunkt sind, kann nach Einführung des Zeichens S_i, $i = 1, 2$, nur eine Ableitung gemäß G_i simuliert werden. Ein Übergang von z.B. G_1 nach G_2 ist nicht möglich. Damit erhalten wir $L(G) = L_1 \cup L_2$.

(b) Der Abschluß unter ε-freier Iteration wird mit Hilfe des ET0L-Systems

$$G = (\Sigma, H, S, \Delta_1) \text{ mit } \Sigma = \Sigma_1 \cup \{S\} \text{ und } H = \{h_0\} \cup \{\widehat{h} \mid h \in H_1\}$$

bewiesen, wobei die Tafeln h_0 und $\widehat{h}$ für $h \in H_1$ durch

$$\begin{aligned} &h_0(S) = \{S, S^2, S_1\},\ h_0(a) = \{a\} \text{ für } a \in \Sigma_1 \text{ und} \\ &\widehat{h}(S) = \{S\},\ \widehat{h}(a) = h(a) \text{ für } a \in \Sigma_1 \end{aligned}$$

definiert sind.

(c) Für den Nachweis des Abschlusses unter Durchschnitt mit regulären Sprachen wird ein beliebiger endlicher erkennender Automat

$$M = (V, Q, \delta, q_{in}, F)$$

betrachtet. Mit Hilfe der Menge

$$\Theta = \{[q, a, \widehat{q}] \mid q, \widehat{q} \in Q, a \in \Sigma_1\}$$

und der neuen Symbole F und S wird zunächst das Alphabet

$$\Sigma = \Theta \cup \Delta_1 \cup \{F, S\}$$

definiert. Für jede Tafel $h \in H_1$ wird eine Tafel $\widehat{h}$ durch

$$\begin{aligned} &\widehat{h}([q, a, \widehat{q}]) = \{[q, a_1, q_{i_1}][q_{i_1}, a_2, q_{i_2}] \ldots [q_{i_{n-1}}, a_n, \widehat{q}] \mid \\ &\qquad\qquad a_1 \ldots a_n \in h(a),\ q_{i_1}, \ldots, q_{i_n} \in Q, n \geq 1\} \text{ für } [q, a, \widehat{q}] \in \Theta \\ &\widehat{h}(S) = \{[q_{in}, S_1, \widehat{q}] \mid \widehat{q} \in F\} \cup \{\varepsilon \mid \varepsilon \in L(G_1), q_{in} \in F\} \text{ und} \\ &\widehat{h}(a) = \{a\} \text{ für } a \in \Delta_1 \cup \{F\} \end{aligned}$$

bestimmt. Man beachte dabei, daß eine eventuell vorhandene Produktion $S \to \varepsilon$ in $\widehat{h}([q,a,\widehat{q}])$ nicht berücksichtigt wird, die zugehörige Produktion jedoch in $\widehat{h}(S)$ aufgenommen wird. Außerdem wird eine finale Tafel h_{fin} durch

$$h_{\text{fin}}([q,a,\widehat{q}]) = \begin{cases} \{a\}, & \text{falls } a \in V \cap \Delta_1, \delta(q,a) = \widehat{q} \\ \{F\} & \text{sonst} \end{cases}$$

für $[q,a,\widehat{q}] \in \Theta$ und

$$h_{\text{fin}}(a) = \{F\}$$

für $a \in \Sigma - \Theta$ gegeben. Mit

$$H = \{h_{\text{fin}}\} \cup \{\widehat{h} \mid h \in H_1\}$$

erhalten wir ein ET0L-System $G = (\Sigma, H, S, \Delta_1 \cap V)$. Die Tafel h_{fin} sorgt am Abschluß einer Ableitung dafür, daß genau die Wörter aus $L(G_1) \cap L(M)$ durch G erzeugt werden.

(d) Wir betrachten den Abschluß von $\mathfrak{L}(\text{ET0L})$ unter Homomorphismus. Es sei $\varphi : \Delta_1^* \to \Theta^*$ ein Homomorphismus. Wir setzen

$$\Sigma = (\Sigma_1 - \Delta_1) \cup \Theta.$$

Mit der terminalen Tafel h_T von G_1 werde

$$\widehat{h}_T(a) = \begin{cases} \{\varphi(\alpha) \mid \alpha \in h_T(a) \cap \Delta_1\} \cup \{F\} & \text{für } a \in \Sigma_1 - \Delta_1 \\ \{a\} & \text{für } a \in \Theta \end{cases}$$

definiert. Für $h \in H_1 - \{h_T\}$ gelte

$$\widehat{h}(a) = \begin{cases} h(a) & \text{für } a \in \Sigma_1 - \Delta_1 \\ \{a\} & \text{für } a \in \Theta. \end{cases}$$

Setzen wir

$$G = (\Sigma, \widehat{H}, S_1, \Theta) \text{ mit } \widehat{H} = \{\widehat{h} \mid h \in H_1\},$$

so folgt $L(G) = \varphi(L_1)$. Wir bemerken, daß im Falle einer vorhandenen Produktion $S_1 \to \varepsilon$ das Wort $\varepsilon \in \varphi(L(G_1))$ durch eine Tafel $\widehat{h}$ mit $h \neq h_T$ erzeugt wird.

(e) Für den Nachweis des Abschlusses unter inversem Homomorphismus wird ein Homomorphismus $\varphi : \Delta^* \to \Delta_1^*$ betrachtet. Dabei können wir ohne Beschränkung der Allgemeinheit annehmen, daß $\Delta \cap \Sigma_1 = \emptyset$ gilt. Wir definieren eine Menge $K \subset (\Delta \cup \Delta_1)^*$ durch

$$K = \{y \in (\Delta \cup \Delta_1)^* \mid y = z_0 x_1 z_1 x_2 z_2 \dots x_k z_k,\ x_1 \dots x_k \in L(G_1),\ x_1, \dots, x_k \in \Delta_1,\ z_0, \dots, z_k \in \Delta^*, k \in \mathbb{N}_0\}.$$

Dann konstruieren wir ein ET0L-System

$$G = (\Sigma, \widehat{H}, S_1, \Delta_1 \cup \Delta)$$

mit $L(G) = K$ wie folgt. Es wird

$$\Sigma = \Sigma_1 \cup \Delta \text{ und } \widehat{H} = \{h_0\} \cup \{\widehat{h} \mid h \in H_1\}$$

gesetzt, wobei

$$\begin{aligned} &h_0(a) = \{xay \mid x, y \in \Delta \cup \{\varepsilon\}\} \text{ für } a \in \Delta_1, \\ &h_0(a) = \{a\} \text{ für } a \in (\Sigma_1 - \Delta_1) \cup \Delta,\ a \neq S_1, \\ &h_0(S_1) = \{xS_1y \mid x, y \in \Delta \cup \{\varepsilon\}\} \end{aligned}$$

und

$$\widehat{h}(a) = h(a) \text{ für } a \in \Sigma_1 \text{ sowie } \widehat{h}(a) = \{a\} \text{ für } a \in \Delta$$

für alle $h \in H_1$ gilt. Mit Hilfe der Tafeln $\widehat{h}$ erzeugt G zunächst $L_1 = L(G_1)$, worauf durch wiederholte Anwendung von h_0 die z_i „eingestreut" werden. Aufgrund der Eigenschaften von G_1 hat eine spätere Anwendung von $\widehat{h}$ keine Wirkung. $h_0(S_1)$ wird benötigt, um auch im Fall $\varepsilon \in L_1$ ein $z_0 \in K$ zu erhalten.
Als nächstes wird die Menge

$$M = \{\varphi(y_1)y_1\varphi(y_2)y_2 \ldots \varphi(y_n)y_n \mid n \geq 0, y_i \in \Delta, 1 \leq i \leq n\}$$

betrachtet. Sie ist regulär, da sie von einer Grammatik mit den Produktionen

$$X_0 \to X_0\varphi(y)y \text{ für alle } y \in \Delta \text{ sowie } X_0 \to \varepsilon$$

erzeugt wird. Offenbar gilt

$$K \cap M = \{\varphi(y_1)y_1\varphi(y_2)y_2 \ldots \varphi(y_n)y_n \mid \begin{array}{l} n \geq 0, y_i \in \Delta, 1 \leq i \leq n, \\ \varphi(y_1) \ldots \varphi(y_n) \in L(G_1)\}. \end{array}$$

Das bedeutet, daß für $y = y_1 \ldots y_n$ mit $y_1, \ldots, y_n \in \Delta$ die Äquivalenz

$$\varphi(y) \in L(G_1) \Longleftrightarrow \varphi(y_1)y_1 \ldots \varphi(y_n)y_n \in K \cap M$$

erfüllt ist. Wird jetzt ein Homomorphismus $\psi : (\Delta \cup \Delta_1)^* \to \Delta$ durch

$$\psi(a) = \begin{cases} a & \text{für } a \in \Delta \\ \varepsilon & \text{für } a \in \Delta_1 \end{cases}$$

definiert, so erhalten wir

$$\psi(K \cap M) = \varphi^{-1}(L(G_1)).$$

Da $K \in \mathfrak{L}(\text{ET0L})$ und $M \in \mathfrak{L}_3$ bereits nachgewiesen worden ist, folgt aus (c) $K \cap M \in \mathfrak{L}(\text{ET0L})$, woraus sich wegen (d) die Beziehung

$$\psi(K \cap M) = \varphi^{-1}(L(G_1)) \in \mathfrak{L}(\text{ET0L})$$

ergibt. □

13.3 Kombinatorische Eigenschaften von ET0L-Sprachen

Es seien Δ_1 und Δ_2 Alphabete mit $\Delta_1 \subset \Delta_2$. Weiter gelte $v \in \Delta_2^*$. Mit $\#_{\Delta_1} v$ bezeichnen wir die Anzahl der Vorkommen von Symbolen aus Δ_1 in v. Speziell für $\Delta_1 = \{a\}$ schreiben wir $\#_a v$.

Mit Hilfe des nächsten Satzes kann gezeigt werden, daß einige Sprachen keine ET0L-Sprachen sind. Er spielt also eine ähnliche Rolle wie das Iterationstheorem (Satz 5.8) für kontextfreie Sprachen.

Satz 13.20 Es sei $L \subset \Delta^*$ eine ET0L-Sprache. Dann existiert für jede nichtleere Teilmenge $\Delta_1 \subset \Delta$ eine Zahl $k \in \mathbb{N}$, so daß jedes Wort $v \in L$ eine der folgenden Eigenschaften erfüllt:

(a) $\#_{\Delta_1} v \leq 1$,
(b) v enthält ein Teilwort w mit $|w| \leq k$ und $\#_{\Delta_1} w \geq 2$,
(c) es existiert eine unendliche Teilmenge $M \subset L$, so daß $\#_{\Delta_1} w = \#_{\Delta_1} v$ für alle $w \in M$ gilt.

Beweis: Falls L endlich ist, wird für k die Länge des längsten Wortes von L gewählt. Offenbar ist dann (a) oder (b) erfüllt. Daher sei L im folgenden unendlich, und L werde von einem ET0L-System $G = (\Sigma, H, S, \Delta)$ erzeugt. Da das leere Wort ε die Eigenschaft (a) hat, kann ohne Beschränkung der Allgemeinheit angenommen werden, daß L ε-frei ist. Nach Satz 13.16 kann somit G als propagierend vorausgesetzt werden, wobei G sogar die Form eines nach Satz 13.18 konstruierten synchronisierten ET0L-Systems hat. G besitzt also die folgenden Eigenschaften:

(α) $S \in \Sigma - \Delta$.
(β) Es existiert ein $F \in \Sigma - (\Delta \cup \{S\})$, so daß $h(F) = \{F\}$ und $h(a) = \{F\}$ für alle $a \in \Delta$ und $h \in H$ gilt.
(γ) Für alle $a \in \Sigma$ und $h \in H$ gilt $h(a) \subset \Delta^+$, $h(a) = \{F\}$ oder $h(a) \subset (\Sigma - (\Delta \cup \{S, F\}))^+$.

Wir konstruieren ein äquivalentes EPT0L-System $\bar{G} = (\bar{\Sigma}, \bar{H}, \bar{S}, \Delta)$, mit dessen Hilfe die Aussage des Satzes nachgewiesen wird. Wir setzen

$$\bar{\Sigma} = \Delta \cup \{F, \bar{S}\} \cup \{[a, t] \mid a \in \Sigma - (\Delta \cup \{F\}),\ t = 0, 1, 2\}$$

mit dem neuen Symbol $\bar{S}$ und

$$\bar{H} = \{\bar{h} \mid h \in H\}.$$

Für jedes $h \in H$ wird $\bar{h} \in \bar{H}$ durch die folgenden Produktionen definiert. Für das Axiom $\bar{S}$ gibt es die Produktionen

$$\bar{S} \to_{\bar{h}} [S, 0],\ \bar{S} \to_{\bar{h}} [S, 1],\ \bar{S} \to_{\bar{h}} [S, 2].$$

Weiter werde

$$a \to_{\bar{h}} F \text{ für } a \in \Delta \cup \{F\}$$

gesetzt. Für $a \in \Sigma - (\Delta \cup \{F\})$ sind die Produktionen durch

$[a,0] \to_{\bar{h}} F$,
$[a,0] \to_{\bar{h}} \alpha$ für $\alpha \in \Delta^+$ mit $a \to_h \alpha$, $\#_{\Delta_1}\alpha = 0$,
$[a,0] \to_{\bar{h}} [b_1,0]\dots[b_r,0]$ für $b_1,\dots,b_r \in \Sigma - (\Delta \cup \{F\})$ mit $a \to_h b_1 \dots b_r$,
$[a,1] \to_{\bar{h}} F$,
$[a,1] \to_{\bar{h}} \alpha$ für $\alpha \in \Delta^+$ mit $a \to_h \alpha$, $\#_{\Delta_1}\alpha = 1$,
$[a,1] \to_{\bar{h}} [b_1,t_1]\dots[b_r,t_r]$ für $b_1,\dots,b_r \in \Sigma - (\Delta \cup \{F\})$ mit $a \to_h b_1 \dots b_r$
und $t_i \in \{0,1\}$, $\sum_{i=1}^{r} t_i = 1$,
$[a,2] \to_{\bar{h}} F$,
$[a,2] \to_{\bar{h}} \alpha$ für $\alpha \in \Delta^+$ mit $a \to_h \alpha$, $\#_{\Delta_1}\alpha > 1$,
$[a,2] \to_{\bar{h}} [b_1,t_1]\dots[b_r,t_r]$ für $b_1,\dots,b_r \in \Sigma - (\Delta \cup \{F\})$ mit $a \to_h b_1 \dots b_r$
und $t_i \in \{0,1,2\}$, $\sum_{i=1}^{r} t_i \geq 2$

gegeben. Die Nichtterminalsymbole $[a,0]$, $[a,1]$ und $[a,2]$ haben in einer erfolgreichen Ableitung gemäß $\bar{G}$ die folgende Bedeutung.

$[a,0]$: a trägt zum Ergebnis dieser Ableitung gemäß $\bar{G}$ *kein* Vorkommen eines Symbols aus Δ_1 bei.

$[a,1]$: a trägt zum Ergebnis dieser Ableitung gemäß $\bar{G}$ *genau ein* Vorkommen eines Symbols aus Δ_1 bei.

$[a,2]$: a trägt zum Ergebnis dieser Ableitung gemäß $\bar{G}$ *mindestens zwei* Vorkommen eines Symbols aus Δ_1 bei.

Wir betrachten dazu ein Beispiel. Es sei $\Delta_1 = \{a,b\}$, und es existieren die Produktionen

$$S \to xy, \quad x \to xz, \quad y \to y^2, \quad y \to c, \quad x \to a, \quad z \to b.$$

Dann gibt es eine Ableitung

$$S \Longrightarrow xy \Longrightarrow xzy^2 \Longrightarrow abc^2 \text{ gemäß } G,$$

die durch eine Ableitung

$$\bar{S} \Longrightarrow [S,2] \Longrightarrow [x,2][y,0] \Longrightarrow [x,1][z,1][y,0]^2 \Longrightarrow abc^2 \text{ gemäß } \bar{G}$$

simuliert wird.

Die Konstruktion von $\bar{G}$ zeigt, daß $L(G) = L(\bar{G})$ gilt.

Die Ableitungen gemäß $\bar{G}$ werden nun genauer betrachtet. Falls eine Ableitung mit den Produktionen $\bar{S} \to [S,0]$ oder $\bar{S} \to [S,1]$ beginnt, erfüllt das Ergebnis der Ableitung offensichtlich die Eigenschaft (a). Daher nehmen wir im folgenden an, daß im ersten Schritt einer Ableitung D die Produktion $\bar{S} \to [S,2]$ verwendet wird.

$$\text{ftr}(D) = ((w_0,h_0),\dots,(w_{m-1},h_{m-1}),w_m = v)$$

wird als *volle Spur* von D bezeichnet, die aus dem jeweiligen j-ten Wort der Ableitung sowie der Tafel für den nächsten Schritt besteht. Es sei i die größte Zahl, für die w_i ein Vorkommen eines Typ-2-Symbols, d.h. eines Symbols der Form $[a,2]$, enthält. Im obigen Beispiel ist dies $i = 2$. Wir betrachten im folgenden zwei Fälle.

(1) Es existieren zwei Zahlen $r, s \in \{i+1, \ldots, m-1\}$ mit $r < s$, so daß

$$\text{alph}(w_r) = \text{alph}(w_s)$$

gilt und mindestens ein Vorkommen eines Symbols aus w_r, etwa c, zu dem Wort w_s ein Wort der Form $\alpha c \beta$ mit $\alpha\beta \neq \varepsilon$ beiträgt, d.h., es gilt

$$w_r = w' c w'' \Longrightarrow^+ w_s = \bar{w}\alpha c \beta \bar{\bar{w}} \quad \text{mit } w' \Longrightarrow^* \bar{w},\ w'' \Longrightarrow^* \bar{\bar{w}} \text{ und } \alpha\beta \neq \varepsilon.$$

(2) Es existieren keine Zahlen $r, s \in \{i+1, \ldots, m-1\}$, die die Eigenschaften aus (1) erfüllen.

Wir betrachten zunächst den Fall (1). Für alle $n \in \mathbb{N}_0$ ändern wir D ab zu einer Ableitungen D_n, die wie folgt konstruiert wird. Zunächst werden die Tafeln $h_0, \ldots, h_{r-1}$ wie in D angewendet, so daß das Wort w_r entsteht. Auf w_r wird dann die Folge der Tafeln $(h_r \ldots h_{s-1})^n$ derart angewendet, daß jedes Vorkommen eines Symbols a (außer dem gegebenen festen Vorkommen c) bei jeder Iteration von $h_r \ldots h_{s-1}$ ein beliebiges Wort liefert, das aus einem Vorkommen von a in w_r beim Übergang nach w_s gemäß D erzeugt wird. Dies ist wegen $\text{alph}(w_r) = \text{alph}(w_s)$ immer möglich. Außerdem wird das gegebene Vorkommen des Symbols c so ersetzt, daß es in jeder Iteration von $h_r \ldots h_{s-1}$ das Wort $\alpha c \beta$ liefert. Nach Anwendung von $(h_r \ldots h_{s-1})^n$ erhalten wir das Wort z_n. Wir machen darauf aufmerksam, daß es für die weiteren Überlegungen keine Bedeutung hat, daß z_n nicht jedes Symbol von w_r enthalten muß.

Wegen $\text{alph}(z_n) \subset \text{alph}(w_r) = \text{alph}(w_s)$ kann abschließend die Folge der Tafeln $h_s \ldots h_{m-1}$ so auf z_n angewendet werden, daß jedes Vorkommen eines Symbols a in z_n ein beliebiges Wort liefert, das aus einem Vorkommen von a in w_s gemäß D erzeugt wird. Das Wort

$$h_0 \ldots h_{r-1} (h_r \ldots h_{s-1})^n h_s \ldots h_{m-1}$$

kann somit als Kontrollwort von D_n aufgefaßt werden. Falls $\rho(D_n)$ das Ergebnis der Ableitung D_n bezeichnet, erhalten wir

$$\#_{\Delta_1}(\rho(D_n)) = \#_{\Delta_1} v,$$

da die Wörter $w_r, w_{r+1}, \ldots, w_s, \ldots$ keine Typ-2-Symbole enthalten und daher wegen der Gestalt der Produktionen von $\bar{G}$ die Anzahl der Symbole aus Δ_1 in $\rho(D_n)$ ebenso wie in v durch w_r bestimmt und gleich ist. So bedeutet zum Beispiel $w_r = [a,1][b,0][c,1]$, daß diese Anzahl gleich 2 ist.

Da $\bar{G}$ propagierend ist und nach Voraussetzung des betrachteten Falls $\alpha\beta \neq \varepsilon$ gilt, folgt

$$|z_0| < |z_1| < |z_2| < \ldots.$$

Wir erhalten eine Folge von Zahlen $j_1, j_2, \ldots \in \mathbb{N}$ mit

$$\begin{aligned} |v| &< |z_{j_1}| \leq |\rho(D_{j_1})|, \\ |\rho(D_{j_1})| &< |z_{j_2}| \leq |\rho(D_{j_2})|, \\ |\rho(D_{j_2})| &< |z_{j_3}| \leq |\rho(D_{j_3})|, \\ &\vdots \qquad\qquad . \end{aligned}$$

Die Menge $\{\rho(D_{j_i}) \mid i \in I\!N\} \subset L$ erfüllt somit die Eigenschaft (c).

Wir kommen zum Fall (2) und betrachten ein Zeichen $[a,2]$ aus w_i. Es wird gezeigt, daß der Beitrag von $[a,2]$ zu $v = w_m$ aus höchstens $k_{\bar{G}}$ Symbolen besteht, wobei $k_{\bar{G}}$ eine nur von $\bar{G}$ abhängige Konstante ist. Andererseits liefert $[a,2]$ nach Definition mindestens zwei Symbole aus Δ_1, so daß insgesamt die Eigenschaft (b) erfüllt ist.

Es sei E der Baum der Teilableitung von D mit der Wurzel in $[a,2]$. Dann werden die Marken von E so abgeändert, daß jeder Knoten mit der Marke d die Marke $(d, \mathrm{alph}(w_j))$ erhält, wobei dieser Knoten dem Vorkommen von d im Wort w_j, $j = 1, \ldots, m-1$, von D entspricht. Dadurch entsteht der Baum $\bar{E}$, dessen Wurzel mit $([a,2], \mathrm{alph}(w_i))$ markiert ist. Aus $[a,2]$ können in einem Schritt höchstens

$$p = \max\{|\alpha| \mid \bigvee_{\bar{h} \in \bar{H},\, a \in \bar{\Sigma}} \alpha \in \bar{h}(a)\}$$

Symbole abgeleitet werden. Von diesen gehen entsprechend viele Teilbäume $\bar{E}_j$ aus, für die jeweils die folgende Eigenschaft gilt:

$(*)$ Es sei $(e_1, e_2, \ldots, e_t)$ mit $t \geq 2$ ein Weg in $\bar{E}_j$, wobei der Knoten e_1 näher an der Wurzel ist als der Knoten e_t und beide Knoten die gleiche Marke besitzen. Dann hat jeder der Knoten $e_1, \ldots, e_{t-1}$ den ausgehenden Grad 1.

Zum Nachweis von $(*)$ betrachten wir die Knoten e_1 und e_t und ihre Marken $(d, \mathrm{alph}(w_r)) = (d, \mathrm{alph}(w_s))$. Hätte einer der Knoten $e_1, \ldots, e_{t-1}$ einen ausgehenden Grad > 1, so würde, da $\bar{G}$ propagierend ist, das Symbol d aus w_r einen Beitrag $\alpha d \beta$ zu w_s mit $\alpha\beta \neq \varepsilon$ liefern, was der Voraussetzung des Falls (2) widerspricht.

Behauptung: Es sei T ein Baum, der $(*)$ erfüllt und q Marken benutzt. Ist der ausgehende Grad eines jeden Knotens durch p beschränkt, dann ist die Anzahl der Blätter von T durch p^q beschränkt.

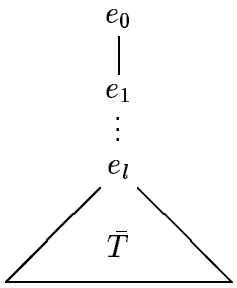
Der Beweis der Behauptung erfolgt durch Induktion über q. Für $q = 1$ ist die Aussage offensichtlich erfüllt, da nach $(*)$ alle Knoten bis auf e_t $(t \geq 2)$ den ausgehenden Grad 1 besitzen. Die Induktionsannahme ist, daß die Behauptung für alle Bäume gilt, die $(*)$ erfüllen und nicht mehr als k Marken benutzen. Es sei nun $q = k+1$ und c die Marke der Wurzel von T.

(α) Falls kein anderer Knoten von T mit c markiert ist, betrachten wir die $\leq p$ Teilbäume von T, die von den Kindern von c ausgehen. Die Anzahl der Blätter dieser Teilbäume ist nach der Induktionsannahme jeweils durch p^k beschränkt. Insgesamt ist also die Anzahl der Blätter von T durch

$$p \cdot p^k = p^{k+1} = p^q$$

beschränkt.

(β) Anderenfalls sei $(e_0, e_1, \ldots, e_l)$ ein längster Weg in T, so daß e_0 die Wurzel von T ist und e_0 und e_l mit c markiert sind. Nach $(*)$ hat T die Form



Für den Teilbaum $\bar{T}$ mit der Wurzel e_l gilt, daß keiner seiner Knoten mit Ausnahme von e_l mit c markiert ist. Daher ist die Anzahl der Blätter von $\bar{T}$ und damit von T wie in (α) durch p^q beschränkt.

Die Konstruktion von $\bar{E}$ aus E zeigt, daß nicht mehr als

$$q = \text{card}(\bar{\Sigma}) \cdot 2^{\text{card}(\bar{\Sigma})}$$

Marken benutzt werden. Dabei gibt $\text{card}(\bar{\Sigma})$ die Anzahl der möglichen d und $2^{\text{card}(\bar{\Sigma})}$ die Anzahl der möglichen $\text{alph}(w_s)$ an. Jeder der $\leq p$ Teilbäume $\bar{E}_j$ hat nach der Behauptung höchstens p^q Blätter, so daß insgesamt die Anzahl der Blätter durch

$$k_{\bar{G}} = p^q \cdot p$$

beschränkt ist. Dies ist eine nur von $\bar{G}$ und damit von G abhängige Konstante. Da $[a, 2]$ die Wurzel von E ist, folgt die Eigenschaft (b) des Satzes. □

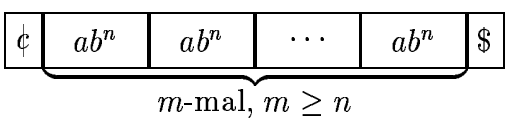
Satz 13.21 Die Sprache $L = \{(ab^n)^m \mid m \geq n \geq 1\}$ ist keine ET0L-Sprache.

Beweis: Es seien $\Delta = \{a, b\}$ und $\Delta_1 = \{a\}$ Alphabete. Für eine beliebige Zahl $k \in \mathbb{N}$ erfüllt das Wort $v = (ab^{k+1})^{k+1} \in L$ offenbar weder die Eigenschaft (a) noch die Eigenschaft (b) aus dem vorangegangenen Satz 13.20. Da jedoch nur endlich viele Wörter $w \in L$ mit

$$\#_{\{a\}} w = \#_{\{a\}} v = k + 1$$

existieren, ist für v auch die Eigenschaft (c) nicht erfüllt. Folglich kann L keine ET0L-Sprache sein. □

Es kann gezeigt werden, daß die Sprache $L = \{(ab^n)^m \mid m \geq n \geq 1\}$ kontextsensitiv ist. Der Beweis erfolgt durch die Angabe eines linear beschränkten Automaten M mit $L(M) = L$, dessen Band zu Beginn der Arbeit wie folgt beschrieben ist:



Durch einen n-maligen Hin- und Herlauf bei jeweiliger Markierung eines unmarkierten Vorkommen von b (zu b') in jedem Abschnitt ab^n kann überprüft werden, ob ein

Symbol a jeweils von n Symbolen b gefolgt wird. Anschließend markiert man das a des ersten Abschnitts und zählt diesen Abschnitt durch eine Markierung eines Zeichen b' (zu b'') dieses ersten Abschnitts. Dann sucht man das nächste unmarkierte a, markiert es und läuft zurück zum ersten Abschnitt, wo der zweite Abschnitt durch die Markierung des nächsten b' gezählt wird. Dann läuft man wieder nach rechts zum nächsten unmarkierten Vorkommen von a usw. Sind nach mehrmaligem Hin- und Herlaufen am Ende alle Symbole b' im ersten Abschnitt markiert, so gilt $m \geq n$, und das Wort wird akzeptiert. Anderenfalls ist kein Wort aus L vorgegeben worden.

Definition 13.9 Es sei $L \subset \Delta^*$ eine nichtleere Sprache und $\Theta \subset \Delta$ eine nichtleere Teilmenge. Dann *steht* Θ *in* L *beieinander* (Θ *clustered in* L), wenn $n, m \in \mathbb{N}$, $n, m \geq 2$, existieren, so daß es in jedem Wort $w \in L$ mit $\#_\Theta w \geq n$ ein Teilwort v mit $|v| \leq m$ und $\#_\Theta v \geq 2$ gibt. □

Beispiel 13.7 Für die Sprache $L = \{(aba^2)^{2^r} \mid r \in \mathbb{N}_0\}$ gelten die folgenden Eigenschaften:

(1) Die Menge $\{a\}$ steht in L beieinander, da mit $n = m = 2$ alle Wörter $w \in L$ die Eigenschaft $\#_a w \geq n = 2$ erfüllen und das Teilwort $v = a^2$ mit $|v| \leq m = 2$ und $\#_a v \geq 2$ enthält.

(2) Die Menge $\{b\}$ steht in L beieinander, da mit $n = 2$ und $m = 5$ alle Wörter $w \in L$ mit $\#_b w \geq n = 2$ die Eigenschaft $r \geq 1$ erfüllen und somit das Teilwort $v = ba^2ab$ mit $|v| = m = 5$ und $\#_b v = 2$ enthalten. □

Beispiel 13.8 In der Sprache $L = \{w \in \{a, b\}^+ \mid \#_a w = 2^r, r \in \mathbb{N}_0\}$ stehen weder $\{a\}$ noch $\{b\}$ beieinander. Für $\{a\}$ betrachten wir $w = (ab^m)^{2^n} \in L$ für zwei beliebige Zahlen $n, m \in \mathbb{N}$. Dann gilt $\#_a w \geq n$, und es gibt kein Teilwort v mit $|v| \leq m$ und $\#_a v \geq 2$.

Ein ähnlicher Beweis existiert für $\{b\}$ mit dem Wort $w = (a^{2^m}b)^{2^n} \in L$. □

Satz 13.22 Es sei $L \subset \Delta^*$ eine ET0L-Sprache, $\Delta_1 \cup \Delta_2$ eine Partition von Δ, und es existiere eine Abbildung $\varphi : \mathbb{N} \to \mathbb{N}$, so daß für alle Wörter $w \in L$ die Relation

$$\#_{\Delta_2} w < \varphi(\#_{\Delta_1} w)$$

gilt. Dann steht Δ_1 in L beieinander.

Beweis: Für $w \in \Delta^*$ betrachten wir die Sprache $L_w = \{v \in L \mid \#_{\Delta_1} v = \#_{\Delta_1} w\} \subset \Delta^*$. Für jedes Wort $v \in L_w$ gilt

$$|v| = \#_{\Delta_1} v + \#_{\Delta_2} v < \#_{\Delta_1} v + \varphi(\#_{\Delta_1} v) = \#_{\Delta_1} w + \varphi(\#_{\Delta_1} w) = k_w.$$

Das bedeutet, daß für ein festes w die Länge der Wörter $v \in L_w$ durch k_w beschränkt und somit L_w endlich ist. Folglich ist die Eigenschaft (c) aus Satz 13.20 nicht erfüllbar, so daß eine der Eigenschaften (a) oder (b) gelten muß. Es sei k die Konstante aus Satz 13.20. Da Wörter $w \in L$, die die Eigenschaft (a) erfüllen, in Definition 13.9 nicht betrachtet werden, werden nur noch Wörter w berücksichtigt, für die $\#_{\Delta_1} w \geq 2$ gilt

und die die Eigenschaft (b) erfüllen. Für solche Wörter w gelten mit $m = k$ und $\Theta = \Delta_1$ die Bedingungen von Definition 13.9 (wenn es nur Wörter mit $\#_{\Delta_1} w \leq 1$ gibt, ist diese Bedingung trivialer Weise erfüllt). Damit ist der Satz bewiesen. □

Wir wollen zur Verdeutlichung ein Beispiel eines trivialen Falls angeben. Es gilt offenbar $\{ab\} \in \mathfrak{L}(\text{ET0L})$. Mit $\Delta_1 = \{a\}$, $\Delta_2 = \{b\}$ und $\varphi(n) = n + 1$ sind die Voraussetzungen von Satz 13.22 erfüllt, so daß $\{a\}$ in $\{ab\}$ beieinander steht. Dies ergibt sich auch in trivialer Weise aus Definition 13.9.

Satz 13.23 Die Sprache $L = \{w \in \{a,b\}^* \mid \#_b w = 2^{\#_a w}\}$ ist keine ET0L-Sprache.

Beweis: Es seien $\Delta = \{a,b\}$, $\Delta_1 = \{a\}$ und $\Delta_2 = \{b\}$ Alphabete, und durch $\varphi(n) = 2^n + 1$ für $n \in \mathbb{N}$ sei eine Abbildung $\varphi : \mathbb{N} \to \mathbb{N}$ definiert. Dann gilt für alle Wörter $w \in L$

$$\#_{\Delta_2} w = 2^{\#_a w} = 2^{\#_{\Delta_1} w} < \varphi(\#_{\Delta_1} w).$$

Falls L eine ET0L-Sprache ist, steht nach Satz 13.22 $\Delta_1 = \{a\}$ in L beieinander. Folglich existieren $n, m \in \mathbb{N}$ mit den Eigenschaften aus Definition 13.9. Wir wählen $n' \geq n$ mit $m \cdot n' \leq 2^{n'}$ und bestimmen damit $r = 2^{n'} - m \cdot n'$. Mit diesen Werten betrachten wir das Wort

$$w = (ab^m)^{n'} b^r \in L.$$

Es besitzt kein Teilwort v mit $|v| \leq m$ und $\#_a v \geq 2$, was der Definition 13.9 widerspricht. Folglich kann L keine ET0L-Sprache sein. □

Definition 13.10 Es sei $L \subset \Delta^*$ eine nichtleere Sprache und $\Delta_1 \cup \Delta_2$ eine Partition von Δ. Δ_1 und Δ_2 heißen *L-äquivalent*, wenn für alle Wörter $v, w \in L$

$$\#_{\Delta_1} v = \#_{\Delta_1} w \Longleftrightarrow \#_{\Delta_2} v = \#_{\Delta_2} w$$

gilt. □

Beispiel 13.9 Für die Sprache $L = \{w \in \{a,b\}^+ \mid \#_a w = 2^n, \quad \#_b w = 3^n, n \in \mathbb{N}_0\}$ sind $\{a\}$ und $\{b\}$ offensichtlich L-äquivalent. □

Satz 13.24 Es sei $L \subset \Delta^*$ eine nichtleere ET0L-Sprache und $\Delta_1 \cup \Delta_2$ eine Partition von Δ. Sind Δ_1 und Δ_2 L-äquivalent, dann stehen Δ_1 und Δ_2 in L beieinander.

Beweis: Es sei k die Konstante aus Satz 13.20. Falls Δ_1 und Δ_2 L-äquivalent sind, sind die Sprachen

$$L_w^i = \{v \in L \mid \#_{\Delta_i} v = \#_{\Delta_i} w\}, \; i = 1, 2,$$

für jedes Wort $w \in L$ endlich. Somit kommt die Eigenschaft (c) aus Satz 13.20 nicht in Frage, so daß eine der Eigenschaften (a) oder (b) gelten muß. Da Wörter $w \in L$, die die Eigenschaft (a) erfüllen, in Definition 13.9 nicht betrachtet werden, werden im Falle von Δ_1 nur noch Wörter w berücksichtigt, für die $\#_{\Delta_1} w \geq 2$ gilt und die die Eigenschaft (b) erfüllen. Für solche Wörter w gelten mit $m = k$ und $\Theta = \Delta_1$ die Bedingungen von Definition 13.9 (wenn es nur Wörter mit $\#_{\Delta_1} w \leq 1$ gibt, ist die Bedingung trivialer Weise erfüllt). Aus Symmetriegründen gelten dieselben Aussagen auch für Δ_2. Damit ist der Satz bewiesen. □

Satz 13.25 $L = \{w \in \{a,b\}^+ \mid \#_a w = 2^n, \#_b w = 3^n, n \in \mathbb{N}_0\} \notin \mathfrak{L}(\text{ET0L})$.

Beweis: Die Menge $\{a\}$ steht in L nicht beieinander, da für zwei beliebige Zahlen $n, m \in \mathbb{N}$ in dem Wort

$$w = (ab^m)^{2^{n'}} b^r \in L$$

für ein $n' \in \mathbb{N}$ mit $2^{n'} \geq n$ und $m \cdot 2^{n'} \leq 3^{n'}$ sowie $r = 3^{n'} - m2^{n'}$ kein Teilwort v mit $|v| \leq m$ und $\#_a v \geq 2$ existiert, was der Definition 13.9 widerspricht.

Nach Beispiel 13.9 sind $\{a\}$ und $\{b\}$ L-äquivalent. Ist L eine ET0L-Sprache, so steht nach Satz 13.24 speziell $\{a\}$ in L beieinander, was ein Widerspruch zur ersten Aussage des Beweises ist. □

Definition 13.11 Es seien L_1 und L_2 Sprachen über den Alphabeten Δ_1 bzw. Δ_2. Dann heißt

$$L_1 \perp L_2 = \{v_1 w_1 \ldots v_r w_r \mid r \in \mathbb{N},\ v_1, \ldots, v_r \in \Delta_1^*,\ w_1, \ldots, w_r \in \Delta_2^*,\ v_1 \ldots v_r \in L_1,\ w_1 \ldots w_r \in L_2\}$$

das *Shuffle-Produkt* von L_1 und L_2.

Beispiel 13.10 Das Shuffle-Produkt der Sprachen $L_1 = \{a^{2^n} \mid n \in \mathbb{N}_0\}$ und $L_2 = \{b^{3^n} \mid n \in \mathbb{N}_0\}$ ist

$$L_1 \perp L_2 = \{w \in \{a,b\}^* \mid \#_a w = 2^n,\ \#_b w = 3^n,\ n, m \in \mathbb{N}_0\}. \quad \square$$

Satz 13.26 $\mathfrak{L}(\text{ET0L})$ ist nicht unter Shuffle-Produkt abgeschlossen.

Beweis: Wir betrachten die beiden D0L-Sprachen

$$L_1 = \{a^{2^n} b^{3^n} \mid n \in \mathbb{N}_0\} \quad \text{und} \quad L_2 = \{b^{3^n} a^{2^n} \mid n \in \mathbb{N}_0\}$$

und zeigen, daß $L = L_1 \perp L_2$ keine ET0L-Sprache ist.

Die Mengen $\{a\}$ und $\{b\}$ sind L-äquivalent. Gilt nämlich $\#_a v = \#_a w = 2^n + 2^m$ für $v, w \in L$, wobei sich zum Beispiel 2^n auf den Anteil der Vorkommen von a aus L_1 und 2^m auf den der Vorkommen von a aus L_2 bezieht, so sind für diese Zahl die Werte n und m eindeutig festgelegt und bestimmen damit auch eindeutig $\#_b v = \#_b w = 3^n + 3^m$. Dieser Schluß ist umkehrbar. Ist L eine ET0L-Sprache, so stehen nach Satz 13.24 sowohl $\{a\}$ als auch $\{b\}$ beieinander.

Andererseits kann mit dem Wort

$$w = (ab^m)^{2^n} b^r (ab^m)^{2^n} b^r \text{ mit } n', m, r \text{ wie im Beweis von Satz 13.25,}$$

das ein Shuffle-Produkt von $v = a^{2^{n'}} b^{3^{n'}}$ mit $w = b^{3^{n'}} a^{2^{n'}}$ ist, ähnlich wie im Beweis von Satz 13.25 gezeigt werden, daß $\{a\}$ in L nicht beieinander steht. Dies ist ein Widerspruch zur ersten Aussage des Beweises, so daß L keine ET0L-Sprache ist. □

13.4 Inklusionsbeziehungen der Familien der ET0L-Sprachen

Auch für E0L-Sprachen können kombinatorische Eigenschaften betrachtet werden. Zunächst bezeichnen wir für ein Wort $w \in \Sigma^*$ und ein Teilalphabet $\Theta \subset \Sigma$ mit

$$\text{pres}_\Theta(w)$$

das Wort, das sich aus w durch Entfernung aller Symbole ergibt, die nicht zu Θ gehören.

Definition 13.12 Es sei $L \subset \Sigma^*$ eine Sprache und $\Theta \subset \Sigma$, $\Theta \neq \emptyset$. L heißt Θ-*determiniert*, wenn für alle $k \in \mathbb{N}$ eine Zahl $n_k \in \mathbb{N}$ existiert, so daß für alle $v, w \in L$ mit $|v|, |w| > n_k$ und $v = v_1 v' v_2$, $w = v_1 w' v_2$ und $|v'|, |w'| < k$ die Beziehung

$$\text{pres}_\Theta(v) = \text{pres}_\Theta(w)$$

gilt. □

Beispiel 13.11 Die Sprache

$$L = \{¢w'¢w'¢w' \mid w \in \{a, b\}^*\}$$

ist $\{a\}$-determiniert. Dies erkennen wir wie folgt. Zu $k \in \mathbb{N}$ wählen wir $n_k = 3k + 3$ und betrachten $v = v_1 u v_2, w = v_1 u' v_2 \in L$ mit $|v|, |w| > 3k + 3$. Die Wörter v und w haben die Gestalt $v = ¢w'¢w'¢w'$ bzw. $w = ¢w''¢w''¢w''$. Wegen $|v|, |w| > 3k + 3$ gilt $|w'|, |w''| > k$. Da $|u| < k$ gilt, kann somit w' kein echtes Teilwort von u sein. Daher ist $¢w'¢$ ein Präfix von v_1 oder $¢w'$ ein Postfix von v_2. Aufgrund der jeweils beiden Darstellungen von v und w muß dann $w' = w''$, also $v = w$ und somit erst recht $\text{pres}_a(v) = \text{pres}_a(w)$ gelten. □

Satz 13.27 Es sei $L \in \mathfrak{L}(\text{E0L})$ eine Θ-determinierte Sprache. Dann existieren $C, D \in \mathbb{N}$, so daß für jedes Wort $w \in L$ mit $\#_\Theta w > C$ die Beziehung $|w| < D^{\#_\Theta w}$ gilt.

Beweis: [22], Theorem II.4.6, S. 88. □

Beispiel 13.12 Wir zeigen, daß die Sprache L aus Beispiel 13.11 keine E0L-Sprache ist. Anderenfalls existieren $C, D \in \mathbb{N}$, die Satz 13.27 erfüllen. Wir wählen $n, m \in \mathbb{N}$ mit $n > \frac{C}{3}$ und $m = D^{3n}$ und betrachten das Wort

$$w = ¢a^n b^m ¢a^n b^m ¢a^n b^m \in L.$$

Einerseits gilt offensichtlich $\#_a w = 3n > C$, andererseits ist jedoch $|w| = 3+3n+3m > D^{3n} = D^{\#_a w}$, was im Widerspruch zu $|w| < D^{\#_a w}$ steht. □

Satz 13.28 Es gelten die Inklusionsbeziehungen

$$\mathfrak{L}_2 \subsetneqq \mathfrak{L}(\text{E0L}) \subsetneqq \mathfrak{L}(\text{ET0L}) \subsetneqq \mathfrak{L}_1 \text{ und } \mathfrak{L}(\text{0L}) \subsetneqq \mathfrak{L}(\text{E0L}).$$

Die Sprachfamilien $\mathfrak{L}_2$ und $\mathfrak{L}(\text{0L})$ sind unvergleichbar.

Beweis: Die Beziehungen $\mathfrak{L}(0L) \subset \mathfrak{L}(E0L) \subset \mathfrak{L}(ET0L)$ sind trivial (siehe Satz 13.9), und $\mathfrak{L}_2 \subset \mathfrak{L}(E0L)$ gilt nach Satz 13.5(b).

In Satz 13.4 haben wir bereits $\mathfrak{L}_2 \not\subset \mathfrak{L}(0L)$ gezeigt. Außerdem gilt

$$\{a^{2^n} \mid n \in I\!N_0\} \in \mathfrak{L}(E0L) - \mathfrak{L}_2,$$

woraus $\mathfrak{L}(0L) \not\subset \mathfrak{L}_2$ und damit die Unvergleichbarkeit der Familien $\mathfrak{L}_2$ und $\mathfrak{L}(0L)$ folgt. Diese Sprache beweist auch $\mathfrak{L}_2 \subsetneqq \mathfrak{L}(E0L)$.

Da es endliche Sprachen gibt, die nicht zu $\mathfrak{L}(0L)$ gehören (siehe Beweis von Satz 13.2), folgt nach Satz 13.10 die echte Inklusion $\mathfrak{L}(0L) \subsetneqq \mathfrak{L}(E0L)$.

Um $\mathfrak{L}(E0L) \subsetneqq \mathfrak{L}(ET0L)$ zu beweisen, betrachten wir ein EDT0L-System $G = (\Sigma, H, \omega, \Delta)$ mit

$$\Sigma = \{S, A, B, ¢, a, b\},\ \Delta = \{¢, a, b\},\ \omega = SSS \text{ und } H = \{h_1, h_2\},$$

wobei die Tafel h_1 durch

$$h_1(S) = A,\ h_1(A) = Sa,\ h_1(B) = ¢ \text{ und } h_1(x) = x \text{ für } x \in \{a, b, ¢\}$$

definiert ist und die Tafel h_2 durch

$$h_2(S) = B,\ h_2(A) = ¢,\ h_2(B) = Sb \text{ und } h_2(x) = x \text{ für } x \in \{a, b, ¢\}.$$

In G beginnen die Ableitungen wie folgt:

$$SSS \begin{cases} \overset{h_1}{\Longrightarrow} AAA & \begin{cases} \overset{h_1}{\Longrightarrow} SaSaSa \\ \overset{h_2}{\Longrightarrow} ¢¢¢ \end{cases} \\ \overset{h_2}{\Longrightarrow} BBB & \begin{cases} \overset{h_1}{\Longrightarrow} ¢¢¢ \\ \overset{h_2}{\Longrightarrow} SbSbSb \end{cases} \end{cases}$$

Mit der Tafel h_1 wird, sofern nicht zum Abschluß der Ableitung dreimal das Symbol ¢ erzeugt wird, in den drei Teilen des Wortes jeweils das Symbol a eingeführt, mit der Tafel h_2 entsprechend das Symbol b. Das ET0L-System G liefert somit die Sprache

$$L(G) = \{¢w¢w¢w \mid w \in \{a, b\}^*\},$$

die nach Beispiel 13.12 keine E0L-Sprache ist, so daß insgesamt $\mathfrak{L}(E0L) \subsetneqq \mathfrak{L}(ET0L)$ gezeigt ist.

Um $\mathfrak{L}(ET0L) \subset \mathfrak{L}_1$ zu beweisen, betrachten wir ein beliebiges ET0L-System $G = (\Sigma, H, \omega, \Delta)$. Nach Satz 13.14 ist entscheidbar, ob $L(G)$ ε-frei ist. Daher kann nach Satz 13.16 ohne Beschränkung der Allgemeinheit angenommen werden, daß G ein EPT0L-System mit den Eigenschaften dieses Satzes ist, für das speziell $L(G)$ ε-frei ist und $\omega = S \in \Sigma - \Delta$ gilt.

Im folgenden konstruieren wir eine monotone Grammatik $G' = (V_N, V_T, S, Q)$ mit $L(G) = L(G')$. Wir setzen

$$V_T = \Delta,$$
$$V_N = (\Sigma - \Delta) \cup \{\bar{a} \mid a \in \Sigma\} \cup \{(a, b) \mid a, b \in \Sigma\} \cup \{R_h \mid h \in H\} \cup \{L\},$$

wobei R_h als *Botschaftssymbol* bezeichnet wird. Die Produktionen aus Q definieren wir wie folgt:

$S \to \alpha$ für alle $\alpha \in L(G)$ mit $|\alpha| \leq 2$,

$S \to \bar{a}_1 L a_2 \dots a_{n-2}(a_{n-1}, a_n)$ für alle $\alpha = a_1 \dots a_n \in (\Sigma - \Delta)^+$, $|\alpha| \geq 3$, für die ein $t \in \mathbb{N}_0$ und Wörter $\alpha_1, \dots, \alpha_t \in (\Sigma - \Delta)^+$ mit $|\alpha_i| < 3$, $1 \leq i \leq t-1$, existieren, so daß

$$S \Longrightarrow_G \alpha_1 \dots \Longrightarrow_G \alpha_t \Longrightarrow_G \alpha$$

gilt (man beachte, daß G propagierend ist),

$\bar{a}L \to \bar{a}_1 a_2 \dots a_n R_h$ für alle $h \in H - \{h_T\}$, $a \in \Sigma - \{S\}$ und $\alpha = a_1 \dots a_n \in \Sigma^+$ mit $a \to_h \alpha$,

$R_h a \to \alpha R_h$ für alle $h \in H$, $a \in \Sigma - \{S\}$ und $\alpha \in \Sigma^+$ mit $a \to_h \alpha$, (1)

$R_h(a,b) \to L a_1 \dots a_{n-2}(a_{n-1}, a_n)$ für alle $h \in H - \{h_T\}$, $a, b \in \Sigma - \{S\}$ und $a_1, \dots, a_n \in \Sigma$ mit der Eigenschaft, daß ein $t \in \mathbb{N}$, $t < n$, existiert mit Produktionen $a \to_h a_1 \dots a_t$ und $b \to_h a_{t+1} \dots a_n$,

$\bar{a}L \to b R_{h_T}$ für alle $a, b \in \Sigma$ mit $a \to_{h_T} b$, (2)

$R_{h_T}(a,b) \to cd$ für alle $a, b, c, d \in \Sigma$ mit $a \to_{h_T} c$ und $b \to_{h_T} d$, (3)

$aL \to La$ für alle $a \in \Sigma$.

Offenbar enthält jedes Wort α mit $S \Longrightarrow^* \alpha$ höchstens ein Symbol aus der Menge $\{L\} \cup \{R_h \mid h \in H\}$. Folglich läßt sich jedes solche α auf höchstens eine Weise als $\alpha = \gamma\bar{\alpha}\delta$ schreiben, wobei $\bar{\alpha}$ die linke Seite einer Produktion aus Q ist.

Nach Konstruktion der Grammatik gilt für Wörter α mit $|\alpha| \leq 2$ die Äquivalenz

$$\alpha \in L(G) \iff \alpha \in L(G').$$

Wir betrachten nun eine Ableitung eines Wortes α mit $|\alpha| \geq 3$ in G. Aufgrund der Form der Grammatik, die gemäß Satz 13.16 bestimmt ist, erscheint in jeder solchen Ableitung erstmalig ein Wort $\beta \in (\Sigma - \Delta)^+$ mit $|\beta| \geq 3$, d.h., es gilt $S \Longrightarrow_G^+ \beta$. Ist $\beta = a_1 \dots a_n$, so ergibt sich nach Konstruktion von G' die Produktion

$$S \to \bar{a}_1 L a_2 \dots a_{n-2}(a_{n-1}, a_n)$$

von Q. Das geklammerte Zeichen (a_{n-1}, a_n) ist zur Erkennung des Randes des betrachteten Wortes erforderlich. Es sei $\beta \Longrightarrow_h \beta'$ ein weiterer Ableitungsschritt in G mit $h \in H - \{h_T\}$, der unter Verwendung von Produktionen der Form $a_i \to_h \alpha_i = a_{i1} \dots a_{in_i}$, $i = 1, \dots, n$, erfolgt. In der Grammatik G' wird dieser Schritt durch

$$\begin{array}{l}
\bar{a}_1 L a_2 \dots a_{n-2}(a_{n-1}, a_n) \\
\Downarrow \\
\bar{a}_{11} a_{12} \dots a_{1n_1} R_h a_2 \dots a_{n-2}(a_{n-1}, a_n) \\
\Downarrow \\
\bar{a}_{11} a_{12} \dots a_{1n_1} \alpha_2 R_h a_3 \dots a_{n-2}(a_{n-1}, a_n) \\
\Downarrow_* \\
\bar{a}_{11} a_{12} \dots a_{1n_1} \alpha_2 \dots \alpha_{n-2} R_h(a_{n-1}, a_n) \\
\Downarrow \\
\bar{a}_{11} a_{12} \dots a_{1n_1} \alpha_2 \dots \alpha_{n-2} L b_1 \dots b_{k-2}(b_{k-1}, b_k) \qquad \text{mit } b_1 \dots b_k = \alpha_{n-1}\alpha_n \\
\Downarrow_* \\
\bar{a}_{11} L a_{12} \dots a_{1n_1} \alpha_2 \dots \alpha_{n-2} b_1 \dots b_{k-2}(b_{k-1}, b_k).
\end{array}$$

simuliert. Umgekehrt betrachten wir eine Ableitung

$$\bar{a}_1 L a_2 \dots a_{n-2}(a_{n-1}, a_n) \Longrightarrow_{G'}^* \bar{b}_1 L b_2 \dots b_{m-2}(b_{m-1}, b_m).$$

Dabei kommt das Symbol L nur im ersten und im letzten Wort vor, und zwar als jeweils zweites Symbol. Dies entspricht einem Ableitungsschritt $\beta \Longrightarrow_G \beta'$. Daher erhalten wir für jedes Wort $\beta = b_1 \dots b_n \in (\Sigma - \Delta)^+$ mit $|\beta| \geq 3$ die Äquivalenz

$$S \Longrightarrow_G^* \beta \quad \Longleftrightarrow \quad S \Longrightarrow_{G'}^* \bar{b}_1 L b_2 \dots b_{n-2}(b_{n-1}, b_n).$$

Nach Konstruktion der Grammatik G' und der Eigenschaft aus Satz 13.17(a) gilt damit

$$\begin{aligned}
\alpha \in L(G),\ |\alpha| \geq 3 \quad &\Longleftrightarrow \quad \bigvee_{\beta = b_1 \dots b_n \in (\Sigma - \Delta)^+} |\alpha| = |\beta|,\ S \Longrightarrow_G^* \beta \Longrightarrow_{h_T} \alpha \\
&\Longleftrightarrow \quad \bigvee_{\beta = b_1 \dots b_n \in (\Sigma - \Delta)^+} |\alpha| = |\beta|,\ S \Longrightarrow_{G'}^* \bar{b}_1 L b_2 \dots b_{n-2}(b_{n-1}, b_n) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \Longrightarrow_{G'}^+ \alpha \\
&\Longleftrightarrow \quad \alpha \in L(G').
\end{aligned}$$

Dabei wird der Schritt $\beta \Longrightarrow_{h_T} \alpha$ äquivalent durch $\bar{b}_1 L b_2 \dots b_{n-2}(b_{n-1}, b_n) \Longrightarrow_{G'}^+ \alpha$ mit Hilfe der Produktionen der Form (2), (1) und dann (3) simuliert. Es folgt $L(G) = L(G')$.

Die Beziehung $\mathfrak{L}(\text{ET0L}) \subsetneq \mathfrak{L}_1$ ergibt sich dann aus der Tatsache, daß $\mathfrak{L}(\text{ET0L})$ nach Satz 13.19 eine volle AFL ist, dies jedoch nach Satz 10.5 für $\mathfrak{L}_1$ nicht zutrifft. Ein direkterer Beweis folgt aus der Aussage von Satz 13.21, daß die Sprache $L = \{(ab^n)^m \mid m \geq n \geq 1\}$ keine ET0L-Sprache ist. Im Anschluß daran wurde jedoch gezeigt, daß diese Sprache eine Typ-1-Sprache ist. □

Kapitel 14

Limitierte Lindenmayersysteme

Die beliebige parallele Ersetzung beim Ableitungsprozeß von Lindenmayersystemen entspricht aus biologischer Sicht nur zum Teil der Wirklichkeit. Sind die Organismen genügend groß, so müssen Einschränkungen der natürlichen Resourcen berücksichtigt werden. Unter diesem Gesichtspunkt ist eine Beschränkung der parallelen Ersetzung von Lindenmayersystemen ganz natürlich. Zuerst hat *Frings* in seiner Diplomarbeit [8] einen Spezialfall solcher Systeme betrachtet. In [28] wurden diese k-limitierten L-Systeme dann allgemeiner definiert. Inzwischen gibt es eine Reihe weiterer Veröffentlichungen, die sich mit solchen limitierten Lindenmayersystemen befassen.

14.1 Limitierte 0L-Systeme

Definition 14.1 Es sei $k \in \mathbb{N}$ und $h: \Sigma \to \mathfrak{P}(\Sigma^*)$ eine endliche Substitution. Dann heißt die Abbildung $h_k: \Sigma^* \to \mathfrak{P}(\Sigma^*)$ eine *k-Limitierung von h*, wenn für alle Wörter $w \in \Sigma^*$ die Menge $h_k(w)$ definiert ist als die Menge derjenigen Wörter, die sich aus w ergeben, indem genau $\min\{\#_a w, k\}$ Vorkommen eines jeden Symbols $a \in \Sigma$ in w durch die Menge $h(a)$ ersetzt wird. □

Definition 14.2 Es sei $k \in \mathbb{N}$. $G = (\Sigma, H, \omega, \Delta, k)$ heißt *k-limitiertes* ET0L-*System (*klET0L-*System)*, wenn $G = (\Sigma, H, \omega, \Delta)$ ein ET0L-System ist.

$$L(G) = \{w \in \Sigma^* \mid w = \omega \text{ oder } w \in h_k^t \ldots h_k^1(\omega),\ h^1, \ldots, h^t \in H,\ t \in \mathbb{N}\}$$

heißt die *von G erzeugte* klET0L-*Sprache.* □

Analog zu den ET0L-Systemen werden klT0L-, kl0L-, klP0L-, klEPDT0L-Systeme usw. definiert. Die erzeugten Sprachfamilien werden entsprechend mit $\mathfrak{L}(k\text{lT0L})$, $\mathfrak{L}(k\text{l0L})$, $\mathfrak{L}(k\text{lP0L})$, $\mathfrak{L}(k\text{lEPDT0L})$ usw. bezeichnet. Wir schreiben z.B. auch $\mathfrak{L}(k\text{l(E)(T)0L})$ und meinen damit, daß ein Vorkommen dieses Ausdrucks in einer Aussage ihre Gültigkeit für diejenigen vier Möglichkeiten bedeutet, die wir erhalten, wenn wir wahlweise die eingeklammerten Buchstaben berücksichtigen oder fortlassen. Aufgrund der Definitionen ergibt sich unmittelbar

Satz 14.1 Für alle $k \in \mathbb{N}$ gilt

$$\mathfrak{L}(k\text{l0L}) \subset \mathfrak{L}(k\text{lT0L}) \subset \mathfrak{L}(k\text{lET0L}) \text{ und}$$
$$\mathfrak{L}(k\text{l0L}) \subset \mathfrak{L}(k\text{lE0L}) \subset \mathfrak{L}(k\text{lET0L}). \quad \square$$

Beispiel 14.1 Für jedes $k \in \mathbb{N}$ ist

$$G = (\{a,b\}, \{\{a \to a^3,\ b \to b^3\}, \{a \to a^4,\ b \to b^4\}\}, ab, k)$$

ein klPDT0L-System. Die kürzesten Wörter von $L(G)$ sind ab, a^3b^3 und a^4b^4, und alle Wörter w von $L(G)$ haben die Form $w = a^n b^n$ für geeignete, von k abhängige Zahlen $n \geq 1$. $\square$

Beispiel 14.2 Für jedes $k \in \mathbb{N}$ ist

$$G = (\{a,b,c\}, h, abc, k) \text{ mit } h(a) = a^2, h(b) = b^2, h(c) = c^2$$

ein klPD0L-System, das die Sprache

$$L(G) = \{a^{2^n} b^{2^n} c^{2^n} \mid n = 0, \ldots, r\} \cup \{a^{2^{r+1}+nk} b^{2^{r+1}+nk} c^{2^{r+1}+nk} \mid n \in \mathbb{N}_0\}$$

für $k = 2^r + m$ mit $0 \leq m < 2^r$ erzeugt. $\square$

Beispiel 14.3 Ein *Turtle* T (turtle = Schildkröte) ist ein Tripel (Z, d, δ) mit dem *Zustand* $Z = (x, y, \alpha)$ des Turtles. Hierbei sind x und y die kartesischen Koordinaten einer Position in der Ebene und α die Ausrichtung des Turtles gegenüber der positiven x-Halbachse in mathematisch positiver Richtung. Das Turtle interpretiert die einzelnen Zeichen eines Wortes als Komandos, die nacheinander von ihm ausgeführt werden:

(a) Ein kleiner Buchstabe bewegt das Turtle um einen Schritt der Länge d von der aktuellen Position Z in Richtung des Winkels α, so daß es den neuen Zustand $Z' = (x + d\cos\alpha, y + d\sin\alpha, \alpha)$ einnimmt. Die alte und die neue Position werden durch eine Gerade verbunden.

(b) Das Zeichen $+$ $(-)$ dreht das Turtle um den Winkel δ nach links (rechts), d.h., versetzt es in den Zustand $Z' = (x, y, \alpha + \delta)$ $(Z' = (x, y, \alpha - \delta))$.

Ein Viertupel $T = (Z, K, d, \delta)$ mit einem *Keller* K über Z heißt *erweitertes Turtle*, wenn (Z, d, δ) ein Turtle ist, das zusätzlich die folgenden beiden Kommandos versteht:

(a) Das Zeichen „(“ veranlaßt das Turtle, seinen aktuellen Zustand in den Keller zu schreiben.

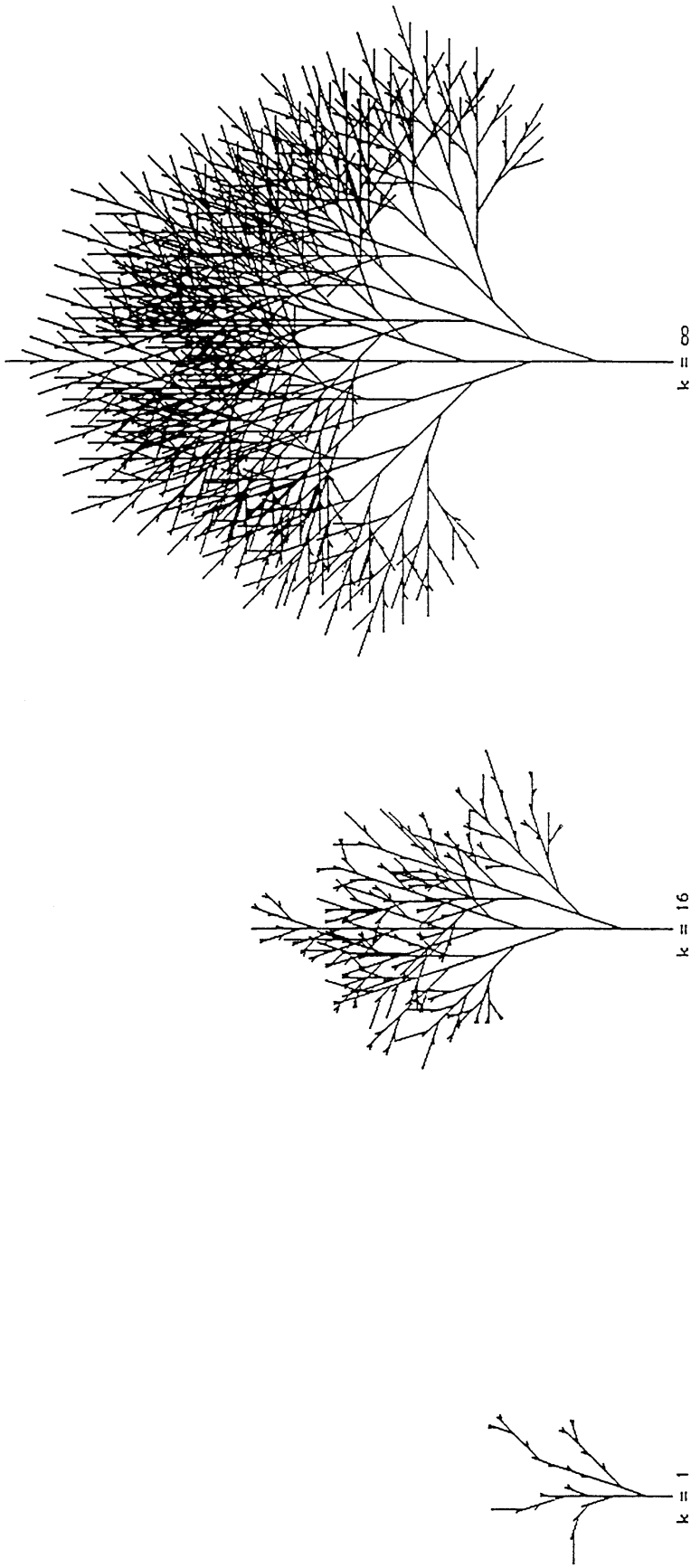
(b) Das Zeichen „)“ veranlaßt das Turtle, den letzten Zustand aus dem Keller zu lesen und zu seinem aktuellen Zustand zu machen.

Turtles können z.B. für die graphische Interpretation der Limitierung von Lindenmayersystemen benutzt werden. Wir betrachten z.B. das D0L-System (Σ, h, a) mit

$$\Sigma = \{a, b, c, l, r, w, x, (,)\}$$

und der Substitution h, die durch die Produktionen

$$a \to cwlb, \quad l \to (+a), \quad w \to x, \quad (\to (, \quad + \to +, \quad c \to c,$$
$$b \to cwra, \quad r \to (-b), \quad x \to wc, \quad) \to), \quad - \to -$$


k = 8
k = 16
k = 1

definiert ist. Bei einer k-Limitierung von $k = 1, 16$ bzw. ∞ ($k = \infty$ bedeutet, daß wir ein übliches D0L-System betrachten) liefert das System nach 15 Ableitungsschritten jeweils ein Wort, das bei Wahl von $\delta = 23°$ durch die Turtle-Graphik auf Seite 238 interpretiert werden kann.

Satz 14.2 Es sei $G = (\Sigma, H, \omega, k)$ ein klT0L-System mit $n = \text{card}(\Sigma)$ und

$$s = \max\{|w| \mid w \in h(a),\ h \in H,\ a \in \Sigma\}.$$

Dann gilt für alle Wörter $w_1, w_2 \in \Sigma^*$ mit $w_1 \Longrightarrow_G w_2$ die Abschätzung

$$|w_1| - nk \leq |w_2| \leq |w_1| + nk \cdot (s-1). \quad \square$$

Dieser Satz bedeutet, daß Sprachen, bei denen die Längen der Wörter exponentiell wachsen, keine klT0L-Sprachen sein können.

Satz 14.3 Für jedes $k \in \mathbb{N}$ ist die Familie $\mathfrak{L}(k\text{l(T)0L})$ eine Anti-AFL und außerdem nicht unter Konkatenation abgeschlossen.

Beweis: Wir müssen beweisen, daß $\mathfrak{L}(k\text{l(T)0L})$ unter keiner der AFL-Operationen aus Definition 10.9 und auch nicht unter Konkatenation abgeschlossen ist. Dabei gehen wir jeweils von geeigneten Sprachen aus $\mathfrak{L}(k\text{l0L}) \subset \mathfrak{L}(k\text{lT0L})$ aus und zeigen, daß die Anwendung der entsprechenden Operationen zu Sprachen führt, die nicht zu $\mathfrak{L}(k\text{lT0L})$ gehören.

(a) Wir zeigen, daß $\mathfrak{L}(k\text{l(T)0L})$ nicht unter Vereinigung abgeschlossen ist. Offenbar gilt $\{a\}, \{a^3\} \in \mathfrak{L}(k\text{l0L})$. Wäre $\{a\} \cup \{a^3\} = \{a, a^3\} \in \mathfrak{L}(k\text{lT0L})$, so würde ein klT0L-System $G = (\{a\}, H, \omega, k)$ mit $L(G) = \{a, a^3\}$ existieren. Für $\omega \in L(G)$ gibt es nun zwei Möglichkeiten.

(1) Ist $\omega = a$, so muß $a \Longrightarrow a^3$ und damit $a^3 \in h(a)$ für eine Tafel $h \in H$ gelten. Daraus folgt jedoch $a^3 \Longrightarrow_h a^5$ (falls $k = 1$), $a^3 \Longrightarrow_h a^7$ (falls $k = 2$) oder $a^3 \Longrightarrow_h a^9$ (falls $k \geq 3$), was in jedem Fall ein Widerspruch ist.

(2) Ist $\omega = a^3$, so muß $a^3 \Longrightarrow a$ gelten. Dann folgt $\varepsilon, a \in h(a)$ für ein $h \in H$. Damit erhalten wir für jedes $k \in \mathbb{N}$ den Widerspruch $a^3 \Longrightarrow_h a^2 \notin L(G)$.

(b) Für den Nichtabschluß von $\mathfrak{L}(k\text{l(T)0L})$ unter Durchschnitt mit regulären Sprachen betrachten wir das kl0L-System

$$G = (\{a\}, h, a, k) \text{ mit } h(a) = \{a, a^3\}$$

und die unter (a) betrachtete reguläre Sprache $\{a, a^3\}$. Dann gilt

$$L(G) \cap \{a, a^3\} = \{a, a^3\} \notin \mathfrak{L}(k\text{lT0L}).$$

(c) $\mathfrak{L}(k\text{l(T)0L})$ ist nicht unter ε-freier Iteration abgeschlossen. Zum Nachweis betrachten wir die kl0L-Sprache $\{b^2c^2\}$. Falls $L = \{b^2c^2\}^+$ von einem klT0L-System $G = (\Sigma, H, \omega, k)$ erzeugt wird, muß offenbar $\Sigma = \{b, c\}$ gelten. Zur Erzeugung

von Wörtern, die länger als das Axiom ω sind, wird eine Tafel $h \in H$ mit zwei Wörtern $\beta \in h(b)$ und $\gamma \in h(c)$ benötigt, von denen mindestens eines eine Länge ≥ 2 besitzt. Ohne Beschränkung der Allgemeinheit gelte $|\beta| \geq 2$. Dann existiert zu dem Wort $(b^2c^2)^{k+1} \in L$ die Ableitung

$$(b^2c^2)^{k+1} \Longrightarrow_h \beta bc^2u \in L,$$

wobei u aus dem Wort $(b^2c^2)^k$ entsteht durch Ersetzen von k Vorkommen des Symbols c und von $k-1$ Vorkommen des Symbols b. Da β aus mindestens zwei Zeichen besteht, ist b^2 ein Präfix von β. Bei einer anderen Wahl der zu ersetzenden Vorkommen von Symbolen in $(b^2c^2)^{k+1}$ erhalten wir die Ableitung

$$(b^2c^2)^{k+1} \stackrel{h}{\Longrightarrow} b\beta c^2u.$$

Das bedeutet, daß b^3 ein Präfix von $b\beta c^2u \in L$ ist, was jedoch der Definition von L widerspricht.

(d) Zum Nachweis des Nichtabschlusses von $\mathfrak{L}(k\text{l(T)0L})$ unter ε-freiem Homomorphismus betrachten wir die Sprache $\{c,d\}$, die von dem klD0L-System

$$G = (\{c,d\}, h, c, k) \quad \text{mit} \quad h(c) = d,\ h(d) = d$$

erzeugt wird. Definieren wir einen Homomorphismus $g : \{c,d\}^* \to \{a\}^*$ durch

$$g(c) = a \quad \text{und} \quad g(d) = a^3,$$

so erhalten wir die Sprache $g(L(G)) = \{a, a^3\}$, die nach (a) keine klT0L-Sprache ist.

(e) $\mathfrak{L}(k\text{l(T)0L})$ ist nicht abgeschlossen unter inversem Homomorphismus. Zum Beweis betrachten wir die Sprache $\{c, b, b^3\}$, die von dem kl0L-System

$$G = (\{b,c\}, h, c, k) \quad \text{mit} \quad h(b) = \{b\},\ h(c) = \{b, b^3\}$$

erzeugt wird. Definieren wir einen Homomorphismus $g : \{a\}^* \to \{b,c\}^*$ durch $g(a) = b$, so erhalten wir die Sprache $g^{-1}(L(G)) = \{a, a^3\} \notin \mathfrak{L}(k\text{lT0L})$.

(f) $\mathfrak{L}(k\text{l(T)0L})$ ist nicht abgeschlossen unter Konkatenation, da $\{a\}$ und $\{\varepsilon, a\}$ offensichtlich kl0L-Sprachen sind, jedoch $\{a\}\{\varepsilon, a\} = \{a, a^2\} \notin \mathfrak{L}(k\text{lT0L})$ gilt, was ähnlich wie $\{a, a^3\} \notin \mathfrak{L}(k\text{lT0L})$ bewiesen wird. □

Satz 14.4 Für alle $k \in \mathbb{N}$ existieren in $\mathfrak{L}(\text{fin})$, $\mathfrak{L}_3 - \mathfrak{L}(\text{fin})$, $\mathfrak{L}_2 - \mathfrak{L}_3$ und $\mathfrak{L}_1 - \mathfrak{L}_2$ sowohl Sprachen, die in $\mathfrak{L}(k\text{l(T)0L})$ liegen, als auch solche Sprachen, die nicht in $\mathfrak{L}(k\text{l(T)0L})$ liegen.

Beweis: (a) Die Sprache

$$L_1 = \{a\} \in \mathfrak{L}(\text{fin})$$

ist offensichtlich eine kl0L-Sprache. Die Sprache

$$L_2 = \{a, a^2, \ldots, a^{2^n}\} \cup \{a^{2^n}\}\{a^k\}^+ \in \mathfrak{L}_3 - \mathfrak{L}(\text{fin})$$

wird von dem kl0L-System $G_2 = (\{a\}, \{a \to a^2\}, a, k)$ erzeugt, wobei $n \in \mathbb{N}_0$ die kleinste Zahl mit $k \leq 2^n$ ist. Die Sprache

$$L_3 = \{ab, a^2b^2, \ldots, a^{2^n}b^{2^n}\} \cup \{a^{2^n+rk}b^{2^n+rk} \mid r \in \mathbb{N}\}$$

wird von dem kl0L-System $G_3 = (\{a,b\}, \{a \to a^2,\ b \to b^2\}, ab, k)$ erzeugt, wobei $n \in \mathbb{N}_0$ wieder die kleinste Zahl mit $k \leq 2^n$ ist. Offensichtlich ist L_3 nicht vom Typ 3. Für die Typ-2-Grammatik $H_3 = (\{S, X\}, \{a, b\}, S, F)$ mit der Produktionenmenge

$$F = \{S \to ab, \ldots, S \to a^{2^{n-1}}b^{2^{n-1}}, S \to a^{2^n}Xb^{2^n}, X \to a^kXb^k, X \to \varepsilon\}$$

gilt $L(H_3) = L_3$. Somit folgt $L_3 \in \mathfrak{L}_2 - \mathfrak{L}_3$. Schließlich betrachten wir die Sprache $L_4 = L(G)$ aus Beispiel 14.2, die nach dem Lemma von Bar-Hillel (Satz 5.9) nicht kontextfrei ist, jedoch von der monotonen Grammatik $H_4 = (\{S, X, B\}, \{a, b, c\}, S, P)$ mit den Produktionen

$$P = \{S \to abc,\ S \to a^2b^2c^2, \ldots,\ S \to a^{2^{r+1}}b^{2^{r+1}}c^{2^{r+1}},\ S \to (aB)^{2^{r+1}}Xc^{2^{r+1}},$$
$$X \to (aB)^kc^k,\ X \to (aB)^kXc^k,\ Ba \to aB,\ Bc \to bc,\ Bb \to bb\}$$

erzeugt wird. Damit erhalten wir $L_4 \in \mathfrak{L}_1 - \mathfrak{L}_2$.

(b) Wir geben Sprachen der genannten Familien an, die keine klT0L-Sprachen sind. In $\mathfrak{L}$(fin) ist dies nach dem Beweis von Satz 14.3 die Sprache

$$L_5 = \{a, a^3\}.$$

In $\mathfrak{L}_3 - \mathfrak{L}$(fin) liegt offenbar die Sprache

$$L_6 = \{a, a^2\} \cup \{a^5\}^+.$$

Wir nehmen an, daß L_6 von einem klT0L-System $G_6 = (\Sigma, H, \omega, k)$ erzeugt wird. Da L_6 ε-frei ist, muß G_6 propagierend sein und somit $\omega = a$ gelten. Um a^2 abzuleiten, ist die Produktion $a^2 \in h(a)$ für eine Tafel $h \in H$ nötig. Dann erhalten wir für $k = 1$ die Ableitung $a^2 \Longrightarrow_h a^3 \notin L_6$ und für $k \geq 2$ die Ableitung $a^2 \Longrightarrow_h a^4 \notin L_6$. In beiden Fällen ergibt sich ein Widerspruch.
Offensichtlich ist

$$L_7 = \{a^nb^n \mid n \in \mathbb{N}\} \cup \{a^{n+1}b^n \mid n \in \mathbb{N}\} \in \mathfrak{L}_2 - \mathfrak{L}_3.$$

Wir nehmen an, daß L_7 von einem klT0L-System $G_7 = (\{a,b\}, H, \omega, k)$ erzeugt wird. Da L_7 ε-frei ist, ist G_7 propagierend, und wir erhalten somit $\omega = ab$. Aufgrund der Reihenfolge der Symbole in jedem Wort aus L_7 folgt weiter $h(a) \subset \{a\}^+$ und $h(b) \subset \{b\}^+$. Da $a^2b \in L_7$ das zweitkürzeste Wort von L_7 ist, muß die Ableitung $\omega = ab \Longrightarrow_{h'} a^2b$ für eine Tafel $h' \in H$ existieren. Es folgt $a^2 \in h'(a)$ und $b \in h'(b)$. Daraus ergibt sich $a^2b \Longrightarrow_{h'} a^4b \notin L_7$ für $k \geq 2$ und $a^2b \Longrightarrow_{h'} a^3b \notin L_7$ für $k = 1$, ein Widerspruch.
Schließlich betrachten wir die Sprache

$$L_8 = \{a^{2^n} \mid n \in \mathbb{N}_0\} \in \mathfrak{L}_1 - \mathfrak{L}_2.$$

Sie kann nach Satz 14.2 durch kein klT0L-System erzeugt werden. □

Satz 14.5 Für alle $k \in \mathbb{N}$ sind die Familien $\mathfrak{L}(k\text{l0L})$ und $\mathfrak{L}(k\text{lT0L})$ unvergleichbar mit den Familien $\mathfrak{L}(\text{0L})$ und $\mathfrak{L}(\text{T0L})$. Die Familien sind jedoch nicht disjunkt.

Beweis: Die Sprache $\{a\}$ liegt in allen betrachteten Familien. Weiter gilt

$$L' = \{a^{2^n} \mid n \in \mathbb{N}_0\} \in \mathfrak{L}(\text{0L}) \subset \mathfrak{L}(\text{T0L}),$$

jedoch ist L' nach Satz 14.2 keine klT0L-Sprache.

Um den Beweis abzuschließen, müssen wir $\mathfrak{L}(k\text{l0L}) \not\subset \mathfrak{L}(\text{T0L})$ zeigen. Wir betrachten die Sprache

$$L = \{ab, a^2b^2, \ldots, a^{2^n}b^{2^n}\} \cup \{a^{2^n+rk}b^{2^n+rk} \mid r \in \mathbb{N}\}$$

mit $n = \min\{m \in \mathbb{N}_0 \mid k \leq 2^m\}$, die nach dem Beweis von Satz 14.4 eine kl0L-Sprache ist. Wir nehmen an, daß sie durch ein T0L-System $G = (\{a, b\}, H, \omega)$ erzeugt wird. Aufgrund der Gestalt von L gilt $\#_a w = \#_b w$ für alle $w \in L$, und für jede Tafel $h \in H$ muß $h(a) \subset \{a\}^*$ und $h(b) \subset \{b\}^*$ gelten. Folglich ist G deterministisch, da sonst Wörter w mit unterschiedlich vielen Vorkommen von a und b in w erzeugt werden könnten. Wir können daher ohne Beschränkung der Allgemeinheit annehmen, daß für jedes $h \in H$ genau eine natürliche Zahl $n_h > 1$ mit $h(a) = a^{n_h}$ und $h(b) = b^{n_h}$ existiert. Es seien $n_1, \ldots, n_s$ alle derartigen Exponenten. Da das Wort ab aus keinem Wort $a^{n'}b^{n'}$ mit $n' \neq 1$ erzeugt werden kann, folgt $\omega = ab$. Somit erhalten wir

$$L(G) = \{a^{n_1^{\alpha_1}\cdots n_s^{\alpha_s}} b^{n_1^{\alpha_1}\cdots n_s^{\alpha_s}} \mid \alpha_\sigma \in \mathbb{N}_0,\ \sigma = 1, \ldots, s\}.$$

Wir wählen $r = n_1 \cdots n_s \cdot 2^n$ und betrachten $a^{2^n+rk}b^{2^n+rk} \in L$. Der Exponent von a und b dieses Wortes hat die Darstellung

$$2^n + rk = 2^n(1 + n_1 \cdots n_s \cdot k).$$

Da $1 + n_1 \cdots n_s \cdot k$ durch keinen Primfaktor von $n_1 \cdots n_s$ teilbar ist, enthält $2^n + rk$ Primfaktoren, die nicht in $n_1 \cdots n_s$ enthalten sind. Somit gilt

$$a^{2^n+rk}b^{2^n+rk} \notin L(G). \quad \square$$

Satz 14.6 Für $k, k' \in \mathbb{N}$, $k \neq k'$, sind die Familien $\mathfrak{L}(k\text{l0L})$ und $\mathfrak{L}(k\text{lT0L})$ unvergleichbar mit den Familien $\mathfrak{L}(k'\text{l0L})$ und $\mathfrak{L}(k'\text{lT0L})$. Die Familien sind jedoch nicht disjunkt.

Beweis: Die Sprache $\{a\}$ liegt in allen angegebenen Familien. Weiter betrachten wir das klPD0L-System

$$G' = (\{a, b, c\}, h, a^{k+k'}b^{k+1}, k) \text{ mit } h(a) = a^{k'+2},\ h(b) = c \text{ und } h(c) = c,$$

das die Sprache

$$\begin{aligned} L = \ & \{a^{k+k'}b^{k+1}\} \cup \{a^{k+k'+k(k'+1)}c^{k-i}bc^i \mid i = 0, \ldots, k\} \\ & \cup \{a^{k+k'+\nu k(k'+1)}c^{k+1} \mid \nu = 2, 3, \ldots\} \end{aligned}$$

erzeugt. Wir nehmen an, daß

$$G = (\{a, b, c\}, H, \omega, k')$$

ein k'lT0L-System mit $L(G) = L$ ist. Aufgrund der Gestalt von L gilt $h(a) \subset \{a\}^*$ für jedes $h \in H$. Für Wörter $w, w' \in L$ mit $|w| \neq |w'|$ gilt immer

$$|\#_a w - \#_a w'| \geq k(k'+1) > k'.$$

Da in einem Ableitungsschritt höchstens k' Vorkommen von a gelöscht werden können, kann man gemäß G kein kürzeres Wort aus einem längeren Wort ableiten. Somit muß $\omega = a^{k+k'}b^{k+1}$ gelten, und es folgt die Existenz einer Ableitung

$$\omega \Longrightarrow_h a^{k+k'+k(k'+1)}c^{k-i}bc^i$$

für mindestens ein $i \in \{0, \ldots, k\}$ und eine Tafel $h \in H$. Für $k' < k$ ist dies gar nicht möglich, da von den $k+1$ Vorkommen von b in ω mindestens zwei übrigbleiben würden. Für $k' > k$ betrachten wir mit dieser Tafel h eine beliebige Produktion $w \in h(b)$. Würde $w \in \{a\}^*$ gelten, so könnten wir aus ω Wörter ableiten, die mit a enden und daher in L nicht vorkommen. Wäre w eine Mischung aus Vorkommen von Symbolen a und mindestens einem Vorkommen eines Symbols aus $\{b, c\}$, so wäre bei jedem $k' \in \mathbb{N}$ wegen $k+1 \geq 2$ ein Wort erzeugbar, das ein Vorkommen von a nach einem Vorkommen von $\{b, c\}$ hätte, was ebenfalls ein Widerspruch ist. Folglich muß $\#_a w = 0$ gelten. Mit der Tafel h werden alle Vorkommen des Symbols b in ω ersetzt, so daß b in $w \in h(b)$ enthalten sein muß. Folglich existiert eine Ableitung $\omega \Longrightarrow_h w_1$ mit $\#_a w_1 = k + k' + k(k'+1)$ und $\#_b w_1 \geq k+1$. Ein solches Wort w_1 kommt aber in L nicht vor. □

14.2 Limitierte ET0L-Systeme

Wir stellen zunächst fest, daß die Einführung von Terminalzeichen wie bei den üblichen Lindenmayersystemen zu einer Vergrößerung der betrachteten Sprachfamilien führt.

Satz 14.7 Für jedes $k \in \mathbb{N}$ gilt

$$\mathfrak{L}(k\text{l0L}) \subsetneq \mathfrak{L}(k\text{lE0L}) \quad \text{und} \quad \mathfrak{L}(k\text{lT0L}) \subsetneq \mathfrak{L}(k\text{lET0L}).$$

Beweis: (a) Das klEPDT0L-System

$$G = (\{a, b, F\}, \{a \to a^2b, b \to F, F \to F\}, a, \{a, b\}, k)$$

erzeugt offensichtlich die Sprache $L = \{a, a^2b\}$. Wir nehmen an, daß es ein kl0L-System $G' = (\{a, b\}, h, \omega, k)$ mit $L(G') = L$ gibt. Das Axiom ω muß a oder a^2b sein. Im Fall $\omega = a$ folgt $a^2b \in h(a)$, so daß für jedes $k \in \mathbb{N}$ eine Ableitung $a^2b \Longrightarrow_h w$ mit $\#_a w \geq 3$ existiert. Ein solches Wort w kommt jedoch in L nicht vor. Im Fall $\omega = a^2b$ muß eine Ableitung $a^2b \Longrightarrow_h a$ existieren. Dann ist $\varepsilon \in h(a)$, und wir erhalten $a \Longrightarrow_h \varepsilon \in L(G')$, was ebenfalls ein Widerspruch ist.

(b) Wir betrachten das klEPDT0L-System

$$G = (\{X, Y, F, a\}, \{h_1, h_2, h_3\}, a, \{a\}, k)$$

mit

$$\begin{array}{llll} h_1(a) = Y, & h_1(Y) = Y, & h_1(X) = F, & h_1(F) = F, \\ h_2(a) = F, & h_2(Y) = X, & h_2(X) = X, & h_2(F) = F, \\ h_3(a) = a, & h_3(Y) = F, & h_3(X) = a^2, & h_3(F) = F. \end{array}$$

Ohne F zu erzeugen und ohne triviale Schritte $w \Longrightarrow w$ sind nur Ableitungen der Form

$$a \Longrightarrow_{h_1}^* Y \Longrightarrow_{h_2}^* X \Longrightarrow_{h_3}^* a^2 \Longrightarrow_{h_1}^* Y^2 \Longrightarrow_{h_2}^* X^2 \Longrightarrow_{h_3}^* a^4 \Longrightarrow_{h_1}^* \ldots$$

möglich, wobei die Anzahl der notwendigen Schritte von k abhängt. G erzeugt folglich die Sprache $\{a^{2^n} \mid n \in \mathbb{N}_0\}$, die wegen Satz 14.2 keine klT0L-Sprache ist. □

Es sei $\Longrightarrow$ ein Ableitungsschritt gemäß einem klE0L-System. Dann bezeichnen wir mit $\Longrightarrow_{(1)}$ einen Ableitungsschritt des entsprechenden unterliegenden E0S-Systems. Dies ist ein *sequentieller Ableitungsschritt* wie bei einer kontextfreien Grammatik, bei dem jedoch nicht zwischen Terminal- und Nichtterminalzeichen unterschieden wird. Aus $v_1 \Longrightarrow^* v_2$ folgt offenbar $v_1 \Longrightarrow_{(1)}^* v_2$. Wir sagen, daß die Ableitung $v_1 \Longrightarrow^* v_2$ die Ableitung $x_1 \Longrightarrow_{(1)}^* x_2$ impliziert, wenn $v_1 = u_1 x_1 u_1'$ gilt, x_2 ein Teilwort von v_2 ist und nach Streichung von u_1 und u_1' die Ableitung $x_1 \Longrightarrow_{(1)}^* x_2$ eine Beschränkung der Ableitung $v_1 \Longrightarrow_{(1)}^* v_2$ in dem Sinn ist, daß mit x_1 anstelle von v_1 gestartet wird.

Der folgende Satz ist ein schwacher Iteratiossatz, mit dessen Hilfe für einige Sprachen nachgewiesen werden kann, daß sie keine klE0L-Sprachen sind.

Satz 14.8 Es sei $k \in \mathbb{N}$ und $G = (\Sigma, h, \omega, \Delta, k)$ ein klE0L-System. Es existiere eine Teilsprache $L' \subset L = L(G)$ über $\Delta' = \{a_1, \ldots, a_n\} \subset \Delta$ mit einem geeigneten $n \in \mathbb{N}$, so daß für alle $w \in L'$ ein $w' \in L'$ existiert mit

$$\#_a w < \#_a w'$$

für alle $a \in \Delta'$. Dann ist mindestens eine der folgenden Eigenschaften erfüllt:

(a) Für alle $m \in \mathbb{N}$ existieren Wörter $w_0, \ldots, w_m \in L \cap \Delta'^*$, so daß
ein $c \in \mathbb{Z}$, $c \neq 0$, existiert mit $|w_{i+1}| = |w_i| + c$ für alle $i \in \{0, \ldots, m-1\}$
oder aber
$|w_{i+1}| = |w_i|$, $\#_a w_{i+1} = \#_a w_i - k$, für ein $a \in \Delta'$ und alle $i \in \{0, \ldots, m-1\}$
gilt.

(b) Für alle $j \in \{1, \ldots, n\}$ gibt es ein Wort $t_{a_j} \in h(a_j) \cap \Delta'^*$, so daß das Wort $t_{a_1} \ldots t_{a_n}$ eine Permutation des Wortes $a_1 \ldots a_n$ ist und ein Index $i \in \{1, \ldots, n\}$ mit $t_{a_i} = \varepsilon$ existiert.

Beweis: Es sei $s = \max\{|w| \mid w \in h(b), b \in \Sigma\}$. Da L nicht endlich ist, muß $s \geq 2$ gelten. Aus den Voraussetzungen entnehmen wir, daß es für die Anzahl der Vorkommen von Symbolen aus Δ' keine obere Schranke gibt. Aus den Voraussetzungen für L folgt, daß für alle $m \in \mathbb{N}$ ein Wort $w_0 \in L'$ existiert, so daß für alle $a \in \Delta'$

$$\#_a w_0 > \sigma = k \cdot m \cdot \mathrm{card}(\Sigma) \cdot s^{\mathrm{card}(\Sigma)} \cdot |\omega|$$

gilt. Somit gibt es ein Wort $\widetilde{w} \in \Sigma^*$ mit

$$\#_a \widetilde{w} \geq 1 \text{ für alle } a \in \Delta' \text{ und } \omega \Longrightarrow^* \widetilde{w} \Longrightarrow w_0.$$

Da in w_0 jedes Symbol $a \in \Delta'$ mehr als σ-mal vorkommt und damit wegen $\sigma \geq k \cdot s \cdot \mathrm{card}(\Sigma)$ in einer größeren Anzahl vorkommt, als in einem Schritt maximal erzeugt werden könnten, muß jedes a schon in $\widetilde{w}$ vorkommen. Daraus schließen wir, daß in der Ableitung $\widetilde{w} \Longrightarrow w_0$ für alle $a \in \Delta'$ eine Produktion $t_a \in h(a) \cap \Delta'^*$ benutzt wird. Mit diesen Produktionen gelte

$$a_1 \ldots a_n \Longrightarrow_G t_{a_1} \ldots t_{a_n} = w' \in \Delta'^*.$$

Werden diese Produktionen auf w_0 angewendet, so ergibt sich

$$w_0 \Longrightarrow w_1 \in \Delta'^* \quad \text{mit} \quad |w_1| = |w_0| + k \cdot (|w'| - n).$$

Wegen $\#_a w_0 > \sigma \geq k \cdot m$ für alle $a \in \Delta'$ kann ein solcher Ersetzungsschritt mindestens m-mal ausgeführt werden. Wir erhalten folglich die Ableitung

$$w_0 \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_i \Longrightarrow \ldots \Longrightarrow w_m$$

mit

$$w_i \in \Delta'^*, \ |w_i| = |w_0| + ik \cdot (|w'| - n), \ i = 0, \ldots, m.$$

Für $|w'| \neq n$ wählen wir $c = k \cdot (|w'| - n)$. Dann ist offensichtlich die erste Alternative der Eigenschaft (a) erfüllt. Wir betrachten jetzt den Fall $|w'| = n$. Zunächst sei w' keine Permutation der Symbole aus Δ'. Folglich existiert ein Symbol $a \in \Delta'$ mit $\#_a w' = 0$, so daß für alle $i \in \{0, \ldots, m-1\}$

$$|w_{i+1}| = |w_i|, \ \#_a w_{i+1} = \#_a w_i - k$$

gilt, also die zweite Alternative der Eigenschaft (a) erfüllt ist. Es bleibt der Fall $|w'| = n$, wobei w' eine Permutation der Symbole aus Δ' ist. Dann ist die Eigenschaft (b) erfüllt, oder anderenfalls existiert für jedes $\nu \in \{1, \ldots, n\}$ eine Zahl $\mu_\nu \in \{1, \ldots, n\}$ mit

$$a_{\mu_\nu} \in h(a_\nu).$$

Wir betrachten eine Ableitung

$$D: \ \omega \Longrightarrow^* w_0.$$

Wir sagen, daß ein zyklisches Vorkommen eines Symbols $x \in \Sigma$ in D existiert, wenn sich D als

$$D: \ \omega \Longrightarrow^* \widetilde{w}_1 x \widetilde{w}_2 \Longrightarrow^* \widetilde{w}_1' v_1' x v_2' \widetilde{w}_2' \Longrightarrow^* \widetilde{w}_1'' v_1 v_0 v_2 \widetilde{w}_2'' = w_0$$

darstellen läßt, wobei $v_1 v_2 \neq \varepsilon$ und

$$\begin{gathered} x \Longrightarrow_{(1)}^* v_1' x v_2', \quad x \Longrightarrow_{(1)}^* v_0, \\ \widetilde{w}_i \Longrightarrow_{(1)}^* \widetilde{w}_i', \quad \widetilde{w}_i' \Longrightarrow_{(1)}^* \widetilde{w}_i'', \quad v_i' \Longrightarrow_{(1)}^* v_i, \quad (i = 1, 2), \end{gathered}$$

gelten und die Ableitungen $\Longrightarrow_{(1)}^*$ durch die entsprechenden Teilableitungen von D impliziert werden. Da aus dem Axiom ω ohne zyklisches Vorkommen nur Wörter der Länge höchstens

$$|\omega| \cdot s^{\mathrm{card}(\Sigma)-1} < \sigma$$

erzeugt werden können und $\#_a w_0 > \sigma$ gilt, muß D ein zyklisches Vorkommen enthalten und kann somit wie angegeben dargestellt werden. Wir betrachten das Wort $\widetilde{w}_1' v_1' x v_2' \widetilde{w}_2'$ der Ableitung. Von diesem Wort ausgehend konstruieren wir eine Ableitung gemäß dem gegebenen klE0L-System G wie folgt:

$\widetilde{w}_i'$	für $i = 1, 2$	wird gemäß	$\widetilde{w}_i' \Longrightarrow_{(1)}^* \widetilde{w}_i'' \in \Delta'^*$ ersetzt,
v_i'	für $i = 1, 2$	wird gemäß	$v_i' \Longrightarrow_{(1)}^* v_i \in \Delta'^*$ ersetzt,
x		wird gemäß	$x \Longrightarrow_{(1)}^* v_1' x v_2'$ ersetzt.

Dann wird erneut das Teilwort $v_1' x v_2'$ ersetzt, diesmal gemäß $v_i' \Longrightarrow_{(1)}^* v_i$ und $x \Longrightarrow_{(1)}^* v_0$. Das bedeutet, daß in bezug auf w_0 die Teilwörter v_1 und v_2 zusätzlich erzeugt werden. Da alle diese Ersetzungen in k-limitierter Form durchgeführt werden müssen, ist es immer möglich, daß die Simulation einiger der obigen sequentiellen Ersetzungen schon abgeschlossen sind und zu Symbolen aus Δ' geführt haben, während andere noch beendet werden müssen. Werden jedoch auf Symbole aus Δ' in weiteren Ersetzungsschritten nur Produktionen der Form $a_\nu \to a_{\mu_\nu}$ angewendet, so wird dadurch die Länge des erzeugten Wortes aus Δ' nicht beeinflußt. Wir erhalten so eine Ableitung

$$\omega \Longrightarrow^* \widetilde{w}_1' v_1' x v_2' \widetilde{w}_2' \Longrightarrow^* w_1 \text{ mit } w_1 \in \Delta'^* \text{ und } |w_1| = |w_0| + |v_1 v_2|.$$

Da dieses Vorgehen iteriert werden kann, folgt mit $c = |v_1 v_2| > 0$ die Eigenschaft (a) des Satzes. □

Beispiel 14.4 Die folgenden Sprachen besitzen jeweils eine Teilsprache L', die die Vorausetzung aus Satz 14.8 erfüllen. Da $m \in \mathbb{N}$ beliebig gewählt werden kann, ist aufgrund der jeweiligen Gestalt der Sprachen die Folgerung des Satzes nicht richtig. Daher sind die Sprachen keine klE0L-Sprachen.

(a) $\{a^{2^n} \mid n \in \mathbb{N}_0\}$,
(b) $\{a^{2^n} \mid n \in \mathbb{N}_0\} \cup \{cde^n \mid n \in \mathbb{N}\}$,
(c) $\{a, ab^2, ab^2a^3, ab^2a^3b^4, \ldots\}$,
(d) $\{a^p \mid p \text{ prim}\}$,
(e) $\{a^{n^2} \mid n \in \mathbb{N}\}$. □

Satz 14.9 Für alle $k \in \mathbb{N}$ gilt $\mathfrak{L}(k\text{lE0L}) \subsetneq \mathfrak{L}(k\text{lET0L})$.

Beweis: Nach dem Beweis von Satz 14.7 ist die Sprache $L = \{a^{2^n} \mid n \in \mathbb{N}_0\}$ eine klET0L-Sprache, nach Beispiel 14.4(a) ist sie jedoch keine klE0L-Sprache. □

Satz 14.10 Für $k \in \mathbb{N}$ und $m, n \in \mathbb{N}$ sei $G_k = (\{a,b\}, h, a^n b^m, k)$ ein klD0L-System mit $h(a) = b$ und $h(b) = a$. Ist

$$k \leq \min\{n,m\} \text{ für } n \neq m,\ k < n = m \text{ oder } k = n = m = 1,$$

so folgt $L(G_k) = S_{n,m} = \{w \in \{a,b\}^+ \mid \#_a w = n, \#_b w = m\}$.

Beweis: Für $k = 1$ bedeutet die einmalige Anwendung von h das Vertauschen von einem Vorkommen von a mit einem Vorkommen von b. Wir sehen sofort, daß $L(G_1) = S_{n,m}$ folgt. Für $1 < k$ mit den übrigen Voraussetzungen des Satzes gilt offensichtlich $L(G_k) \subset S_{n,m} = L(G_1)$. Wir müssen zeigen, daß ein Ableitungsschritt gemäß G_1 durch (zwei) Ableitungsschritte gemäß G_k simuliert werden kann. Wir können ohne Beschränkung der Allgemeinheit annehmen, daß $k < n = m$ oder $k \leq n < m$ gilt und somit $m - k - 1 \geq 0$ erfüllt ist. Wir betrachten einen Ableitungsschritt

$$a^n b^m \Longrightarrow_{G_1} ba^{n-1}ab^{m-1}.$$

Dieser wird durch

$$a^n b^m \Longrightarrow_{G_k} b^k a^{n-k} b^{m-k} a^k \Longrightarrow_{G_k} ba^{k-1}a^{n-k}ab^{m-k-1}b^k = ba^{n-1}ab^{m-1}$$

simuliert. Entsprechend können andere Vertauschungen von G_1 durch G_k simuliert werden. □

Satz 14.11 Für alle $k \in \mathbb{N}$ gilt $\mathfrak{L}(\text{ET0L}) \subsetneq \mathfrak{L}(k\text{lET0L})$.

Beweis: Wir gehen von einem beliebigen ET0L-System $G = (\Sigma, H, \omega, \Delta)$ mit $H = \{h_1, \ldots, h_n\}$ und $L = L(G)$ aus. Dazu definieren wir ein klET0L-System $G' = (\Sigma', H', \omega, \Delta, k)$ mit

$$\begin{aligned}\Sigma' &= \Sigma \cup \bigcup_{i=1}^{n} \{X_x^i \mid x \in \Sigma\} \cup \bigcup_{i=1}^{n} \{Y_x^i \mid x \in \Sigma\} \cup \{F\} \text{ und}\\ H' &= \bigcup_{i=1}^{n} \{h_{i1}, h_{i2}, h_{i3}\},\end{aligned}$$

wobei die Tafeln für $i, i' = 1, \ldots, n$, $i \neq i'$, $j = 1, 2$ und $x \in \Sigma$ durch

$$\begin{aligned}&h_{i1}(x) = \{Y_x^i\}, \quad h_{i1}(Y_x^i) = \{Y_x^i\}, \quad h_{i1}(X_x^i) = \{F\},\\ &h_{i2}(x) = \{F\}, \quad h_{i2}(Y_x^i) = \{X_x^i\}, \quad h_{i2}(X_x^i) = \{X_x^i\},\\ &h_{i3}(x) = \{x\}, \quad h_{i3}(Y_x^i) = \{F\}, \quad h_{i3}(X_x^i) = h_i(x) \text{ und}\\ &h_{ij}(Y_x^{i'}) = h_{ij}(X_x^{i'}) = h_{ij}(F) = \{F\}\end{aligned}$$

gegeben sind (vergleiche Beweisteil (b) von Satz 14.7) Ohne F zu erzeugen und ohne triviale Schritte $w \Longrightarrow w$ zu benutzen, sind für $i = 1, \ldots, n$ nur Ableitungen der Form

$$x_1 \ldots x_m \Longrightarrow_{h_{i1}}^* Y_{x_1}^i \ldots Y_{x_m}^i \Longrightarrow_{h_{i2}}^* X_{x_1}^i \ldots X_{x_m}^i \Longrightarrow_{h_{i3}}^* w_1 \ldots w_m$$

mit $w_\mu \in h_i(x_\mu)$ für $\mu = 1, \ldots, m$ möglich. Wir erhalten $L(G') = L$ und schließen, daß $\mathfrak{L}(\text{ET0L}) \subset \mathfrak{L}(k\text{lET0L})$ gilt. Wir betrachten die EPDT0L-Sprache

$$K = \{a^{2^n} b^{3^n} \mid n \in \mathbb{N}_0\},$$

die von dem PD0L-System

$$G_1 = (\{a, b\}, h_1, ab) \quad \text{mit} \quad h_1(a) = a^2,\ h_1(b) = b^3$$

erzeugt wird. Nach der Konstruktion aus der ersten Hälfte des Beweises wird K von einem klET0L-System

$$G_1' = (\Sigma, H', ab, \{a, b\}, k)$$

mit $\text{card}(H') \geq 3$ erzeugt. Mit Hilfe von G_1' bilden wir ein weiteres klET0L-System

$$G_2 = (\Sigma, H, a^{2^r} b^{2^r}, \{a, b\}, k),$$

wobei $r \in \mathbb{N}$ die kleinste Zahl mit $k \leq 2^r$ ist und $H = H' \cup \{h\}$ mit

$$h(a) = b, \quad h(b) = a, \quad h(x) = F \text{ für } x \in \Sigma - \{a, b\}$$

gilt. Um nicht das Fehlschlagsymbol F einzuführen, darf h nur nach der Erzeugung eines Wortes aus $\{a, b\}^+$ angewendet werden. Nach Satz 14.10 folgt

$$L(G_2) = \{w \in \{a, b\}^+ \mid \#_a w = 2^n,\ \#_b w = 3^n,\ k \leq 2^n,\ n \in \mathbb{N}_0\}.$$

Die Sprache

$$\hat{L} = \{w \in \{a, b\}^+ \mid \#_a w = 2^n,\ \#_b w = 3^n,\ 2^n < k,\ n \in \mathbb{N}_0\}$$

ist endlich und somit nach Satz 13.10 eine ET0L-Sprache. Wäre $L(G_2) \in \mathfrak{L}(\text{ET0L})$, so würde wegen der AFL-Eigenschaft von $\mathfrak{L}(\text{ET0L})$ (Satz 13.19)

$$L' = L(G_2) \cup \hat{L} = \{w \in \{a, b\}^+ \mid \#_a w = 2^n,\ \#_b w = 3^n,\ n \in \mathbb{N}_0\} \in \mathfrak{L}(\text{ET0L})$$

folgen. Nach Satz 13.25 gilt jedoch $L' \notin \mathfrak{L}(\text{ET0L})$. □

Satz 14.12 Für alle $k \in \mathbb{N}$ existiert ein Algorithmus, der zu jedem klET0L-System G' ein äquivalentes klET0L-System $G = (\Sigma, H, \omega, \Delta, k)$ konstruiert, so daß $\omega \in \Sigma - \Delta$ gilt und ein Symbol $F \in \Sigma - (\Delta \cup \{\omega\})$ sowie Tafeln $h_I, h_T \in H$ (Fehlschlagsymbol, initiale und terminale Tafel) existieren mit folgenden Eigenschaften:

(a) Es ist $h_I(\omega) \subset (\Sigma - (\Delta \cup \{\omega\}))^*$, und für $a \in \Sigma$, $a \neq \omega$, gilt $h_I(a) = \{a\}$.

(b) Es gilt $h_T(a) \subset (\Delta \cup \{F\})^*$ für $a \in \Sigma - (\Delta \cup \{\omega, F\})$ und $h_T(a) = \{a\}$ für $a \in \Delta \cup \{\omega, F\}$.

(c) Ist $h \in H - \{h_I, h_T\}$, so ist $h(a) \subset (\Sigma - (\Delta \cup \{\omega\}))^*$ für $a \in \Sigma - (\Delta \cup \{\omega, F\})$ und $h(a) = \{a\}$ für $a \in \Delta \cup \{\omega, F\}$.

Beweis: Es sei $G' = (\Sigma_1, H_1, \omega_1, \Delta, k)$ ein beliebiges klET0L-System. Wir nehmen an, daß $L(G') \neq \emptyset$ gilt, da anderenfalls die Aussage des Satzes trivialer Weise erfüllt ist. Mit Hilfe der neuen Alphabete

$$\Delta' = \{a' \mid a \in \Delta\} \quad \text{und} \quad \Delta'' = \{a'' \mid a \in \Delta\}$$

und den neuen Symbolen ω und F wird das Alphabet

$$\Sigma = \Sigma_1 \cup \Delta' \cup \Delta'' \cup \{\omega, F\}$$

bestimmt. Für jedes Wort $\alpha \in \Sigma_1^*$ definieren wir α'', indem ein jedes Symbol $a \in \Delta$ in α durch das Symbol $a'' \in \Delta''$ ersetzt wird, die Symbole aus $\Sigma_1 - \Delta$ jedoch unverändert bleiben. Speziell wird $\varepsilon'' = \varepsilon$ gesetzt. Auf entsprechende Weise erhalten wir α'. Die Menge

$$H = \{h_I, h_T, h_C\} \cup \{h' \mid h \in H_1\}$$

der Tafeln wird im folgenden bestimmt. Es werde

$$\begin{aligned} h_I(\omega) &= \{\omega_1''\}, \\ h_I(x) &= \{x\} \text{ für } x \in \Sigma - \{\omega\}, \end{aligned}$$

$$\begin{aligned} h_T(a') &= \{a\} \text{ für } a \in \Delta, \\ h_T(x) &= \{x\} \text{ für } x \in \Delta \cup \{\omega\}, \\ h_T(x) &= \{F\} \text{ für } x \in \Sigma - (\Delta \cup \Delta' \cup \{\omega\}) \text{ und} \end{aligned}$$

$$\begin{aligned} h_C(a'') &= \{a'\} \text{ für } a \in \Delta, \\ h_C(x) &= \{x\} \text{ für } x \in \Delta \cup \Delta' \cup \{\omega, F\} \text{ sowie} \\ h_C(x) &= \{F\} \text{ für } x \in \Sigma_1 - \Delta \end{aligned}$$

gesetzt. Für jedes $h \in H_1$ definieren wir

$$\begin{aligned} h'(x) &= \{\alpha'' \mid \alpha \in h(x)\} \text{ für } x \in \Sigma_1 - \Delta, \\ h'(a'') &= \{\alpha'' \mid \alpha \in h(a)\} \text{ für } a'' \in \Delta'', \\ h'(x) &= \{x\} \text{ für } x \in \Delta \cup \{\omega, F\} \text{ und} \\ h'(a') &= \{F\} \text{ für } a' \in \Delta'. \end{aligned}$$

Dann ist durch $G = (\Sigma, H, \omega, \Delta, k)$ ein klET0L-System festgelegt. Ohne triviale Anwendungen von Tafeln erhalten wir

$$\omega_1 \Longrightarrow_{h_1} w_1 \Longrightarrow_{h_2} w_2 \Longrightarrow_{h_3} \ldots \Longrightarrow_{h_r} w_r \in \Delta^* \text{ gemäß } G'$$

genau dann, wenn

$$\omega \Longrightarrow_{h_I} \omega_1'' \Longrightarrow_{h_1'} w_1'' \Longrightarrow_{h_2'} w_2'' \Longrightarrow_{h_3'} \ldots \Longrightarrow_{h_r'} w_r'' \Longrightarrow_{h_C}^* w_r' \Longrightarrow_{h_T}^* w_r \in \Delta^* \text{ gemäß } G$$

erfüllt ist. Dabei werden für $w_r = \varepsilon$ die Tafeln h_C und h_T nicht angewendet.

Zusätzlich zum ähnlichen Beweis von Satz 13.16 benötigen wir hier wegen der k-Limitierung die Kontrolltafel h_C. Nur mit ihrer Hilfe können wir die zweimal gestrichenen Symbole in Terminalsymbole verwandeln. □

Satz 14.13 Für alle $k_1, k_2 \in \mathbb{N}$ gilt $\mathfrak{L}((k_1 \cdot k_2)\text{lET0L}) \subset \mathfrak{L}(k_1\text{lET0L})$.

Beweis: Nach Satz 14.12 wird jede $(k_1 \cdot k_2)$lET0L-Sprache von einem $(k_1 \cdot k_2)$lET0L-System

$$G = (\Sigma, H, \omega, \Delta, k_1 \cdot k_2)$$

mit $h(x) = \{x\}$ für alle $x \in \Delta$ und $h \in H$ erzeugt. Wir definieren

$$\Sigma_{(i)} = \{x_{(i)} \mid x \in \Sigma - \Delta\}, \quad i = 1, \ldots, k_2,$$

und setzen damit

$$\Sigma' = \Sigma \cup \bigcup_{i=1}^{k_2} \Sigma_{(i)} \cup \{0, \ldots, k_2\} \cup \{F\},$$

wobei alle neuen Symbole paarweise verschieden und nicht in Σ enthalten sind. Aus G wird das k_1lET0L-System

$$G' = (\Sigma', H', 0\omega, \Delta, k_1)$$

mit

$$H' = \{h_1, \ldots, h_{k_2}, h_T\} \cup \{h' \mid h \in H\}$$

konstruiert, wobei die Tafeln wie folgt gegeben sind. Für alle $h \in H$ setzen wir

$$\begin{aligned} h'(k_2) &= \{0\}, \\ h'(i) &= \{F\} \text{ für } i = 0, \ldots, k_2 - 1, \\ h'(x) &= \{x\} \text{ für } x \in \Sigma \cup \{F\} \text{ und} \\ h'(x_{(i)}) &= h(x) \text{ für } x_{(i)} \in \bigcup_{j=1}^{k_2} \Sigma_{(j)}. \end{aligned}$$

Weiter definieren wir

$$\begin{aligned} h_j(i) &= \{F\} \text{ für } i = 0, \ldots, k_2,\ i \neq j-1, \\ h_j(j-1) &= \{j\}, \\ h_j(x) &= \{x_{(j)}\} \text{ für } x \in \Sigma - \Delta, \\ h_j(x) &= \{x\} \text{ für } x \in \Delta \cup \bigcup_{i=1}^{k_2} \Sigma_{(i)} \cup \{F\} \end{aligned}$$

für alle $j = 1, \ldots, k_2$ und schließlich

$$\begin{aligned} h_T(0) &= \{\varepsilon\}, \\ h_T(x) &= \{F\} \text{ für } x \in \Sigma' - (\Delta \cup \{0\}) \text{ und} \\ h_T(x) &= \{x\} \text{ für } x \in \Delta. \end{aligned}$$

Zu zeigen ist $L(G) = L(G')$. Für Ableitungen gemäß G' gelten die folgenden Überlegungen.

(1) F ist das Fehlschlagssymbol.

(2) Für jedes Wort $w \in \Delta^*$ mit $w \Longrightarrow_{G'} w'$ gilt $w = w'$.

(3) Auf jedes Wort $w' = 0w$ mit $w \in \Sigma^*$ können ohne Einführung des Fehlschlagsymbols F nur die Tafeln h_1 oder h_T angewendet werden. Da h_T jedoch für jedes Wort $w \notin \Delta^*$ das Symbol F einführt, ist h_1 die einzige Tafel, deren Anwendung auf ein Wort w' mit $w \notin \Delta^*$ in späteren Schritten eventuell zu einem Terminalwort führt.

(4) Für jedes Wort $w \in (\Sigma - \Delta)^*$ existieren ohne Erzeugung von F nur Ableitungen der Form

$$0w \Longrightarrow_{h_1} 1w_1 \Longrightarrow_{h_2} 2w_2 \Longrightarrow_{h_3} \ldots \Longrightarrow_{h_{k_2}} k_2 w_{k_2} \Longrightarrow_{h'} 0w'$$

mit einer Tafel h' für $h \in H$. Wegen der Konstruktion der Tafeln gilt dabei

$$w' \in \Sigma^* \text{ und } \min\{k_1 \cdot k_2, \#_x w\} = \min\{k_1 \cdot k_2, \sum_{i=1}^{k_2} \#_{x_{(i)}} w_{k_2}\} \text{ für alle } x \in \Sigma.$$

Es wird somit genau der Ableitungsschritt $w \Longrightarrow_G w'$ simuliert.
Aus diesen Überlegungen erhalten wir $L(G) = L(G')$. □

Unmittelbar folgt

Satz 14.14 Für alle $k \in \mathbb{N}$ gilt $\mathfrak{L}(k\text{lET0L}) \subset \mathfrak{L}(1\text{lET0L})$. □

Ein offenes Problem ist dabei die Frage, ob diese Inklusion echt ist oder nicht.

Satz 14.15 Für alle $k \in \mathbb{N}$ ist $\mathfrak{L}(k\text{lET0L})$ unter Bildung von Vereinigung, Konkatenation, Homomorphismus, ε-freier Iteration und Spiegelbild abgeschlossen.

Beweis: Der Abschluß unter Spiegelbild ist trivial. Der Abschluß unter Vereinigung, konkatenation und Homomorphismus wird unter Verwendung von Satz 14.12 wie im Beweis von Satz 13.19 bewiesen. Es bleibt der Abschluß unter ε-freier Iteration nachzuweisen.

Es sei L eine beliebige klET0L-Sprache, die von einem klET0L-System

$$G' = (\Sigma', H', \omega', \Delta, k)$$

mit den Eigenschaften aus Satz 14.12 erzeugt wird. Wir konstruieren ein klET0L-System

$$G = (\Sigma, H, S, \Delta, k)$$

durch

$$\Sigma = \Sigma' \cup \{S\} \quad \text{und} \quad H = \{h_0\} \cup \{h \mid h' \in H'\},$$

wobei

$$\begin{aligned} h_0(S) &= \{S\omega', \omega'\}, \\ h_0(a) &= \{a\} \text{ für } a \in \Delta, \\ h_0(x) &= \{F\} \text{ für } x \in \Sigma - (\Delta \cup \{S\}) \end{aligned}$$

gilt und für alle h mit $h' \in H'$

$$h(S) = \{S\} \text{ und } h(x) = h'(x) \text{ für } x \in \Sigma'$$

gesetzt wird. Nach einer nichttrivialen Anwendung von h_0, die das Axiom ω' von G' als Teilwort liefert, kann ohne Einführung von F die Tafel h_0 erst dann erneut angewendet werden, wenn aus ω' ein Terminalwort abgeleitet wurde. Offensichtlich gilt $L(G) = L^+$. $\square$

Für alle $k \in \mathbb{N}$ ist es offen, ob die Familie $\mathfrak{L}(k\text{lET0L})$ unter inversem Homomorphismus und unter Durchschnitt mit regulären Sprachen abgeschlossen ist. Man kann jedoch zeigen, daß $\mathfrak{L}(k\text{lET0L})$ unter endlicher Substitution abgeschlossen ist.

In [7] wurde gezeigt, daß es Sprachen aus $\mathfrak{L}(1\text{lET0L})$ gibt, die nicht rekursiv sind. Damit gilt dann, im Gegensatz zu Satz 13.19, $\mathfrak{L}(1\text{lET0L}) \not\subset \mathfrak{L}_1$. Ob jede Typ-1-Sprache zu $\mathfrak{L}(1\text{lET0L})$ gehört, ist nicht bekannt.

Es werden auch andere Formen der Limitierung von L-Systemen untersucht. Bei *uniform-k-limitierten* ET0L-*Systemen* (siehe [29]) werden in jedem Ersetzungsschritt im jeweils vorgelegten Wort w genau $\min\{k, |w|\}$ Symbole von w ersetzt. Das bedeutet, daß für die Auswahl der zu ersetzenden Symbole die Art der Symbole gleichgültig ist, in diesem Sinn also von einer gleichmäßigen Limitierung gesprochen werden kann.

Literaturverzeichnis

[1] *H. Becker, H. Walter:* Formale Sprachen. Vieweg, Braunschweig 1977.

[2] *W. Brauer:* Automatentheorie. Teubner, Stuttgart 1984.

[3] *W. Bucher, H. Maurer:* Theoretische Grundlagen der Programmiersprachen. Automaten und Sprachen. Bibliographisches Institut. Mannheim 1984.

[4] *V. Claus:* Stochastische Automaten. Teubner, Stuttgart 1971.

[5] *J. Dassow, Gh. Păun:* Regulated Rewriting in Formal Language Theory. Akademie-Verlag, Berlin 1989 und Springer, Berlin 1989.

[6] *S. Eilenberg:* Automata, Languages, and Machines,Volume A. Academic Press. New York 1974.

[7] *H. Fernau:* Membership for 1-Limited ET0L Languages Is Not Decidable. J. Inform. Process. Cybernet. EIK **30**(1994), 191–211.

[8] *M. Frings:* Systeme mit eingeschränkt paralleler Ersetzung. Diplomarbeit, Technische Universität Braunschweig, 1985.

[9] *S. Ginsburg:* The mathematical theory of context-free languages. McGraw-Hill, New York 1966.

[10] *S. Ginsburg:* Algebraic and automata-theoretic properties of formal language. North-Holland, Amsterdam 1975.

[11] *M.A. Harrison:* Introduction to Formal Language Theory. Addison-Wesley, Reading 1978.

[12] *G.T. Herman, G. Rozenberg:* Developmental Systems and Languages. North-Holland, 1975.

[13] *H. Hermes:* Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit. Springer, Berlin 1961.

[14] *W.M.L. Holcombe:* Algebraic automata theory. Cambridge University Press, Cambridge 1982.

[15] *J.E. Hopcroft, J.D. Ullman:* Introduction to automata theory, languages, and computation, Addison-Wesley, Reading 1979.

[16] *G. Hotz, Th. Kretschmer:* The Power of the Greibach Normal Form. J. Inf. Process. Cybernet. **EIK 25**(1989), 507–512.

[17] *R.N. Moll, M.A. Arbib, A.J. Kfoury:* An Introduction to Formal Language Theory. Springer, New York 1988.

[18] *M. Penttonen:* One-Sided and Two-Sided Context in Formal Grammars. Information and Control **25**(1974), 371–392.

[19] *P. Prusienkiewicz, J. Hanan:* Lindenmayer Systems, Fractals, and Plants. Lecture Notes in Biomathematics, Vol **79**. Springer. Berlin 1989.

[20] *P. Prusienkiewicz, A. Lindenmayer:* The Algorithmic Beauty of Plants. Springer, Berlin 1990.

[21] *G.E. Révész:* Introduction to Formal Languages. McGraw-Hill, New York 1983.

[22] *G. Rozenberg, A. Salomaa:* The mathematical theory of L Systems. Academic Press, New York 1980.

[23] *A. Salomaa:* Formal Languages. Academic Press, New York 1973.

[24] *A. Salomaa:* Formale Sprachen. Springer, Berlin 1978.

[25] *A. Salomaa:* Computation and Automata. Cambridge University Press, Cambridge 1985.

[26] *P.H. Starke:* Abstrakte Automaten. Deutscher Verlag der Wissenschaften, Berlin 1969.

[27] *R. Szelepcsényi:* The Method of Forcing for Nondeterministic Automata. EATCS **33**(1987), 96–100.

[28] *D. Wätjen:* k-limited 0L Systems and Languages. J. Inf. Process. Cybernet. EIK **24**(1988), 267–285.

[29] *D. Wätjen:* On k-uniformly-limited T0L Systems and Languages. J. Inf. Process. Cybernet. EIK **26**(1990), 229–238.

[30] *D. Wätjen:* Theoretische Informatik. Eine Einführung. Oldenbourg Verlag, München 1994.

Index