Inhaltsverzeichnis

3. Beschreibungskomplexität von Grammatiken **77**
3.1. Definitionen 77
3.2. Anzahl der Nichtterminale 83
3.3. Anzahl der Regeln 97
3.4. Anzahl der Symbole 105

4. Beschreibungskomplexität endlicher Automaten **111**
4.1. Eine algebraische Charakterisierung der regulären Sprachen 111
4.2. Minimierung endlicher Automaten 114

Literaturverzeichnis **121**

Kapitel 3

Beschreibungskomplexität von Grammatiken

3.1. Definitionen

Wir geben in diesem Abschnitt erst eine kurze Wiederholung von Begriffen zur Theorie der formalen Grammatiken, vor allem um Bezeichnungen zu klären, und führen danach drei Maße für die Beschreibungskomplexität von Grammatiken ein.

Für ein Alphabet V bezeichnen wir mit V^* die Menge aller Wörter über V (einschließlich des Leerworts λ) und mit V^+ die Menge aller nichtleeren Wörter über V. Für ein Wort $w \in V^*$ und eine Menge $U \subseteq V$ bezeichnen $|w|$ die Länge des Wörtes w und $|w|_U$ die Anzahl der Vorkommen von Buchstaben aus U in w. Für $a \in V$ schreiben wir einfach $|w|_a$ anstelle von $|w|_{\{a\}}$. Eine Sprache über V ist eine Teilmenge von V^*. Eine Sprache L heißt λ-frei, falls $L \subseteq V^+$ gilt.

Eine *Regelgrammatik* (oder kurz *Grammatik*) ist ein Quadrupel

$$G = (N, T, P, S),$$

wobei

- N und T endliche, disjunkte Alphabete sind, deren Vereinigung wir mit V bezeichnen,
- P eine endliche Teilmenge von $(V^* \setminus T^*) \times V^*$ ist, und
- $S \in N$ gilt.

Die Menge N ist das Alphabet der Nichtterminale oder Hilfssymbole oder Variablen und T das der Terminale. Im Folgenden werden wir meist große lateinische Buchstaben zur Bezeichnung der Nichtterminale und kleine lateinische Buchstaben für die Terminale verwenden, wobei die Buchstaben auch indiziert sein können. Die Elemente aus P heißen Regeln. Meistens werden wir das Paar (α, β) aus P in der Form $\alpha \to \beta$ schreiben, da diese Notation der Anwendung von Regeln als Ersetzung entspricht. Das Symbol S heißt Axiom oder Startwort.

Wir sagen, dass in der Grammatik $G = (N, T, P, S)$ aus dem Wort $\gamma \in V^+$ das Wort $\gamma' \in V^*$ *direkt erzeugt* wird, wenn

$$\gamma = \gamma_1 \alpha \gamma_2, \quad \gamma' = \gamma_1 \beta \gamma_2, \quad \alpha \to \beta \in P$$

für gewisse $\gamma_1, \gamma_2 \in V^*$ gelten. Wir schreiben dann

$$\gamma \Longrightarrow \gamma'.$$

Falls die bei der Erzeugung verwendete Regel $p = \alpha \to \beta$ betont werden soll, so schreiben wir $\gamma \Longrightarrow_p \gamma'$. Durch $\Longrightarrow$ wird offenbar eine Relation definiert. Wie üblich kann hiervon der reflexive und transitive Abschluss $\Longrightarrow^*$ gebildet werden, d. h. es gilt

$$\gamma \Longrightarrow^* \gamma'$$

genau dann, wenn es eine natürliche Zahl $n \geq 0$ und Wörter $\delta_0, \delta_1, \delta_2, \ldots, \delta_{n-1}, \delta_n$ mit

$$\gamma = \delta_0 \Longrightarrow \delta_1 \Longrightarrow \delta_2 \Longrightarrow \cdots \Longrightarrow \delta_{n-1} \Longrightarrow \delta_n = \gamma'$$

gibt (im Fall $n = 0$ gilt $\gamma = \gamma'$, und im Fall $n = 1$ haben wir $\gamma \Longrightarrow \gamma'$); $\gamma \Longrightarrow^* \gamma'$ gilt genau dann, wenn γ' durch iterierte Anwendung von (im Allgemeinen verschiedenen) Regeln aus γ entsteht. Gilt $\gamma \Longrightarrow^* \gamma'$, so sagen wir auch, γ' ist aus γ (in mehreren Schritten) *ableitbar* oder *erzeugbar*.

Ein Wort $w \in V^*$ heißt *Satzform* von G, wenn $S \Longrightarrow^* w$ gilt, d. h. wenn w aus S erzeugt werden kann. Mit $S(G)$ bezeichnen wir die Menge aller Satzformen von G.

Für eine Grammatik $G = (N, T, P, S)$ ist die *von G erzeugte Sprache* $L(G)$ durch

$$L(G) = \{w \mid w \in T^* \text{ und } S \Longrightarrow^* w\}$$

definiert. Die von G erzeugte Sprache besteht also aus allen Satzformen von G, die nur Terminale enthalten.

Für eine Grammatik G und ein Nichtterminal A von G setzen wir noch

$$L(G, A) = \{w \mid w \in T^* \text{ und } A \Longrightarrow^* w\}.$$

$L(G, A)$ ist also die Sprache, die bei Verwendung von A als Axiom erzeugt wird. Offenbar gilt $L(G) = L(G, S)$.

Zwei Grammatiken sind einander äquivalent, wenn sie die gleiche Sprache erzeugen.

Wir sagen, die Regelgrammatik $G = (N, T, P, S)$ ist

- *monoton*, wenn für alle Regeln $\alpha \to \beta \in P$ die Bedingung $|\alpha| \leq |\beta|$ erfüllt ist, wobei als Ausnahme $S \to \lambda$ zugelassen ist, wenn $|\beta'|_S = 0$ für alle Regeln $\alpha' \to \beta' \in P$ gilt,

- *kontextfrei*, wenn alle Regeln in P von der Form $A \to w$ mit $A \in N$ und $w \in V^*$ sind,

- *regulär*, wenn alle Regeln in P von der Form $A \to wB$ oder $A \to w$ mit $A, B \in N$ und $w \in T^*$ sind.

Eine kontextfreie Grammatik $G = (N, T, P, S)$ ist

- *reduziert*, wenn für jedes $A \in N$ sowohl eine Ableitung $S \Longrightarrow^* w_1 A w_2$ als auch eine Ableitung $A \Longrightarrow^* w$ mit $w_1, w_2 \in V^*$ und $w \in T^*$ existieren,

- λ-*frei*, wenn alle Regeln in P von der Form $A \to w$ mit $w \in V^+$ sind, wobei als Ausnahme $S \to \lambda$ zugelassen ist, wenn $|v|_S = 0$ für alle Regeln $B \to v \in P$ gilt,

- in *Chomsky-Normalform*, wenn alle Regeln in P von der Form $A \to BC$ oder $A \to a$ mit $A, B, C \in N$ und $a \in T$ sind, wobei als Ausnahme $S \to \lambda$ zugelassen ist, wenn $|v|_S = 0$ für alle Regeln $D \to v \in P$ gilt.

Bei einer reduzierten Grammatik lässt sich also jede Satzform so weiter ableiten, dass ein Wort über der Terminalmenge entsteht.

Eine reguläre Grammatik $G = (N, T, P, S)$ ist in *regulärer Normalform*, wenn alle Regeln aus P von der Form $A \to aB$ oder $A \to a$ mit $A, B \in N$ und $a \in T$ sind, wobei als Ausnahme $S \to \lambda$ zugelassen ist, wenn $|v|_S = 0$ für alle Regeln $C \to v \in P$ gilt.

Es ist bekannt, dass es zu jeder kontextfreien Grammatik G eine reduzierte kontextfreie Grammatik G_1, eine kontextfreie λ-freie Grammatik G_2 und eine Grammatik G_3 in Chomsky-Normalform derart gibt, dass $L(G) = L(G_1) = L(G_2) = L(G_3)$ gilt. Außerdem gibt es zu jeder regulären Grammatik H eine Grammatik H' in regulärer Normalform, so dass $L(H) = L(H')$ erfüllt ist.

Im Folgenden sei eine abzählbar unendliche Menge $\mathcal{U}$ gegeben und für jede Grammatik $G = (N, T, P, S)$ gelte stets $N \subset \mathcal{U}$ und $T \subset \mathcal{U}$. Wir bezeichnen mit

RE	die Menge aller Regelgrammatiken,
MON	die Menge aller monotonen Grammatiken,
CF	die Menge aller kontextfreien Grammatiken,
CF-λ	die Menge aller kontextfreien λ-freien Grammatiken,
redCF	die Menge aller reduzierten kontextfreien Grammatiken,
ChCF	die Menge aller kontextfreien Grammatiken in Chomsky-Normalform,
REG	die Menge aller regulären Grammatiken,
nfREG	die Menge aller Grammatiken in regulärer Normalform.

Für eine Menge X von Grammatiken (im Folgenden wird stets

$$X \in \{RE, MON, CF, CF\text{-}\lambda, redCF, ChCF, REG, nfREG\}$$

gelten) bezeichnen wir mit $\mathcal{L}(X)$ die Menge aller Sprachen, die von Grammatiken aus X erzeugt werden können.

Es gilt

$$\begin{aligned}\mathcal{L}(nfREG) &= \mathcal{L}(REG)\\ &\subset \mathcal{L}(ChCF) = \mathcal{L}(CF\text{-}\lambda) = \mathcal{L}(redCF) = \mathcal{L}(CF)\\ &\subset \mathcal{L}(MON)\\ &\subset \mathcal{L}(RE).\end{aligned}$$

Für eine Grammatik $G = (N, T, P, S)$ führen wir die folgenden drei Komplexitätsmaße ein:

$$\begin{aligned} \mathrm{Var}(G) &= \#(N), \\ \mathrm{Prod}(G) &= \#(P), \\ \mathrm{Symb}(G) &= \sum_{\alpha \to \beta \in P} (|\alpha| + |\beta| + 1). \end{aligned}$$

Es sei nun X eine Menge von Grammatiken. Wir erweitern nun die obigen Komplexitätsmaße auf Sprachen $L \in \mathcal{L}(X)$ mittels

$$\begin{aligned} \mathrm{Var}_X(L) &= \min\{\mathrm{Var}(G) \mid L = L(G),\ G \in X\}, \\ \mathrm{Prod}_X(L) &= \min\{\mathrm{Prod}(G) \mid L = L(G),\ G \in X\}, \\ \mathrm{Symb}_X(L) &= \min\{\mathrm{Symb}(G) \mid L = L(G),\ G \in X\}. \end{aligned}$$

Wir sagen, eine Grammatik $G \in X$ ist *minimal* für eine Sprache $L \in \mathcal{L}(X)$ bezüglich eines Komplexitätsmaßes $K \in \{\,\mathrm{Var}, \mathrm{Prod}, \mathrm{Symb}\,\}$, falls $L(G) = L$ und $K(G) = K_X(L)$ gelten.

Beispiel 3.1 *Wir betrachten die Grammatik*

$$G_1 = (\{S, U, X, X', Y, Z\}, \{a, b\}, P_1, S)$$

mit

$$\begin{aligned} P_1 = \{&S \to YX'aZ,\ YX' \to YX,\ YX' \to b^2U,\ Ua \to aU,\ UZ \to b^2, \\ &Xa \to aaX,\ XZ \to X'Z,\ aX' \to X'a\}. \end{aligned}$$

Es ist leicht zu sehen, dass die Satzformen von G_1 eine der folgenden Formen

$$S,\ Ya^{2n}Xa^mZ,\ Ya^kX'a^lZ,\ b^2a^kUa^lZ,\ b^2a^{2^r}b^2$$

mit $n \geq 0$, $m \geq 0$, $2n + 2m = 2^r$, $k \geq 0$, $l \geq 0$, $k + l = 2^r$, $r \geq 0$ haben, und dass umgekehrt jedes Wort dieser Form auch Satzform ist. Damit gilt

$$L_1 = L(G_1) = \{b^2a^{2^r}b^2 \mid r \geq 0\}.$$

Als Komplexitätsmaße von G_1 erhält man durch einfaches Auszählen

$$\mathrm{Var}(G_1) = 6, \quad \mathrm{Prod}(G_1) = 8 \quad \textit{und} \quad \mathrm{Symb}(G_1) = 43.$$

Damit ergibt sich auch

$$\mathrm{Var}_{MON}(L_1) \leq 6, \quad \mathrm{Prod}_{MON}(L_1) \leq 8 \quad \textit{und} \quad \mathrm{Symb}_{MON}(L_1) \leq 43.$$

Die monotone Grammatik

$$G_1' = (\{S, U, X, X'\}, \{a, b, Y, Z\}, P_1, S)$$

erzeugt offensichtlich auch L_1, denn wir erhalten die gleichen Ableitungen wie in G_1. Der Unterschied zwischen G_1 und G_1' liegt darin, dass wir bei G_1' die Symbole Y und Z als

Terminale auffassen (obwohl sie in keinem Wort der Sprache vorkommen). Für G_1' erhalten wir

$$\mathrm{Var}(G_1') = 4, \quad \mathrm{Prod}(G_1) = 8 \quad \textit{und} \quad \mathrm{Symb}(G_1) = 43.$$

Folglich ergibt sich für die Anzahl der Variablen die Verschärfung $\mathrm{Var}_{MON}(L_1) \leq 4$.

Beispiel 3.2 *Es sei r eine beliebige positive natürliche Zahl. Für die kontextfreie Grammatik*

$$G_2 = (\{S, A\}, \{a, b\}, \{S \to A^r,\ A \to aA,\ A \to Ab,\ A \to ab\}, S)$$

ergeben sich offenbar

$$L_{2,r} = L(G_2) = \{a^n b^m \mid n \geq 1,\ m \geq 1\}^r$$

und

$$\mathrm{Var}(G_2) = 2, \quad \mathrm{Prod}(G_2) = 4 \quad \textit{und} \quad \mathrm{Symb}(G_2) = 14 + r.$$

$L_{2,r}$ ist für jedes $r \geq 1$ eine reguläre Sprache, da z. B. die reguläre Grammatik

$$G_2' = (\{A_i \mid 1 \leq i \leq r\} \cup \{B_i \mid 1 \leq i \leq r\}, \{a, b\}, P_2, A_1)$$

mit

$$\begin{aligned} P_2 = {} & \{A_i \to aA_i \mid 1 \leq i \leq r\} \cup \{A_i \to aB_i \mid 1 \leq i \leq r\} \\ & \cup \{B_i \to bB_i \mid 1 \leq i \leq r\} \cup \{B_i \to bA_{i+1} \mid 1 \leq i \leq r-1\} \cup \{B_r \to b\} \end{aligned}$$

ebenfalls $L_{2,r}$ erzeugt. Wegen $\mathrm{Var}(G_2') = 2r$ gilt $\mathrm{Var}_{REG}(L_{2,r}) \leq 2r$. Wir wollen nun zeigen, dass hinsichtlich der Anzahl der Nichtterminale G_2' sogar optimal ist.

Es sei daher $G = (N, \{a, b\}, P, S)$ eine (beliebige) reguläre Grammatik mit $L(G) = L_{2,r}$ und $\mathrm{Var}(G) = \mathrm{Var}_{REG}(L_{2,r})$.

Wir bemerken zuerst, dass für jedes $A \in N$ mit $A \neq S$ eine Ableitung der Form $A \Longrightarrow^* uA$ mit $u \neq \lambda$ existiert. Angenommen, es gibt ein $A \neq S$, für das es keine Ableitung dieses Typs gibt. Dann gibt es nur endlich viele Wörter über T, die von A (in beliebig vielen Schritten) erzeugt werden können, d. h. $L(G, A)$ ist endlich. Jeder Regel $p = X \to vA$ mit $v \in T^*$ ordnen wir die Menge

$$P_p = \{X \to vz \mid z \in L(G, A)\}$$

zu, und setzen $G_A = (N_A, T, P_A, S)$ mit

$$N_A = N \setminus \{A\} \quad \textit{und} \quad P_A = \{\, p \,|\, p = B \to vC \in P, B \neq A, C \neq A \,\} \cup \bigcup_{\substack{p = B \to w \in P \\ B \neq A}} P_p.$$

Da wir jedes Vorkommen von A in einer rechten Seite einer Regel durch ein terminales Wort ersetzen, dass aus A erzeugt werden kann, ist leicht zu sehen, dass

$$L(G_A) = L(G) = L_{2,r}$$

gilt. Da aber $\mathrm{Var}(G_A) = \mathrm{Var}(G) - 1 = \mathrm{Var}_{REG}(L_{2,r}) - 1$ *gilt, erhalten wir einen Widerspruch.*

Es sei nun

$$S \Longrightarrow^* vA \Longrightarrow^* vuA \Longrightarrow^* vuw$$

eine Ableitung eines Terminalwortes. Falls für die Ableitung $A \Longrightarrow^* uA$ *eine der Beziehungen* $u = u_1abu_2$ *oder* $u = u_1bau_2$ *gilt, so liefert die Ableitung*

$$S \Longrightarrow^* vA \Longrightarrow^* vuA \Longrightarrow^* vuuA \Longrightarrow^* vu^3A \Longrightarrow^* \cdots \Longrightarrow^* vu^{2r}A \Longrightarrow^* vu^{2r}w$$

ein Wort in dem mindestens $2r$*-mal ein Wechsel von* a *nach* b *oder von* b *nach* a *erfolgt. Dies steht im Widerspruch zur Struktur der Wörter in* $L_{2,r}$*, bei denen höchstens* r *derartige Wechsel auftreten. Folglich gilt bei* $A \Longrightarrow^* uA$ *entweder* $u \in \{a\}^+$ *oder* $u \in \{b\}^+$*. Somit erfordert jeder Wechsel von* a *nach* b *oder von* b *nach* a *mindestens einen Wechsel der Nichtterminale. Weiterhin sind diese Nichtterminale paarweise voneinander verschieden, da sonst eine Ableitung* $A \Longrightarrow^* uA$ *derart existiert, dass* ab *oder* ba *ein Teilwort von* u *ist. Damit hat* G *mindestens* $2r$ *Nichtterminale. Deshalb gilt* $\mathrm{Var}_{REG}(L_{2,r}) \geq 2r$*.*

Aufgrund der regulären Grammatik G_2' *mit* $L(G_2') = L_{2,r}$ *und* $\mathrm{Var}(G_2') = 2r$ *erhalten wir*

$$\mathrm{Var}_{REG}(L_{2,r}) = 2r.$$

Das Beispiel 3.2 belegt, dass kontextfreie Grammatik erheblich effizienter als reguläre Grammatiken bei der Beschreibung von Sprachen sein können. Wir definieren daher einige mögliche Relationen zwischen den Komplexitätsmaßen.

Es sei $K \in \{\mathrm{Var}, \mathrm{Prod}, \mathrm{Symb}\}$ ein Komplexitätsmaß, und seien X und Y zwei Mengen von Grammatiken, für die $\mathcal{L}(X) \subset \mathcal{L}(Y)$ gilt. Wir schreiben

- $X \approx_K Y$, falls es eine Konstante k derart gibt, dass $|K_X(L) - K_Y(L)| \leq k$ für alle $L \in \mathcal{L}(X)$ gilt,
- $Y \leq_K^1 X$, falls es eine Folge L_n, $n \geq 1$, von Sprachen mit $L_n \in \mathcal{L}(X)$ derart gibt, dass $\lim_{n\to\infty}(K_X(L_n) - K_Y(L_n)) = \infty$ gilt,
- $Y \leq_K^2 X$, falls es eine Folge L_n, $n \geq 1$, von Sprachen $L_n \in \mathcal{L}(X)$ derart gibt, dass $\lim_{n\to\infty} \frac{K_X(L_n)}{K_Y(L_n)} = \infty$ gilt,
- $Y \leq_K^3 X$, falls es eine Folge L_n, $n \geq 1$, von Sprachen $L_n \in \mathcal{L}(X)$ und eine Konstante k derart gibt, dass $K_X(L_n) \geq n$ und $K_Y(L_n) \leq k$ gilt.

Intuitiv bedeutet, die approximative Gleichheit $X \approx_K Y$, dass sich die Komplexitäten $K_X(L_n)$ und $K_Y(L_n)$ einer Sprache, die von Grammatiken aus X und Y erzeugt werden kann, höchstens um eine Konstante unterscheiden können. Die Relation $\leq_K^1$ bedeutet, dass der Unterschied zwischen den beiden Komplexitäten $K_X(L_n)$ und $K_Y(L_n)$ beliebig groß werden kann. Bei $\leq_K^2$ wird der Unterschied zwischen den Komplexitäten nicht nur beliebig groß, sondern für jede Konstanten c gilt für hinreichend großes n sogar $c \cdot K_Y(L_n) \leq K_X(L_n) - K_Y(L_n)$, d. h. der Unterschied ist durch keine lineare Funktionen in $K_Y(L_n)$ beschränkt. Bei $\leq_K^3$ kann die Differenz $K_X(L_n) - K_Y(L_n)$ durch keinerlei Funktion mehr beschränkt werden, da für jede Funktion f die Differenz größer als der maximale der Werte $f(1), f(2), \ldots, f(k)$ wird. Offenbar gelten

aus $\leq_K^2$ folgt $\leq_K^1$,
aus $\leq_K^3$ folgt $\leq_K^2$.

Wegen der in Beispiel 3.2 gezeigten Beziehungen

$$\mathrm{Var}_{CF}(L_{2,r}) = 2 \quad \text{und} \quad \mathrm{Var}_{REG}(L_{2,r}) = 2r \geq r$$

für $r \geq 1$ gilt in dieser Terminologie

$$CF \leq_{\mathrm{Var}}^3 REG. \tag{3.1}$$

Für den Nachweis einer Relation vom Typ $X \leq_K^i Y$, $i \in \{1, 2, 3\}$ benötigt man Sprachen aus $\mathcal{L}(Y)$ beliebig großer Komplexität. Eine Verschärfung dieser Forderung besteht darin, dass jede beliebige positive natürliche Zahl im Wertevorrat von K_Y auftaucht.

Definition 3.3 *Wir sagen, dass ein Komplexitätsmaß $K \in \{\mathrm{Var}, \mathrm{Prod}, \mathrm{Symb}\}$ bezüglich der Menge Y von Grammatiken vollständig ist, wenn es für jede natürliche Zahl n eine Sprache $L_n \in \mathcal{L}(Y)$ derart gibt, dass $K_Y(L_n) = n$ gilt.*

3.2. Anzahl der Nichtterminale

In diesem Abschnitt untersuchen wir die Anzahl der zur Erzeugung einer Sprache notwendigen Nichtterminale. Wir behandeln zuerst die Beziehungen zwischen den Maßen Var_X und Var_Y für verschiedene Klassen X und Y von Grammatiken.

Lemma 3.4 *Für je zwei Mengen X und Y von Grammatiken mit $X \subseteq Y$ und jede Sprache $L \in \mathcal{L}(X)$ gilt $\mathrm{Var}_Y(L) \leq \mathrm{Var}_X(L)$.*

Beweis. Es sei $G \in X$ eine Grammatik mit $L(G) = L$ und $\mathrm{Var}(G) = \mathrm{Var}_X(L)$. Da auch $G \in Y$ nach Voraussetzung gilt, erhalten wir $\mathrm{Var}_Y(L) \leq \mathrm{Var}(G) = \mathrm{Var}_X(L)$. □

Lemma 3.5

i) *Für jede kontextfreie Sprache L gilt $\mathrm{Var}_{CF}(L) = \mathrm{Var}_{redCF}(L)$.*

ii) *Für jede kontextfreie λ-freie Sprache L gilt $\mathrm{Var}_{CF}(L) = \mathrm{Var}_{CF\text{-}\lambda}(L)$.*

Beweis. i) Da jede reduzierte kontextfreie Grammatik auch eine kontextfreie Grammatik ist, gilt nach Lemma 3.4

$$\mathrm{Var}_{CF}(L) \leq \mathrm{Var}_{redCF}(L). \tag{3.2}$$

Ist umgekehrt $L \in \mathcal{L}(CF)$ und $G = (N, T, P, S)$ eine kontextfreie Grammatik mit $L = L(G)$ und $\mathrm{Var}(G) = \mathrm{Var}_{CF}(L)$, so konstruieren wir die zugehörige reduzierte Grammatik G' durch Streichen aller Nichtterminale A von N, für die keine Ableitung der Form $S \Longrightarrow^* uAv$ oder $A \Longrightarrow^* w \in T^*$ existiert, und aller Regeln, in denen A vorkommt. Folglich gilt $\mathrm{Var}(G') \leq \mathrm{Var}(G)$. Dies impliziert

$$\mathrm{Var}_{redCF}(L) \leq \mathrm{Var}(G') \leq \mathrm{Var}(G) = \mathrm{Var}_{CF}(L),$$

woraus mit (3.2) sofort die Behauptung folgt.

ii) Der Beweis verläuft völlig analog. Wir haben nur zu beachten, dass die Standardkonstruktion einer λ-freien Grammatik aus einer Grammatik, die eine λ-freie Sprache erzeugt, die Anzahl der Nichtterminale nicht erhöht. □

Folgerung 3.6 *Für* $X \in \{CF, redCF, CF\text{-}\lambda, MON, RE\}$ *gilt* $X \leq^3_{\mathrm{Var}} REG$.

Beweis. Wegen $CF \subseteq RE$ und $CF\text{-}\lambda \subseteq MON$ folgt die Aussage aus (3.1) und den Lemmata 3.4 und 3.5. □

Satz 3.7 Var *ist bezüglich REG ein vollständiges Maß.*

Beweis. Es sei $n \geq 1$ eine gerade Zahl. Dann setzen wir

$$L_n = L_{2,\frac{n}{2}} = (\{a\}^+\{b\}^+)^{\frac{n}{2}}$$

und erhalten aus Beispiel 3.2 $\mathrm{Var}_{REG}(L_n) = n$.

Für ungerades $n \geq 1$ setzen wir

$$L_n = (\{a\}^+\{b\}^+)^{\frac{n-1}{2}}\{a\}^+$$

und können dafür wie in Beispiel 3.2 $\mathrm{Var}_{REG}(L_n) = n$ nachweisen. □

Bevor wir das Analogon zu Satz 3.7 für kontextfreie Sprachen angeben und beweisen, zeigen wir zwei Lemmata, die es uns erlauben, zukünftig für bezüglich der Anzahl der Nichtterminale minimale Grammatiken gewisse Voraussetzungen zu machen.

Lemma 3.8

i) *Zu einer kontextfreien Grammatik* $G = (N, T, P, S)$ *mit* $\mathrm{Var}(G) = \mathrm{Var}_{CF}(L(G))$ *gibt es für jedes* $A \in N$ *mit* $A \neq S$ *eine Regel* $A \to uAv$ *mit* $uv \neq \lambda$ *in* P.

ii) *Für eine kontextfreie Grammatik* $G = (N, T, P, S)$ *mit* $\mathrm{Var}(G) = \mathrm{Var}_{CF}(L(G))$ *ist für jedes* $A \in N$ *mit* $A \neq S$ *die Sprache* $L(G, A)$ *unendlich.*

Beweis. i) Wir nehmen an, dass es ein $A \in N$, $A \neq S$, so gibt, dass die rechten Seiten aller Regeln mit linker Seite A den Buchstaben A nicht enthalten. Wir setzen

$$P_A = \{w \mid A \to w \in P\}.$$

Wir zerlegen die rechten Seiten einer jeden Regel $p = B \to u_1Au_2A\ldots u_rAu_{r+1} \in P$ mit $B \neq A$ so, dass $u_i \in (V \setminus \{A\})^*$ für $1 \leq i \leq r+1$ ist, und setzen

$$P_p = \{B \to u_1w_1u_2w_2\ldots u_rw_ru_{r+1} \mid r \geq 0,\ w_i \in P_A,\ 1 \leq i \leq r\}.$$

(Falls die rechte Seite von p kein A enthält, ist $P_p = \{p\}$.) Wir betrachten nun die (kontextfreie) Grammatik

$$G' = (N \setminus \{A\}, T, \bigcup_{\substack{p=B\to z\in P\\ B\neq A}} P_p, S).$$

Es ist leicht zu sehen, dass $L(G') = L(G)$ gilt, denn die in G bzw. G' möglichen Ableitungen

$$\begin{aligned} S &\Longrightarrow^* v_1Bv_2 \Longrightarrow v_1u_1Au_2A\ldots u_rAu_{r+1}v_2 \\ &\Longrightarrow v_1u_1w_1u_2A\ldots u_rAu_{r+1}v_2 \\ &\Longrightarrow v_1u_1w_1u_2w_2u_3A\ldots u_rAu_{r+1}v_2 \\ &\quad\vdots \\ &\Longrightarrow v_1u_1w_1u_2w_2\ldots u_rw_ru_{r+1}v_2 \end{aligned}$$

und

$$S \Longrightarrow^* v_1Bv_2 \Longrightarrow v_1u_1w_1u_2w_2\ldots u_rw_ru_{r+1}v_2$$

sind gleichwertig. Da offenbar $\mathrm{Var}(G') = \mathrm{Var}(G) - 1 = \mathrm{Var}_{CF}(L) - 1$ gilt, erhalten wir einen Widerspruch.

ii) Der Beweis wird völlig analog zu dem für den Teil i) geführt, wobei wir nur P_A durch (die nach Annahme endliche Menge) $L(G, A)$ ersetzen. □

Satz 3.9 Var *ist bezüglich CF ein vollständiges Maß.*

Beweis. Wir betrachten die Sprachen

$$\begin{aligned} L_1 &= \{a\}, \\ L_2 &= \{a\}^+\{b\}^+\{a\}^+, \\ L_n &= \bigcup_{i=1}^{n-1}\{a^ib\}^+ \quad \text{für} \quad n \geq 3. \end{aligned}$$

Da die kontextfreien Grammatiken

$$\begin{aligned} G_1 &= (\{S\}, \{a\}, \{S \to a\}, S), \\ G_2 &= (\{S, A\}, \{a, b\}, \{S \to aS,\ S \to Sa,\ S \to aAa,\ A \to bA, A \to b\}, S), \\ G_n &= (\{S, A_1, A_2, \ldots, A_{n-1}\}, \{a, b\}, \bigcup_{i=1}^{n-1}\{S \to A_i,\ A_i \to a^ibA_i,\ A_i \to a^ib\}, S),\ n \geq 3, \end{aligned}$$

mit

$$\mathrm{Var}(G_1) = 1, \quad \mathrm{Var}(G_2) = 2 \quad \text{und} \quad \mathrm{Var}(G_n) = n \text{ für } n \geq 3$$

die Sprachen L_1, L_2 und L_n für $n \geq 3$ erzeugen, ist

$$\mathrm{Var}_{CF}(L_1) \leq 1, \quad \mathrm{Var}_{CF}(L_2) \leq 2 \quad \text{und} \quad \mathrm{Var}_{CF}(G_n) \leq n \text{ für } n \geq 3 \tag{3.3}$$

bereits gezeigt.

Es gilt $\mathrm{Var}(L_1) \geq 1$ offensichtlich, da jede L_1 erzeugende Grammatik mindestens ein Nichtterminal (das Axiom) hat. Folglich gilt $\mathrm{Var}_{CF}(L_1) = 1$.

Angenommen, es gibt eine kontextfreie Grammatik $G = (\{S\}, \{a, b\}, P, S)$ mit einem einzigen Nichtterminal S und $L_2 = L(G)$. Wegen Lemma 3.5 können wir ferner ohne

Beschränkung der Allgemeinheit annehmen, dass G reduziert ist. Wenn es eine Satzform gibt, in der S mindestens zweimal vorkommt, so gibt es in G eine Ableitung

$$S \Longrightarrow^* xSySz \Longrightarrow^* xabayabaz \Longrightarrow^* x'abay'abaz' = w$$

mit $x', y', z' \in T^*$. Die Struktur von w widerspricht aber der der Wörter von L_2, womit wir einen Widerspruch erhalten haben.

Folglich enthalten alle Satzformen von G höchstens ein S. Es sei nun p die maximale Länge der rechten Seiten von Regeln in P. L_2 enthält das Wort $ab^{2p}a$. Die Ableitung dieses Wortes erfordert daher die Existenz von Regeln der Form $S \to b^rSb^s$ oder $S \to b^t$ mit $r+s > 0$ bzw. $t > 0$, da sonst der Abstand von zwei Vorkommen von a durch p beschränkt wäre. Im ersten Fall gibt es in G die Ableitung $S \Longrightarrow b^rSb^s \Longrightarrow^* b^rababb^s$ und im zweiten Fall die Ableitung $S \Longrightarrow b^t$, die beide nicht zu Wörtern in L_2 führen. Erneut erhalten wir einen Widerspruch, der beweist, dass wir mindestens zwei Nichtterminale zur Erzeugung von G benötigen. Das bedeutet $\mathrm{Var}_{CF}(L_2) \geq 2$, woraus mit (3.3) dann $\mathrm{Var}_{CF}(L_2) = 2$ folgt.

Wir behandeln nun den Fall $n \geq 3$. $G = (N, T, P, S)$ sei eine kontextfreie Grammatik mit $L(G) = L_n$ und $\mathrm{Var}(G) = \mathrm{Var}_{CF}(L_n)$. Wegen Lemma 3.5 können wir annehmen, dass G λ-frei und reduziert ist.

Es sei A ein Nichtterminal mit $A \neq S$. Dann gibt es nach Lemma 3.8 eine Regel $A \to uAv$ mit $uv \neq \lambda$. Da G reduziert ist, gibt es in G die Ableitungen

$$\begin{aligned} S &\Longrightarrow^* z_1Az_2 \Longrightarrow z_1uAvz_2 \Longrightarrow z_1u^2Av^2z_2 \Longrightarrow \cdots \Longrightarrow z_1u^mAv^mz_2 \\ &\Longrightarrow^* z_1'(u')^mw(v')^mz_2' \end{aligned} \tag{3.4}$$

von Wörtern in $L(G)$, bei denen m eine beliebige positive natürliche Zahl und $w \in L(G, A)$ sind sowie (wegen der λ-Freiheit) $|(u')^m| \geq m$ oder $|(v')^m| \geq m$ gelten. Wir wählen nun $m \geq 2n$. Dann muss $(u')^m$ oder $(v')^m$ ein Teilwort $z = ba^ib$ für ein $i \in \{1, 2, \ldots, n-1\}$ enthalten. Ohne Beschränkung der Allgemeinheit nehmen wir $(u')^m = x_1ba^ibx_2$ an.

Nach Lemma 3.8 ist $L(G, A)$ unendlich. Es sei $y \in L(G, A)$ und $|y| \geq n+1$. Dann gibt es eine Zerlegung $y = y_1ba^jby_2$ mit gewissen Wörtern $y_1, y_2 \in T^*$. Nach (3.4) gilt dann $z_1'x_1ba^ibx_2y_1ba^jby_2(v')^mz_2' \in L(G) = L_n$. Aufgrund der Struktur der Wörter aus L_n erhalten wir $i = j$. Daher gibt es für jedes $A \neq S$ genau ein $i \in \{1, 2, \ldots, n-1\}$ derart, dass alle Wörter in $L(A, G)$, deren Länge mindestens $2n$ ist, Teilwörter aus $\{a^ib\}^+$ haben.

Die gleichen Überlegungen gelten auch für S, falls es eine Ableitung $S \Longrightarrow^* t_1St_2$ mit $t_1, t_2 \in V^*$ und $t_1t_2 \neq \lambda$ gibt. Damit gibt es keine Ableitungen dieser Form, da $L(G, S) = L_n$ gilt.

Angenommen, $N \setminus \{S\}$ enthält weniger als $n-1$ Nichtterminale. Dann gibt es ein $j \in \{0, 1, 2, \ldots, n-1\}$ so, dass G nur endlich viele Wörter aus $\{a^jb\}^+$ erzeugen kann. Dies widerspricht $L(G) = L_n$, womit $\mathrm{Var}_{CF}(L_n) \geq n$ gezeigt ist. Mit (3.3) folgt die Behauptung. □

Bezüglich der beliebigen Regelgrammatiken liegt eine grundsätzlich andere Situation als bei regulären und kontextfreien Grammatiken vor. Hier ist die Anzahl der zur Erzeugung einer Sprache notwendigen Nichtterminale generell beschränkt.

Satz 3.10 *Für jede Sprache $L \in \mathcal{L}(RE)$ gilt* $\mathrm{Var}_{RE}(L) \leq 4$.

Beweis. Zu jeder Sprache $L \in \mathcal{L}(RE)$ gibt es eine Grammatik $G = (N, T, P, S)$, die $L(G) = L$ erfüllt und nur Regeln der Form $\alpha \to \beta$ und $A \to a$ mit $\alpha, \beta \in N^*$, $A \in N$, $a \in T$ hat.[1] Wir betrachten die Grammatik

$$G' = (\{S', \$, [,]\}, N \cup T, P_1 \cup P_2 \cup P_3 \cup P_4, S')$$

mit

$$\begin{aligned}
P_1 &= \{S' \to [\#S],\ [\#] \to \lambda\}, \\
P_2 &= \bigcup_{A \in N} \{\#A \to A\#,\ A\# \to \#A\}, \\
P_3 &= \{\, \#\alpha \to \#\beta \mid \alpha \to \beta \in P \,\}, \\
P_4 &= \{\, [A \to a[\mid A \to a \in P \,\}.
\end{aligned}$$

Die Nichtterminale [und] fungieren im wesentlichen als Markierungen des linken bzw. rechten Endes der Satzform. Jede Ableitung in G' beginnt mit einer Anwendung der Regel $S' \to [\#S]$, da dies die einzige Regel für das Axiom ist, und endet mit einer Anwendung der Regel $[\#] \to \lambda$, da dies die einzige Regel ist, bei der auf der rechten Seite kein Nichtterminal von G' steht. Mittels der Regeln aus P_2 kann das Symbol # nach links und rechts verschoben werden. Die Regeln aus P_3 erlauben die Simulation der Ableitungen in G. Durch die Regeln aus P_4 können wir den linken Endmarker nach rechts verschieben, wobei jede Verschiebung aber mit einer gleichzeitigen Simulation einer terminierenden Regel aus P verbunden ist. Aus diesen Erläuterungen folgt, dass jede nichtterminale Satzform von G' die Form $u[v_1\#v_2]$ hat, wobei $u \in T^*$ gilt und uv_1v_2 eine Satzform von G ist. Der letzte Schritt einer Ableitung in G' hat daher die Form $u[\#] \Longrightarrow u$ mit $u \in L(G)$. Daher erhalten wir $L(G') = L(G) = L$. Wegen $\mathrm{Var}(G') = 4$ folgt die Behauptung. □

Für monotone Grammatiken können wir keine generelle Beschränkung der Anzahl der Nichtterminale wie in Satz 3.10 angeben, aber wir können eine Schranke angeben, die nur von der Größe des Alphabets abhängt, über dem die Sprache definiert ist.

Satz 3.11 *Für jede Sprache $L \in \mathcal{L}(MON)$ mit $L \subseteq T^*$ gilt* $\mathrm{Var}_{MON}(L) \leq 3\#(T) + 1$.

Beweis. Es sei L eine Sprache über dem Alphabet T. Dann gibt es zu je drei Buchstaben a, b und c eine Menge L_{abc} derart, dass die Sprache $L_{abc}\{abc\}$ alle Wörter mit einer Länge von mindestens vier enthält, die auf abc enden. Damit lässt sich L wie folgt zerlegen

$$L = \{\, w \mid w \in L,\ |w| \leq 3 \,\} \ \cup \bigcup_{(a,b,c) \in T^3} L_{abc}\{abc\}.$$

Wir benutzen ohne Beweis den folgenden Satz: *Ist L eine monotone Sprache, so gibt es zu jedem Wort $abc \in T^3$ eine monotone Grammatik G_{abc} mit $L(G_{abc}) = L_{abc}$.* Es sei jeweils $G_{abc} = (N_{abc}, T, P_{abc}, S_{abc})$. Ohne Beschränkung der Allgemeinheit können wir annehmen, dass

- die Mengen N_{abc} der Nichttermiale dieser Grammatiken paarweise disjunkt sind,

[1] Sollte eine Grammatik nicht in dieser Form vorliegen, ersetze man zuerst jedes Vorkommen eines Terminals a in Regeln durch ein neues Nichtterminal a' und füge dann die Regeln $a' \to a$ hinzu.

- alle Grammatiken nur Regeln der Form $\alpha \to \beta$ und $A \to a$ mit $\alpha, \beta \in N_{abc}^*$, $A \in N_{abc}$ und $a \in T$ haben.

Wir setzen dann

$$\begin{aligned}
N' &= \bigcup_{(a,b,c)\in T^3} N_{abc},\\
P' &= \bigcup_{(a,b,c)\in T^3} P_{abc},\\
G' &= (\{S'\} \cup \bigcup_{a\in T} \{[^a, \#^a,]^a\}, N' \cup T, P_1 \cup P_2 \cup P_3 \cup P_4, S'),\\
P_1 &= \{\, S' \to w \mid w \in L(G),\ |w| \le 3 \,\} \ \cup \bigcup_{(a,b,c)\in T^3} \{S' \to [^a\#^b S_{abc}]^c,\ [^a\#^b]^c \to abc\},\\
P_2 &= \bigcup_{A\in N} \bigcup_{a\in T} \{\#^a A \to A\#^a,\ A\#^a \to \#A\},\\
P_3 &= \{\, \#^a\alpha \to \#^a\beta \mid \alpha \to \beta \in P',\ a \in T \,\},\\
P_4 &= \{\, [^b A \to a[^b \mid A \to a \in P',\ b \in T \,\}.
\end{aligned}$$

Wegen der Disjunktheit der Mengen der Nichtterminale und Regeln erhält man in Analogie zum Beweis von Satz 3.10

$$L(G') = \{\, w \mid w \in L,\ |w| \le 3 \,\} \ \cup \bigcup_{(a,b,c)\in T^3} L_{abc}\{abc\} = L.$$

Wegen $\mathrm{Var}(G') = 3\#(T) + 1$ folgt die Behauptung. □

Folgerung 3.12 *Für $X \in \{MON, RE\}$ gilt $X \le^3_{\mathrm{Var}} CF$.*

Beweis. Wir betrachten die Sprachen $L_n \subseteq \{a,b\}^*$, $n \ge 1$, aus dem Beweis von Satz 3.9. Nach diesem Beweis gilt $\mathrm{Var}_{CF}(L_n) = n$. Aus den Sätzen 3.10 und 3.11 erhalten wir $\mathrm{Var}_{RE}(L_n) \le 4$ und $\mathrm{Var}_{MON}(L_n) \le 7$. □

Lemma 3.13 *Es sei w ein Wort der Länge n über T. Dann gilt $\mathrm{Var}_{nfREG}(\{w\}) = n$.*

Beweis. Es sei $w = a_1a_2\ldots a_n$ mit $a_i \in T$, $1 \le i \le n$. Dann erzeugt die Grammatik

$$G = (\{A_i \mid 1 \le i \le n\}, T, \{A_i \to a_iA_{i+1} \mid 1 \le i \le n-1\} \cup \{A_n \to a_n\}, A_1)$$

in regulärer Normalform offenbar die Sprache $\{a_1a_2\ldots a_n\} = \{w\}$. Wegen $\mathrm{Var}(G) = n$ ist die Beziehung $\mathrm{Var}_{nfREG}(\{w\}) \le n$ bereits gezeigt.

Es sei nun $H = (N, T, P, S)$ eine Grammatik in regulärer Normalform, die die Sprache $L(H) = \{w\}$ erzeugt. Jede Ableitung von w hat offensichtlich die Form

$$\begin{aligned}
S = B_1 &\Longrightarrow a_1B_2 \Longrightarrow a_1a_2B_3 \Longrightarrow \cdots \Longrightarrow a_1a_2\ldots a_{i-1}B_i \Longrightarrow \cdots \Longrightarrow a_1a_2\ldots a_{n-1}B_n\\
&\Longrightarrow a_1a_2\ldots a_{n-1}a_n = w. \qquad (3.5)
\end{aligned}$$

Angenommen, es wären zwei der Nichtterminale in solch einer Ableitung gleich, also $B_i = B_j$ für gewisse i und j mit $1 \leq i < j \leq n$. Dann gibt es eine Ableitung

$$B_i \Longrightarrow a_i B_{i+1} \Longrightarrow \cdots \Longrightarrow a_i a_{i+1} \ldots a_{j-1} B_j,$$

und folglich die Ableitung

$$B_i \Longrightarrow^* a_i a_{i+1} \ldots a_{j-1} B_i.$$

Außerdem folgen aus (3.5) die Ableitungen

$$S \Longrightarrow^* a_1 a_2 \ldots a_{i-1} B_i \quad \text{und} \quad B_i = B_j \Longrightarrow^* a_j a_{j+1} \ldots a_n.$$

Damit gibt es in H die Ableitung

$$\begin{aligned} S &\Longrightarrow^* a_1 a_2 \ldots a_{i-1} B_i \\ &\Longrightarrow^* a_1 a_2 \ldots a_{i-1} a_i \ldots a_{j-1} B_i \\ &\Longrightarrow^* a_1 a_2 \ldots a_{i-1} a_i \ldots a_{j-1} a_i \ldots a_{j-1} B_i \\ &\Longrightarrow^* a_1 a_2 \ldots a_{i-1} a_i \ldots a_{j-1} a_i \ldots a_{j-1} a_j \ldots a_n \end{aligned}$$

eines Wortes der Länge $i-1+(j-i)+(j-i)+(n-(j-1)) = n+(j-i) > n$. Dies widerspricht $L(H) = \{w\}$. Folglich sind alle Nichtterminale in der Ableitung (3.5) paarweise verschieden. Daher gilt $\mathrm{Var}(H) \geq n$ für jede Grammatik H in regulärer Normalform mit $L(H) = \{w\}$, womit auch $\mathrm{Var}_{nfREG}(\{w\}) \geq n$ gezeigt ist. □

Folgerung 3.14 *REG* $\leq^3_{\mathrm{Var}}$ *nfREG*.

Beweis. Es seien n eine natürliche Zahl und w ein Wort der Länge n über T. Die reguläre Grammatik $G' = (\{S\}, T, \{S \to w\}, S)$ erzeugt $\{w\}$, und damit gilt $\mathrm{Var}_{REG}(\{w\}) = 1$ (da jede Grammatik mindestens ein Nichtterminal hat). Nun folgt die Behauptung aus Lemma 3.13. □

Folgerung 3.14 besagt, dass wir den Übergang zur regulären Normalform mit einer drastischen Erhöhung der Anzahl der Nichtterminale erkaufen. Die Aussagen des obigen Lemmas 3.5 zeigen dagegen, dass der Übergang zur reduzierten Grammatik bzw. zur λ-freien Grammatik kein Erhöhung der Anzahl der Nichtterminale mit sich führt. Ohne Beweis geben wir nun das entsprechende Resultat für den Fall der Chomsky-Normalform an, bei der ebenfalls eine beliebig große Erhöhung eintreten kann.

Lemma 3.15 *CF* $\leq^3_{\mathrm{Var}}$ *ChCF*. □

Zusammenfassend erhalten wir folgende Beziehung zwischen den Klassen bezüglich des Maßes Var:

$$MON \leq^3_{\mathrm{Var}} CF \approx_{\mathrm{Var}} redCF \approx_{\mathrm{Var}} CF\text{-}\lambda \leq^3_{\mathrm{Var}} REG \leq^3_{\mathrm{Var}} nfREG$$

und

$$MON \leq^3_{\mathrm{Var}} CF \approx_{\mathrm{Var}} redCF \approx_{\mathrm{Var}} CF\text{-}\lambda \leq^3_{\mathrm{Var}} ChCF.$$

Uns sind zur Zeit außer dem trivialen Fakt $\mathrm{Var}_{RE}(L) \leq \mathrm{Var}_{MON}(L)$ (siehe Lemma 3.4) keine Aussagen über das Verhältnis zwischen *RE* und *MON* bzw. zwischen *ChCF* und *REG* hinsichtlich der Anzahl der Nichtterminale bekannt.

Bisher haben wir stets ausgehend von einer Sprache die zugehörige minimale Anzahl von Nichtterminalen für die Erzeugung der Sprache bestimmt. Man kann das Problem auch umgekehrt betrachten. Gegeben sei eine Zahl n, und wir versuchen die Menge der Sprachen zu bestimmen, die mit höchstens n Nichtterminalen erzeugt werden können. Wir geben zwei Resultate dieser Art an.

Lemma 3.16 *Für eine reguläre Sprache L gilt genau dann $\mathrm{Var}_{REG}(L) = 1$, wenn es zwei endliche Mengen R_1 und R_2 von Wörtern so gibt, dass $L = R_1^* R_2$ gilt.*

Beweis. Es sei $G = (N, T, P, S)$ eine reguläre Grammatik mit $\mathrm{Var}(G) = 1$, d. h. $N = \{S\}$. Dann gibt es in P nur Regeln der Form $S \to wS$ und $S \to w$ mit $w \in T^*$. Wir setzen

$$R_1 = \{w \mid w \in T^*,\ S \to wS \in P\},$$
$$R_2 = \{w \mid w \in T^*,\ S \to w \in P\}.$$

Jede Ableitung in G hat die Form

$$S \Longrightarrow w_1 S \Longrightarrow w_1 w_2 S \Longrightarrow \cdots \Longrightarrow w_1 w_2 \ldots w_t S \Longrightarrow w_1 w_2 \ldots w_t v$$

mit $t \geq 0$, $w_1, w_2, \ldots, w_t \in R_1$ und $v \in R_2$. Folglich gilt $L(G) \subseteq R_1^* R_2$. Umgekehrt ist leicht zu sehen, dass für jedes Wort $w_1' w_2' \ldots w_s' v' \in R_1^* R_2$ eine Ableitung in G existiert, sodass sogar $L(G) = R_1^* R_2$ gilt.

Ist umgekehrt $L \subset T^*$ und $L = R_1^* R_2$ für gewisse endliche Sprachen R_1 und R_2, so wird L offenbar von der (regulären) Grammatik

$$G' = (\{S\}, T, \{S \to wS \mid w \in R_1\} \cup \{S \to w \mid w \in R_2\}, S)$$

erzeugt, womit $\mathrm{Var}_{REG}(L) = 1$ gezeigt ist. □

Lemma 3.17 *Für eine Sprache L gilt genau dann $\mathrm{Var}_{REG}(L) \leq 2$, wenn es endliche Sprachen $R_1, R_2, R_3, R_4, R_5, R_6$ so gibt, dass*

$$L = (R_1^* R_2 R_4^* R_5)^* (R_1^* R_3 \cup R_1^* R_2 R_4^* R_6)$$

gilt.

Beweis. Es sei $G = (\{S, A\}, T, P, S)$. Mit den Mengen

$$R_1 = \{w \mid S \to wS \in P\},$$
$$R_2 = \{w \mid S \to wA \in P\},$$
$$R_3 = \{w \mid S \to w \in P\},$$
$$R_4 = \{w \mid A \to wA \in P\},$$
$$R_5 = \{w \mid A \to wS \in P\},$$
$$R_6 = \{w \mid A \to w \in P\}.$$

kann analog zum vorhergehenden Beweis gezeigt werden, dass $L(G)$ die gewünschte Form hat. Auch der Beweis für $\mathrm{Var}_{REG}(L) \leq 2$ für Sprachen der angegebenen Form ist analog leicht zu führen. □

Man beachte, dass bei Grammatiken mit einem Nichtterminal die Mengen R_2, R_4, R_5 und R_6 leer sind. Dadurch geht eine Sprache mit der Darstellung aus Lemma 3.17 dann in eine Sprache mit der Darstellung $R_1^* R_3$ über, die der aus Lemma 3.16 entspricht.

Lemma 3.18 *Eine Sprache L über einem einelementigen Alphabet $\{a\}$ ist genau dann regulär, wenn es endliche Mengen $U_1 \subseteq \{a\}^*$ und $U_2 \subseteq \{a\}^*$ sowie eine natürliche Zahl $p > 0$ mit $L = U_1 \cup U_2\{a^p\}^*$ gibt.*

Beweis. Ist L endlich, so erhalten wir mit $U_1 = L$ und $U_2 = \emptyset$ die gewünschte Form.

Es sei nun L eine unendliche reguläre Sprache über einem Alphabet $\{a\}$. Dann gibt es einen deterministischen endlichen Automaten $\mathcal{A} = (\{a\}, Z, z_0, \delta, F)$ mit dem Eingabesymbol a, einer Menge $Z = \{z_0, z_1, \ldots, z_n\}$ von Zuständen, dem Anfangszustand z_0, einer Menge $F \subseteq Z$ von akzeptierenden Zuständen und einer Überführungsfunktion $\delta : Z \times X \to Z$, der L akzeptiert.

Die Überführungsfunktion δ ist durch den Graphen in Abbildung 3.1 charakterisiert.

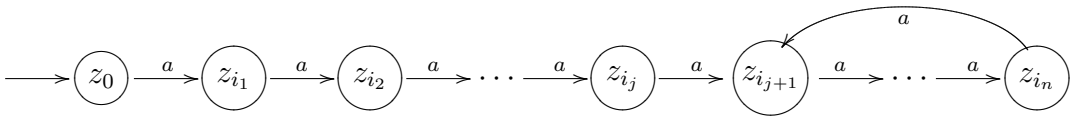

Abbildung 3.1: Transitionsgraph von $\mathcal{A}$

Für die akzeptierte Sprache gilt $L = L_1 \cup L_2$ mit

$$L_1 = \{\, a^k \mid 0 \leq k \leq j \text{ und } z_{i_k} \in F \,\} \text{ und}$$
$$L_2 = \{\, a^k \mid j+1 \leq k \leq n \text{ und } z_{i_k} \in F \,\}\{a^{n-j}\}^*.$$

Mit $U_1 = L_1$, $U_2 = \{\, a^k \mid j+1 \leq k \leq n \text{ und } z_{i_k} \in F \,\}$ und $p = n - j$ erhalten wir die gewünschte Form. □

Folgerung 3.19 *Für jede reguläre Sprache L über einem Alphabet, das aus genau einem Buchstaben besteht, gilt* $\mathrm{Var}_{REG}(L) \leq 2$.

Beweis. Nach Lemma 3.18 gibt es zwei endliche Sprachen U_1 und U_2 sowie eine natürliche Zahl p so, dass $L = U_1 \cup U_2\{a^p\}^*$ gilt. Mit $R_1 = \emptyset$, $R_2 = U_2$, $R_3 = U_1$, $R_4 = \{a^p\}$, $R_5 = \emptyset$ und $R_6 = \{\lambda\}$ geht die Darstellung von L durch U_1, U_2 und a^p in die Form aus Lemma 3.17 über, womit die Aussage bewiesen ist. □

Mit jedem Komplexitätsmaß $K \in \{\mathrm{Var}, \mathrm{Prod}, \mathrm{Symb}\}$ ist auch das Problem der algorithmischen Berechnung von $K(G)$ bzw. $K_X(L(G))$ (für eine Menge X von Grammatiken) verbunden. Die folgenden Probleme sind in diesem Zusammenhang von Interesse:

- Bestimme für eine Grammatik G den Wert $K(G)$.

- Bestimme für eine Grammatik G aus der Menge X den Wert $K_X(L(G))$.
- Entscheide für eine Grammatik G aus X und eine natürliche Zahl $n \geq 1$, ob $K_X(L(G)) = n$ gilt.
- Entscheide für eine Grammatik G aus X, ob $K(G) = K_X(L(G))$ gilt, d. h. entscheide ob G eine bez. K minimale Grammatik aus X ist.
- Bestimme für eine Grammatik G aus X eine bez. K minimale Grammatik G' aus X für $L(G)$, d. h. eine Grammatik $G' \in X$ mit $L(G') = L(G)$ und $K(G') = K_X(L(G))$.

Das erste Problem betrifft die Komplexität der Grammatik, während die anderen die Komplexität der erzeugten Sprache betreffen.

Wir wollen nun den Status der Entscheidbarkeit oben genannter Probleme bez. des Maßes Var untersuchen.

Satz 3.20 *Zu jeder der Mengen $X \in \{nfREG, REG, CF, ChCF, CF\text{-}\lambda, MON, RE\}$ und jeder Grammatik $G \in X$ gibt es einen Algorithmus, der* $\mathrm{Var}(G)$ *berechnet.*

Beweis. Wir haben die Anzahl der Nichtterminale der angegebenen Grammatik zu ermitteln. Dies kann durch Zählen während eines Durchlaufs durch die Grammatikbeschreibung geschehen. □

Satz 3.21 *Es gibt keinen Algorithmus, der für eine kontextfreie Grammatik G entscheidet, ob* $\mathrm{Var}_{CF}(L(G)) = 1$ *gilt.*

Beweis. Es seien zwei Mengen $X = \{x_1, x_2, \ldots, x_n\}$ und $Y = \{y_1, y_2, \ldots, y_n\}$ von nichtleeren Wörtern über dem Alphabet $\{a, b\}$ gegeben. Wir betrachten nun folgende Sprachen

$$\begin{aligned}
L_X &= \{ba^{i_1}ba^{i_2}\ldots ba^{i_{k-1}}ba^{i_k}cx_{i_k}x_{i_{k-1}}\ldots x_{i_2}x_{i_1} \mid k \geq 1,\ 1 \leq i_j \leq n \text{ für } 1 \leq j \leq k\},\\
L_Y &= \{ba^{i_1}ba^{i_2}\ldots ba^{i_{k-1}}ba^{i_k}cy_{i_k}y_{i_{k-1}}\ldots y_{i_2}y_{i_1} \mid k \geq 1,\ 1 \leq i_j \leq n \text{ für } 1 \leq j \leq k\},\\
L_{X,Y} &= \{wcv^R \mid w \in L_X,\ v \in L_Y\},\\
L_s &= \{w_1cw_2cw_2^Rcw_1^R \mid w_1, w_2 \in \{a,b\}^*\},\\
L(X,Y) &= \{a,b,c\}^* \setminus (L_{X,Y} \cap L_s).
\end{aligned}$$

Aus den gegebenen Mengen X und Y kann eine kontextfreie Grammatik $G(X,Y)$ mit $L(G(X,Y)) = L(X,Y)$ konstruiert werden. [2]

Das Postsche Korrespondenzproblem (PKP) bez. X und Y besteht darin, festzustellen, ob es eine Folge $i_1i_2\ldots i_k$ mit $1 \leq i_j \leq n$ für $1 \leq j \leq k$ so gibt, dass

$$x_{i_1}x_{i_2}\ldots x_{i_k} = y_{i_1}y_{i_2}\ldots y_{i_k}$$

[2] Wir verzichten auf einen vollständigen Beweis dieser Aussage und bemerken nur folgende Fakten: Es lassen sich leicht deterministische Kellerautomaten angeben, die $L_{X,Y}$ bzw. L_s akzeptieren. Da die Menge der deterministischen kontextfreien Sprachen unter Komplementbildung abgeschlossen ist, ergibt sich, dass auch $\{a,b,c\}^* \setminus L_{X,Y}$ und $\{a,b,c\}^* \setminus L_s$ kontextfrei sind. Als Vereinigung dieser Mengen ist auch $L(X,Y)$ kontextfrei. Da zu den Operationen entsprechende Konstruktionen der zugehörigen Kellerautomaten bzw. Grammatiken existieren und der Übergang von Kellerautomaten zu Grammatiken und umgekehrt jeweils algorithmisch vollzogen werden können, lässt sich $G(X,Y)$ algorithmisch bestimmen.

gilt. Die Folge $i_1 i_2 \dots i_k$ heißt dann Lösung des PKPs. Es ist bekannt, dass das PKP für beliebige (gleichmächtige) Mengen X und Y von Wörtern über $\{a, b\}$ unentscheidbar ist.

Folgende Aussagen setzen die obigen Sprachen in Relation zum PKP. $L_{X,Y} \cap L_s \neq \emptyset$ ist genau dann gültig, wenn das PKP für X und Y eine Lösung hat. Folglich ist $L(X, Y) = \{a, b, c\}^*$ genau dann gültig, wenn das PKP bez. X und Y keine Lösung hat. Somit ist $L(G(X, Y)) = \{a, b, c\}^*$ genau dann gültig, wenn das PKP bez. X und Y keine Lösung hat. Damit ergibt sich aus der Unentscheidbarkeit des PKPs die Unentscheidbarkeit der Frage, ob $L(G(X, Y))$ die Menge aller Wörter über $\{a, b, c\}$ ist.

Gilt $L(X, Y) = \{a, b, c\}^*$, so wird $L(X, Y)$ offenbar von der Grammatik

$$G(X, Y) = (\{S\}, \{a, b, c\}, \{S \to aS,\ S \to bS,\ S \to cS,\ S \to \lambda\}, S)$$

erzeugt, woraus wir sofort $\mathrm{Var}_{CF}(L(X, Y)) = 1 = \mathrm{Var}_{CF}(L(G(X, Y))$ erhalten.

Wir werden nun zeigen, dass bei $L(X, Y) \neq \{a, b, c\}^*$ die Beziehung

$$\mathrm{Var}_{CF}(L(X, Y)) = \mathrm{Var}_{CF}(L(G(X, Y))) \geq 2$$

gilt. Daher würde eine Entscheidbarkeit der Frage, ob $\mathrm{Var}_{CF}(L(G)) = 1$ ist, die Entscheidbarkeit von $L(G(X, Y)) = \{a, b, c\}^*$ implizieren. Vom letzteren Problem wissen wir aber schon, dass es unentscheidbar ist. Daher muss es unentscheidbar sein, ob $\mathrm{Var}_{CF}(L(G)) = 1$ gilt.

Wir nehmen nun an, dass es eine kontextfreie Grammatik $H = (\{S\}, \{a, b, c\}, P, S)$ mit nur einem Nichtterminal gibt, die $L(X, Y) \neq \{a, b, c\}^*$ erzeugt. Wegen $L_{X,Y} \cap L_s \neq \emptyset$ gibt es eine Lösung $i_1 i_2 \dots i_{k-1} i_k$ des PKPs. Wir setzen

$$I = ba^{i_1} ba^{i_2} \dots ba^{i_k},\ X = x_{i_1} x_{i_2} \dots x_{i_k},\ J = I^R \text{ und } Y = X^R.$$

Dann gilt

$$I^m c X^m c Y^m c J^m \in L_{X,Y} \cap L_s \quad \text{für alle } m \geq 1 \tag{3.6}$$

und folglich $w_m = I^{m-1} c X^m c Y^m c J^m \in L(X, Y)$ für alle $m \geq 1$. Wir betrachten eine Ableitung von $w = w_m$ mit $m \geq \max\{|u| \mid S \to u \in P\}$. Dann gibt es ein α so, dass $S \Longrightarrow \alpha \Longrightarrow^* w$ und $\alpha = \alpha_0 S \alpha_1$ und $w = \alpha_0 w_0 w_1$ für gewisse $\alpha_0, w_0, w_1 \in T^*$ mit $S \Longrightarrow^* w_0$, $\alpha_1 \Longrightarrow^* w_1$ und $\alpha_0 w_1 \neq \lambda$ gelten. Aufgrund der Wahl von m enthält α_0 kein c. Daher gibt es Wörter z_0 und z_1 so, dass $z_0 z_1 = I$ und $\alpha_0 = I^{i_0} z_0$ gelten. Wegen $\alpha_0 w_0 w_1 = w$ erhalten wir

$$\alpha_0 z_1 z_0 w_0 w_1 = I^m c X^m c Y^m c J^m. \tag{3.7}$$

Weiterhin gilt $z_1 z_0 w_0 \in L(X, Y)$. Dies ist klar, falls $z_1 z_0 w_0$ keine drei Vorkommen des Buchstaben c hat. Kommen aber drei c vor, so muss

$$z_1 z_0 w_0 = z_1 z_0 u_1 c u_2 c u_3 c u_4 \tag{3.8}$$

sein. Aus (3.7) folgt dann $\alpha_0 z_1 z_0 u_1 c u_2 c u_3 c u_4 w_1 = I^m c X^m c Y^m c J^m$ und damit

$$\alpha_0 z_1 z_0 u_1 = I^m,\ u_2 = X^m,\ u_3 = Y^m \text{ und } u_4 w_1 = J^m. \tag{3.9}$$

Nehmen wir nun an, dass $z_1z_0w_0 \notin L(X,Y)$ gilt. Dann muss $z_1z_0w_0 \in L_{X,Y} \cap L_s$ sein, woraus mit (3.8) und (3.9) $z_1z_0u_1cX^mcY^mcu_4 \in L_{X,Y} \cap L_s$ folgt. Beachten wir die Struktur der Wörter in $L_{X,Y} \cap L_s$, so erhalten wir $z_1z_0u_1 = I^m$ und $u_4 = J^m$. Damit ergibt sich aus (3.9) $\alpha_0 = \lambda$ und $w_1 = \lambda$. Dies ist ein Widerspruch zu $\alpha_0 w_1 \neq \lambda$.

Deshalb gibt es eine Ableitung $S \Longrightarrow^* z_1z_0w_0$. Hieraus resultiert die Ableitung

$$S \Longrightarrow \alpha_0 S \alpha_1 \Longrightarrow^* \alpha_0 z_1 z_0 w_0 \alpha_1 \Longrightarrow^* \alpha_0 z_1 z_0 w_0 w_1.$$

Mittels (3.9) ergibt dies $I^mcX^mcY^mcJ^m \in L(H) = L(X,Y)$ im Widerspruch zu (3.6). □

Folgerung 3.22 *Für eine gegebene kontextfreie Grammatik G und eine natürliche Zahl $n \geq 1$ ist es unentscheidbar, ob* $\mathrm{Var}_{CF}(L(G)) = n$ *gilt.*

Beweis. Für $n = 1$ stimmt die Behauptung mit der Aussage von Satz 3.21 überein und ist damit bewiesen.

Es sei nun $n \geq 2$. Wir betrachten anstelle von $G(X,Y)$ eine Grammatik $G_n(X,Y)$ mit

$$L(G_n(X,Y)) = L(X,Y) \cup \{de^2\}^* \cup \{de^3\}^* \cup \cdots \cup \{de^n\}^*.$$

Unter Verwendung von Satz 3.9 und den Betrachtungen beim Beweis von Satz 3.21 zeigt man sehr leicht, dass $\mathrm{Var}_{CF}(L(G_n(X,Y)) = n$ genau dann gilt, wenn $L(X,Y) = \{a,b,c\}^*$ ist, also wenn das PKP bez. X und Y keine Lösung hat. □

Folgerung 3.23 *Es gibt keinen Algorithmus, der zu einer kontextfreien Grammatik G den Wert* $\mathrm{Var}_{CF}(L(G))$ *berechnet.*

Beweis. Angenommen, es gäbe einen solchen Algorithmus. Dann kann $\mathrm{Var}_{CF}(L(G))$ algorithmisch ermittelt werden und durch Vergleich mit einer gegebenen natürlichen Zahl n kann $\mathrm{Var}_{CF}(L(G)) = n$ algorithmisch entschieden werden.

Dies widerspricht Folgerung 3.22. □

Folgerung 3.24 *Es gibt keinen Algorithmus, der zu einer kontextfreien Grammatik G eine bez.* Var *minimale kontextfreie Grammatik G' für $L(G)$, d. h. G' erfüllt $L(G') = L(G)$ und* $\mathrm{Var}(G') = \mathrm{Var}_{CF}(L(G))$, *liefert.*

Beweis. Angenommen es gäbe einen solchen Algorithmus. Dann bestimmen wir zu G zuerst die minimale Grammatik G' und dann von dieser $\mathrm{Var}(G')$. Dies liefert wegen $\mathrm{Var}(G') = \mathrm{Var}_{CF}(L(G))$ einen Algorithmus zur Berechnung von $\mathrm{Var}_{CF}(L(G))$, der nach Folgerung 3.23 nicht existiert. □

Satz 3.25 *Es ist unentscheidbar, ob für eine gegebene kontextfreie Grammatik G die Beziehung* $\mathrm{Var}(G) = \mathrm{Var}_{CF}(L(G))$ *gilt.*

Beweis. Wir betrachten erneut das PKP bezüglich der Mengen $X = \{x_1, x_2, \ldots, x_n\}$ und $Y = \{y_1, y_2, \ldots, y_n\}$ über dem Alphabet $\{a, b\}$ und dazu die Grammatik

$$H(X,Y) = (\{S, S', T, T', R\}, \{a, b, c, d\}, P, S)$$

mit

$$\begin{aligned} P = \{&S \to aSa,\ S \to bSb,\ S \to bT'RT'a,\ S \to aT'RT'b,\\ &S \to aTRa,\ S \to bTRb,\ S \to aRTa,\ S \to bRTb,\\ &T \to Ta,\ T \to Tb,\ T \to a,\ T \to b,\\ &T' \to T,\ T' \to \lambda,\\ &R \to cSc,\ R \to cS'c,\ R \to dTd\}\\ &\cup \{S' \to x_i S' y_i^R \mid 1 \le i \le n\} \cup \{S' \to x_i dTdy_i^R \mid 1 \le i \le n\}. \end{aligned}$$

Es ist leicht zu sehen, dass

$$\begin{aligned} L = L(H(X,Y)) = \{&u_1cu_2c\ldots cu_{k-1}cu_kdw_0dv_kcv_{k-1}c\ldots cv_2cv_1 \mid\\ &w_0 \in \{a,b\}^+,\ k \ge 1,\ u_i, v_i \in \{a,b\}^+ \text{ für } 1 \le i \le k,\\ &u_j \ne v_j^R \text{ für } 1 \le j < k \text{ und entweder } u_k \ne v_k^R \text{ oder}\\ &u_k = x_{i_1}x_{i_2}\ldots x_{i_l} = y_{i_1}y_{i_2}\ldots y_{i_l} = v_k^R \text{ für gewisse } i_1, i_2, \ldots, i_l\}. \end{aligned}$$

Dabei tritt der Fall $u_k = v_k^R$ offenbar nur ein, wenn es eine Lösung des PKPs bez. X und Y gibt ($i_1i_2\ldots i_l$ ist gerade eine Lösung).

Wir stellen nun erst einmal fest, dass die Nichtterminale T' und R eigentlich nicht benötigt werden, wie im Beweis von Lemma 3.8 gezeigt wurde. Entsprechend der dortigen Konstruktion erhalten wir eine Grammatik $H'(X,Y)$ mit 3 Nichtterminalen und $L(H'(X,Y)) = L$. Wir zeigen nun

$$\mathrm{Var}_{CF}(L) = \begin{cases} 2, & \text{wenn das PKP bez. } X \text{ und } Y \text{ keine Lösung hat,}\\ 3, & \text{wenn das PKP bez. } X \text{ und } Y \text{ eine Lösung hat.} \end{cases} \tag{3.10}$$

Habe das PKP bez. X und Y zuerst keine Lösung. Dann können wir auf das Nichtterminal S' und alle zugehörigen Regeln verzichten, da man aus S' wegen der Nichtexistenz einer Lösung nur Wörter der Form $u_kdw_0dv_k$ mit $u_k \ne v_k^R$ erzeugen kann, die auch aus S erzeugt werden können. Damit gilt $\mathrm{Var}_{CF}(L) \le 2$.

Angenommen, es wäre $\mathrm{Var}_{CF}(L) = 1$. Dann gibt es eine kontextfreie Grammatik $H' = (\{A\}, \{a, b, c, d\}, P', A)$ mit $L(H') = L$. Wegen $w_m = ada^mda^m \in L$ für alle $m \ge 2$ gibt es eine Ableitung $A \Longrightarrow \alpha \Longrightarrow^* ada^mda^m$ von w_m in H'. Es sei m hinreichend groß. Ohne Beschränkung der Allgemeinheit können wir annehmen, dass $\alpha \ne A$ gilt. Enthält α zweimal den Buchstaben A, so ist aus α und damit aus A ein Wort mit mindestens vier Vorkommen von d ableitbar. Ein solches Wort liegt aber nicht in L. Folglich erhalten wir $\alpha = \alpha_1 A \alpha_2$ für gewisse $\alpha_1 \in T^*$ und $\alpha_2 \in T^*$ (da $\alpha \ne w_m$ wegen der Wahl von m sein muss). Wegen der hinreichend großen Wahl von m muss auch $\alpha_2 = a^j$ mit $0 \le j < m$ gelten.

Ist $\alpha_1 = \lambda$, so ist $j \ge 1$. Damit erhalten wir wegen $a^{j+1}dada \in L$ die Ableitung $A \Longrightarrow Aa^j \Longrightarrow^* a^{j+1}dadaa^j \notin L$, womit ein Widerspruch erreicht wurde.

Ist $\alpha_1 = a$, so muss aus A ein Wort abgeleitet werden, dass mit d beginnt. Dies widerspricht erneut der Form der Wörter in L.

Gilt $\alpha_1 = ada^r$, so erhalten wir die Ableitung

$$A \Longrightarrow ada^r Aa^j \Longrightarrow^* ada^r ada^m da^m a^j \notin L.$$

Weitere Möglichkeiten gibt es aufgrund der Wahl von m für α_1 nicht, womit gezeigt ist, dass zur Erzeugung von L mindestens 2 Nichtterminale benötigt werden.

Analog zeigt man, dass drei Nichtterminale erforderlich sind, wenn das PKP eine Lösung hat (der Beweis ist aber erheblich umfangreicher, siehe [7]).

Wegen (3.10) erhalten wir, dass $G(X, Y)$ nach Ersetzen von T' und R' genau dann minimal ist, wenn das PKP bez. X und Y eine Lösung hat. Da letzteres unentscheidbar ist, ist also auch unentscheidbar, ob G minimal ist. □

Hinsichtlich regulärer Grammatiken in Normalform stellt sich die Situation völlig anders dar. Alle oben erwähnten Entscheidungsprobleme werden entscheidbar. Wir beweisen zuerst, dass es einen Algorithmus gibt, der zu einer Grammatik G in Normalform eine für $L(G)$ minimale Grammatik in regulärer Normalform bestimmt.

Satz 3.26 *Es gibt einen Algorithmus, der zu einer gegebenen regulären Grammatik G in Normalform eine reguläre Grammatik G' in Normalform bestimmt, für die $L(G') = L(G)$ und $\mathrm{Var}(G') = \mathrm{Var}_{nfREG}(L(G))$ gelten.*

Beweis. Es sei $G = (N, T, P, S)$ gegeben. Wir bestimmen zuerst $n = \mathrm{Var}(G)$. Nach Satz 3.20 ist dies algorithmisch möglich. Wir geben nun eine Konstruktion aller Grammatiken mit k Nichtterminalen, $k < n$, und dem Terminalalphabet. Dazu setzen wir

$$N_k = \{A_1, A_2, \ldots, A_k\}.$$

Weiterhin wählen wir für jede natürliche Zahl i mit $1 \leq i \leq k$ eine Teilmenge $M(k, i)$ von T und für jedes Paar (i, j) mit $1 \leq i, j \leq k$ eine Teilmenge $M(k, i, j)$ von T und setzen

$$\begin{aligned} P(k, i) &= \{A_i \to x \mid x \in M(k, i)\}, \\ P(k, i, j) &= \{A_i \to xA_j \mid x \in M(k, i, j)\}. \end{aligned}$$

Wir betrachten die Grammatik

$$H = \left(N_k, T, \bigcup_{1 \leq i \leq k} P(k, i) \cup \bigcup_{1 \leq i, j \leq k} P(k, i, j), A_1\right).$$

Es können alle Grammatiken mit k Nichtterminalen in dieser Weise erhalten werden. Beginnend mit $k = 1$ testen wir alle möglichen Grammatiken H, ob $L(H) = L(G)$ gilt. Trifft dies auf eine Grammatik H zu, so ist $\mathrm{Var}_{nfREG}(L(G)) = \mathrm{Var}(H)$. Gilt dagegen $L(H) \neq L(G)$ für alle H, so setzen wir mit $k = 2$ fort. Haben wir auch für $k = \mathrm{Var}(G) - 1$ noch keine zu G äquivalente Grammatik, so ist G selbst eine minimale Grammatik für die Sprache $L(G)$. □

Aus Satz 3.26 erhalten wir durch einfache Schlüsse, dass auch die anderen Entscheidbarkeitsprobleme für reguläre Grammatiken in Normalform entscheidbar sind.

Folgerung 3.27

1. *Es gibt einen Algorithmus, der zu gegebener regulärer Grammatik G in Normalform* $\mathrm{Var}_{nfREG}(L(G))$ *bestimmt.*

2. *Es ist entscheidbar, ob eine gegebene Grammatik G in regulärer Normalform minimal für $L(G)$ ist, d. h. ob* $\mathrm{Var}(G) = \mathrm{Var}_{nfREG}(L(G))$ *gilt.* □

Es ist den Autoren nicht bekannt, welchen Entscheidbarkeitsstatus die betrachteten Probleme bei beliebigen regulären Grammatiken haben.

3.3. Anzahl der Regeln

Wir beginnen mit Lemmata, die denen vom Beginn von Abschnitt 2.2 entsprechen.

Lemma 3.28 *Für je zwei Mengen X und Y von Grammatiken mit $X \subseteq Y$ und jede Sprache $L \in \mathcal{L}(X)$ gilt* $\mathrm{Prod}_Y(L) \leq \mathrm{Prod}_X(L)$.

Beweis. Der Beweis wird analog zu dem von Lemma 3.4 geführt. □

Lemma 3.29

i) *Es sei $X \in \{nfREG, REG, ChCF, CF\text{-}\lambda, CF\}$. Zu jeder Sprache $L \in \mathcal{L}(X)$ gibt es eine reduzierte Grammatik $G = (N, T, P, S)$, die die Sprache L erzeugt und für die* $\mathrm{Prod}_X(L) = \mathrm{Prod}(G)$ *gilt.*

ii) *Es sei $X \in \{REG, CF\text{-}\lambda, CF\}$. Ist $G \in X$ eine reduzierte Grammatik mit den Eigenschaften* $\mathrm{Prod}(G) = \mathrm{Prod}_X(L(G))$ *und $\#(L(G)) \geq 2$, so gilt für jedes Nichtterminal A von G, dass auch $L(G, A)$ mindestens zwei Wörter enthält.*

Beweis. i) Der Beweis erfolgt analog zu dem von Lemma 3.5 i).

ii) Angenommen, es gibt ein Nichtterminal A mit $\#(L(G, A)) \leq 1$. Wegen der vorausgesetzten Reduziertheit gilt $L(G, A) \neq \emptyset$. Somit enthält $L(G, A)$ genau ein Wort w. Wegen der Reduziertheit gibt es auch eine Regel mit linker Seite A. Ersetzen wir A in allen rechten Seiten durch w und Streichen alle Regeln mit linker Seite A, so entsteht eine Grammatik G' mit weniger Regeln und $L(G') = L(G)$, was nach Voraussetzung nicht möglich ist. □

Wir bestimmen nun für einige Sprachen die Anzahl der notwendigen Regeln bei Verwendung gewisser Typen von Grammatiken.

Lemma 3.30 *Für $n \geq 1$ sei*

$$L_n = \{a^{2^i} \mid 1 \leq i \leq n\}.$$

Dann gilt

$$\mathrm{Prod}_{CF}(L_n) = n.$$

Beweis. Offensichtlich erzeugt die Grammatik

$$G_n = (\{S\}, \{a\}, \{S \to a^{2^i} \mid 1 \leq i \leq n\}, S)$$

die Sprache L_n, womit $\mathrm{Prod}_{CF}(L_n) \leq n$ nachgewiesen ist.

Nach Lemma 3.29 können wir voraussetzen, dass es eine reduzierte Grammatik H_n mit $L(H_n) = L_n$ und $\mathrm{Prod}(H_n) = \mathrm{Prod}_{CF}(L_n)$ gibt, bei der jedes Nichtterminal mindestens zwei Wörter erzeugt.

Wir zeigen zuerst, dass es in H_n keine Ableitung $S \Longrightarrow^* xAyBz$ mit Nichtterminalen A und B und terminalen Wörtern x, y und z gibt. Angenommen, dies wäre der Fall. Dann können wir aus A zwei Wörter a^{i_1} und a^{i_2} und aus B zwei Wörter a^{j_1} und a^{j_2} mit $i_1 < i_2$ und $j_1 < j_2$ ableiten. Setzen wir noch $s = |xyz|$, so sind aus S die Wörter a^{s_1}, a^{s_2}, a^{s_3} und a^{s_4} mit

$$\begin{aligned} s_1 &= s + i_1 + j_1, \\ s_2 &= s + i_2 + j_1, \\ s_3 &= s + i_1 + j_2, \\ s_4 &= s + i_2 + j_2 \end{aligned}$$

ableitbar. Offensichtlich gelten weiterhin

$$s_4 - s_3 = s_2 - s_1 > 0 \quad \text{und} \quad s_3 > s_1. \tag{3.11}$$

Beachten wir noch, dass die Wörter $a^{s_i} \in L_n$ liegen, so muss $s_i = 2^{t_i}$ für $1 \leq i \leq 4$ gelten. Somit ergibt sich aus (3.11)

$$2^{t_4} = 2^{t_3} + (2^{t_2} - 2^{t_1}) = 2^{t_1}(2^{t_3 - t_1} + 2^{t_2 - t_1} - 1),$$

womit wir einen Widerspruch erhalten haben, da wegen $t_3 - t_1 \geq 1$ und $t_2 - t_1 \geq 1$ die ungerade Zahl $2^{t_3 - t_1} + 2^{t_2 - t_1} - 1$ ein Teiler von 2^{t_4} ist.

Wir haben damit gezeigt, dass es keine Satzformen gibt, die mindestens zwei Nichtterminale enthalten (da die restlichen Nichtterminale wegen der Reduziertheit zu terminalen Wörtern abgeleitet werden können). Nehmen wir daher nun an, dass es zwei Ableitungen verschiedener Wörter gibt, in denen das gleiche Nichtterminal auftritt. Es seien also

$$S \Longrightarrow^* x_1 A y_1 \quad \text{und} \quad S \Longrightarrow^* x_2 A y_2$$

zwei derartige Ableitungen. Wir setzen $u_1 = |x_1 y_1|$ und $u_2 = |x_2 y_2|$. Weiterhin sei wie oben $\{a^{i_1}, a^{i_2}\} \subseteq L(G, A)$. Dann erhalten wir, dass die Wörter a^{v_i}, $1 \leq i \leq 4$, mit

$$v_1 = u_1 + i_1,\ v_2 = u_1 + i_2,\ v_3 = u_2 + i_1 \text{ und } v_4 = u_2 + i_2$$

in L_n liegen. Ohne Beschränkung der Allgemeinheit können wir $u_1 < u_2$ und $i_1 < i_2$ annehmen und wie oben ein Widerspruch herleiten.

Damit ist gezeigt, dass die Ableitungen der Wörter a^{2^i} und a^{2^j}, $1 \leq i < j \leq n$, abgesehen von S keine gemeinsamen Nichtterminale haben. Erfolgt die Ableitung direkt, d. h. mittels $S \to a^{2^i}$ bzw. über ein Nichtterminal A_i, so wird jeweils mindestens eine Regel benutzt, die für einen anderen Wert j nicht in Frage kommt. Daher gibt es mindestens n Regeln, womit $\mathrm{Prod}_{CF}(L_n) \geq n$ ebenfalls gezeigt ist. □

Lemma 3.31 *Für $n \geq 2$ sei*

$$L_n = \{a^{2^i} \mid 1 \leq i \leq n\} \cup \{a^{j2^{n+1}} \mid j \geq 1\}.$$

Dann gelten

$$\mathrm{Prod}_{MON}(L_n) \leq 12 \quad \textit{und} \quad \mathrm{Prod}_{CF}(L_n) \geq n.$$

Beweis. Die monotone Grammatik

$$G_n = (\{S, \#, A, A', B, C\}, \{a\}, P_n, S)$$

mit

$$\begin{aligned} P_n = \{&S \to aa,\ S \to \#Aa\#,\ S \to C, \\ &\#A \to \#aaA',\ A'a \to aaA',\ A'\# \to Aa\#,\ aA \to Aa, \\ &\#A \to aB,\ Ba \to aB,\ B\# \to aa, \\ &C \to Ca^{2^{n+1}},\ C \to a^{2^{n+1}}\} \end{aligned}$$

erzeugt L_n, wie man leicht nachrechnet. Damit gilt $\mathrm{Prod}_{MON}(L_n) \leq 12$.

Um die Aussage bezüglich kontextfreier Grammatiken zu beweisen, gehen wir wie beim Beweis des vorhergehenden Lemmas vor. Wir haben nur jeweils mehr Fälle zu unterscheiden. So erhalten wir wie beim Beweis des Lemmas 3.30 die Relationen

$$\begin{aligned} s_1 &= s + i_1 + j_1, \\ s_2 &= s + i_2 + j_1, \\ s_3 &= s + i_1 + j_2, \\ s_4 &= s + i_2 + j_2, \end{aligned}$$

nur dass jetzt nicht alle der s_i, $1 \leq i \leq 4$, Potenzen von 2 sein müssen.

Sind alle s_i Potenzen von 2, erhalten wir wie oben einen Widerspruch.

Wir betrachten nun den Fall, dass $s_i \leq 2^n$ für $i \in \{1, 2, 3\}$ und $s_4 \geq 2^{n+1}$ gelten. Damit erhalten wir $s_i = 2^{t_i}$ mit $1 \leq t_i \leq n$ für $i \in \{1, 2, 3\}$ und $s_4 = j2^{n+1}$ für ein $j \geq 1$. Ist $j = 1$ haben wir es wiederum mit vier Zweierpotenzen zu tun und wir erhalten analog einen Widerspruch. Es sei daher $j \geq 2$ Weiter ergibt sich erneut wegen $s_4 - s_3 = s_2 - s_1$die Beziehung

$$2^{n+2} \leq j2^{n+1} = 2^{t_3} + 2^{t_2} - 2^{t_1} \leq 2^{t_3} + 2^{t_2} \leq 2^n + 2^n = 2^{n+1},$$

womit der gewünschte Widerspruch hergestellt ist.

Die Herleitung der Widersprüche in den anderen Fällen bleibt dem Leser überlassen (siehe [17]).

Damit erhalten wir wie im Beweis von Lemma 3.30, dass mindestens n Regeln erforderlich sind. □

Die Aussage des folgenden Lemmas unterscheidet sich von denen der bisherigen Lemmata, weil in ihm nicht nur eine Aussage zur Anzahl der notwendigen Regeln für eine Sprache gemacht wird, sondern eine diesbezügliche Aussage für *alle* endlichen Sprachen getroffen wird.

Lemma 3.32 *Es seien L eine endliche Sprache, die mehr als n Wörter enthält und G eine reguläre Grammatik mit $L(G) = L$. Dann gilt* $\mathrm{Prod}(G) > \log_2(n) + 1$.

Beweis. Wie beweisen die zum Lemma äquivalente Formulierung: *Erzeugt eine reguläre Grammatik G mit höchstens n Regeln eine endliche Sprache L, so enthält L höchstens 2^{n-1} Wörter.*

Wir beweisen diese Aussage mittels vollständiger Induktion über die Anzahl n der Regeln.

Ist $n = 1$, so erzeugt $G = (N, T, P, S)$ genau ein Wort, wenn die Regel die Form $S \to w \in T^*$ hat, oder die leere Menge, wenn die einzige Regel aus P auf der rechten Seite mindestens ein Nichtterminal hat. In beiden Fällen gilt $\#(L(G)) \leq 1 = 2^0$.

Es sei nun $n \geq 2$ und es gelte die Aussage für alle Werte i mit $i < n$. Wir zeigen, dass sie dann auch für n gilt. Es sei also $G = (N, T, P, S)$ eine reguläre Grammatik mit n Regeln.

Wir nehmen zuerst an, dass es eine Ableitung $S \Longrightarrow^* uS \Longrightarrow^* uv$ mit $uv \in T^*$ gibt. Ist $u \neq \lambda$, so können wir $u^n v$ für alle $n \geq 1$ ableiten. Dies impliziert, dass $L(G)$ im Widerspruch zu unserer Voraussetzung unendlich ist. Ist dagegen $u = \lambda$, so gibt es ein Nichtterminal A ($A = S$ ist zugelassen) mit $A \to S$. Um Unendlichkeit der Sprache zu vermeiden, darf es dann auch keine Ableitung $S \Longrightarrow^* u'A$ mit $u' \neq \lambda$ geben. Folglich wird die erzeugte Sprache nicht verändert, wenn wir die Regel $A \to S$ streichen. Die dadurch entstandene Grammatik G' enthält nur $n - 1$ Regeln und erzeugt ebenfalls $L(G)$. Nach Induktionsvoraussetzung gilt dann

$$\#(L(G)) = \#(L(G')) \leq 2^{n-2} < 2^{n-1}.$$

Wir nehmen nun an, dass S in keiner Ableitung – außer als Startsymbol – vorkommt. Es seien $S \to w_1$, $S \to w_2$, $\ldots$, $S \to w_r$ die Regeln von P mit linker Seite S. Dann kommt S in keinem der Wörter w_i, $1 \leq i \leq r$ vor.

Gilt $r = n$, so ist

$$L(G) = \{z \mid z \in \{w_1, w_2, \ldots, w_r\},\ z \in T^*\},$$

da für Nichtterminale in den Wörtern w_i keine Regeln zur Weiterableitung existieren. Damit gilt

$$\#(L(G)) \leq r = n \leq 2^{n-1}$$

für $n \geq 2$.

Es sei nun $r < n$. Für $1 \leq i \leq r$ setzen wir

$$L_i = \{w \mid w_i \Longrightarrow^* w,\ w \in T^*\}.$$

Offenbar gilt

$$L = \bigcup_{i=1}^{r} L_i.$$

Gilt $w_i \in T^*$ und damit $L_i = \{w_i\}$ für ein i, $1 \leq i \leq n$, so erhalten wir $\#(L_i) = 1 \leq 2^{n-r-1}$.

Gilt $w_i = v_i A$ für ein i, $1 \leq i \leq n$, so setzen wir

$$L'_i = \{v \mid A \Longrightarrow^* v \in T^*\}.$$

Nach Konstruktion gilt dann

$$L_i = \{v_i v \mid v \in L'_i\}$$

und damit $\#(L_i) = \#(L'_i)$. Beachten wir noch, dass in den Ableitungen der Wörter aus L'_i aus A das Startsymbol S nicht mehr vorkommt und daher höchstens die $n-r$ Regeln aus $P \setminus \{S \to w_1, S \to r_2, \ldots, S \to w_r\}$ zur Anwendung kommen, so ergibt sich nach Induktionsvoraussetzung $\#(L_i) = \#(L'_i) \leq 2^{n-r-1}$.

Somit ergibt sich

$$\#(L(G)) = \#\left(\bigcup_{i=1}^{r} L_i\right) \leq r \cdot 2^{n-r-1} \leq 2^r \cdot 2^{n-r-1} = 2^{n-1}. \qquad \square$$

Lemma 3.33 *Für $n \geq 1$ sei*

$$L_n = \{a^{2^n}, a^{2^n+1}, a^{2^n+2}, \ldots, a^{2^{n+1}}\}.$$

Dann gelten

$$\mathrm{Prod}_{CF}(L_n) \leq 3 \quad \textit{und} \quad \mathrm{Prod}_{REG}(L_n) \geq n.$$

Beweis. Die kontextfreie Grammatik

$$G = (\{S, A\}, \{a\}, \{S \to A^{2^n},\ A \to a,\ A \to a^2\}, S)$$

mit drei Regeln erzeugt offenbar L_n, womit $\mathrm{Prod}_{CF}(L_n) \leq 3$ gezeigt ist.

Für $n \geq 1$ hat L_n $2^n + 1$ Elemente. Damit gilt nach Lemma 3.32 für jede reguläre Grammatik G, die L_n erzeugt, $\mathrm{Prod}(G) \geq \log_2(2^n + 2) \geq n$, woraus $\mathrm{Prod}_{REG}(L_n) \geq n$ folgt. $\square$

Lemma 3.34 *Für $n \geq 1$ sei $L_n = \{a^n\}$. Dann gelten*

$$\mathrm{Prod}_{REG}(L_n) = 1 \quad \textit{und} \quad \mathrm{Prod}_{nfREG}(L_n) \geq n.$$

Beweis. Die reguläre Grammatik $G = (\{S\}, \{a\}, \{S \to a^n\}, S)$ erzeugt L_n, und somit gilt $\mathrm{Prod}_{REG}(L_n) = 1$.

Andererseits folgt aus Lemma 3.13, dass zur Erzeugung von L_n durch reguläre Grammatiken in Normalform mindestens n Nichtterminale notwendig sind. Da für jedes dieser Nichtterminale auch eine Regel erforderlich ist, werden mindestens n Regeln zur Erzeugung von L_n durch reguläre Grammatiken in Normalform benötigt. $\square$

Aus den obigen Abschätzungen der notwendigen Anzahl von Regeln für gewisse Sprachen ergeben sich die folgenden Sätze.

Satz 3.35 Prod *ist bez. CF und REG vollständig.*

Beweis. Für *CF* folgt die Aussage sofort aus Lemma 3.30.

Sie gilt für *REG* ebenfalls. Zum einen ist die zur Erzeugung von L_n im Beweis von Lemma 3.30 verwendete Grammatik regulär, woraus $\mathrm{Prod}_{REG}(L_n) \leq n$ folgt. Zum anderen gilt aber sicher $n = \mathrm{Prod}_{CF}(L_n) \leq \mathrm{Prod}_{REG}(L_n)$. □

Satz 3.36 $MON \leq^3_{\mathrm{Prod}} CF \leq^3_{\mathrm{Prod}} REG \leq^3_{\mathrm{Prod}} nfREG$.

Beweis. $MON \leq^3_{\mathrm{Prod}} CF$ folgt aus Lemma 3.31. $CF \leq^3_{\mathrm{Prod}} REG$ folgt aus Lemma 3.33. $REG \leq^3_{\mathrm{Prod}} nfREG$ folgt aus Lemma 3.34. □

Wir haben bisher keine Aussagen zu Einschränkungen der kontextfreien Grammatiken gemacht. Aus Lemma 3.29 folgt sofort, dass

$$CF \approx_{\mathrm{Prod}} redCF$$

gilt. Dagegen ergibt sich für λ-freie Grammatiken die Relation

$$CF \leq^3_{\mathrm{Prod}} CF\text{-}\lambda,$$

auf deren Beweis wir verzichten. Man beachte, dass hinsichtlich Var und Prod das Verhältnis zwischen *CF* und *CF*-λ sehr unterschiedlich ist. Ebenfalls ohne Beweis geben wir an, dass

$$CF \leq^3_{\mathrm{Prod}} ChCF$$

gilt.

Wir kommen nun zu den Entscheidbarkeitsfragen bez. des Maßes Prod. Offensichtlich gilt der folgende Satz.

Satz 3.37 *Für jede Grammatik ist* $\mathrm{Prod}(G)$ *berechenbar.* □

Satz 3.38

i) *Es ist entscheidbar, ob für eine gegebene kontextfreie Grammatik G die Beziehung* $\mathrm{Prod}_{CF}(L(G)) = 1$ *gilt.*

ii) *Es ist unentscheidbar, ob für eine gegebene kontextfreie Grammatik G und eine gegebene Zahl $n \geq 2$ die Beziehung* $\mathrm{Prod}_{CF}(L(G)) = n$ *gilt.*

Beweis. i) Enthält G nur eine einzige Regel, so ist $L(G)$ entweder leer oder enthält genau ein Wort. Für eine kontextfreie Grammatik ist es entscheidbar, ob sie leer bzw. endlich ist. Ferner liefert im Fall der Endlichkeit die Konstante aus dem Pumping-Lemma, die aus G berechenbar ist, eine obere Schranke für die Wörter aus $L(G)$. Wir testen nun alle Wörter bis zu dieser Länge, ob sie von G erzeugt werden. Werden mindestens zwei Wörter erzeugt, so ist $\mathrm{Prod}_{CF}(L(G)) \neq 1$.

ii) Es sei $n = 2$. Wir betrachten erneut die Sprachen $L_{X,Y}$ und L_s aus dem Beweis von Satz 3.21 zu gegebenen Mengen X und Y. Ferner sei die Abbildung $h : \{a, b, c\}^* \to \{a, b\}^*$ durch

$$\begin{aligned} &h(\lambda) = \lambda,\ h(a) = ab,\ h(b) = aabb,\ h(c) = aaabbb, \\ &h(x_1x_2) = h(x_1)h(x_2) \text{ für } x_1, x_2 \in \{a, b, c\}^* \end{aligned}$$

gegeben. Weiterhin betrachten wir die Grammatik

$$G = (\{S\}, \{a, b\}, \{S \to SaSb,\ S \to \lambda\}, S)$$

und die von G erzeugte Sprache L und setzen

$$L' = L \setminus h(L_{X,Y} \cap L_s).$$

Wir zeigen

$$\mathrm{Prod}_{CF}(L') \begin{cases} = 2, & \text{wenn das PKP bez. } X \text{ und } Y \text{ keine Lösung hat,} \\ \geq 3, & \text{wenn das PKP bez. } X \text{ und } Y \text{ eine Lösung hat.} \end{cases}$$

Hat das PKP bez. X und Y keine Lösung, so gilt $L' = L$ und damit nach Konstruktion $\mathrm{Prod}_{CF}(L') = 2$.

Hat das PKP eine Lösung $i_1 i_2 \ldots i_k$, so liegt

$$w = ba^{i_1}b \ldots ba^{i_k}cx_{i_k} \ldots x_{i_1}cy_{i_1} \ldots y_{i_k}ca^{i_k}b \ldots a^{i_1}b$$

in $L_{X,Y} \cap L_s$. Ist $h(w)$ in L, so sind wegen der Ableitungen

$$\begin{aligned} &S \Longrightarrow SaSb \Longrightarrow^* waSb \Longrightarrow wab, \\ &S \Longrightarrow SaSb \Longrightarrow^* waSb \Longrightarrow waSaSbb \Longrightarrow^* waabb, \\ &S \Longrightarrow SaSb \Longrightarrow^* waSb \Longrightarrow waSaSbb \Longrightarrow waSaSaSbbb \Longrightarrow^* waaabbb \end{aligned}$$

in G auch $wab, waabb, waaabbb$ in L. Wegen $\lambda \in L$ folgt daher

$$\{ab, aabb, aaabbb\}^* \subset L.$$

Wegen $h(w) \in \{ab, aabb, aaabbb\}^*$ folgt damit $L' \subset L$. Andererseits enthält L' aber sicher alle Wörter aus L mit einer Länge ≤ 30, da die Wörter in $L_{X,Y} \cap L_s$ mindestens drei c, zwei a und zwei b enthalten. Angenommen, es gibt eine Grammatik $H = (N, T, P, S)$ zur Erzeugung von L' mit nur 2 Regeln. Wegen $\lambda \in L'$ muss es eine Regel $A \to \lambda$ geben. Ist $A \neq S$, so muss die zweite Regel $S \to A^k$ für ein $k \geq 1$ sein, da sonst λ nicht erzeugt werden kann. Dann gilt aber $L(H) = \{\lambda\}$ im Widerspruch zu $L(H) = L' \neq \{\lambda\}$. Folglich ist eine der beiden Regeln $S \to \lambda$. Da jede Anwendung der Regeln $S \to SaSb$ und $S \to \lambda$ aus G die gleiche Anzahl von as und bs produziert, muss dies auch für die Regeln aus P gelten. Ferner muss $ab \in L'$ herleitbar sein. Damit muss die zweite Regel in P die Form $S \to S^r aS^s bS^t$ haben.

Es seien zuerst $r \geq 1$, $s \geq 1$ und $t \geq 0$. Dann gibt es die Ableitung

$$S \Longrightarrow S^r aS^s bS^t \Longrightarrow^* SaS^s bS^t \Longrightarrow^* SaSb.$$

Verwenden wir stets diese Ableitung anstelle der Regel selbst, so folgt offenbar $L \subseteq L(H)$. Dies widerspricht wegen $L' \subset L$ der Forderung $L(H) = L'$.

Bei $r = t = 0$ und $s \geq 1$ gilt $abab \in L'$ aber $abab \notin L(H)$. In den Fällen $s = 0$ und $r = 0$, $t \geq 1$ oder $r \geq 1$, $t = 0$ oder $r \geq 1$, $t \geq 1$ ergibt sich $L(H) = \{ab\}^*$. In diesen Fällen erhalten wir wegen $aabb \in L'$ jeweils einen Widerspruch zu $L(H) = L'$.

Bei $r = s = t = 0$ ergibt sich $L(H) = \{ab, \lambda\}$. Dies widerspricht ebenfalls $L(H) = L'$.

Für den verbleibenden Fall $r0$, $s \geq 1$, $t \geq 1$ können wir wie oben zeigen, dass $\{ab, aabb, aaabbb\}^* \subseteq L(H)$ gilt. Wegen $h(w) \in \{ab, aabb, aaabbb\}^*$ ist damit $h(w) \in L(H)$, während $h(w) \notin L'$ gilt.

Da wir in allen Fällen einen Widerspruch hergeleitet haben, sind zum Erzeugen von L' mindestens drei Regeln nötig.

Es sei $n = 3$. In diesem Fall setzen wir

$$L'' = \{a, b\}^* \setminus h(L_{X,Y} \cap L_s)$$

und beweisen analog

$$\mathrm{Prod}_{CF}(L'') \begin{cases} = 3, & \text{wenn das PKP bez. } X \text{ und } Y \text{ keine Lösung hat,} \\ \geq 4, & \text{wenn das PKP bez. } X \text{ und } Y \text{ eine Lösung hat.} \end{cases}$$

Es sei nun $n \geq 4$. Wir setzen $m = n - 3$ und

$$U_m = \{d^{2^i} \mid 1 \leq i \leq m\}.$$

Wegen Lemma 3.30 ist eine bez. Prod minimale Grammatik für U_m dadurch gegeben, dass wir die Wörter aus U_m jeweils in einem Schritt direkt aus dem Axiom Z herleiten. Es ist leicht zu sehen, dass dann

$$\mathrm{Prod}_{CF}(L' \cup U_m) = \mathrm{Prod}_{CF}(L') + m + 1$$

gilt (zusätzlich zu den Wörtern aus U_m erzeugen wir aus Z noch das Axiom der minimalen Grammatik für L'). Damit gilt

$$\mathrm{Prod}_{CF}(L' \cup U_m) \begin{cases} = n, & \text{wenn das PKP bez. } X \text{ und } Y \text{ keine Lösung hat,} \\ > n, & \text{wenn das PKP bez. } X \text{ und } Y \text{ eine Lösung hat.} \end{cases}$$

Damit haben wir nachgewiesen, dass es für jede natürliche Zahl $n \geq 2$ unentscheidbar ist, ob eine gegebene kontextfreie Grammatik G die Bedingung $\mathrm{Prod}_{CF}(L(G)) = n$ erfüllt. □

Wie bei den Betrachtungen zur Anzahl der Nichtterminale lassen sich nun aus Satz 3.38 die folgenden Aussagen ableiten.

Folgerung 3.39

i) *Es gibt keinen Algorithmus, der zu einer beliebigen kontextfreien Grammatik G den Wert* $\mathrm{Prod}_{CF}(L(G))$ *berechnet.*

ii) *Es gibt keinen Algorithmus, der zu einer beliebigen kontextfreien Grammatik G eine bez.* Prod *minimale kontextfreie Grammatik G' für $L(G)$ liefert.* □

Es ist noch offen, ob es entscheidbar ist, ob eine gegebene kontextfreie Grammatik G bez. Prod minimal für $L(G)$ ist.

In Analogie zu den Beweisen bezüglich des Maßes Var können wir nun zeigen, dass die angegebenen Probleme alle entscheidbar sind, wenn wir uns auf reguläre Grammatiken in Normalform beschränken.

Satz 3.40

i) *Es gibt einen Algorithmus, der zu einer gegebenen regulären Grammatik G in Normalform eine reguläre Grammatik G' in Normalform bestimmt, für die $L(G') = L(G)$ und* $\mathrm{Prod}(G') = \mathrm{Prod}_{nfREG}(L(G))$ *gelten.*

ii) *Es gibt einen Algorithmus, der zu gegebener regulärer Grammatik G in Normalform* $\mathrm{Prod}_{nfREG}(L(G))$ *bestimmt.*

iii) *Es ist entscheidbar, ob eine gegebene Grammatik G in regulärer Normalform minimal für $L(G)$ ist, d. h. ob* $\mathrm{Prod}(G) = \mathrm{Prod}_{nfREG}(L(G))$ *gilt.* □

Erneut ist es offen, ob diese Probleme für reguläre Grammatiken entscheidbar oder unentscheidbar sind.

3.4. Anzahl der Symbole

Wir untersuchen jetzt Grammatiken hinsichtlich des Komplexitätsmaßes Symb.

Lemma 3.41 *Für je zwei Mengen X und Y von Grammatiken mit $X \subseteq Y$ und jede Sprache $L \in \mathcal{L}(X)$ gilt* $\mathrm{Symb}_Y(L) \leq \mathrm{Symb}_X(L)$.

Beweis. Der Beweis wird analog zu dem von Lemma 3.4 geführt. □

Lemma 3.42 *Für jede natürliche Zahl n und jede Sprache L, die aus genau einem Wort der Länge n besteht, gilt* $\mathrm{Symb}_{REG}(L) = n + 2$.

Beweis. Es sei $L = \{w\}$ für ein Wort $w \in T^*$ der Länge n.

Da die Grammatik $G = (\{S\}, T, \{S \to w\}, S)$ nur das Wort w erzeugt, gilt offenbar $\mathrm{Symb}_{REG}(L) \leq \mathrm{Symb}(G) = n + 2$.

Es sei nun $H = (N, T, P, S)$ eine reguläre Grammatik, die minimal für L ist. Enthält sie die Regel $S \to w$, so gilt $\mathrm{Symb}(H) \geq n + 2$.

Enthält P die Regel $S \to w$ nicht, so gibt es für w eine Ableitung

$$\begin{aligned} S = A_0 &\Longrightarrow w_1 A_1 \Longrightarrow w_1 w_2 A_2 \Longrightarrow \cdots \Longrightarrow w_1 w_2 \ldots w_{m-1} A_{m-1} \\ &\Longrightarrow w_1 w_2 \ldots w_{m-1} w_m = w, \end{aligned}$$

wobei der Reihe nach die Regeln

$$S \to w_1 A_1, \quad A_{i-1} \to w_i A_i \text{ für } 2 \leq i \leq m-1, \quad A_{m-1} \to w_m$$

angewendet werden. Falls alle A_i, $1 \leq i \leq m-1$, voneinander und von S verschieden sind, so ergibt sich

$$\mathrm{Symb}_{REG}(L) = \mathrm{Symb}(H) \geq |w_m| + 2 + \sum_{i=1}^{m-1}(|w_i| + 3) = |w| + 3m - 1 \geq n + 2.$$

Es sei $A_i = A_j$ mit $0 \leq i < j \leq m-1$. Wir betrachten die Ableitung

$$A_i \Longrightarrow^* w_{i+1}w_{i+2}\dots w_jA_j.$$

Gilt $w_{i+1}w_{i+2}\dots w_j \neq \lambda$, so kann dieser Teil iteriert werden, und wir können ein Wort der Länge $k > n$ erzeugen, was $H(L) = \{w\}$ widerspricht. Ist dagegen, $w_{i+1}w_{i+2}\dots w_j = \lambda$, so streichen wir in P die Regel $A_i \to A_{i+1}$ und erhalten P'. Offensichtlich gilt dann wegen der Existenz der Ableitung

$$\begin{aligned} S &\Longrightarrow w_1A_1 \Longrightarrow w_1w_2A_2 \Longrightarrow \cdots \Longrightarrow w_1w_2\dots w_iA_i = w_1w_2\dots w_iA_j \\ &\Longrightarrow^* w_1w_2\dots w_{m-1}w_m = w \end{aligned}$$

für die Grammatik $H' = (N, T, P', S)$ ebenfalls $L(H) = \{w\}$. Da H' drei Symbole weniger als H hat, ist H für L nicht minimal wie vorausgesetzt.

Damit ist auch $\mathrm{Symb}(H) \geq n + 2$ gezeigt. □

Hieraus ergibt sich sofort das folgende Resultat.

Folgerung 3.43 Symb *ist bezüglich REG vollständig.* □

Ohne Beweis geben wir an, dass diese Aussage auch für kontextfreie Grammatiken richtig ist.

Satz 3.44 Symb *ist bezüglich CF vollständig.* □

Satz 3.45 $CF \leq^2_{\mathrm{Symb}} REG$

Beweis. Wir betrachten für $n \geq 1$ die Sprache $L_n = \{a^{2^n}\}$. Nach Lemma 3.42 ergibt sich $\mathrm{Symb}_{REG}(L_n) = 2^n + 2$. Andererseits erzeugt die kontextfreie Grammatik

$$G_n = (\{A_0, A_1, \dots, A_{n-1}\}, \{a\}, P_n, A_0)$$

mit

$$P_n = \{A_{n-1} \to a^2\} \cup \bigcup_{i=0}^{n-2}\{A_i \to A_{i+1}^2\}$$

die Sprache L_n. Damit gilt $\mathrm{Symb}_{CF}(L_n) \leq \mathrm{Symb}(G_n) = 4n$. □

Den Autoren ist nicht bekannt, ob dieses Resultat optimal ist oder ob sogar

$$CF \leq^d_{\mathrm{Symb}} REG$$

gilt. Auch ist ihnen keine nichttriviale Relation zwischen *CF* und *MON* bekannt.

Wir betrachten nun den Übergang von kontextfreien Grammatiken zu λ-freien kontextfreien Grammatiken.

Satz 3.46 *Für jede kontextfreie Sprache L gilt* $\mathrm{Symb}_{CF\text{-}\lambda}(L) \leq 10 \cdot \mathrm{Symb}_{CF}(L)$.

Beweis. Es sei $G = (N, T, P, S)$ eine kontextfreie Grammatik, die bez. Symb minimal für L ist. Wir setzen

$$E = \{A \mid A \in N,\ A \Longrightarrow^* \lambda\}.$$

Für jede Regel $p = A \to \alpha \in P$ mit $\alpha \neq \lambda$ definieren nun eine Menge $\varphi(p)$ von Regeln in der folgenden Weise:

- Enthält α kein Symbol aus E, so setzen wir $\varphi(p) = \{p\}$. Damit gilt

  $$\mathrm{Symb}(\varphi(A \to \alpha)) = \mathrm{Symb}(A \to \alpha),$$

  wobei sich hier Symb auf die Anzahl der in der oder den angegebenen Regeln vorkommenden Symbole bezieht.

- Ist $\alpha = E_1 E_2 \ldots E_k \in E^+$, so besteht $\varphi(p)$ aus den Regeln

  $$\begin{aligned} A &\to E_1 \mid E_1 R_2 \mid R_2, \\ R_2 &\to E_2 \mid E_2 R_3 \mid R_3, \\ &\vdots \\ R_{k-2} &\to E_{k-2} \mid E_{k-2} R_{k-1} \mid R_{k-1}, \\ R_{k-1} &\to E_{k-1} \mid E_{k-1} E_k \mid E_k \text{ und} \\ S &\to \lambda, \text{ falls } A = S, \end{aligned}$$

  wobei $R_1, R_2, \ldots, R_{k-1}$ neue Nichtterminale sind, die auch von den neuen Nichtterminalen anderer Regeln verschieden sind. Man rechnet leicht nach, dass

  $$\mathrm{Symb}(\varphi(A \to \alpha)) = 10(k-1) \leq 10(k+2) = 10 \cdot \mathrm{Symb}(A \to \alpha)$$

  gilt.

- Falls α sowohl Elemente aus E als auch welche aus $(N \cup T) \setminus E$ enthält, so stellen wir zuerst fest, dass es Wörter $u_i \in ((N \cup T) \setminus E)^*$, $0 \leq i \leq n$, und $\alpha_j \in E^+$, $1 \leq j \leq n$, mit $u_k \neq \lambda$ für $1 \leq k \leq n-1$ und $\alpha = u_0 \alpha_1 u_1 \alpha_2 u_2 \ldots \alpha_n u_n$ gibt. Wir setzen nun

  $$\varphi(p) = \{A \to u_0 A_1 u_1 A_2 u_2 \ldots A_n u_n\} \cup \bigcup_{i=1}^{n} \varphi(A_i \to \alpha_i),$$

  wobei erneut die A_i, $1 \leq i \leq n$, neue Nichtterminale sind und $\varphi(A_i \to \alpha_i)$, $1 \leq i \leq n$, wie im vorhergehenden Punkt definiert wird. Dann gilt

  $$\begin{aligned} \mathrm{Symb}(\varphi(p)) &\leq n + 2 + \sum_{i=0}^{n} |u_i| + 10 \sum_{i=1}^{n} |\alpha_i| \\ &\leq 10(2 + \sum_{i=0}^{n} |u_i| + \sum_{i=1}^{n} |\alpha_i|) \\ &= 10 \cdot \mathrm{Symb}(p). \end{aligned}$$

Es seien nun $P' = \bigcup_{p \in P} \varphi(p)$ und N' die Menge aller Nichtterminale, die in Regeln von P' vorkommen. Dann erfüllt die λ-freie Grammatik $G' = (N', T, P', S)$ die Bedingungen $L(G') = L(G)$ und $\mathrm{Symb}(G') \leq 10 \cdot \mathrm{Symb}(G)$, woraus die Behauptung folgt. □

Hinsichtlich der Entscheidbarkeitsfragen beginnen wir erneut mit dem trivialen Resultat, dass die Anzahl von Symbolen von Grammatiken berechenbar ist.

Satz 3.47 *Für eine kontextfreie Grammatik G ist* $\mathrm{Symb}(G)$ *berechenbar.* □

Für die Entscheidbarkeitsfragen bezüglich der kontextfreien Sprachen geben wir ohne Beweis den folgenden Satz an.

Satz 3.48

i) *Es gibt keinen Algorithmus, der zu einer beliebigen kontextfreien Grammatik G eine kontextfreie Grammatik G' bestimmt, die für $L(G)$ minimal ist.*

ii) *Es gibt keinen Algorithmus, der zu einer beliebigen kontextfreien Grammatik G den Wert* $\mathrm{Symb}_{CF}(L(G))$ *bestimmt.*

iii) *Es ist unentscheidbar, ob für eine gegebene kontextfreie Grammatik G und eine gegebene natürliche Zahl $n \geq 8$ die Beziehung* $\mathrm{Symb}_{CF}(L(G)) = n$ *gilt; bis $n = 7$ ist die Frage entscheidbar.*

iv) *Es ist unentscheidbar, ob eine gegebene kontextfreie Grammatik G minimal für $L(G)$ ist, d. h. ob* $\mathrm{Symb}(G) = \mathrm{Symb}_{CF}(L(G))$ *gilt.* □

Wir merken an, dass die Entscheidbarkeit $\mathrm{Symb}_{CF}(L(G)) = n$ für $n \leq 7$ – wie bei Prod – daraus resultiert, dass bei $n \leq 7$ nur sehr spezielle reguläre Mengen erzeugt werden können.

Für Symb erhalten wir bereits für reguläre Grammatiken die Entscheidbarkeit der Probleme. Bei Var und Prod ist die Entscheidbarkeit nur für reguläre Grammatiken in Normalform nachgewiesen, während für beliebige reguläre Grammatiken die Entscheidbarkeit offen ist.

Satz 3.49

i) *Es gibt einen Algorithmus, der zu einer gegebenen regulären Grammatik G eine reguläre Grammatik G' bestimmt, für die $L(G') = L(G)$ und* $\mathrm{Symb}(G') = \mathrm{Symb}_{REG}(L(G))$ *gelten.*

ii) *Es gibt einen Algorithmus, der zu gegebener regulärer Grammatik G* $\mathrm{Symb}_{REG}(L(G))$ *bestimmt.*

iii) *Es ist entscheidbar, ob eine gegebene reguläre Grammatik G minimal für $L(G)$ ist, d. h. ob* $\mathrm{Symb}(G) = \mathrm{Symb}_{REG}(L(G))$ *gilt.*

Beweis. i) Es sei $G = (N, T, P, S)$ gegeben. Wir berechnen zuerst $\mathrm{Symb}(G)$. Dann konstruieren wir alle regulären Grammatiken H mit den Nichtterminalen aus der Menge $A_1, A_2, \ldots, A_{\mathrm{Symb}(G)}$, der Terminalmenge T und $\mathrm{Symb}(H) \leq \mathrm{Symb}(G)$. Dies sind offensichtlich nur endlich viele Grammatiken. Es sei M die Menge aller dieser regulären Grammatiken. Dann bestimmen wir

$$k = \min\{\mathrm{Symb}(H) \mid H \in M,\ L(H) = L(G)\}.$$

Jede Grammatik G' aus M mit $L(G') = L(G)$ und $\mathrm{Symb}(G') = k$ ist eine minimale Grammatik für $L(G)$.

ii) Der im Teil i) dieses Beweises erhaltene Wert k ist $\mathrm{Symb}_{REG}(L(G))$.

iii) Die Aussage folgt aus i). □

Kapitel 4

Beschreibungskomplexität endlicher Automaten

Die Ausführungen in diesem Kapitel stammen im Wesentlichen aus [9] (englische Version) bzw. [10] (deutsche Version).

4.1. Eine algebraische Charakterisierung der regulären Sprachen

Ein deterministischer *endlicher Automat* ist ein Quintupel

$$\mathcal{A} = (X, Z, z_0, \delta, F),$$

wobei X und Z Alphabete sind, z_0 ein Element von Z ist, δ eine Funktion von $Z \times X$ in Z ist, und F eine nichtleere Teilmenge von Z ist. Das Alphabet X ist die Menge der Eingabesymbole, Z ist die Menge der Zustände, z_0 ist der Anfangszustand, δ ist die Überführungsfunktion und F ist die Menge der akzeptierenden Zustände.

Im Unterschied dazu ist bei nichtdeterministischen endlichen Automaten die Überführungsfunktion als Abbildung in die Potenzmenge von Z erklärt (einem Paar $(z, x) \in Z \times X$ wird nicht ein einzelner Zustand zugeordnet sondern eine Menge von Zuständen). Im weiteren Verlauf werden wir stets deterministische endliche Automaten betrachten.

Durch

$$\begin{aligned} \delta^*(z, \lambda) &= z, \text{ für } z \in Z, \\ \delta^*(z, wa) &= \delta(\delta^*(z, w), a) \text{ für } w \in X^*,\ a \in X \end{aligned}$$

erweitern wir δ zu einer Funktion δ^* von $Z \times X^*$ in Z.

Die von $\mathcal{A}$ *akzeptierte* Sprache ist durch

$$T(\mathcal{A}) = \{w \in X^* \mid \delta^*(z_0, w) \in F\}$$

definiert.

Es ist bekannt, dass eine Sprache L genau dann regulär ist, wenn es einen endlichen Automaten $\mathcal{A}$ gibt, der L akzeptiert.

Wir wollen zuerst zwei algebraische Charakterisierungen regulärer Sprachen herleiten. Dazu benötigen wir die folgenden Definitionen.

Eine Äquivalenzrelation $\sim$ auf der Menge X^* heißt *Rechtskongruenz*, falls für beliebige Wörter $x, y \in X^*$ und $a \in X$ aus $x \sim y$ auch $xa \sim ya$ folgt. Bei Multiplikation von rechts werden äquivalente Wörter wieder in äquivalente Wörter überführt.

Eine Äquivalenzrelation $\sim$ auf X^* heißt *Verfeinerung* der Menge $R \subseteq X^*$, wenn für beliebige Wörter $x, y \in X^*$ aus $x \sim y$ folgt, dass $x \in R$ genau dann gilt, wenn auch $y \in R$ gilt. Dies bedeutet, dass mit einem Wort $x \in R$ auch alle zu x äquivalenten Wörter in R liegen. Folglich liegt die zu $x \in R$ gehörige Äquivalenzklasse vollständig in R. Hieraus resultiert die Bezeichnung Verfeinerung für diese Eigenschaft.

Für eine Äquivalenzrelation $\sim$ bezeichnen wir die Anzahl der Äquivalenzklassen von $\sim$ als Index von $\sim$ und bezeichnen sie mit $Ind(\sim)$. Die Äquivalenzrelation $\sim$ hat *endlichen Index*, falls $Ind(\sim)$ endlich ist.

Wir nennen eine Äquivalenzrelation $\sim$ eine *R-Relation*, falls sie eine Rechtskongruenz mit endlichem Index ist und eine Verfeinerung von R ist.

Beispiel 4.1 *Es seien $\mathcal{A} = (X, Z, z_0, \delta, F)$ ein endlicher Automat und $R = T(\mathcal{A})$. Wir definieren auf X^* die Relation $\sim_{\mathcal{A}}$ durch*

$$x \sim_{\mathcal{A}} y \quad \text{genau dann, wenn} \quad \delta^*(z_0, x) = \delta^*(z_0, y)$$

gilt. Offenbar ist $\sim_{\mathcal{A}}$ eine Äquivalenzrelation. Wir zeigen nun, dass $\sim_{\mathcal{A}}$ eine R-Relation ist.

Es sei $x \sim_{\mathcal{A}} y$. Dann gilt nach Definition $\delta^*(z_0, x) = \delta^*(z_0, y)$. Damit erhalten wir

$$\delta^*(z_0, xa) = \delta(\delta^*(z_0, x), a) = \delta(\delta^*(z_0, y), a) = \delta^*(z_0, ya)$$

und somit $xa \sim_{\mathcal{A}} ya$, womit nachgewiesen ist, dass $\sim_{\mathcal{A}}$ eine Rechtskongruenz ist.

Es sei erneut $x \sim_{\mathcal{A}} y$. Ferner sei $x \in R$. Dies bedeutet $\delta^*(z_0, x) = \delta^*(z_0, y)$ und $\delta^*(z_0, x) \in F$. Folglich gilt $\delta^*(z_0, y) \in F$, woraus $y \in R$ folgt. Analog folgt aus $y \in R$ auch $x \in R$. Damit ist $\sim_{\mathcal{A}}$ Verfeinerung von R.

Es sei $x \in X^*$. Ferner sei $\delta^*(z_0, x) = z$. Dann ergibt sich für die zu x bez. $\sim_{\mathcal{A}}$ gehörende Äquivalenzklasse $[x]_{\mathcal{A}}$

$$\begin{aligned}
[x]_{\mathcal{A}} &= \{y \mid x \sim_{\mathcal{A}} y\} \\
&= \{y \mid \delta^*(z_0, x) = \delta^*(z_0, y)\} \\
&= \{y \mid \delta^*(z_0, y) = z\}.
\end{aligned}$$

Es sei nun $z \in Z$ ein von z_0 erreichbarer Zustand. Dann gibt es ein Wort x mit $\delta^*(z_0, x) = z$ und die Beziehungen

$$\begin{aligned}
\{y \mid \delta^*(z_0, y) = z\} &= \{y \mid \delta^*(z_0, y) = \delta^*(z_0, x)\} \\
&= \{y \mid y \sim_{\mathcal{A}} x\} \\
&= [x]_{\mathcal{A}}
\end{aligned}$$

sind gültig. Daher gibt es eine eineindeutige Beziehung zwischen den Äquivalenzklassen von $\sim_{\mathcal{A}}$ und den Zuständen von $\mathcal{A}$, die vom Anfangszustand erreichbar sind. Dies bedeutet, dass die Anzahl der Zustände von $\mathcal{A}$ mit dem Index von $\sim_{\mathcal{A}}$ übereinstimmt. Wegen der Endlichkeit der Zustandsmenge Z ist also auch der Index von $\sim_{\mathcal{A}}$ endlich.

Beispiel 4.2 *Für eine Sprache $R \subseteq X^*$ definieren wir die Relation $\sim_R$ wie folgt: $x \sim_R y$ gilt genau dann, wenn für alle $w \in X^*$ das Wort xw genau dann in R ist, falls dies auch für yw der Fall ist. Wir zeigen, dass $\sim_R$ eine Rechtskongruenz ist, die R verfeinert.*

Es seien dazu $x \sim_R y$, $a \in X$ und $w \in X^*$. Dann ist $aw \in X^*$ und nach Definition der Relation $\sim_R$ gilt $xaw \in R$ genau dann, wenn $yaw \in R$ gültig ist. Dies liefert, da w beliebig ist, die Aussage $xa \sim_R ya$. Somit ist $\sim_R$ eine Rechtskongruenz.

Wählen wir $w = \lambda$, so ergibt sich aus $x \sim_R y$ nach Definition, dass $x \in R$ genau dann gilt, wenn auch $y \in R$ erfüllt ist. Deshalb ist $\sim_R$ eine Verfeinerung von R.

Wir bemerken, dass $\sim_R$ nicht notwendigerweise einen endlichen Index haben muss. Dazu betrachten wir

$$R = \{a^n b^n \mid n \geq 0\}$$

und zwei Wörter a^k und a^ℓ mit $k \neq \ell$. Wegen $a^k b^k \in R$ und $a^\ell b^k \notin R$ sind offensichtlich a^ℓ und a^k nicht äquivalent. Somit gibt es mindestens soviel Äquivalenzklassen wie Potenzen von a und damit soviel wie natürliche Zahlen. Dies zeigt die Unendlichkeit des Index.

Ziel dieses Abschnitts ist eine Charakterisierung der regulären Sprachen durch Äquivalenzrelationen mit den oben definierten Eigenschaften.

Satz 4.3 *Für eine Sprache $R \subseteq X^*$ sind die folgenden Aussagen gleichwertig:*

i) *R ist regulär.*

ii) *Es gibt eine R-Relation.*

iii) *Die Relation $\sim_R$ (aus Beispiel 4.2) hat endlichen Index.*

Beweis. i) $\Longrightarrow$ ii). Zu einer regulären Sprache R gibt es einen endlichen Automaten $\mathcal{A}$ mit $T(\mathcal{A}) = R$. Dann können wir entsprechend Beispiel 4.1 die Relation $\sim_{\mathcal{A}}$ konstruieren. Diese ist nach Beispiel 4.1 eine R-Relation.

ii) $\Longrightarrow$ iii) Nach Voraussetzung gibt es eine R-Relation $\sim$ mit endlichem Index. Wir beweisen nun $Ind(\sim_R) \leq Ind(\sim)$.

Dazu zeigen wir zuerst, dass aus $x \sim y$ auch $xw \sim yw$ für alle $w \in X^*$ folgt. Für $w = \lambda$ ist dies klar. Für $w \in X$ ist es klar, da $\sim$ eine Rechtskongruenz ist. Es sei die Aussage nun schon für $|w| < n$ bewiesen. Wir betrachten das Wort v der Länge n. Dann gibt es ein w der Länge $n-1$ und ein $a \in X$ mit $v = wa$. Nach Induktionsvoraussetzung haben wir $xw \sim yw$. Nach der Definition der Rechtskongruenz folgt daraus $xwa \sim ywa$ und damit $xv \sim yv$.

Es sei nun $x \sim y$ und $w \in X^*$. Dann gilt auch $xw \sim yw$. Da $\sim$ eine R-Relation ist, folgt $xw \in R$ gilt genau dann, wenn $yw \in R$ gilt. Nach Definition $\sim_R$ bedeutet dies aber $x \sim_R y$. Damit ist gezeigt, dass $x \sim y$ auch $x \sim_R y$ impliziert. Hieraus ergibt sich aber

$$\{y \mid y \sim x\} \subseteq \{y \mid y \sim_R x\}.$$

Jede Äquivalenzklasse von $\sim$ ist also in einer Äquivalenzklasse von $\sim_R$ enthalten. Damit gilt $Ind(\sim_R) \leq Ind(\sim)$. Da $Ind(\sim)$ endlich ist, hat auch $\sim_R$ endlichen Index.

iii) $\Longrightarrow$ i) Mit $[x]_R$ bezeichnen wir die Äquivalenzklasse von $x \in X^*$ bez. $\sim_R$. Wir betrachten den endlichen Automaten

$$\mathcal{A} = (X, \{[x]_R \mid x \in X^*\}, [\lambda]_R, \delta, \{[y]_R \mid y \in R\})$$

mit

$$\delta([x]_R, a) = [xa]_R.$$

Wir bemerken zuerst, dass aufgrund der Voraussetzung, dass $\sim_R$ eine R-Relation ist, die Definition von $\mathcal{A}$ zulässig ist. (Die Endlichkeit der Zustandsmenge von $\mathcal{A}$ folgt aus der Endlichkeit des Index von $\sim_R$. Falls $[x]_R = [y]_R$, so ist auch $[xa]_R = [ya]_R$, denn $\sim_R$ ist eine Rechtskongruenz und daher gilt mit $x \sim_R y$, d. h. $[x]_R = [y]_R$, auch $xa \sim_R ya$.) Ferner beweist man mittels vollständiger Induktion über die Länge von x leicht, dass $\delta^*([\lambda]_R, x) = [x]_R$ für alle $x \in X^*$ gilt. Damit folgt

$$T(\mathcal{A}) = \{x \mid \delta^*([\lambda]_R, x) \in \{[y]_R \mid y \in R\}\} = \{x \mid [x]_R \in \{[y]_R \mid y \in R\}\} = R.$$

Damit ist R als regulär nachgewiesen. □

Im Beweisteil ii) $\Longrightarrow$ iii) wurde eigentlich die folgende Aussage bewiesen.

Folgerung 4.4 *Für jede R-Relation $\sim$ einer beliebigen regulären Sprache R gilt $Ind(\sim) \geq Ind(\sim_R)$.* □

4.2. Minimierung endlicher Automaten

In diesem Abschnitt diskutieren wir als Komplexitätsmaß die Anzahl der Zustände eines endlichen Automaten. Dazu setzen wir für einen endlichen Automaten $\mathcal{A} = (X, Z, z_0, \delta, F)$ und eine reguläre Sprache R

$$z(\mathcal{A}) = \#(Z),$$
$$z(R) = \min\{z(\mathcal{A}) \mid T(\mathcal{A}) = R\}.$$

Wir sagen, dass $\mathcal{A}$ *minimaler* Automat für R ist, falls $T(\mathcal{A}) = R$ und $z(\mathcal{A}) = z(R)$ gelten.

Als erstes geben wir ein Resultat an, dass die Größe $z(R)$ zu einer Sprache R bestimmt.

Satz 4.5 *Für jede reguläre Sprache R gilt $z(R) = Ind(\sim_R)$.*

Beweis. Wir bemerken zuerst, dass aus dem Teil iii) $\Longrightarrow$ i) des Beweises von Satz 4.3 folgt, dass es einen Automaten $\mathcal{A}$ mit $z(\mathcal{A}) = Ind(\sim_R)$ gibt.

Es sei nun $\mathcal{A} = (X, Z, z_0, \delta, F)$ ein beliebiger endlicher Automat mit $T(\mathcal{A}) = R$. Dann ist nach Beispiel 4.1 die Relation $\sim_{\mathcal{A}}$ eine R-Relation, für die $z(\mathcal{A}) = Ind(\sim_{\mathcal{A}})$ gilt. Wegen Folgerung 4.4 erhalten wir daraus sofort $z(\mathcal{A}) \geq Ind(\sim_R)$.

Die Kombination dieser beiden Erkenntnisse besagt gerade die Behauptung. □

Wir beschreiben nun für einen Automaten $\mathcal{A}$ einen Automaten, der minimal für $T(\mathcal{A})$ ist.

Es sei $\mathcal{A} = (X, Z, z_0, \delta, F)$ ein endlicher Automat. Für $z \in Z$ und $z' \in Z$ setzen wir $z \approx_{\mathcal{A}} z'$ genau dann, wenn für alle $x \in X^*$ genau dann $\delta^*(z, x)$ in F liegt, wenn auch $\delta^*(z', x)$ in F liegt. Falls der Automat $\mathcal{A}$ aus dem Kontext klar ist, schreiben wir einfach $\approx$ anstelle von $\approx_{\mathcal{A}}$

Lemma 4.6 *Die Relation $\approx_{\mathcal{A}}$ ist eine Äquivalenzrelation auf der Menge der Zustände des Automaten $\mathcal{A}$.*

Beweis. Wir verzichten auf den einfachen Standardbeweis. □

Mit $[z]_\approx$ bezeichnen wir die Äquivalenzklasse von $z \in Z$ bezüglich $\approx$ $(= \approx_{\mathcal{A}})$. Wegen $\lambda \in X^*$ impliziert $z \approx z'$ für zwei Zustände z und z', dass $z \in F$ genau dann gilt, wenn auch $z' \in F$ gilt. Daraus folgt $[z]_\approx \subseteq F$.

Wir setzen

$$Z_\approx = \{[z]_\approx \mid z \in Z\} \quad \text{und} \quad F_\approx = \{[z]_\approx \mid z \in F\}$$

und konstruieren den Automaten

$$\mathcal{A}_\approx = (X, Z_\approx, [z_0]_\approx, \delta_\approx, F_\approx)$$

mit

$$\delta_\approx([z]_\approx, a) = [\delta(z, a)]_\approx.$$

Wir zeigen, dass die Definition von $\mathcal{A}_\approx$ zulässig ist. Dazu haben wir nachzuweisen, dass die Definition von $\delta_\approx$ unabhängig von der Auswahl des Repräsentanten der Äquivalenzklasse ist. Es seien daher z und z' zwei Zustände mit $[z]_\approx = [z']_\approx$ und a ein beliebiges Element aus X. Dann muss $z \approx z'$ gelten. Wir betrachten ein beliebiges Wort $w \in X^*$. Dann liegt $\delta^*(z, aw)$ in F genau dann wenn $\delta^*(z', aw)$ in F liegt. Wegen $\delta^*(z, aw) = \delta^*(\delta(z, a), w)$ und $\delta^*(z', aw) = \delta^*(\delta(z', a), w)$ folgt $\delta(z, a) \approx \delta(z', a)$ und damit auch $[\delta(z, a)]_\approx = [\delta(z', a)]_\approx$, was zu zeigen war.

Wir zeigen nun, dass $\mathcal{A}_\approx$ minimal für die von $\mathcal{A}$ akzeptierte reguläre Sprache ist.

Satz 4.7 *Für jeden Automaten $\mathcal{A}$ ist $\mathcal{A}_\approx$ minimaler Automat für $T(\mathcal{A})$.*

Beweis. Wir zeigen zuerst mittels vollständiger Induktion über die Wortlänge, dass sich die Definition von $\delta_\approx : Z_\approx \times X \to Z_\approx$ auch auf die Erweiterung $\delta^*_\approx : Z_\approx \times X^* \to Z_\approx$ übertragen lässt, d. h. dass $\delta^*_\approx([z]_\approx, y) = [\delta^*(z, y)]_\approx$ für alle $y \in X^*$ gilt. Nach (genereller) Definition der Erweiterung haben wir

$$\begin{aligned}
&\delta^*_\approx([z]_\approx, \lambda) = [z]_\approx = [\delta^*(z, \lambda)]_\approx, \\
&\delta^*_\approx([z]_\approx, a) = \delta_\approx([z]_\approx, a) = [\delta(z, a)]_\approx = [\delta^*(z, a)]_\approx,
\end{aligned}$$

womit der Induktionsanfang gesichert ist. Der Induktionsschluss ist durch

$$\begin{aligned}
\delta^*_\approx([z]_\approx, ya) &= \delta_\approx(\delta^*_\approx([z]_\approx, y), a) \\
&= \delta_\approx([\delta^*(z, y)]_\approx, a) \\
&= [\delta(\delta^*(z, y), a)]_\approx \\
&= [\delta^*(z, ya)]_\approx
\end{aligned}$$

gegeben.

Hieraus erhalten wir sofort

$$\begin{aligned} T(\mathcal{A}_\approx) &= \{x \mid \delta^*_\approx([z_0]_\approx, x) \in F_\approx\} \\ &= \{x \mid [\delta^*(z_0, x)]_\approx \in F_\approx\} \\ &= \{x \mid \delta^*(z_0, x) \in F\} \\ &= T(\mathcal{A}). \end{aligned}$$

Damit bleibt nur noch $z(\mathcal{A}_\approx) = z(T(\mathcal{A}))$ zu zeigen. Wegen Satz 4.5 reicht es nachzuweisen, dass $z(\mathcal{A}_\approx) = Ind(\sim_{T(\mathcal{A})})$ erfüllt ist. Dazu betrachten wir die Relation $\sim_{\mathcal{A}_\approx}$ entsprechend Beispiel 4.1, für die $z(\mathcal{A}_\approx) = Ind(\sim_{\mathcal{A}_\approx})$ gilt, und beweisen $\sim_{\mathcal{A}_\approx} = \sim_{T(\mathcal{A})}$, womit der Beweis vollständig ist.

Es seien zuerst $x \in X^*$ und $y \in X^*$ zwei Wörter mit $x \not\sim_{\mathcal{A}_\approx} y$. Dann gilt

$$\delta^*_\approx([z_0]_\approx, x) \neq \delta^*_\approx([z_0]_\approx, y).$$

Hieraus erhalten wir $[\delta^*(z_0, x)]_\approx \neq [\delta^*(z_0, y)]_\approx$ und deshalb $\delta^*(z_0, x) \not\approx \delta^*(z_0, y)$. Nach der Definition von $\approx$ heißt dies, dass es ein Wort $w \in X^*$ gibt, für das entweder

$$\delta^*(\delta^*(z_0, x), w) \in F \quad \text{und} \quad \delta^*(\delta^*(z_0, y), w) \notin F$$

oder

$$\delta^*(\delta^*(z_0, x), w) \notin F \quad \text{und} \quad \delta^*(\delta^*(z_0, y), w) \in F$$

erfüllt ist. Daher gilt entweder $\delta^*(z_0, xw) \in F$ und $\delta^*(z_0, yw) \notin F$ oder $\delta^*(z_0, xw) \notin F$ und $\delta^*(z_0, yw) \in F$ und folglich entweder $xw \in T(\mathcal{A})$ und $yw \notin T(\mathcal{A})$ oder $xw \notin T(\mathcal{A})$ und $yw \in T(\mathcal{A})$. Letzteres bedeutet aber gerade $x \not\sim_{T(\mathcal{A})} y$.

Analog beweist man, dass $x \sim_{\mathcal{A}_\approx} y$ für zwei Wörter $x \in X^*$ und $y \in X^*$ auch $x \sim_{T(\mathcal{A})} y$ zur Folge hat.

Somit stimmen die Äquivalenzrelationen $\sim_{\mathcal{A}_\approx}$ und $\sim_{T(\mathcal{A})}$ überein, was zu zeigen war.□

Leider geben die bisherigen Betrachtungen keine Hinweis, wie der Index von $\sim_R$ zu gegebenem R zu ermitteln ist, denn nach Definition von $\sim_R$ sind unendlich viele Wörter zu untersuchen, um $x \sim_R y$ festzustellen. Analog ist die Konstruktion von $\mathcal{A}_\approx$ nicht algorithmisch, denn auch die Feststellung, ob $x \approx y$ gilt, erfordert die Betrachtung von unendlich vielen Wörtern.

Deshalb beschreiben wir jetzt einen Algorithmus zum Bestimmen der Relation $\approx = \approx_{\mathcal{A}}$ für einen endlichen Automaten $\mathcal{A}$. Dies versetzt uns dann in die Lage, zu $\mathcal{A}$ den minimalen Automaten $\mathcal{A}_\approx$ zu konstruieren. Dazu beginnen wir mit einer Liste aller Paare von Zuständen und markieren jeweils ein Paar, wenn sich aus dem Verhalten der beiden Zustände des Paares entsprechend der Überführungsfunktion oder der Menge der akzeptierenden Zustände und der aktuellen Liste ergibt, dass die beiden Zustände des Paares nicht äquivalent sind.

Der *Reduktionsalgorithmus* besteht aus den folgenden Schritten:

1. Erstelle eine Liste aller Paare (z, z') von Zuständen $z \in Z$ und $z' \in Z$.

2. Streiche ein Paar (z, z') falls entweder $z \in F$ und $z' \notin F$ oder $z \notin F$ und $z' \in F$ gilt.

3. Führe die folgende Aktion solange aus, bis die Liste nicht mehr geändert wird: Falls für ein Paar (z, z') und ein $a \in X$ das Paar $(\delta(z, a), \delta(z', a))$ nicht in der Liste ist, so streiche (z, z').

Beispiel 4.8 *Wir betrachten den endlichen Automaten*

$$\mathcal{A} = (\{a, b\}, \{0, 1, 2, 3, 4, 5\}, 0, \delta, \{1, 2, 5\}),$$

dessen Überführungsfunktion δ dem Zustandsgraphen aus Abbildung 4.1 zu entnehmen ist. Wir erhalten entsprechend Schritt 1 des Algorithmus zuerst die Liste

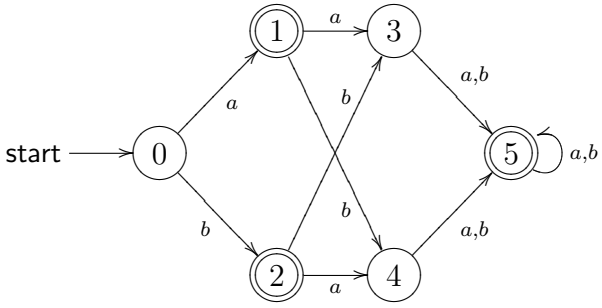

Abbildung 4.1: Zustandsgraph des Automaten $\mathcal{A}$ aus Beispiel 4.8

$$\begin{array}{llllll}
(0,0), & (0,1), & (0,2), & (0,3), & (0,4), & (0,5), \\
(1,0), & (1,1), & (1,2), & (1,3), & (1,4), & (1,5), \\
(2,0), & (2,1), & (2,2), & (2,3), & (2,4), & (2,5), \\
(3,0), & (3,1), & (3,2), & (3,3), & (3,4), & (3,5), \\
(4,0), & (4,1), & (4,2), & (4,3), & (4,4), & (4,5), \\
(5,0), & (5,1), & (5,2), & (5,3), & (5,4), & (5,5).
\end{array}$$

Hieraus entsteht nach Schritt 2 die Liste

$$\begin{array}{llllll}
(0,0), & (0,3), & (0,4), & (1,1), & (1,2), & (1,5), \\
(2,1), & (2,2), & (2,5), & (3,0), & (3,3), & (3,4), \\
(4,0), & (4,3), & (4,4), & (5,1), & (5,2), & (5,5).
\end{array}$$

Wir haben nun solange wie möglich Schritt 3 anzuwenden. Hierzu gehen wir stets die Liste durch und streichen die entsprechenden Paare und wiederholen diesen Prozess. Nach dem ersten Durchlauf streichen wir nur die Paare $(1, 5)$ (da $(\delta(1, a), \delta(5, a)) = (3, 5)$ nicht mehr in der Liste ist), $(2, 5)$, $(5, 1)$ und $(5, 2)$. Beim zweiten Durchlauf werden die Paare $(0, 3)$ (da das Paar $(\delta(0, a), \delta(3, a)) = (1, 5)$ beim ersten Durchlauf gerade gestrichen wurde), $(0, 4)$, $(3, 0)$ und $(4, 0)$ gestrichen. Beim dritten Durchlauf wird nichts mehr gestrichen. Damit ergibt sich abschließend die Liste

$$(0,0),\ (1,1),\ (1,2),\ (2,1),\ (2,2),\ (3,3),\ (3,4),\ (4,3),\ (4,4),\ (5,5).$$

Wir zeigen nun, dass mittels des oben angegebenen Algorithmus wirklich die Relation $\approx_{\mathcal{A}}$ berechnet wird. Dies ist im Wesentlichen die Aussage des folgenden Lemmas.

Lemma 4.9 *Es seien $\mathcal{A} = (X, Z, z_0, \delta, F)$ ein endlicher Automat sowie $z \in Z$ und $z' \in Z$ zwei Zustände des Automaten. Dann ist das Paar (z, z') genau dann in der durch den Reduktionsalgorithmus erzeugten Liste enthalten, wenn $z \approx_{\mathcal{A}} z'$ gilt.*

Beweis. Wir beweisen mittels vollständiger Induktion über die Anzahl der Schritte des Reduktionsalgorithmus die folgende äquivalente Aussage: Das Paar (z, z') wird genau dann durch den Reduktionsalgorithmus aus der Liste gestrichen, wenn es ein Wort $x \in X^*$ gibt, für das entweder $\delta^*(z, x) \in F$ und $\delta^*(z', x) \notin F$ oder $\delta^*(z, x) \notin F$ und $\delta^*(z', x) \in F$ gelten. Im Folgenden nehmen wir stets ohne Beschränkung der Allgemeinheit an, dass der erste dieser beiden Fälle eintritt.

Erfolgt ein Streichen des Paares (z, z') im zweiten Schritt, so hat wegen $\delta^*(z, \lambda) = z$ und $\delta^*(z', \lambda) = z'$ das leere Wort die gewünschte Eigenschaft. Hat umgekehrt das Leerwort die Eigenschaft, so wird das Paar im zweiten Schritt gestrichen.

Das Paar (z, z') werde nun im dritten Schritt gestrichen. Dann gibt es ein Element $a \in X$ so, dass das Paar $(\delta(z, a), \delta(z', a))$ bereits früher gestrichen wurde. Nach Induktionsannahme gibt es ein Wort $x \in X^*$ mit $\delta^*(\delta(z, a), x) \in F$ und $\delta^*(\delta(z', a), x) \notin F$. Damit haben wir auch $\delta^*(z, ax) \in F$ und $\delta^*(z', ax) \notin F$, womit die Behauptung bewiesen ist. Durch Umkehrung der Schlüsse zeigt man, dass aus der Existenz eines Wortes ax mit $\delta^*(z, ax) \in F$ und $\delta^*(z', ax) \notin F$ folgt, dass (z, z') gestrichen wird. □

Beispiel 4.8 (Fortsetzung) *Wir erhalten aus der bereits oben erzeugten Liste die folgenden Äquivalenzklassen bez. $\approx$:*

$$[0]_\approx = \{0\},\ [1]_\approx = [2]_\approx = \{1, 2\},\ [3]_\approx = [4]_\approx = \{3, 4\},\ [5]_\approx = \{5\}.$$

Hieraus resultiert nun entsprechend obiger Konstruktion der minimale Automat

$$\mathcal{A}_\approx = (X, \{[0]_\approx, [1]_\approx, [3]_\approx, [5]_\approx\}, [0]_\approx, \delta_\approx, \{[1]_\approx, [5]_\approx\}),$$

dessen Überführungsfunktion aus Abbildung 4.2, in der wir $[i]$ anstelle von $[i]_\approx$, $1 \le i \le 5$, schreiben, entnommen werden kann.

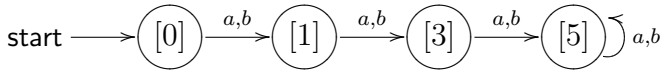

Abbildung 4.2: Zustandsgraph des minimalen Automaten zu $\mathcal{A}$ aus Beispiel 4.8

Wir bemerken, dass aus der Darstellung des minimalen Automaten sofort

$$T(\mathcal{A}) = T(\mathcal{A}_\approx) = X \cup X^3 X^*$$

zu sehen ist.

Wir untersuchen nun die Komplexität des Reduktionsalgorithmus. Es gibt n^2 Paare von Zuständen. Bei Schritt 2 oder jedem Durchlauf entsprechend Schritt 3 streichen wir ein Paar oder stoppen. Das Durchmustern einer Liste der Länge r erfordert $O(r)$ Schritte. Damit ist durch

$$\sum_{i=1}^{n^2} O(i) = O\left(\sum_{i=1}^{n^2} i\right) = O(n^4)$$

eine obere Schranke für die Komplexität gegeben.

Wir merken hier ohne Beweis an, dass es einen erheblich effizienteren Algorithmus zum Minimieren von endlichen Automaten gibt.

Satz 4.10 *Die Konstruktion des minimalen Automaten zu einem gegebenem endlichen Automaten mit n Zuständen ist in $O(n \cdot \log(n))$ Schritten möglich.* □

Wir wollen nun beweisen, dass der minimale Automat im Wesentlichen eindeutig bestimmt ist. Um das „im Wesentlichen" exakt zu fassen, geben wir die folgende Definition.

Definition 4.11 *Zwei Automaten $\mathcal{A} = (X, Z, z_0, \delta, F)$ und $\mathcal{A}' = (X, Z', z_0', \delta', F')$ heißen zu einander* isomorph*, wenn es eine eineindeutige Funktion φ von Z auf Z' mit folgenden Eigenschaften gibt:*

i) $\varphi(z_0) = z_0'$.

ii) *Für $z \in Z$ gilt $z \in F$ genau dann, wenn auch $\varphi(z) \in F'$ gilt.*

iii) *Für alle $z \in Z$ und $a \in X$ gilt $\delta'(\varphi(z), a) = \varphi(\delta(z, a))$.*

Intuitiv liegt Isomorphie zwischen zwei Automaten vor, wenn diese sich nur in der Bezeichnung der Zustände unterscheiden (und ansonsten das gleiche Verhalten zeigen).

Satz 4.12 *Sind $\mathcal{A}$ und $\mathcal{A}'$ zwei minimale Automaten für eine reguläre Menge R, so sind $\mathcal{A}$ und $\mathcal{A}'$ zu einander isomorph.*

Beweis. Es seien $\mathcal{A} = (X, Z, z_0, \delta, F)$ und $\mathcal{A}' = (X, Z', z_0', \delta', F')$ zwei minimale Automaten für R.

Für zwei Zustände $z \in Z$ und $z' \in Z$ von $\mathcal{A}$ ist $z \not\approx_{\mathcal{A}} z'$. Dies folgt daraus, dass aufgrund der Minimalität von $\mathcal{A}$ der Automat $\mathcal{A}_{\approx}$ genauso viel Zustände wie $\mathcal{A}$ hat. Also können keine zwei Zustände von $\mathcal{A}$ in einer Äquivalenzklasse bez. $\approx_{\mathcal{A}}$ liegen. Nun beweist man analog zum letzten Teil des Beweises von Satz 4.7, dass die Relationen $\sim_{\mathcal{A}}$ und $\sim_R$ übereinstimmen.

Entsprechend ergibt sich auch $\sim_{\mathcal{A}'} = \sim_R$ und damit $\sim_{\mathcal{A}} = \sim_{\mathcal{A}'}$.

Es sei nun $z \in Z$. Dann gibt es ein Wort $x \in X^*$ mit $\delta^*(z_0, x) = z$. Wir setzen

$$\varphi(z) = (\delta')^*(z_0', x).$$

Dies liefert eine zulässige Definition von φ, denn der Wert von $\varphi(z)$ hängt nicht von der Wahl von x ab. Sind nämlich $x \in X^*$ und $y \in X^*$ mit $\delta^*(z_0, x) = \delta^*(z_0, y)$, so ergibt sich zuerst $x \sim_{\mathcal{A}} y$ und damit auch $x \sim_{\mathcal{A}'} y$, woraus $(\delta')^*(z_0', x) = (\delta')^*(z_0', y)$ folgt.

Wir zeigen, dass φ ein Isomorphismus ist.
Wegen $z_0 = \delta^*(z_0, \lambda)$ ergibt sich $\varphi(z_0) = (\delta')^*(z_0', \lambda) = z_0'$.
Es sei $z \in F$. Dann gibt es ein Wort $x \in R$ mit $z = \delta^*(z_0, x)$ und es gilt $\varphi(z) = (\delta')^*(z_0', x)$. Wegen $x \in R$ und $R = T(\mathcal{A}')$ erhalten wir $\varphi(z) \in F'$. Es sei umgekehrt $\varphi(z) \in F'$. Dann liegt x mit $\varphi(z) = (\delta')^*(z_0', x)$ in R und somit ist $z = \delta(z_0, x) \in F$.

Außerdem gilt

$$\begin{aligned}\delta'(\varphi(z), a) &= \delta'((\delta')^*(z_0', x), a) = (\delta')^*(z_0', xa) \\ &= \varphi(\delta^*(z_0, xa)) = \varphi(\delta(\delta^*(z_0, x), a)) \\ &= \varphi(\delta(z, a)).\end{aligned}$$

□

Minimale deterministische endliche Automaten sind also bis auf „Umbenennen“ der Zustände eindeutig bestimmt. Dies gestattet uns, von *dem minimalen Automaten* für eine Sprache zu sprechen.

Abschließend merken wir an, dass bei nichtdeterministischen endlichen Automaten ein minimaler Automat nicht eindeutig bestimmt ist.

Literaturverzeichnis

[1] J. Dassow, Theoretische Informatik. Vorlesungsskript, 2000.

[2] W. Dörfler, *Mathematik für Informatiker*. Hanser-Verlag München, 1977.

[3] M. R. Garey und D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[4] S. B. Gaskov, The depth of Boolean functions. *Probl. Kibernetiki* **34** (1978) 265–268.

[5] J. Gruska, On a classification of context-free languages. ßem Kybernetika **3** (1967) 22–29.

[6] J. Gruska, Some classifications of context-free languages. *Information and Control* **14** (1969) 152–179.

[7] J. Gruska, Complexity and unambiguity of context-free grammars and languages. *Information and Control* **18** (1971) 502–519.

[8] J. Gruska, Descriptional complexity (of languages) – a short survey. In: *Mathematical Foundations of Computer Science 1976* (Hrsg. A. Mazurkiewicz), Lecture Notes in Computer Science 45, Springer-Verlag, 1976, 65–80.

[9] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Adddison Wesley, Reading, Massachusetts, 1979.

[10] John E. Hopcroft and Jeffrey D. Ullman. *Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie*. Addison Wesley, Reading, Bonn, 1990.

[11] S. W. Jablonski, Einführung in die Theorie der Funktionen der k-wertigen Logik. In: *Diskrete Mathematik und mathematische Fragen der Kybernetik* (Hrsg. S. W. Jablonski und O. B. Lupanov), Akademie-Verlag, 1980.

[12] D. C. Kozen, *Automata and Computability*. Springer-Verlag, 1997.

[13] O. B. Lupanov, Complexity of formulae realisation of functions of logical algebra. *Probl. Kibernetiki* **3** (1962) 782–811.

[14] O. B. Lupanov, A method of synthesis of control systems. *Probl. Kibernetiki* **23** (1970) 31–110.

[15] W. F. McColl und M. Paterson, The depth of all Boolean functions. *SIAM J. Comp.* **6** (1977) 373–380.

[16] T. Ottmann und P. Widmayer, *Algorithmen und Datenstrukturen.* B.I.-Wissenschaftsverlag, Mannheim, 1993.

[17] B. Reichel, Some classifications of Indian parallel languages. *J. Inf. Process. Cybern.* (EIK) **26** (1990) 85–99.

[18] U. Schöning, *Theoretische Informatik kurz gefaßt.* B.I.-Wissenschaftsverlag, 1992.

[19] R. Sedgewick, *Algorithmen.* Addison-Wesley, 1990.

[20] K. Wagner, *Einführung in die Theoretische Informatik.* Springer-Verlag, 1994.

[21] I. Wegener, *The Complexity of Boolean Functions.* Teubner, Stuttgart und Wiley, New York, 1987.

[22] I. Wegener, *Theoretische Informatik.* Teubner-Verlag, 1993.