

Prof. Dr. Jürgen Dassow

und

Dr. Bianca Truthe

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

**FORMALE MODELLE FÜR
NATÜRLICHE SPRACHEN**

Vorlesungsmanuskript

Magdeburg, Oktober 2007 – Januar 2008

Inhaltsverzeichnis

Einleitung	1
1. Modelle auf der Basis von verallgemeinerten Regelgrammatiken	5
1.1. Kontextfreie Grammatiken – Rückblick und Ergänzungen	5
1.2. Schwach kontextabhängige Grammatikklassen	20
1.3. Indizierte Grammatiken	23
1.3.1. Definition, Beispiele und Eigenschaften	23
1.3.2. Lineare indizierte Grammatiken	31
1.4. Baum einfügende Grammatiken	37
1.4.1. Definition, Beispiele und Eigenschaften	37
1.4.2. Lexikalische Grammatiken	46
1.5. Kopf-Grammatiken	50
1.6. Beziehungen zwischen den Grammatikklassen	56
2. Einfügende Grammatiken und streichende Automaten	73
2.1. Kontextuale Grammatiken	73
2.1.1. Allgemeine kontextuale Grammatiken	73
2.1.2. Lokale und maximal lokale kontextuale Grammatiken	91
2.2. Automaten mit Neustart	97
2.2.1. Definitionen und Beispiele	98
2.2.2. Hierarchieresultate	109
Literatur	119

Einleitung

Für das automatische Verarbeiten von Texten ist es erforderlich, dass man für eine Vielzahl von linguistischen Problemen und Aufgaben Algorithmen zu deren Lösung zur Verfügung hat. Dazu gehören unter anderem festzustellen, ob ein gegebener Satz syntaktisch richtig ist, aus gegebenen (z. B. durch Übersetzung erhaltenen) Wörtern einen syntaktisch richtigen Satz zusammensetzen, das Prädikat eines Satzes zu erkennen usw. Um die Algorithmen formulieren zu können, ist es erforderlich, Strukturen zu entwickeln, die die Syntax von Sätzen formalisieren.

Ein solcher Ansatz wurde Ende der fünfziger Jahre von NOAM CHOMSKY entwickelt (siehe z. B. [3, 2]). Er basiert darauf, ein Modell für die Grammatik einer natürlichen Sprache zu entwickeln. Dabei wird der natürliche Ansatz verfolgt, dass ein Satz aus Satzteilen besteht, und dass es Regeln gibt, welche Satzteile an welchen Stellen verwendet werden dürfen. So gilt z. B. im Englischen, dass in der Regel in einem Satz die Reihenfolge *Subjekt Prädikat Objekt* einzuhalten ist. Als Prädikat kann hierbei ein einfaches *Verb* (in der entsprechenden konjugierten Form) oder aber auch eine Wortgruppe *Verb Adverb* fungieren. Folglich sind Regeln aufzustellen, wodurch ein Prädikat ersetzt werden darf. Um wirkliche Sätze der Sprache zu erhalten, müssen am Ende jedes *Substantiv* durch ein solches der Sprache, jedes *Verb* durch ein solches der Sprache usw. ersetzt werden. Von der rein syntaktischen Betrachtungsweise reicht es, die durch eine Folge wie z. B.

Artikel Substantiv Verb Adverb Artikel Substantiv

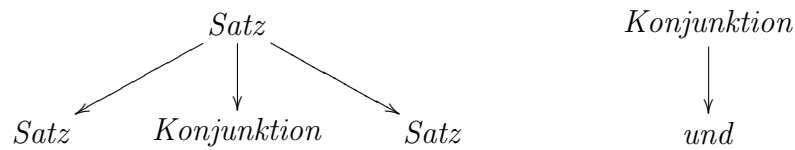
entstehenden Gebilde zu betrachten.

Der Roman *Der alte Mann und das Meer* von *Ernest Hemingway* beginnt in der Übersetzung von Annemarie Horschitz-Horst (erschienen im Verlag Philipp Reclam jun. Leipzig, 1973) wie folgt:

Er war ein alter Mann, der allein in einem kleinen Boot im Golfstrom fischte, und er war jetzt vierundachtzig Tage hintereinander hinausgefahren, ohne einen Fisch zu fangen. In den ersten vierzig Tagen hatte er einen Jungen bei sich gehabt. Aber nach vierzig fischlosen Tagen hatten die Eltern des Jungen ihm gesagt, daß der alte Mann jetzt bestimmt für immer salao sei, was die schlimmste Form von Pechhaben ist, und der Junge war auf ihr Geheiß in einem anderen Boot mitgefahren, das in der erste Woche drei gute Fische gefangen hatte. ...

Wir wollen nun den ersten Satz dieses Textes entsprechend dem Vorgehen von CHOMSKY generieren. Zuerst stellen wir fest, dass sich dieser in seiner Gänze als aus zwei Teilsätzen bestehend erweist, die durch die Konjunktion *und* verbunden sind. Dies wird durch

die Regeln



widergespiegelt. Wir generieren nun nur noch den ersten Teilsatz; für den zweiten Teilsatz ergibt sich ein ähnliches Vorgehen. Unter Verwendung der Abkürzungen

<i>S</i> für Satz	<i>RS</i> für Relativsatz
<i>NP</i> für Nominalphrase	<i>VP</i> für Verbphrase
<i>Sub</i> für Substantiv	<i>Art</i> für Artikel
<i>Pr</i> für Präposition	<i>Adj</i> für Adjektiv
<i>Adv</i> für Adverb	<i>PP</i> für Personalpronomen
<i>OB</i> für Ortsbestimmung	

ergibt sich der in Abbildung 1 angegebene Ableitungsbaum.

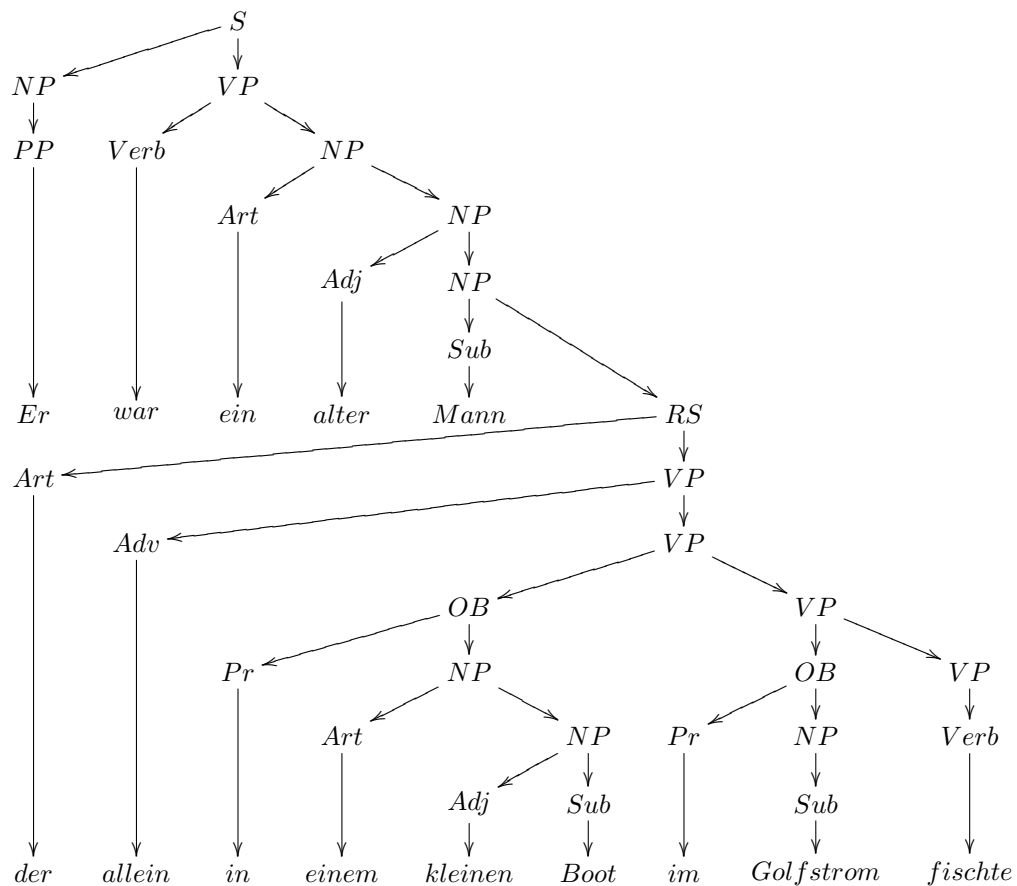


Abbildung 1: Ableitungsbaum eines Satzes aus *Der alte Mann und das Meer* von Ernest Hemingway

CHOMSKY formalisierte diesen Ansatz, indem er formale Grammatiken definierte und diese nach der Art der Regeln klassifizierte. Dabei stellten sich die sogenannten kontextfreien Grammatiken als die hinsichtlich ihrer Handhabbarkeit interessantesten heraus.

Unglücklicherweise erwiesen sie sich aber als zu schwach, um einige in der Linguistik bekannte Phänomene zu modellieren. Daher formulierte ARAVIND K. JOSHI Anforderungen an Erweiterungen der kontextfreien Grammatiken/Sprachen, die von brauchbaren Modellen für natürliche Sprachen erfüllt sein sollten. Im Rahmen des ersten Teils der Vorlesung werden einige Varianten derartiger Modelle behandelt.

Ein gänzlich anderer Ansatz zur Beschreibung natürlicher Sprachen geht auf SOLOMON MARCUS zurück (siehe [13, 14]). Ihm liegt die Beobachtung zugrunde, dass man bei der Synthese von Sätzen an gewissen Stellen oder bei gewissen Kontexten Einschiebungen vornehmen kann und dadurch einen längeren Satz erhält. Ein einfaches Beispiel ist dadurch gegeben, dass man zwischen einem Artikel und einem Substantiv ein Adjektiv einfügen kann bzw. an ein Substantiv einen Relativsatz anhängt. Ausgangspunkt sind dabei kurze korrekte Sätze oder besser Folgen von Wortarten, die zu Sätzen werden, wenn man jede Wortart durch ein passendes Wort ersetzt. Wir illustrieren diese Methode auch anhand des ersten Teilsatzes aus *Der alte Mann und das Meer* von *Ernest Hemingway*. Es ergibt sich mit den obigen Abkürzungen die Abfolge von Einschüben aus Abbildung 2.

PP	Verb	Art		Sub									
PP	Verb	Art	Adj	Sub									
PP	Verb	Art	Adj	Sub	Art								Verb
PP	Verb	Art	Adj	Sub	Art	Adv							Verb
PP	Verb	Art	Adj	Sub	Art	Adv	Pr	Art		Sub			Verb
PP	Verb	Art	Adj	Sub	Art	Adv	Pr	Art	Adj	Sub			Verb
PP	Verb	Art	Adj	Sub	Art	Adv	Pr	Art	Adj	Sub	Pr	Sub	Verb
Er	war	ein	alter	Mann	der	allein	in	einem	kleinen	Boot	im	Golfstrom	fischte

Abbildung 2: Erzeugung eines Satzes aus *Der alte Mann und das Meer* von *Ernest Hemingway* durch Einschübe

Ein ähnliches Vorgehen liegt bei Neustart-Automaten vor, bei denen in gleicher Weise eine Analyse vorgenommen wird, d. h. aus bestehenden Sätzen werden gewisse Teile entfernt, ohne die Satzstruktur zu verletzen. Diese beiden Ansätze werden im zweiten Teil der Vorlesung behandelt.

Darüber hinaus werden noch einige Grammatiken vorgestellt, die zur Beschreibung natürlicher Sprachen eingesetzt werden können, wie z. B. Transformationsgrammatiken, und einige generelle zu berücksichtigende Aspekte für die Modellierung natürlicher Sprachen erörtert.

Im Rahmen dieser Vorlesung gehen wir davon aus, dass der Hörer (oder Leser dieses Skripts) über Grundkenntnisse der Theorie formaler Sprachen verfügt, wie sie etwa im Rahmen einer Grundvorlesung zur Theoretischen Informatik vermittelt werden. Dies betrifft insbesondere Kenntnisse über reguläre und kontextfreie Sprachen sowie Begriffe der Unentscheidbarkeit oder NP-Vollständigkeit von Problemen. Im Abschnitt 1.1. werden einige wenige Konzepte wiederholt, um die Notation zu klären und weil auf diese Fakten direkt zurückgegriffen werden wird. Wir verweisen den Hörer/Leser zur Orientierung auf die Standardwerke [20, 7, 22, 28, 29].

Kapitel 1

Modelle auf der Basis von verallgemeinerten Regelgrammatiken

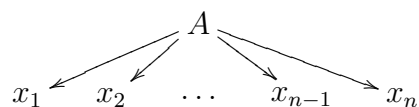
1.1. Kontextfreie Grammatiken – Rückblick und Ergänzungen

Wir wiederholen zuerst einige Begriffe, die bereits im Rahmen der Grundvorlesung zur Theoretischen Informatik eingeführt wurden. Sie werden hier noch einmal behandelt, weil damit zum einen die Notation festgelegt werden soll, zum anderen sollen die aus der Sicht der Modellierung natürlicher Sprachen teilweise notwendigen Modifizierungen vorgenommen werden, und des Weiteren sollen einige ergänzende Definitionen und Resultate angegeben werden, die im Folgenden benutzt werden.

Unter einem Alphabet verstehen wir eine endliche nicht-leere Menge. Die Elemente eines Alphabets heißen Buchstaben. Wörter über einem Alphabet V sind endliche Folgen von Buchstaben aus V . Dabei ist das Alphabet stets so, dass kein zusammengesetztes Wort gleichzeitig ein Buchstabe ist. Das leere Wort wird mit λ bezeichnet (auch dieses Wort tritt nicht als Buchstabe auf); V^* und V^+ bezeichnen die Menge aller bzw. aller nicht-leeren Wörter über V . Die Länge eines Wortes w wird mit $|w|$ bezeichnet; $\#_U(w)$ gibt die Anzahl der Vorkommen von Buchstaben aus $U \subseteq V$ in einem Wort $w \in V^*$ an.

Definition 1.1 *Eine kontextfreie Regelgrammatik 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 Bäumen der Form



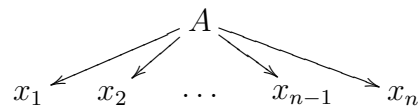
mit $A \in N$ und $x_i \in V$ für $1 \leq i \leq n$ ist, und

- $S \in N$ gilt.

Die Elemente aus N werden Nichtterminale (oder Variable) genannt; sie entsprechen grammatikalischen Objekten, die keine Wörter bezeichnen, also z. B. Nominalphrase, Verbphrase, Ortsbestimmung usw. Die Elemente aus T heißen Terminale; sie entsprechen den syntaktisch nicht mehr spezifizierbaren Bezeichnungen von Wörtern, wie z. B. Substantiv, Verb, Artikel usw. Die Terminale sind aus Sicht der Linguistik nur noch durch konkrete Wörter der Sprache zu interpretieren.

Definition 1.2 *Es sei $G = (N, T, P, S)$ eine kontextfreie Grammatik entsprechend Definition 1.1.*

- a) Wir definieren Ableitungsbäume der Grammatik G induktiv durch die folgenden Bedingungen.
- i) Der Baum, der nur aus der Wurzel S besteht (die dann auch zugleich Blatt ist), ist ein Ableitungsbaum von G .
 - ii) Sind t ein Ableitungsbaum (mit der Wurzel S), $A \in N$ ein Blatt von t und

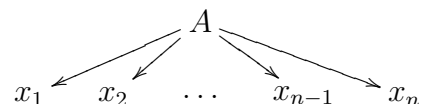


eine Regel von P , so ist auch der Baum, der aus t entsteht, indem man die Wurzel A der Regel mit dem Blatt A von t identifiziert, ein Ableitungsbaum von G .

- iii) Ableitungsbäume von G entstehen nur aufgrund endlich oftmaliger Anwendung von ii) aus i).
- b) Ein Wort $w \in V^+$ heißt Satzform von G , falls es einen Ableitungsbaum t von G derart gibt, dass sich w durch Lesen der Blätter von t von links nach rechts ergibt.
- c) Die von G erzeugte Sprache $L(G)$ ist die Menge aller Satzformen von G , die nur aus Elementen aus T bestehen.

Die Komponente S entspricht daher dem Satz in der Linguistik; S wird auch als Axiom oder Startelement bezeichnet. Ableitungsbäume widerspiegeln die Erzeugung von Sätzen entsprechend dem Ansatz von CHOMSKY. Die Sprache wird aus allen Folgen von Terminalen (also Wortbezeichnungen) gebildet, die durch Lesen der Blätter eines Ableitungsbauums von links nach rechts entstehen. Die Sätze der natürlichen Sprachen entstehen hieraus, indem jedes *Verb* durch ein konkretes Verb der Sprache, jedes *Substantiv* durch ein konkretes Substantiv usw. ersetzt wird.

Sind t ein Ableitungsbaum mit einem Blatt $A \in N$,



eine Regel und t' der durch Anwendung der Regel aus t entstehende Ableitungsbaum, so ergibt sich aus der zu t gehörenden Satzform $y = y_1 A y_2$ die zu t' gehörende Satzform $y' = y_1 x_1 x_2 \dots x_n y_2$. Hinsichtlich der Satzformen wird also einfach eine Ersetzung von A durch $x_1 x_2 \dots x_n$ vorgenommen. Dies ermöglicht eine Verallgemeinerung dahingehend, dass wir entsprechend den Regeln Teilwörter von y durch andere Wörter ersetzen. Regeln sind also dann von der Gestalt $\alpha \rightarrow \beta$ (wobei der Pfeil die durch die Regel mögliche Ersetzung andeuten soll). Hierbei entstehen dann aber keine Ableitungsbäume mehr,

da nicht ein einzelnes Blatt von t sondern mehrere Blätter von t berücksichtigt werden müssen.

Wir formalisieren nun den von dieser Beobachtung ausgehenden Sachverhalt.

Definition 1.3 Eine Regelgrammatik 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 $(V^+ \setminus T^*) \times V^+$ ist, und
- $S \in N$ gilt.

Definition 1.4 Es sei $G = (N, T, P, S)$ eine Regelgrammatik, wie in Definition 1.3 beschrieben.

a) Wir sagen, dass aus einem Wort $\gamma \in V^+$ ein Wort $\gamma' \in V^*$ erzeugt (oder abgeleitet) wird, wenn

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

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

b) Mit \Longrightarrow^* bezeichnen wir den reflexiven und transitiven Abschluss der Relation \Longrightarrow .

c) Ein Wort $w \in V^*$ heißt Satzform von G , wenn $S \Longrightarrow^* w$ gilt.

d) Die von G erzeugte Sprache $L(G)$ ist durch

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

definiert.

Definition 1.5 Es sei $G = (N, T, P, S)$ eine Regelgrammatik wie in Definition 1.3. Wir sagen,

- G ist monoton, wenn alle Regeln $\alpha \rightarrow \beta$ der Bedingung $|\alpha| \leq |\beta|$ genügen,
- G ist kontextabhängig, wenn alle Regeln in P von der Form $uAv \rightarrow uvw$ mit $u, v \in V^*$, $A \in N$ und $w \in V^+$ sind,
- G ist kontextfrei, wenn alle Regeln in P von der Form $A \rightarrow w$ mit $A \in N$ und $w \in V^+$ sind,
- G ist linear, wenn alle Regeln in P von der Form $A \rightarrow uBv$ oder $A \rightarrow w$ mit $A, B \in N$, $uv \in T^+$ und $w \in T^+$ sind,
- G ist regulär, wenn alle Regeln in P von der Form $A \rightarrow wB$ oder $A \rightarrow w$ mit $A, B \in N$ und $w \in T^+$ sind.

In der Literatur ist es manchmal üblich, zu gestatten, dass bei monotonen und kontextabhängigen Grammatiken auch $S \rightarrow \lambda$ als Regel zugelassen ist, falls S in keiner rechten Seite von Regeln in P vorkommt. Hierdurch wird aber nur abgesichert, dass das Leerwort erzeugt werden kann. Da das Leerwort in der Linguistik unbedeutend ist, benötigen wir keine Ableitungsmöglichkeit für das Leerwort und lassen daher die obige zusätzliche Regel nicht zu. Vielfach ist es üblich, bei kontextfreien Grammatiken nur zu fordern, dass

$w \in T^*$ gilt. Es sind daher löschende Regeln $A \rightarrow \lambda$ zugelassen. Es kann dann gezeigt werden (und wird auch meist nachgewiesen), dass es zu jeder kontextfreien Grammatik G mit löschenden Regeln eine kontextfreie Grammatik G' derart gibt, dass $L(G') = L(G) \setminus \{\lambda\}$ gilt (siehe z. B. [7]), d. h. die beiden Sprachen unterscheiden sich höchstens im Leerwort. Da das Leerwort für linguistische Belange nicht von Bedeutung ist, betrachten wir hier nur kontextfreie Grammatiken ohne löschende Regeln. Da überdies beim Übergang von Grammatiken mit löschenden Regeln zu solchen ohne löschenden Regeln die Form der Regeln bei linearen und regulären Grammatiken nicht geändert wird, gilt der eben erwähnte Sachverhalt auch für lineare und reguläre Grammatiken.

Wir bemerken, dass für kontextfreie Grammatiken die Begriffe aus den beiden Ansätzen leicht ineinander überführbar sind, und dabei die gleiche Sprache entsteht.

Entsprechend Definition 1.5 haben die Ableitungsbäume regulärer und linearer Grammatiken die Form aus Abbildung 1.1, wobei wir der Einfachheit halber nur Regeln der Form $A \rightarrow aB$ und $A \rightarrow a$ bzw. $A \rightarrow aBb$ und $A \rightarrow a$ mit $A, B \in N$ und $a, b \in T$ verwenden.

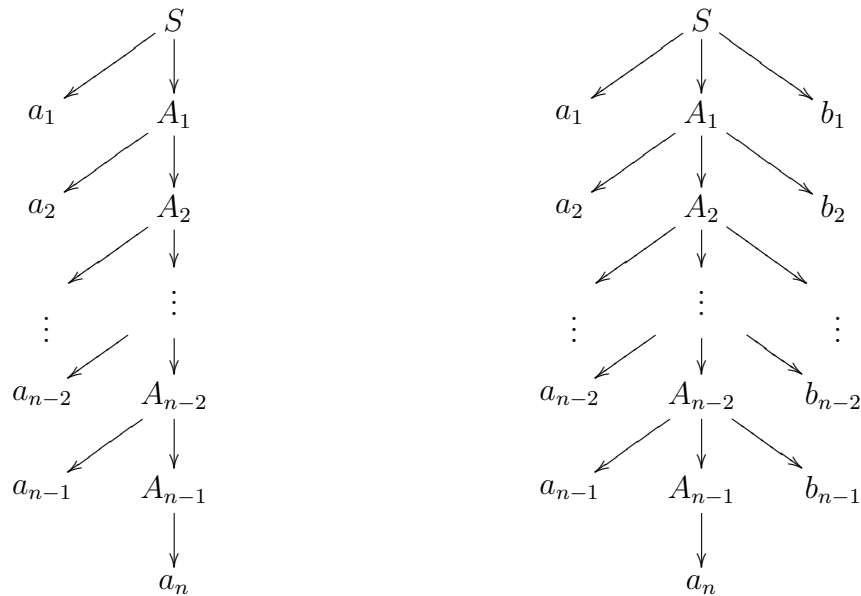


Abbildung 1.1: Ableitungsbäume für reguläre Grammatiken (links) und lineare Grammatiken (rechts)

Definition 1.6 Eine Sprache L heißt regulär (linear, kontextfrei, kontextabhängig oder monoton), wenn es eine reguläre (lineare, kontextfreie, kontextabhängige bzw. monotone) Grammatik G mit $L = L(G)$ gibt.

Mit $\mathcal{L}(REG)$, $\mathcal{L}(LIN)$, $\mathcal{L}(CF)$, $\mathcal{L}(CS)$, $\mathcal{L}(MON)$ und $\mathcal{L}(RE)$ bezeichnen wir die Menge aller Sprachen, die von regulären, linearen, kontextfreien, kontextabhängigen, monotonen bzw. beliebigen Regelgrammatiken erzeugt werden.

Satz 1.7 (Pumping-Lemmata)

- a) Es sei L eine reguläre Sprache. Dann gibt es eine (von L abhängige) Konstante k derart, dass es zu jedem Wort $z \in L$ mit $|z| \geq k$ Wörter u, v, w gibt, die den folgenden Eigenschaften genügen:
- i) $z = uvw$,
 - ii) $|uv| \leq k$, $|v| > 0$, und
 - iii) $uv^i w \in L$ für alle $i \geq 0$.
- b) Es sei L eine lineare Sprache. Dann gibt es eine (von L abhängige) Konstante k derart, dass es zu jedem Wort $z \in L$ mit $|z| \geq k$ Wörter u, v, w, x, y gibt, die den folgenden Eigenschaften genügen:
- i) $z = uvwxy$,
 - ii) $|uvxy| \leq k$, $|vx| > 0$, und
 - iii) $uv^i wx^i y \in L$ für alle $i \geq 0$.
- c) Es sei L eine kontextfreie Sprache. Dann gibt es eine (von L abhängige) Konstante k derart, dass es zu jedem Wort $z \in L$ mit $|z| \geq k$ Wörter u, v, w, x, y gibt, die den folgenden Eigenschaften genügen:
- i) $z = uvwxy$,
 - ii) $|vwx| \leq k$, $|vx| > 0$, und
 - iii) $uv^i wx^i y \in L$ für alle $i \geq 0$.

Beweis. Wir beweisen nur die Aussage b), da a) und c) schon in der Vorlesung zur Theoretischen Informatik angegeben und bewiesen wurden.

Es sei L eine lineare Sprache. Dann gibt es eine lineare Grammatik $G = (N, T, P, S)$ mit $L = L(G)$. Wir setzen

$$\begin{aligned} k_1 &= \#(N), \\ k_2 &= \max\{\max\{|uv| \mid A \rightarrow uBv \in P\}, \max\{|w| \mid A \rightarrow w \in P\}\}, \\ k &= (k_1 + 2)k_2. \end{aligned}$$

Es gelte nun $z \in L$ und $|z| > k$. Dann muss z aus S durch mindestens $k_1 + 2$ Ableitungsschritte entstanden sein, da jeder Ableitungsschritt maximal k_2 zur Länge beiträgt. Es gilt also

$$\begin{aligned} S &\xrightarrow{*} u_1 S_1 v_1 \xrightarrow{*} u_1 u_2 S_2 v_2 v_1 \xrightarrow{*} \cdots \xrightarrow{*} u_1 u_2 \dots u_{k_1+1} S_{k_1+1} v_{k_1+1} v_{k_1} \dots v_1 \\ &\xrightarrow{*} u_1 u_2 \dots u_{k_1+1} z' v_{k_1+1} v_{k_1} \dots v_1 = z \end{aligned}$$

für gewisse Wörter $u_i, v_i, z' \in (N \cup T)^*$ mit $0 < |u_i v_i| \leq k_2$ und Nichtterminale S_i für $1 \leq i \leq k_1 + 1$. Da die Menge N nur k_1 Nichtterminale enthält, muss es Zahlen i und j mit $1 \leq i < j \leq k_1 + 1$ und $S_i = S_j$ geben. Damit gibt es mit

$$\begin{aligned} u &= u_1 u_2 \dots u_i, \quad y = v_i v_{i-1} \dots v_1, \\ v &= u_{i+1} u_{i+2} \dots u_j, \quad x = v_j v_{j-1} \dots v_{i+1}, \\ w &= u_{j+1} u_{j+2} \dots u_{k_1+1} z' v_{k_1+1} v_k \dots v_{j+1} \end{aligned}$$

auch die Ableitungen

$$\begin{aligned} S &\xrightarrow{*} u S_i y \xrightarrow{*} uv S_j x y = uv S_i x y \xrightarrow{*} uvv S_j x x y = uvv S_i x x y \\ &\xrightarrow{*} uv^p S_j x^p y = uv^p w x^p y. \end{aligned}$$

Beachten wir noch, dass nach Wahl von k die Beziehung

$$|uvxy| = |u_1u_2 \dots u_jv_jv_{j-1} \dots v_1| \leq jk_2 < (k_1 + 2)k_2 = k$$

gilt, so ist auch die Aussage für lineare Sprachen gezeigt. \square

Folgerung 1.8 Die Sprache $L = \{ a^n b^n a^m b^m \mid n \geq 1, m \geq 1 \}$ ist keine lineare Sprache.

Beweis. Angenommen, L wäre eine lineare Sprache. Es sei k die nach Satz 1.7 b) existierende Konstante. Wir betrachten das Wort $z = a^{2k} b^{2k} a^{2k} b^{2k} \in L$, dessen Länge $8k$ offenbar größer als k ist. Damit gibt es nach Satz 1.7 b) eine Zerlegung $z = uvwxy$ derart, dass $|uvxy| < k$ ist und für $i \geq 0$ auch uv^iwx^iy in L liegen. Wegen der Längenbeschränkung von $uvxy$ enthalten die Wörter u und v nur den Buchstaben a und die Wörter y und x nur den Buchstaben b , d. h. wir haben $u = a^r$, $v = a^t$, $x = b^p$ und $y = b^q$ für gewisse natürliche Zahlen r, t, p, q , wobei noch $t > 0$ oder $p > 0$ gilt, und $z = a^r a^t a^{2k-r-t} b^{2k} a^{2k} b^{2k-p-q} b^p b^q$. Es ist damit auch

$$z' = a^r a^{2t} a^{2k-r-t} b^{2k} a^{2k} b^{2k-p-q} b^{2p} b^q = a^{2k+t} b^{2k} a^{2k} b^{2k+p} \in L.$$

Dies widerspricht aber der Form der Wörter aus L . \square

Satz 1.9 $\mathcal{L}(REG) \subset \mathcal{L}(LIN) \subset \mathcal{L}(CF) \subset \mathcal{L}(CS) = \mathcal{L}(MON) \subset \mathcal{L}(RE)$.

Beweis. Wir zeigen nur $\mathcal{L}(REG) \subset \mathcal{L}(LIN)$ und $\mathcal{L}(LIN) \subset \mathcal{L}(CF)$, da die anderen echten Inklusionen und die Gleichheit Bestandteil der aus der Vorlesung zur Theoretischen Informatik bekannten Chomsky-Hierarchie sind.

$\mathcal{L}(REG) \subset \mathcal{L}(LIN)$. Die Inklusion folgt aus der Definition der entsprechenden Grammatiken und Sprachen. Es ist bekannt, dass $\{ a^n b^n \mid n \geq 1 \}$ keine reguläre Sprache ist, sie wird aber von der linearen Grammatik $(\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow ab\}, S)$ erzeugt. Damit ist die Inklusion sogar echt.

$\mathcal{L}(LIN) \subset \mathcal{L}(CF)$. Die Inklusion folgt aus der Definition der entsprechenden Grammatiken und Sprachen. Wegen Folgerung 1.8 ist $\{ a^n b^n a^m b^m \mid n \geq 1, m \geq 1 \}$ keine lineare Sprache; sie wird aber von der kontextfreien Grammatik

$$(\{S, A\}, \{a, b\}, \{S \rightarrow AA, A \rightarrow aAb, A \rightarrow ab\}, S)$$

erzeugt. Damit ist die Inklusion sogar echt. \square

Folgerung 1.10 Die Sprachen

$$\begin{aligned} K_1 &= \{ a^n b^n c^n \mid n \geq 1 \}, \\ K_2 &= \{ a^n c^m b^n d^m \mid n \geq 1, m \geq 1 \}, \\ K_3 &= \{ ww \mid w \in \{a, b\}^+ \} \end{aligned}$$

sind nicht kontextfrei.

Beweis. Für K_1 und K_3 wurde die Aussage bereits im Rahmen der Vorlesung und Übungen zur Theoretischen Informatik gezeigt.

Angenommen, K_2 ist eine kontextfreie Sprache. Es sei k die nach Satz 1.7 c) existierende Konstante. Wir betrachten das Wort $z = a^{2k}c^{2k}b^{2k}d^{2k}$ der Länge $8k > k$. Ferner sei $z = uvwxy$ die nach Satz 1.7 c) existierende Zerlegung von z . Falls $v \in \{a\}^+$ gilt, so kommt in vw kein b vor. Folglich enthält uv^2wx^2y mehr Vorkommen des Buchstabens a als Vorkommen des Buchstabens b , was der Eigenschaft iii) aus Satz 1.7 c) widerspricht. In analoger Weise können wir einen Widerspruch für die anderen Fälle für v herleiten. Damit ist dann unsere Annahme als falsch nachgewiesen, d. h., dass K_2 nicht kontextfrei ist. \square

Die nächste Folgerung besagt, dass von einer Stelle an hinsichtlich der Wortlänge in einer kontextfreien Sprache nur noch Lücken von beschränkter Größe auftauchen.

Folgerung 1.11 *Es sei L eine kontextfreie Sprache. Dann gibt es Konstanten k und k' derart, dass zu jedem Wort $z \in L$ mit $|z| > k$ ein Wort z' in der Sprache L so existiert, dass*

$$|z| < |z'| < |z| + k'$$

gilt.

Beweis. Es seien L eine kontextfreie Sprache und k die nach Satz 1.7 c) existierende Konstante. Wir setzen $k' = k + 1$. Für ein Wort z , dessen Länge größer als k ist, gibt es nach Satz 1.7 c) eine Zerlegung $z = uvwxy$ mit $|vw| \leq k$, $|vx| > 0$ und $z' = uv^2wx^2y \in L$. Offensichtlich gilt

$$|z| < |z| + |vx| = |z'| \leq |z| + |vw| \leq |z| + k < |z| + k',$$

womit die Behauptung gezeigt ist. \square

Wir wollen die „Dichtheit“ bezüglich der (eindimensionalen) Länge noch etwas verfeinern, indem wir statt der Länge den (mehrdimensionalen) Parikh-Vektor eines Wortes betrachten. Dazu geben wir zuerst den Begriff einer semi-linearen Menge in einem linearen Vektorraum über den reellen Zahlen an.

Definition 1.12

- i) Eine Teilmenge M des \mathbb{R}^n heißt semi-affin, falls es eine natürliche Zahl r sowie Vektoren $\underline{x} \in \mathbb{R}^n$ und $\underline{y}_j \in \mathbb{R}^n$, $1 \leq j \leq r$ so gibt, dass

$$M = \left\{ \underline{x} + \sum_{j=1}^r \alpha_j \underline{y}_j \mid \alpha_j \in \mathbb{N} \text{ für } 1 \leq j \leq r \right\}$$

gilt.

- ii) Eine Teilmenge des \mathbb{R}^n heißt semi-linear, falls sie die Vereinigung von endlich vielen semi-affinen Mengen ist.

Ersetzt man bei der Definition der semi-affinen Menge die Forderung, dass die Koeffizienten α_j natürliche Zahlen sein müssen, durch die Forderung, dass sie reelle Zahlen sein müssen, so bilden die Vektoren einen affinen Raum (im Sinne der linearen Algebra oder analytischen Geometrie), denn es ist die Verschiebung eines linearen Teilraumes um \underline{x} .

Anschaulich bilden semi-affine Mengen ein Gitter in einem Kegel. Für den zweidimensionalen Fall mit zwei Vektoren \underline{y}_1 und \underline{y}_2 ist dies in Abbildung 1.2 gezeigt.

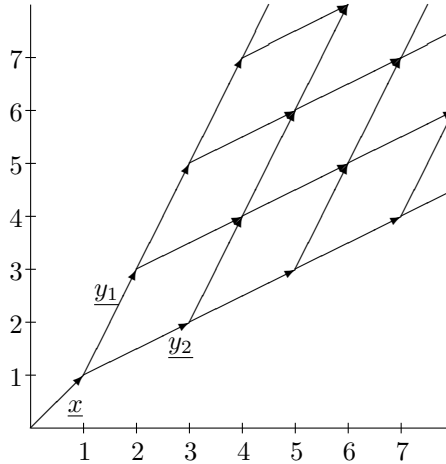


Abbildung 1.2: Semi-affine Menge mit den Vektoren $\underline{x} = (1, 1)$, $\underline{y}_1 = (1, 2)$ und $\underline{y}_2 = (2, 1)$

Innerhalb des Kegels gibt es folglich zu jedem Punkt der semi-affinen Menge in „relativer“ Nähe wieder einen Punkt der semi-affinen Menge. Daher verallgemeinert dieser Begriff die Dichtheit entsprechend Folgerung 1.11.

Ist M eine semi-lineare Menge, so gibt es natürliche Zahlen $m, n, r_1, r_2, \dots, r_m$ sowie Vektoren $\underline{x}_i \in \mathbb{N}^n$, $1 \leq i \leq m$, und $\underline{y}_{ij} \in \mathbb{N}^n$, $1 \leq i \leq m$, $1 \leq j \leq r_i$ so, dass

$$M = \bigcup_{i=1}^m \left\{ \underline{x}_i + \sum_{j=1}^{r_i} \alpha_{ij} \underline{y}_{ij} \mid \alpha_{ij} \in \mathbb{N} \text{ für } 1 \leq j \leq r_i \right\}$$

gilt.

Wir ordnen nun jedem Wort über einem Alphabet aus n Buchstaben einen n -dimensionalen Vektor zu.

Definition 1.13 *Es sei $V = \{a_1, a_2, \dots, a_n\}$ ein Alphabet. Für ein Wort $w \in V^*$ definieren wir den (n -dimensionalen) Parikh-Vektor von w bez. V durch*

$$\pi_V(w) = (\#_{a_1}(w), \#_{a_2}(w), \dots, \#_{a_n}(w)).$$

Die zu einer Sprache $L \subseteq V^*$ gehörende Parikh-Menge wird durch

$$\pi_V(L) = \{ \pi_V(w) \mid w \in L \}$$

definiert.

Falls das Alphabet V aus dem Kontext klar ist, schreiben wir nur π anstelle von π_V .

Nun übertragen wir unter Verwendung der Parikh-Vektoren den Begriff der Semi-Linearität auf Sprachen.

Definition 1.14 Eine Sprache $L \subseteq V^*$ heißt *semi-linear*, falls die Parikh-Menge $\pi_V(L)$ von L eine semi-lineare Menge ist.

Die Sprachen $K_1 \subset \{a, b, c\}^+$, $K_2 \subset \{a, b, c, d\}^+$ und $K_3 \subset \{a, b\}^+$ aus Folgerung 1.10 sind semi-linear, denn es gelten offenbar

$$\begin{aligned}\pi(K_1) &= \{(1, 1, 1) + \alpha(1, 1, 1) \mid \alpha \in \mathbb{N}\}, \\ \pi(K_2) &= \{(1, 1, 1, 1) + \alpha(1, 0, 1, 0) + \beta(0, 1, 0, 1) \mid \alpha \in \mathbb{N}, \beta \in \mathbb{N}\} \text{ und} \\ \pi(K_3) &= \{(2, 0) + \alpha(2, 0) + \beta(0, 2) \mid \alpha \in \mathbb{N}, \beta \in \mathbb{N}\} \\ &\quad \cup \{(0, 2) + \alpha(2, 0) + \beta(0, 2) \mid \alpha \in \mathbb{N}, \beta \in \mathbb{N}\}.\end{aligned}$$

Satz 1.15 Zu jeder semi-linearen Menge M gibt es eine reguläre Grammatik G , für die

$$\pi(L(G)) = M$$

gilt.

Beweis. Wir beweisen zuerst, dass jede semi-affine Menge M sich als $\pi(L(G))$ mit einer regulären Grammatik G darstellen lässt. Es sei eine semi-affine Menge M durch Vektoren $\underline{x}, \underline{y}_1, \underline{y}_2, \dots, \underline{y}_r$ aus dem \mathbb{R}^n gegeben. Wir betrachten ein Alphabet $V = \{a_1, a_2, \dots, a_n\}$, Wörter w, w_1, w_2, \dots, w_r über V mit $\pi_V(w) = \underline{x}$ und $\pi_V(w_i) = \underline{y}_i$ für $1 \leq i \leq r$ (dies kann z. B. durch die Wahl von $w = a_1^{m_1} a_2^{m_2} \dots a_n^{m_n}$ für $x = (m_1, m_2, \dots, m_n)$ geschehen) und die reguläre Grammatik

$$G = (\{S\}, V, \{S \rightarrow w\} \cup \{S \rightarrow w_i S \mid 1 \leq i \leq r\}, S).$$

Offenbar hat jede Ableitung in G die Form

$$S \Longrightarrow w_{i_1} S \Longrightarrow w_{i_1} w_{i_2} S \Longrightarrow w_{i_1} w_{i_2} w_{i_3} S \Longrightarrow \dots \Longrightarrow w_{i_1} w_{i_2} \dots w_{i_k} S \Longrightarrow w_{i_1} w_{i_2} \dots w_{i_k} w,$$

wobei $k \geq 0$ und $1 \leq i_j \leq r$ für $1 \leq j \leq k$ gelten. Damit gilt

$$\begin{aligned}\pi(L(G)) &= \{\pi(w_{i_1} w_{i_2} \dots w_{i_k} w) \mid k \geq 0, 1 \leq i_j \leq r \text{ für } 1 \leq j \leq k\} \\ &= \{\pi(w) + \pi(w_{i_1}) + \pi(w_{i_2}) + \dots + \pi(w_{i_k}) \mid k \geq 0, 1 \leq i_j \leq r \text{ für } 1 \leq j \leq k\} \\ &= \{\underline{x} + \underline{y}_{i_1} + \underline{y}_{i_2} + \dots + \underline{y}_{i_k} \mid k \geq 0, 1 \leq i_j \leq r \text{ für } 1 \leq j \leq k\} \\ &= \{\underline{x} + \sum_{j=1}^r \alpha_j \underline{y}_j \mid \alpha_j \in \mathbb{N} \text{ für } 1 \leq j \leq r\} \\ &= M\end{aligned}$$

(α_j gibt an, wie oft \underline{y}_j unter den $\underline{y}_{i_1}, \underline{y}_{i_2}, \dots, \underline{y}_{i_k}$ vorkommt).

Eine semi-lineare Menge R ist die Vereinigung von endlich vielen semi-affinen Mengen. Die Vereinigung von endlich vielen regulären Sprachen ist wieder regulär. Es sei R die Vereinigung von semi-affinen Mengen M_1, M_2, \dots, M_m . Dann gilt

$$\begin{aligned}R &= M_1 \cup M_2 \cup \dots \cup M_m \\ &= \pi(L(G_1)) \cup \pi(L(G_2)) \cup \dots \cup \pi(L(G_m)) \\ &= \pi(L(G_1) \cup L(G_2) \cup \dots \cup L(G_m)) \\ &= \pi(L(G))\end{aligned}$$

für gewisse reguläre Grammatiken G, G_1, G_2, \dots, G_m . □

Satz 1.16 *Jede kontextfreie Sprache ist semi-linear.*

Beweis. Es sei L eine kontextfreie Sprache, und es sei $G = (N, T, P, S)$ eine kontextfreie Grammatik mit $L(G) = L$. Für eine Teilmenge U von $N \setminus \{S\}$ sei L_U die Menge aller Wörter über T , für die ein Ableitungsbaum bez. G existiert, in dem nur Knoten aus $U \cup \{S\}$ vorkommen und für jedes $X \in U \cup \{S\}$ auch ein mit X markierter Knoten in dem Baum existiert. Ferner sei $u = \#(U) + 1$. Offensichtlich gilt

$$L = L(G) = \bigcup_{U \subseteq N \setminus \{S\}} L_U$$

und damit auch

$$\pi_T(L) = \bigcup_{U \subseteq N \setminus \{S\}} \pi_T(L_U).$$

Da aus der Definition semi-linearer Mengen als endliche Vereinigung semi-affiner Mengen sofort folgt, dass die Vereinigung semi-linearer Sprachen wieder eine semi-lineare Sprache ist, reicht es zum Nachweis der Semi-Linearität von L zu zeigen, dass L_U für jede Menge $U \subseteq N \setminus \{S\}$ semi-linear ist.

Wir bezeichnen mit H die Menge aller Wörter aus L , für die ein Ableitungsbaum B mit folgenden Eigenschaften existiert:

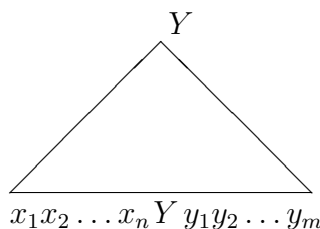
- die Wurzel von B ist mit S markiert,
- alle und nur die Elemente aus $U \cup \{S\}$ kommen als Markierung innerer Knoten von B vor,
- jedes Element aus $U \cup \{S\}$ kommt in jedem Pfad von der Wurzel zu einem Blatt höchstens u -mal als Markierung in B vor.

Nach Definition gilt offenbar $H \subseteq L_U$. Wir setzen

$$H_1 = \pi_T(H).$$

Für jedes $Y \in U \cup \{S\}$ bezeichnen wir mit H_Y die Menge aller Wörter w mit folgenden Eigenschaften:

- $w \in T^*\{Y\}T^*$ (d. h. w enthält bis auf ein Vorkommen des Nichtterminals Y nur Terminale),
- es gibt zu w einen Ableitungsbaum B' mit folgenden Eigenschaften:
 - die Wurzel von B' ist mit Y markiert,
 - nur Elemente aus $U \cup \{S\}$ kommen als Markierung innerer Knoten von B' vor,
 - jedes Element aus $U \cup \{S\}$ kommt in jedem Pfad von der Wurzel zu einem Blatt höchstens u -mal als Markierung in B' vor.

Abbildung 1.3: Ableitungsbaum für ein Wort aus H_Y

Die Ableitungs bäume für Elemente aus H_Y haben die Gestalt aus Abbildung 1.3 mit $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m \in T$.

Wir setzen

$$H_2 = \{\pi_T(v_1 v_2) \mid v_1, v_2 \in T^*, v_1 Y v_2 \in H_Y, Y \in U \cup \{S\}\}.$$

Wegen der dritten Forderung an die Ableitungs bäume für Wörter aus H und H_Y , sind die Mengen H_1 bzw. H_2 endliche Mengen. Es seien

$$H_1 = \{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_r\} \text{ und } H_2 = \{\underline{y}_1, \underline{y}_2, \dots, \underline{y}_s\}.$$

Wir setzen

$$K = \bigcup_{i=1}^r \left\{ \underline{x}_i + \sum_{j=1}^s \alpha_j \underline{y}_j \mid \alpha_j \in \mathbb{N} \text{ für } 1 \leq j \leq s \right\}.$$

Nach Definition ist K eine semi-lineare Menge. Daher reicht es zu zeigen, dass $K = \pi_T(L_U)$ gilt.

$K \subseteq \pi_T(L_U)$. Wir gehen beim Beweis induktiv über den Aufbau von K vor.

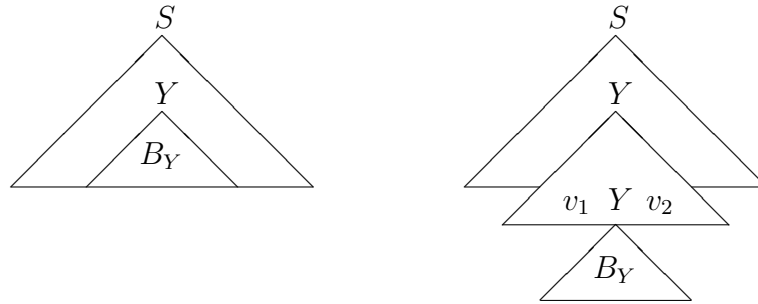
Induktionsanfang: Es sei $\underline{x} \in H_1$. Dann gibt es nach Definition von H_1 ein Wort w aus $H \subseteq L_U$ mit $\pi_T(w) = \underline{x}$. Damit ist $\underline{x} \in \pi_T(L_U)$.

Induktionsschritt: Wir zeigen, dass aus der Beziehung $\underline{x}_i + \sum_{j=1}^s \alpha_j \underline{y}_j \in \pi_T(L_U)$ auch die Relation $(\underline{x}_i + \sum_{j=1}^s \alpha_j \underline{y}_j) + \underline{y}_k \in \pi_T(L_U)$ für $1 \leq k \leq s$ folgt.

Es sei $w \in L_U$ ein Wort mit $\pi_T(w) = \underline{x}_i + \sum_{j=1}^s \alpha_j \underline{y}_j$. Dann gibt es einen Ableitungsbaum B'' für w , in dem alle und nur die Elemente aus $U \cup \{S\}$ vorkommen. Ferner sei $v = v_1 Y v_2$ ein Wort aus H_Y mit $\pi_T(v_1 v_2) = \underline{y}_k$. Dann gibt es für v einen Ableitungsbaum B''' der Form aus Abbildung 1.3 mit $v_1 = x_1 x_2 \dots x_n$ und $v_2 = y_1 y_2 \dots y_m$. In B'' gibt es einen Knoten Y . Es sei B_Y jener Teilbaum von B'' , der diesen Knoten Y als Wurzel hat. Wir schneiden nun B_Y aus B'' heraus, ersetzen ihn durch B''' und fügen dann an das dadurch entstehende Blatt Y den Baum B_Y wieder an. Die Abbildung 1.4 zeigt das Vorgehen schematisch.

Nach Definition der Menge H_2 (oder besser von H_Y) erfüllt der so entstehende Ableitungsbaum wieder alle Bedingungen für Bäume zu Wörtern aus L_U . Damit gehört das Wort z , das die Blätter ergeben, zur Menge L_U . Ferner gilt

$$\pi_T(z) = \pi_T(w) + \pi_T(v_1 v_2) = \left(\underline{x}_i + \sum_{j=1}^s \alpha_j \underline{y}_j \right) + \underline{y}_k.$$

Abbildung 1.4: Induktionsschritt beim Beweis von $K \subseteq \pi_T(L_U)$

Damit ist die Behauptung gezeigt.

$\pi_T(L_U) \subseteq K$. Wir beweisen diese Aussage über die Größe der Ableitungsbäume für die Wörter aus L_U .

Induktionsanfang: Der Ableitungsbaum zu $w \in L_U$ enthalte für jedes $X \in U \cup \{S\}$ höchstens u Markierungen mit X . Dann gehört w zur Menge H und folglich liegt $\pi_T(w)$ in H_1 . Damit liegt $\pi_T(w)$ auch in K (man wähle alle Koeffizienten $\alpha_{ij} = 0$).

Induktionsschritt: Wir zeigen, dass für ein Wort $w \in L_U$, dessen Ableitungsbaum B von einem Element $X \in U \cup \{S\}$ in einem Pfad mindestens $u + 1$ Vorkommen als Markierung enthält, $\pi_T(w) \in K$ gilt, wenn für alle Wörter $v \in L_U$, für die ein Ableitungsbaum B' mit weniger Knoten als B existiert, bereits $\pi_T(v) \in K$ gilt.

Wir betrachten einen Pfad, in dem mindestens $u + 1$ Vorkommen von X sind. Wir wählen $u + 1$ Vorkommen aus und bezeichnen diese vom Blatt ausgehend mit X_1, X_2, \dots, X_{u+1} . Weiterhin bezeichnen wir mit B_i den Teilbaum von B , der X_i als Wurzel hat, und mit N_i die Nichtterminale, die in B_i vorkommen. Da B_i ein Teilbaum von B_{i+1} für $1 \leq i \leq u$ ist, haben wir $N_i \subseteq N_{i+1}$ für $1 \leq i \leq u$. Da N nur u Elemente hat und natürlich $N_i \subseteq N$ für $1 \leq i \leq u + 1$ gilt, gibt es ein j mit $1 \leq j \leq u$ derart, dass $N_j = N_{j+1}$ gilt. Wir streichen nun in B zuerst den Baum B_{j+1} und ersetzen ihn durch B_j . Die Situation ist in Abbildung 1.5 dargestellt.

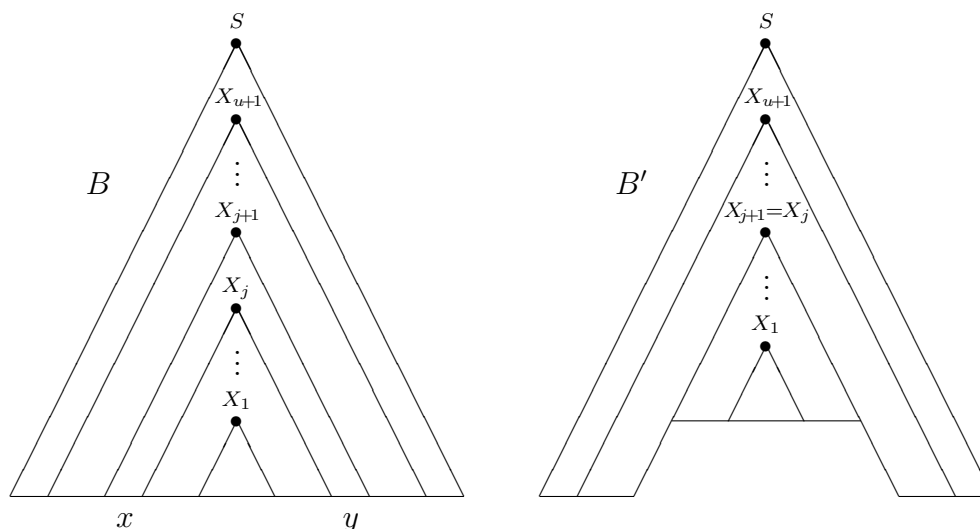
Der so entstehende Baum B' ist erneut ein Ableitungsbaum für ein Wort v in L_U , da die Menge der als Markierung vorkommenden Nichtterminale sich nicht ändert. Da B' offenbar weniger Knoten als B hat, liegt $\pi_T(v)$ nach Induktionsannahme in K , d. h.

$$\pi_T(v) = \underline{x}_i + \sum_{j=1}^s \alpha_j \underline{y}_j \text{ für gewisse } \alpha_j \in \mathbb{N}, 1 \leq j \leq s.$$

Wir betrachten nun den Baum B'' , der aus B_{j+1} entsteht, indem wir alle Knoten von B_j mit Ausnahme der Wurzel von B_j streichen. Die Blätter von B'' ergeben von links nach rechts gelesen ein Wort xXy mit $xy \in T^*$.

Wir nehmen nun an, dass jeder Pfad in B'' von der Wurzel zu einem Blatt von jedem $X' \in U \cup \{S\}$ höchstens u Vorkommen von X' enthält. Dann ist xXy in H_X und damit $\pi_T(xy) \in H_2$, sagen wir $\pi_T(xy) = \underline{y}_k$. Außerdem gilt

$$\pi_T(w) = \pi_T(v) + \pi_T(xy) = \left(\underline{x}_i + \sum_{j=1}^s \alpha_j \underline{y}_j \right) + \underline{y}_k,$$


 Abbildung 1.5: Induktionsschritt beim Beweis von $\pi_T(L_U) \subseteq K$

womit $\pi_T(w) \in K$ gezeigt ist.

Sollte B'' einen Pfad enthalten, in dem es $u+1$ Vorkommen von $X' \in U \cup \{S\}$ gibt, so betrachten wir diese Vorkommen und iterieren den Prozess. Da die Bäume hierbei immer kleiner werden, wird die Situation der Zugehörigkeit zu einem H_Y einmal erreicht, womit der Induktionsschritt vollzogen werden kann. \square

Folgerung 1.17 *Eine Sprache L über einem Alphabet aus genau einem Buchstaben ist genau dann kontextfrei, wenn sie regulär ist.*

Beweis. Es sei a der einzige Buchstabe des Alphabets. Dann besteht zwischen dem Wort a^p und dem zugehörigen Parikh-Vektor (p) eine eindeutige Beziehung. Daher sind die Gleichheiten $\pi_{\{a\}}(L_1) = \pi_{\{a\}}(L_2)$ und $L_1 = L_2$ äquivalent.

Es sei nun L eine kontextfreie Sprache. Nach Satz 1.16 ist $\pi(L)$ semi-linear. Folglich gibt es nach Satz 1.15 eine reguläre Grammatik G mit $\pi(L(G)) = \pi(L)$. Nach Obigem gilt also auch $L(G) = L$. Somit ist L eine reguläre Sprache.

Wir haben also gezeigt, dass jede kontextfreie Sprache über einem einelementigen Alphabet auch regulär ist. Aus der Definition der Sprachtypen folgt aber sofort, dass jede reguläre Sprache kontextfrei ist. Damit ist die Folgerung bewiesen. \square

Während entsprechend Satz 1.16 kontextfreie Sprachen semi-linear sind, gibt es kontextabhängige Sprachen, die nicht semi-linear sind. Um dies zu sehen, betrachten wir die Sprache

$$L = \{ba^{2^n}bb \mid n \geq 0\},$$

die von der monotonen Grammatik

$$G = (\{S, A, B, B', C\}, \{a, b\}, P, S)$$

mit

$$P = \{S \rightarrow bB'Ab, bB' \rightarrow bB, BA \rightarrow AAB, Bb \rightarrow B'b, AB' \rightarrow B'A\} \\ \cup \{bB' \rightarrow bC, CA \rightarrow aC, Cb \rightarrow bb\}$$

erzeugt wird (den Beweis hierfür überlassen wir dem Hörer/Leser). Wegen Satz 1.9 ist L also kontextabhängig. Angenommen, L wäre semi-linear. Dann gibt es natürliche Zahlen m, r_1, r_2, \dots, r_m und zweidimensionale Vektoren \underline{x}_i , $1 \leq i \leq m$, und \underline{y}_{ij} , $1 \leq i \leq m$, $1 \leq j \leq r_i$ so, dass

$$\pi(L) = \bigcup_{i=1}^m \left\{ \underline{x}_i + \sum_{j=1}^{r_i} \alpha_{ij} \underline{y}_{ij} \mid \alpha_{ij} \in \mathbb{N} \text{ für } 1 \leq j \leq r_i \right\}$$

gilt. Wir stellen erst einmal fest, dass $\underline{x}_i = (x_i, 3)$ und $\underline{y}_{ij} = (y_{ij}, 0)$ für $1 \leq i \leq m$ und $1 \leq j \leq r_i$ gelten muss. Wir setzen

$$p = \max \left\{ \max\{x_i \mid 1 \leq i \leq m\}, \max\{y_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq r_i\} \right\}$$

und betrachten ein Wort $ba^{2^q}bb \in L$ mit $2^q > p$. Dann gibt es ein i , $1 \leq i \leq m$, und Koeffizienten α_{ij} , $1 \leq j \leq r_i$, derart, dass

$$2^q = x_i + \sum_{j=1}^{r_i} \alpha_{ij} y_{ij}$$

gilt. Da $2^q > p \geq x_i$ ist, gibt es ein j mit $\alpha_{ij} y_{ij} > 0$. Wir betrachten nun ein Wort $z \in L$ zum Vektor

$$\underline{u} = \underline{x}_i + \left(\sum_{j=1}^{r_i} \alpha_{ij} \underline{y}_{ij} \right) + \underline{y}_{ij}.$$

Offensichtlich haben wir wegen $\pi(z) = \underline{u}$, $z \in L$ und $y_{ij} > 0$ die Beziehungen

$$2^q < 2^q + y_{ij} = \#_a(z) \text{ und } \#_a(z) = 2^q + y_{ij} \leq 2^q + p < 2^q + 2^q = 2^{q+1}.$$

Dies widerspricht aber der Tatsache, dass nach Definition von L die Anzahl der Vorkommen von a in z eine Potenz von 2 sein muss. Damit ist nachgewiesen, dass L nicht semi-linear ist.

Wir ergänzen nun die Abschlusseigenschaften von der Familie der kontextfreien Sprachen.

Es seien X und Y zwei Alphabete. Unter einem Homomorphismus h von X^* in Y^* verstehen wir eine Funktion von X^* in Y^* , die der Bedingung $h(w_1 w_2) = h(w_1) h(w_2)$ genügt. Offensichtlich reicht es zur Angabe eines Homomorphismus, die Bilder der Buchstaben von X zu kennen, da für ein Wort $w = a_1 a_2 \dots a_n$ wegen der Homomorphieeigenschaft $h(w) = h(a_1) h(a_2) \dots h(a_n)$ gelten muss. Für Sprachen $L \subseteq X^*$, $L' \subseteq Y^*$ und einen Homomorphismus $h : X^* \rightarrow Y^*$ setzen wir

$$h(L) = \{h(w) \mid w \in L\} \quad \text{und} \quad h^{-1}(L') = \{w \mid h(w) \in L'\}.$$

Satz 1.18 Die Klasse $\mathcal{L}(CF)$ ist abgeschlossen unter Vereinigung, Durchschnitten mit regulären Sprachen, Produkt (Konkatenation), Kleene-Abschluss, Homomorphismen und inversen Homomorphismen.

Beweis. Die Abgeschlossenheit unter Vereinigung, Produkt und Kleene-Abschluss wurde schon in der Vorlesung zur Theoretischen Informatik gezeigt. Hier beweisen wir nur den Abschluss unter Homomorphismen und Durchschnitten mit regulären Sprachen, da wir diese Eigenschaften im Folgenden benutzen; bez. der Abgeschlossenheit unter inversen Homomorphismen verweisen wir auf [7].

Es sei L eine kontextfreie Sprache. Aus der Vorlesung zur Theoretischen Informatik wissen wir, dass es einen Kellerautomaten \mathcal{A}_L gibt, der L akzeptiert. Ferner sei R eine reguläre Menge. Für diese gibt es einen endlichen Automaten \mathcal{A}_R , der R akzeptiert. Es seien Z_L und Z_R die Zustandsmengen, z_L und z_R die Anfangszustände sowie F_L und F_R die Menge der akzeptierenden Zustände dieser Automaten. Wir konstruieren dann einen Kellerautomaten mit der Zustandsmenge $Z_L \times Z_R$, dem Anfangszustand (z_L, z_R) und der Menge $F_L \times F_R$ der akzeptierenden Zustände, der auf der ersten Komponente der Zustände und bei der Veränderung des Kellers wie \mathcal{A}_L arbeitet und sich auf der zweiten Komponente bei jedem Leseschritt des Kellerautomaten wie \mathcal{A}_R verhält. Erreicht dieser Automat einen Zustand aus $F_L \times F_R$ und ist die Akzeptanzbedingung hinsichtlich des Kellers erfüllt, so liegt das gelesene Wort sowohl in L als auch in R . Somit akzeptiert dieser Kellerautomat die Sprache $L \cap R$. Damit ist $L \cap R$ als kontextfrei nachgewiesen.

Es seien L eine kontextfreie Sprache und $G = (N, X, P, S)$ eine kontextfreie Grammatik mit $L(G) = L$. Ferner sei $h : X^* \rightarrow Y^*$ ein Homomorphismus. Wir erweitern h auf $(X \cup N)^*$, indem wir $h(A) = A$ für $A \in N$ setzen. Wir konstruieren nun die kontextfreie Grammatik $G' = (N, Y, P', S)$ durch Setzen von $P' = \{A \rightarrow h(w) \mid A \rightarrow w \in P\}$. Mittels vollständiger Induktion über die Länge der Ableitung kann man leicht nachweisen, dass $S \xrightarrow{*} z$ genau dann eine Ableitung in G ist, wenn $S \xrightarrow{*} h(z)$ eine in G' ist. Damit gilt $L(G') = \{h(z) \mid z \in L(G)\}$. Folglich gilt $L(G') = h(L)$. Also ist gezeigt, dass $h(L)$ kontextfrei ist. \square

Abschließend geben wir ein paar Bemerkungen zum Mitgliedsproblem für kontextfreie Grammatiken, das durch

Gegeben: kontextfreie Grammatik $G = (N, T, P, S)$, Wort $w \in T^+$
Frage: Liegt w in $L(G)$?

gegeben ist. Hierfür gilt der folgende Satz.

Satz 1.19 Das Mitgliedsproblem für eine kontextfreie Grammatik $G = (N, T, P, S)$ und ein Wort $w \in T^+$ ist in der Zeitschranke $O(|w|^3 \cdot \#(P))$ entscheidbar.

Es seien G eine kontextfreie Grammatik in Chomsky-Normalform und $w = a_1 a_2 \dots a_n$ ein Wort mit $a_i \in T$ für $1 \leq i \leq n$. In dem in der Vorlesung zur Theoretischen Informatik vorgestellten Cocke-Younger-Kasami-Algorithmus werden Mengen $V_{i,j}$ mit $0 \leq i < j \leq n$ wie folgt berechnet: $V_{i-1,i} = \{A \mid A \in N, A \rightarrow a_i \in P\}$ für $1 \leq i \leq n$. Sind dann für $i < k < j$ die Mengen $V_{i,k}$ und $V_{k,j}$ bereits definiert, so setzen wir

$$V_{i,j} = \{A \mid A \in N, A \rightarrow BC \in P, B \in V_{i,k}, C \in V_{k,j}, i < k < j\}.$$

Es wurde bewiesen, dass

$$V_{i,j} = \{A \mid A \in N, A \xrightarrow{*} a_{i+1}a_{i+2} \dots a_j\}$$

gilt. Damit ergibt sich, dass w genau dann in L liegt, wenn $S \in V_{0n}$ gilt.

Da es höchstens n Möglichkeiten für k gibt und für jede Zahl k alle Regeln von P durchzumustern sind, kann jede Menge $V_{i,j}$ in $\#(P) \cdot n$ Schritten konstruiert werden. Da insgesamt $\frac{n(n+1)}{2}$ Mengen zu konstruieren sind, ergibt sich damit ein durch $\frac{\#(P)n^2(n+1)}{2}$ nach oben beschränkter Gesamtaufwand.

1.2. Schwach kontextabhängige Grammatikklassen

Wir zeigen zuerst, dass kontextfreie Sprachen nicht ausreichen, alle Aspekte und Phänomene natürlicher Sprachen zu beschreiben. Wir weisen dafür nach, dass die Menge DS der Sätze der deutschen Sprache nicht kontextfrei ist. Dazu betrachten wir die Sätze

Maria und Jonas sind eine Frau bzw. ein Mann.

Maria, Jonas und Willi sind eine Frau, ein Mann bzw. ein Mann.

Maria, Jonas, Willi und Jenny sind eine Frau, ein Mann, ein Mann bzw. eine Frau.

Wenn wir die Menge aller Vornamen, die in der deutschen Sprache vorkommen, mit *Vorname* und die Menge {ein Mann, eine Frau} mit *Person* bezeichnen, so liegen alle Sätze dieser Form in der regulären Menge

$$R = (\text{Vorname},)^* \text{Vorname} \text{ und } \text{Vorname} \text{ sind } (\text{Person},)^* \text{Person} \text{ bzw. } \text{Person}.$$

Es gilt sogar, dass alle diese Sätze die Menge $DS \cap R$ bilden. Nehmen wir an, dass die deutsche Sprache eine kontextfreie Menge ist, so ist nach Satz 1.18 auch $DS \cap R$, also die Menge aller dieser Sätze kontextfrei. Wir betrachten nun den Homomorphismus h mit

$$h(x) = \begin{cases} a & \text{falls } x \text{ ein weiblicher Vorname oder } x \text{ die Wortgruppe } \textit{eine Frau} \text{ ist,} \\ b & \text{falls } x \text{ ein männlicher Vorname oder } x \text{ die Wortgruppe } \textit{ein Mann} \text{ ist,} \\ \lambda & \text{sonst.} \end{cases}$$

Beachten wir, dass obige Sätze nur korrekt sind, wenn die Anzahl der Vorkommen von Vornamen und von Wortgruppen *eine Frau* und *ein Mann* übereinstimmen, jedem weiblichen Vornamen jeweils ein *eine Frau* entspricht und jedem männlichen Vornamen jeweils ein *ein Mann* entspricht, so ergibt sich

$$h(DS \cap R) = \{ww \mid w \in \{a, b\}^+\}.$$

Da $DS \cap R$ nach unserer Annahme kontextfrei ist, ist nach Satz 1.18 auch $h(DS \cap R)$ kontextfrei. Dies widerspricht aber Folgerung 1.10. Daher muss unsere Annahme falsch sein, d. h. die deutsche Sprache ist nicht kontextfrei.

Dieses Beispiel ist nicht völlig befriedigend, da es semantische Aspekte wegen der geforderten Gleichheit des Geschlechts der durch den Vornamen bezeichneten Person und *eine Frau* bzw. *ein Mann* benötigt. Analoge Beispiele, die nur die Syntax berücksichtigen, gibt es aber in anderen Sprachen. Wir geben hier ein Beispiel aus einem schweizerdeutschen Dialekt:

Jan säit das mer em Hans hälfed.
(Jan sagt, dass wir Hans halfen.)

Jan säit das mer em Hans es Huus hälfed aastriche.
(Jan sagt, dass wir Hans halfen, das Haus zu streichen.)

Jan säit das mer d'chind em Hans es Huus lönd hälfed aastriche.
(Jan sagt, dass wir den Kindern erlaubten, Hans zu helfen, das Haus zu streichen.)

Auch hier erhalten wir analog zu Obigem durch Durchschnittsbildung mit einer regulären Menge und mit einem passenden Homomorphismus die nicht-kontextfreie Sprache $\{ww \mid w \in \{a, b\}^+\}$.

Die verschränkten Abhängigkeiten können weiter gesteigert werden, so dass auch Sätze folgender Form zulässig sind:

Jan säit das mer $(\text{Akk}_{\text{Obj}})^n (\text{Dat}_{\text{Obj}})^m$ es Huus
händ wele $(\text{Akk}_{\text{Verb}})^n (\text{Dat}_{\text{Verb}})^m$ aastriche.
(Jan sagt, dass wir $(\text{Akk}_{\text{Obj}} \text{ Akk}_{\text{Verb}})^n$ wollten,
 $(\text{Dat}_{\text{Obj}}$ zu $\text{Dat}_{\text{Verb}})^m$, das Haus zu streichen.)

Dabei stehen Akk und Dat für Akkusativ bzw. Dativ sowie Obj für Objekt. In solchen Konstruktionen müssen genau so viele Objekte im Akkusativ/Dativ vorkommen wie Verben, die den jeweiligen Kasus verlangen. Außerdem muss die Reihenfolge in dem Sinne übereinstimmen, dass das i -te Objekt in dem Kasus steht, den das i -te Verb verlangt. Durch Durchschnittsbildung mit einer regulären Menge und Anwenden jenes Homomorphismus, der ein Akkusativ-Objekt auf a , ein Dativ-Objekt auf b , ein Akkusativ erforderndes Verb auf c , ein Dativ erforderndes Verb auf d und alles andere auf λ abbildet, erhalten wir die Sprache $\{a^n b^m c^n d^m \mid n \geq 1, m \geq 1\}$, die ebenfalls nicht kontextfrei ist.

Ein weiteres Beispiel zeigt, dass auch die englische Sprache nicht kontextfrei ist. Die folgenden Sätze sind grammatikalisch richtig:

Alice and Bob achieved marks A and B in Mathematics and Biology, respectively.
Alice, Bob, and Mary achieved marks A, B, and A in Mathematics, Biology, and Chemistry, respectively.
Alice, Bob, Mary, and Tom achieved marks A, B, A, and C in Mathematics, Biology, Chemistry, and Spanish, respectively.

Werden ein Vorname auf a , eine Note auf b , ein Fach auf c und alles andere auf λ abgebildet, so erhalten wir aus allen Sätzen der obigen Form die Sprache $\{a^n b^n c^n \mid n \geq 1\}$.

Um also eine natürliche Sprache umfassend durch eine Grammatik beschreiben zu können, müssen wir Sprachklassen betrachten, die die kontextfreien Sprachen aber auch nicht-kontextfreie Sprachen enthalten. Aus der Chomsky-Hierarchie käme hier als erstes die Menge der kontextabhängigen Sprachen in Frage. Diese hat aber einige sehr schlechte Eigenschaften, z. B.

- für kontextabhängige Grammatiken ist das Mitgliedsproblem mindestens NP -hart, d. h. jedes in NP liegende Problem lässt sich in polynomialer Zeit auf das Mitgliedsproblem transformieren,

- für kontextabhängige Grammatiken sind das Leerheits- und das Endlichkeitsproblem unentscheidbar,
- $\mathcal{L}(CS)$ enthält Sprachen, die nicht semi-linear sind, genauer gesagt, bei denen hinsichtlich der Wortlänge beliebig große Lücken auftreten.

Daher erweist sich bereits die Klasse $\mathcal{L}(CS)$ als zu umfangreich für die Beschreibung natürlicher Sprachen. Ziel soll es also sein, eine Klasse von Grammatiken zu definieren, die zum einen einige nicht-kontextfreie Sprachen erzeugt, z. B. $\{ww \mid w \in \{a, b\}^+\}$, da diese Sprachen sich als wichtig für natürliche Sprachen erwiesen haben (siehe oben), zum anderen aber möglichst viele positive Eigenschaften kontextfreier Grammatiken hat. Solch eine Klasse wurde von ARAVIND K. JOSHI und anderen zur folgenden Definition spezifiziert (siehe u. a. [10, 12]).

Definition 1.20 *Eine Klasse \mathcal{G} von Grammatiken heißt schwach kontextabhängig, wenn sie folgende Bedingungen erfüllt:*

- *Zu jeder kontextfreien Sprache L gibt es eine Grammatik $G \in \mathcal{G}$, die die Sprache L erzeugt.*
Diese Forderung wird erhoben, um eine Anreicherung der kontextfreien Sprachen zu erreichen. Sie ist aber nicht durchgängig einsichtig, da es auch kontextfreie Sprachen gibt, deren Struktur nicht unbedingt das Verhalten natürlicher Sprachen erfasst; dies gilt z. B. für die Dyck-Sprache der korrekt geklammerten Ausdrücke (bei Reduktion auf die Klammern), die durch $G = (\{S\}, \{[,]\}, \{S \rightarrow SS, S \rightarrow [S], S \rightarrow []\}, S)$ erzeugt wird.
- *Für die drei Sprachen K_1, K_2 und K_3 (siehe Folgerung 1.10) gibt es in \mathcal{G} Grammatiken G_1, G_2 und G_3 mit $L(G_1) = K_1, L(G_2) = K_2$ und $L(G_3) = K_3$.*
Diese Forderung wird erhoben, weil es gewisse Teile von Sprachen gibt, die im Wesentlichen die Struktur von einer der Sprachen K_1, K_2 oder K_3 aufweisen (wie wir oben gezeigt haben). Daher erfordert die Modellierung dieser Aspekte die Erzeugbarkeit dieser Sprachen.
- *Für jede Grammatik $G \in \mathcal{G}$ ist $L(G)$ semi-linear.*
JOSHI hatte ursprünglich die schwächere Forderung erhoben, dass eine Konstante k derart existiert, dass für jedes hinreichend lange Wort w aus der Sprache ein anderes Wort w' in der Sprache liegt, dessen Länge sich von der von w höchstens um k unterscheidet (siehe Folgerung 1.11). Diese Forderung ist sinnvoll, da es hinsichtlich der Länge von Sätzen einer natürlichen Sprache sicher keine beliebig großen Lücken gibt.
- *Das Mitgliedsproblem ist für Grammatiken aus \mathcal{G} in polynomialer Zeit entscheidbar.*
Da der Mensch die syntaktische Korrektheit eines Satzes sofort erkennt, ist es natürlich zu verlangen, dass dies in relativierter Form auch für schwach kontextabhängige Grammatiken gelten soll.

Wir werden in den folgenden Abschnitten einige Ansätze zum Gewinnen schwach kontextabhängiger Grammatikklassen diskutieren.

1.3. Indizierte Grammatiken

1.3.1. Definition, Beispiele und Eigenschaften

Als ersten Kandidaten für eine schwach kontextabhängige Grammatikklasse betrachten wir die indizierten Grammatiken, die 1968 von ALFRED AHO in der Arbeit [1] eingeführt wurden. Die Idee besteht dabei darin, durch einen Index, der mit einem Nichtterminal assoziiert wird, die Historie der Ableitung dieses Nichtterminals zu speichern.

Definition 1.21

- i) Eine indizierte Grammatik ist ein Quintupel $G = (N, T, I, P, S)$, wobei
- N, T und S wie bei einer Regelgrammatik bestimmt sind,
 - I eine endliche Menge ist und jedem $i \in I$ eine endliche Menge P_i von Regeln der Form $A \rightarrow w$ mit $A \in N$ und $w \in (N \cup T)^+$ zugeordnet ist, und
 - P eine endliche Menge von Regeln $A \rightarrow w$ mit $A \in N$ und $w \in (NI^* \cup T)^+$ ist.
- ii) Es seien $G = (N, T, I, P, S)$ eine indizierte Grammatik und x und y zwei Wörter aus $(NI^* \cup T)^+$. Wir sagen, dass y aus x durch einen Ableitungsschritt entsteht (oder x erzeugt y in einem Ableitungsschritt, geschrieben als $x \Longrightarrow y$), wenn eine der beiden folgenden Situationen ① oder ② zutrifft:
- ① – $x = x_1 A \beta x_2$ mit $A \in N$, $\beta \in I^*$, $x_1, x_2 \in (NI^* \cup T)^*$,
– $A \rightarrow X_1 \beta_1 X_2 \beta_2 \dots X_k \beta_k \in P$ mit $X_j \in N \cup T$ sowie $\beta_j \in I^*$ für $X_j \in N$ und $\beta_j = \lambda$ für $X_j \in T$, $1 \leq j \leq k$ und
– $y = x_1 X_1 \gamma_1 X_2 \gamma_2 \dots X_k \gamma_k x_2$ mit $\gamma_j = \beta_j \beta$ für $X_j \in N$ und $\gamma_j = \lambda$ für $X_j \in T$, $1 \leq j \leq k$,
- oder
- ② – $x = x_1 A_i \beta x_2$ mit $A \in N$, $i \in I$, $\beta \in I^*$, $x_1, x_2 \in (NI^* \cup T)^*$,
– $A \rightarrow X_1 X_2 \dots X_k \in P_i$ mit $X_j \in N \cup T$ für $1 \leq j \leq k$ und
– $y = x_1 X_1 \gamma_1 X_2 \gamma_2 \dots X_k \gamma_k x_2$ mit $\gamma_j = \beta$ für $X_j \in N$ und $\gamma_j = \lambda$ für $X_j \in T$, $1 \leq j \leq k$.

Mit \Longrightarrow^* bezeichnen wir den reflexiven und transitiven Abschluss von \Longrightarrow .

- iii) Die von einer indizierten Grammatik $G = (N, T, I, P, S)$ erzeugte Sprache ist durch

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

definiert.

Die Elemente aus I werden Indices genannt. Während des Ableitens ist jedes Nichtterminal mit einem Wort über I , also einer Indexfolge versehen, die beim Ableiten wie ein Keller bearbeitet wird, wobei der jeweilige erste Index der Folge dem obersten Element im Keller entspricht. Die Indexfolge hinter einem Nichtterminal A speichert in gewisser Weise die Ableitungsschritte, die zum Nichtterminal A führten, da sie bei einem Ableitungsschritt unter Anwendung einer Regel aus P an alle nachfolgenden Nichtterminale übergeben wird. Die Regeln aus P_i dienen dazu, die Indices wieder abzubauen, denn bei

Anwendung einer Regel aus P_i wird der Index i gestrichen; allerdings kann dies nur erfolgen, wenn i der erste Index in der Indexfolge ist.

Anstelle von $Ai_1i_2\dots i_s \in NI^*$ werden wir zukünftig $A[i_1i_2\dots i_s]$ schreiben.

Mit $\mathcal{L}(I)$ bezeichnen wir die Menge der Sprachen, die von indizierten Grammatiken erzeugt werden.

Wir geben nun einige Beispiele für indizierte Grammatiken und die von ihnen erzeugten Sprachen an.

Beispiel 1.22 *Wir betrachten die indizierte Grammatik*

$$G_1 = (\{S, A, B\}, \{a, b, c\}, \{1, 2\}, \{S \rightarrow aA[1]c, A \rightarrow aA[2]c, A \rightarrow B\}, S)$$

mit den assoziierten Regelmengen

$$P_1 = \{B \rightarrow b\} \quad \text{und} \quad P_2 = \{B \rightarrow Bb\}.$$

Die Ableitungen bez. G_1 sind alle von der folgenden Form; sie variieren nur im Parameter n :

$$\begin{aligned} S &\Longrightarrow aA[1]c \Longrightarrow aaA[21]cc \Longrightarrow aaaA[221]ccc \Longrightarrow \dots \Longrightarrow a^n A[2^{n-1}1]c^n \\ &\Longrightarrow a^n B[2^{n-1}1]c^n \Longrightarrow a^n B[2^{n-2}1]bc^n \Longrightarrow \dots \Longrightarrow a^n B[21]b^{n-2}c^n \\ &\Longrightarrow a^n B[1]b^{n-1}c^n \Longrightarrow a^n b^n c^n. \end{aligned}$$

Daher gilt

$$L(G_1) = \{a^n b^n c^n \mid n \geq 1\} = K_1.$$

Wir modifizieren die Grammatik G_1 zur Grammatik G_2 , indem wir zum Terminalalphabet noch den Buchstaben d hinzufügen,

$$P_1 = \{B \rightarrow db\} \quad \text{und} \quad P_2 = \{B \rightarrow dBb\}$$

setzen und die anderen Komponenten von G_1 nicht ändern. Es ist dann offensichtlich

$$L(G_2) = \{a^n d^n b^n c^n \mid n \geq 1\},$$

d. h. wir haben nicht nur die gleiche Anzahl von drei sondern von vier Buchstaben.

Als nächstes betrachten wir die indizierte Grammatik

$$G_3 = (\{S, A, B, C\}, \{a, b, c, d, e\}, \{1, 2\}, \{S \rightarrow aA[1]e, A \rightarrow aA[2]e, A \rightarrow BC\}, S)$$

mit

$$P_1 = \{B \rightarrow b, C \rightarrow cd\} \quad \text{und} \quad P_2 = \{B \rightarrow Bb, C \rightarrow cCd\}.$$

Die typische Form eines Ableitungsbaumes in G_3 ist in Abbildung 1.6 gegeben. Daran sieht man sofort, dass

$$L(G_3) = \{a^n b^n c^n d^n e^n \mid n \geq 1\}$$

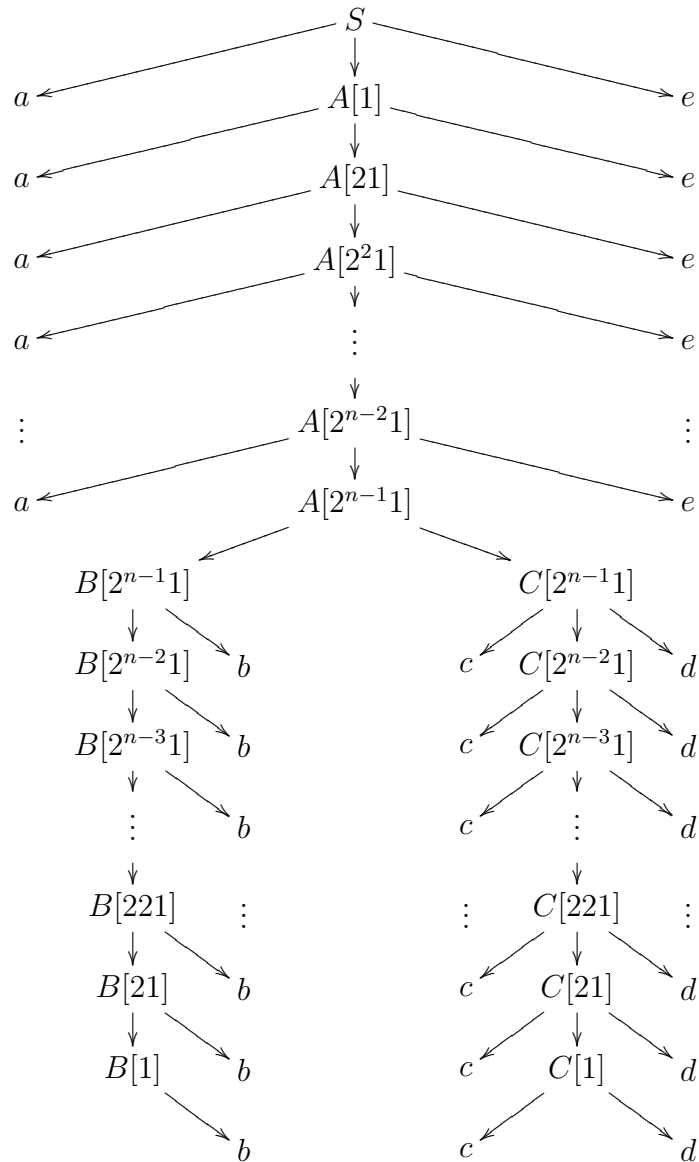


Abbildung 1.6: Ableitungsbaum bezüglich der Grammatik G_3 aus Beispiel 1.22

gilt.

Die indizierte Grammatik

$$G_4 = (\{S, A, A_1, A_2, \dots, A_n\}, \{a_1, a_2, \dots, a_n\}, \{1, 2\}, P, S)$$

mit den Regelmengen

$$P = \{S \rightarrow A[1], A \rightarrow A[2], A \rightarrow A_1 A_2 \dots A_n\},$$

$$P_1 = \bigcup_{i=1}^n \{A_i \rightarrow a_i\} \text{ und } P_2 = \bigcup_{i=1}^n \{A_i \rightarrow A_i a_i\}$$

erzeugt die Sprache

$$L(G_4) = \{a_1^m a_2^m \dots a_n^m \mid m \geq 1\}.$$

Im Folgenden wollen wir die Regelmengen P_i mit $i \in I$ in die Menge P integrieren; wir schreiben für $A \rightarrow w \in P_i$ dann einfach $A[i] \rightarrow w$. Dadurch lassen sich alle Bestandteile für Ableitungen innerhalb von G angeben.

Beispiel 1.23 *Es sei die indizierte Grammatik*

$$G_5 = (\{S, A, B, C\}, \{a, b, c, d\}, \{1, 2\}, P, S)$$

mit

$$P = \{S \rightarrow A[1]d, A \rightarrow A[2]d, A \rightarrow aBc, B \rightarrow aBc, B \rightarrow C, C[1] \rightarrow b, C[2] \rightarrow Cb\}$$

gegeben. Die möglichen Ableitungen haben alle die folgende Form, wobei sie nur hinsichtlich der Parameter n und m variieren:

$$\begin{aligned} S &\Longrightarrow A[1]d \Longrightarrow A[21]dd \Longrightarrow A[221]ddd \Longrightarrow \cdots \Longrightarrow A[2^{n-1}1]d^n \\ &\Longrightarrow aB[2^{n-1}1]cd^n \Longrightarrow aaB[2^{n-1}1]ccd^n \Longrightarrow \cdots \Longrightarrow a^m B[2^{n-1}1]c^m d^n \\ &\Longrightarrow a^m C[2^{n-1}1]c^m d^n \Longrightarrow a^m C[2^{n-2}1]bc^m d^n \Longrightarrow \cdots \Longrightarrow a^m C[21]b^{n-2}c^m d^n \\ &\Longrightarrow a^m C[1]b^{n-1}c^m d^n \Longrightarrow a^m b^n c^m d^n. \end{aligned}$$

Folglich ergibt sich

$$L(G_5) = \{a^m b^n c^m d^n \mid n \geq 1, m \geq 1\} = K_2$$

Beispiel 1.24 *Wir betrachten nun die indizierte Grammatik*

$$G_6 = (\{S, A, B\}, \{a, b\}, \{a, b, a', b'\}, P, S)$$

mit

$$P = \{A \rightarrow B\} \cup \bigcup_{x \in \{a, b\}} \{S \rightarrow xA[x'], A \rightarrow xA[x], B[x'] \rightarrow x, B[x] \rightarrow Bx\}.$$

In diesem Fall ergeben sich Ableitungen der folgenden Form

$$\begin{aligned} S &\Longrightarrow x_1 A[x'_1] \Longrightarrow x_1 x_2 A[x_2 x'_1] \Longrightarrow \cdots \Longrightarrow x_1 x_2 \cdots x_n A[x_n x_{n-1} \cdots x_2 x'_1] \\ &\Longrightarrow x_1 x_2 \cdots x_n B[x_n x_{n-1} \cdots x_2 x'_1] \Longrightarrow x_1 x_2 \cdots x_n B[x_{n-1} x_{n-2} \cdots x_2 x'_1] x_n \\ &\Longrightarrow \cdots \Longrightarrow x_1 x_2 \cdots x_n B[x_2 x'_1] x_3 x_4 \cdots x_n \Longrightarrow x_1 x_2 \cdots x_n B[x'_1] x_2 x_3 \cdots x_n \\ &\Longrightarrow x_1 x_2 \cdots x_n x_1 x_2 \cdots x_n \end{aligned}$$

mit $x_i \in \{a, b\}$ für $1 \leq i \leq n$, woraus

$$L(G_6) = \{ww \mid w \in \{a, b\}^+\} = K_3$$

folgt.

Nach den vorstehenden Beispielen sind wir also in der Lage, jede der Sprachen K_1 , K_2 und K_3 zu erzeugen. Offensichtlich lässt sich aber auch jede kontextfreie Sprache erzeugen, denn die kontextfreie Grammatik $G = (N, T, P, S)$ und die indizierte Grammatik $G' = (N, T, \{1\}, P, S)$ mit $P_1 = \emptyset$ erzeugen die gleiche Sprache, da die Indices nicht wirklich benutzt werden. Aus diesen Bemerkungen ergibt sich der folgende Satz.

Satz 1.25 $\mathcal{L}(CF) \subset \mathcal{L}(I)$. □

Wir geben nun eine Normalform für indizierte Grammatiken an. Sie besagt, dass mit Ausnahme der Regeln, die Indices einführen oder streichen, eine Grammatik in Chomsky-Normalform vorliegt und die Regeln, die Indices einführen bzw. streichen, linear sind.

Definition 1.26 Eine indizierte Grammatik $G = (N, T, I, P, S)$ heißt in Normalform, wenn die Regeln alle von der Form

$$A \rightarrow BC, A \rightarrow B[i], A[i] \rightarrow B \text{ oder } A \rightarrow a \text{ mit } A, B, C \in N, a \in T, i \in I$$

sind.

Satz 1.27 Zu jeder indizierten Grammatik G gibt es eine indizierte Grammatik G' derart in Normalform, dass $L(G') = L(G)$ gilt.

Beweis. Es sei $G = (N, T, I, P, S)$ eine indizierte Grammatik. Wir konstruieren zuerst eine Grammatik $G'' = (N'', T, I, P'', S)$ durch folgendes Vorgehen: Es sei

$$\bar{P} = \{A[i] \rightarrow w \mid A \in N, i \in I, A[i] \rightarrow w \in P, w \notin N\} \subseteq P.$$

Für eine Regel $p = A[i] \rightarrow w \in \bar{P}$ konstruieren wir die Regeln $p' = A[i] \rightarrow A_p$ und $p'' = A_p \rightarrow w$, wobei A_p ein zusätzliches Nichtterminal ist. Wir setzen nun

$$\begin{aligned} N'' &= N \cup \{A_p \mid p \in \bar{P}\}, \\ P'' &= (P \setminus \bar{P}) \cup \bigcup_{p \in \bar{P}} \{p', p''\}. \end{aligned}$$

Da auf jede Anwendung von $A[i] \rightarrow A_p$ eine Anwendung von $A_p \rightarrow w$ folgt (vielleicht erst ein paar Ableitungsschritte später), ergibt sich zusammen die Ableitung $A[i] \xrightarrow{*} w$ mit der richtigen Indexübergabe. Folglich gilt offenbar $L(G'') = L(G)$.

Ausgehend von der Grammatik G'' konstruieren wir nun eine indizierte Grammatik $G' = (N', T, I, P', S)$ wie folgt:

- Für jede Regel $A \rightarrow B$ in P'' mit $A, B \in N''$ wird zu jeder Regel $B \rightarrow w \in P''$ die Regel $A \rightarrow w$ zu P' hinzugenommen. Die Regel $A \rightarrow B$ wird daraufhin aus der Menge P'' entfernt.
- Jede Regel $A \rightarrow a$ aus P'' mit $A \in N''$ und $a \in T$ wird in die Menge P' übernommen.
- Für jede Regel $A \rightarrow X_1[\alpha_1]X_2[\alpha_2]\dots X_k[\alpha_k]$ mit $k \geq 2$ (und $\alpha_i = \lambda$ für $X_i \in T$) aus P'' nehmen wir zum einen in P' die Regeln

$$A \rightarrow B_1C_1, C_1 \rightarrow B_2C_2, \dots, C_{k-3} \rightarrow B_{k-2}C_{k-2}, C_{k-2} \rightarrow B_{k-1}B_k$$

auf, wobei $B_i = X_i$ gilt, falls X_i ein Nichtterminal und der Index α_i leer sind, und alle anderen B_i sowie C_1, C_2, \dots, C_{k-2} neue zusätzliche Nichtterminale sind, und fügen zum anderen die Regel $B_i \rightarrow X_i[\alpha_i]$ für jedes neue Nichtterminal $B_i \neq X_i$ in eine Menge P''' ein (die so gewonnenen Regeln simulieren nacheinander angewandt die Anwendung der Ausgangsregel, und die Anwendung der Regel $A \rightarrow B_1C_1$ impliziert, dass auch die anderen Regeln angewendet werden müssen).

- Für eine Regel $A \rightarrow B[i_1 i_2 \dots i_k]$ aus P'' oder P''' fügen wir in P' die Regeln

$$A \rightarrow D_k[i_k], D_k \rightarrow D_{k-1}[i_{k-1}], D_{k-1} \rightarrow D_{k-2}[i_{k-2}], \dots, D_3 \rightarrow D_2[i_2], D_2 \rightarrow B[i_1]$$

mit neuen Nichtterminalen D_2, D_3, \dots, D_k ein (erneut wird durch das sukzessive Anwenden dieser Regeln die Ausgangsregel simuliert, und die Anwendung der ersten neuen Regel erfordert die vollständige Simulation).

Damit hat jede Regel aus P' eine der in der Normalform zugelassenen Formen. Wir wählen als N' die Vereinigung aus N und der Menge aller bei der Konstruktion der Regeln in P' neu hinzu genommenen Nichtterminale. Auf Grund der obigen Bemerkungen zu den einzelnen Schritten ergibt sich auch noch $L(G') = L(G'') = L(G)$, womit der Satz bewiesen ist. \square

Nach Satz 1.25 und den Beispielen 1.22–1.24 sind die beiden ersten Forderungen an eine schwach kontextabhängige Grammatikklasse erfüllt. Für die Semi-Linearität betrachten wir das folgende Beispiel.

Beispiel 1.28 *Es sei die indizierte Grammatik*

$$G_7 = \{\{S, A, B\}, \{a\}, \{1, 2\}, \{S \rightarrow A[1], A \rightarrow A[2], A \rightarrow B, B[1] \rightarrow aa, B[2] \rightarrow BB\}, S\}$$

gegeben. Jede ihrer Ableitungen beginnt mit

$$S \Longrightarrow A[1] \Longrightarrow A[21] \Longrightarrow A[221] \Longrightarrow \dots \Longrightarrow A[2^{n-1}1] \Longrightarrow B[2^{n-1}1].$$

Beim Abbau der Indices 2 erfolgt stets eine Verdoppelung der Vorkommen von B . Diese Verdoppelungen kann man sich als parallel ablaufend vorstellen, wenn man den zugehörigen Ableitungsbaum betrachtet. In der Schreibweise der Ableitung der Satzformen erhalten wir

$$B[2^{n-1}1] \Longrightarrow B[2^{n-2}1]B[2^{n-2}1] \xrightarrow{*} (B[2^{n-3}1])^4 \xrightarrow{*} \dots \xrightarrow{*} (B[1])^{2^{n-1}}$$

und haben nun jedes Vorkommen von B durch a^2 zu ersetzen, d. h. wir terminieren die Ableitung mit $(B[1])^{2^{n-1}} \xrightarrow{*} a^{2^n}$. Damit erhalten wir

$$L(G_7) = \{a^{2^n} \mid n \geq 1\}.$$

Die in Beispiel 1.28 erzeugte Sprache ist nicht semi-linear. Folglich ist die dritte Forderung bei der schwachen Kontextabhängigkeit (siehe Definition 1.20) nicht erfüllt.

Bevor wir hierfür einen Ausweg suchen, wollen wir noch die Komplexität des Mitgliedsproblems für indizierte Grammatiken untersuchen. Hier gilt folgender Satz.

Satz 1.29 *Das Mitgliedsproblem für indizierte Grammatiken ist NP-vollständig.*

Wir geben hier nicht den vollständigen Beweis. Wir zeigen nur, dass jedes Problem aus NP in polynomialer Zeit auf das Mitgliedsproblem für indizierte Grammatiken transformiert werden kann, d. h. wir weisen damit nach, dass ein polynomialer (deterministischer) Algorithmus für das Mitgliedsproblem nicht zu erwarten ist. Natürlich reicht es, ein NP-vollständiges Problem auf das Mitgliedsproblem für indizierte Grammatiken zu transformieren. Wir geben eine Transformation vom 3-SAT-Problem an. Dieses ist wie folgt definiert:

- Gegeben:* aussagenlogischer Ausdruck A in konjunktiver Normalform, bei dem jede Alternative genau drei Literale enthält
- Frage:* Gibt es eine Belegung der Variablen so, dass A bei dieser Belegung wahr wird?

Lemma 1.30 *Das 3-SAT-Problem ist NP-vollständig.*

Beweis. Aus der Vorlesung zur Theoretischen Informatik ist bekannt, dass das SAT-Problem

- Gegeben:* aussagenlogischer Ausdruck A in konjunktiver Normalform,
- Frage:* Gibt es eine Belegung der Variablen so, dass A bei dieser Belegung wahr wird?

NP-vollständig ist.

Da das 3-SAT-Problem eine Einschränkung des SAT-Problems darstellt, liegt es in der Menge NP. Wir haben also nur noch zu zeigen, dass jedes Problem aus der Menge NP in polynomialer Zeit auf das 3-SAT-Problem transformiert werden kann. Hierfür reicht es wegen der NP-Vollständigkeit des SAT-Problems zu zeigen, dass es in polynomialer Zeit auf das 3-SAT-Problem transformiert werden kann. Hierzu reicht es offenbar zu zeigen, dass jede Alternative in polynomialer Zeit in eine semantisch äquivalente konjunktive Normalform transformierbar ist, bei der jede Alternative genau drei Literale enthält.

Fall 1. Die zu transformierende Alternative enthalte genau ein Literal p_1 . Dann ersetzen wir sie durch den Ausdruck

$$A_1 = (p_1 \vee x \vee y) \wedge (p_1 \vee x \vee \neg y) \wedge (p_1 \vee \neg x \vee y) \wedge (p_1 \vee \neg x \vee \neg y)$$

mit neuen Variablen x und y . Für jede Belegung der Variablen x und y wird eine der Alternativen nur dann wahr, wenn p_1 wahr wird. Daher wird A_1 genau dann wahr, wenn p_1 wahr wird. Damit ist p_1 genau dann erfüllbar, wenn A_1 erfüllbar ist. Außerdem hat A_1 die gewünschte Form.

Fall 2. Die Alternative bestehe aus zwei Literalen, d. h. sie habe die Form $p_1 \vee p_2$. Dann ersetzen wir sie durch den Ausdruck

$$A_2 = (p_1 \vee p_2 \vee z) \wedge (p_1 \vee p_2 \vee \neg z)$$

mit einer neuen Variablen z . Erneut ist A_2 genau dann erfüllbar, wenn $p_1 \vee p_2$ erfüllbar ist, denn für jede Belegung von z wird eine der Alternativen von A_2 nur wahr, wenn $p_1 \vee p_2$ schon wahr wird.

Fall 3. Die Alternative habe die Form $A = p_1 \vee p_2 \vee \dots \vee p_k$ mit $k \geq 4$. Dann ersetzen wir A durch

$$A_3 = (p_1 \vee p_2 \vee x_1) \wedge (p_3 \vee \neg x_1 \vee x_2) \wedge (p_4 \vee \neg x_2 \vee x_3) \wedge \dots \\ \dots \wedge (p_{k-2} \vee \neg x_{k-4} \vee x_{k-3}) \wedge (p_{k-1} \vee p_k \vee \neg x_{k-3})$$

mit neuen Variablen x_1, x_2, \dots, x_{k-3} .

Nehmen wir an, dass A erfüllbar ist. Bei einer erfüllenden Belegung wird ein Literal, sagen wir p_i , wahr. Belegen wir alle zusätzlichen Variablen x_j , $1 \leq j \leq i - 2$ auch mit

wahr und x_r , $i - 1 \leq r \leq k - 3$, mit falsch, so nimmt jede Alternative von A_3 den Wert wahr an. Damit ist A_3 erfüllbar.

Nehmen wir an, dass A_3 erfüllbar ist, und es sei α eine erfüllende Belegung. Falls x_{k-3} bei α mit wahr belegt wird, so ist die letzte Alternative nur dann wahr, wenn $p_{k-1} \vee p_k$ wahr ist. Folglich ist dann auch A unter α wahr, womit A als erfüllbar nachgewiesen ist. Es sei daher x_{k-3} bei α mit falsch belegt. Falls x_{k-4} mit wahr belegt ist, so wird die vorletzte Alternative von A_3 nur wahr, wenn p_{k-2} und damit auch A wahr wird. Erneut erhalten wir die Erfüllbarkeit von A . So fortfahrend ergibt sich aus der Erfüllbarkeit von A_3 die von A . \square

Beweis von Satz 1.29. Wir betrachten nun die indizierte Grammatik

$$G = (\{S, A, B, W, F\}, \{1, (,), \wedge, \vee, \neg\}, \{w, f\}, P, S)$$

mit

$$\begin{aligned} P = & \{S \rightarrow A[w], S \rightarrow A[f], A \rightarrow A[w], A \rightarrow A[f], A \rightarrow B\} \\ & \cup \{B \rightarrow (\alpha \vee \beta \vee \gamma) \wedge B \mid \alpha, \beta, \gamma \in \{W, \neg W, F, \neg F\}, (\alpha, \beta, \gamma) \notin \{\neg W, F\}^3\} \\ & \cup \{B \rightarrow (\alpha \vee \beta \vee \gamma) \mid \alpha, \beta, \gamma \in \{W, \neg W, F, \neg F\}, (\alpha, \beta, \gamma) \notin \{\neg W, F\}^3\} \\ & \cup \{W[w] \rightarrow 1, W[w] \rightarrow W1, F[w] \rightarrow F1, W[f] \rightarrow W1, F[f] \rightarrow 1, F[f] \rightarrow F1\}. \end{aligned}$$

Ohne Benutzung von Index streichenden Regeln ergeben sich Ableitungen der Form

$$\begin{aligned} S & \Longrightarrow A[i_1] \Longrightarrow A[i_2 i_1] \Longrightarrow \cdots \Longrightarrow A[i_t i_{t-1} \dots i_1] \Longrightarrow B[i_t i_{t-1} \dots i_1] \\ & \Longrightarrow (\alpha_1 \vee \beta_1 \vee \gamma_1) \wedge B[i_t i_{t-1} \dots i_1] \\ & \stackrel{*}{\Longrightarrow} (\alpha_1 \vee \beta_1 \vee \gamma_1) \wedge (\alpha_2 \vee \beta_2 \vee \gamma_2) \wedge \cdots \wedge (\alpha_s \vee \beta_s \vee \gamma_s), \end{aligned}$$

wobei für $1 \leq i \leq s$

$$\alpha_i, \beta_i, \gamma_i \in \{W[i_t \dots i_1], \neg W[i_t \dots i_1], F[i_t \dots i_1], \neg F[i_t \dots i_1]\}$$

gilt. Die Folge der Indices ist ein Wort über $\{w, f\}$. Beim Abbau dieser Folge wird aus jedem W und jedem F eine Folge von Einsen produziert. Falls 1^k erzeugt wird, so interpretieren wir dies als Variable x_k . Gehen wir davon aus, dass die Ausgangswerte W und F eine Belegung mit wahr bzw. falsch liefern, so sind alle Alternativen wahr. Wir haben daher nur zu sichern, dass eine Folge 1^k nicht sowohl von einem W als auch einem F abgeleitet werden kann, da dann gesichert ist, dass die Variable x_k mit genau einem Wahrheitswert bei allen ihren Vorkommen belegt wird. Dies ist aber gesichert, da ein Terminieren nur möglich ist, wenn die Situation $W[wu]$ oder $F[fu]$ für ein $u \in \{w, f\}^*$ vorliegt, d. h. bei der Folge $i_t i_{t-1} \dots i_1$ kann die Variable x_k nur dann mit wahr belegt werden, wenn $i_{t-k+1} = w$ gilt.

Somit erzeugt G genau die Menge der erfüllbaren konjunktiven Normalformen, bei denen jede Alternative genau drei Literale hat, wobei der Variablen x_k gerade 1^k entspricht. Daher ist eine gegebene konjunktive Normalform, die in jeder Alternative genau drei Literale enthält genau dann erfüllbar, wenn das zugehörige Wort in $L(G)$ liegt. Da G fest ist und aus der Normalform das zugehörige Wort w in quadratischer Zeit erzeugt werden kann (x_k wird durch 1^k kodiert), haben wir damit 3-SAT polynomial auf das Mitgliedsproblem indizierter Grammatiken transformiert. \square

Zwar ist die Frage, ob für NP-vollständige Probleme ein (deterministischer) polynomialer Algorithmus existiert, noch ungeklärt, aber es wird heute davon ausgegangen, dass dies nicht der Fall ist. Somit ist wegen Satz 1.29 nicht zu erwarten, dass es einen polynomialen Algorithmus für das Mitgliedsproblem indizierter Grammatiken gibt. Damit ist wahrscheinlich auch eine weitere Forderung für eine schwach kontextabhängige Grammatikklasse nicht erfüllt.

1.3.2. Lineare indizierte Grammatiken

Wir betrachten nun unsere Beispiele und Beweise etwas näher. Die negativen Eigenschaften resultieren offenbar daraus, dass wir die Indexfolge an jedes Nichtterminal auf der rechten Seite einer Regel übergeben, wenn wir die Regel anwenden. In Beispiel 1.28 wird zuerst eine Indexfolge aufgebaut, deren Länge im Wesentlichen die Anzahl der nachfolgend auszuführenden Verdoppelungen angibt; anschließend wird die Folge überall so abgebaut, dass in jedem Zweig jeweils Verdopplungen erfolgen, d. h. alle Zweige weisen die gleiche Anzahl von Verdopplungen auf. Beim Beweis von Satz 1.29 wird zuerst eine Folge aufgebaut, die beim Abbau sichert, dass eine Variable nicht sowohl mit *wahr* als auch mit *falsch* belegt werden kann.

Dies legt es nahe, hier eine schwächere Forderung zu erheben. Auf den ersten Blick könnte man denken, dass es reicht, nur lineare Regeln, also Regeln der Form

$$A \rightarrow uB[\alpha]v, \quad A \rightarrow w, \quad A[i] \rightarrow uBv, \quad A[i] \rightarrow w \quad \text{mit } A, B \in N, \quad u, v, w \in T^*, \quad i \in I, \quad \alpha \in I^*,$$

zu betrachten, da diese für die Erzeugung der drei Sprachen K_1 , K_2 und K_3 ausreichend sind. Jedoch wird dann nicht zu garantieren sein, dass alle kontextfreien Sprachen erzeugt werden können.

Aus diesen beiden Betrachtungen resultiert die Idee der linearen indizierten Grammatik, bei der die Regeln wie bei einer indizierten Grammatik sind, aber die Indexfolge während des Ableitens jeweils nur an ein Nichtterminal der rechten Seite übergeben wird. Dieser Spezialfall indizierter Grammatiken wurde von GERALD GAZDAR in [5] eingeführt.

Definition 1.31

i) Eine lineare indizierte Grammatik ist ein 5-Tupel $G = (N, T, I, P, S)$, wobei

- N , T und S wie bei einer Regelgrammatik bestimmt sind,
- I eine endliche Menge von Indices ist,
- P eine endliche Menge von Regeln der Form

$$A \rightarrow u_1 X_1 [\alpha_1] u_2 X_2 [\alpha_2] \dots u_{i-1} X_{i-1} [\alpha_{i-1}] u_i \underline{X_i} [\alpha_i] u_{i+1} X_{i+1} [\alpha_{i+1}] \dots u_n X_n [\alpha_n] u_{n+1}$$

mit $A, X_1 X_2, \dots, X_n \in N, \quad u_1, u_2, \dots, u_{n+1} \in T^*, \quad \alpha_1, \alpha_2, \dots, \alpha_n \in I^*,$

$$A[i] \rightarrow u_1 X_1 u_2 X_2 \dots u_{i-1} X_{i-1} u_i \underline{X_i} u_{i+1} X_{i+1} \dots u_n X_n u_{n+1}$$

mit $A, X_1 X_2, \dots, X_n \in N, \quad u_1, u_2, \dots, u_{n+1} \in T^*$

ist.

ii) Es seien $G = (N, T, I, P, S)$ eine lineare indizierte Grammatik sowie x und y zwei Wörter aus $(NI^* \cup T)^+$. Wir sagen, dass y aus x durch einen Ableitungsschritt entsteht (oder x erzeugt y in einem Ableitungsschritt, geschrieben als $x \Longrightarrow y$), wenn eine der beiden folgenden Situationen ① oder ② zutrifft:

- ① – $x = x_1 A [\beta] x_2$ mit $A \in N$, $\beta \in I^*$, $x_1, x_2 \in (NI^* \cup T)^*$,
– $A \rightarrow w_1 \underline{X}_i [\alpha_i] w_2 \in P$ mit

$$w_1 = u_1 X_1 [\alpha_1] u_2 X_2 [\alpha_2] \dots u_{i-1} X_{i-1} [\alpha_{i-1}] u_i \text{ und}$$

$$w_2 = u_{i+1} X_{i+1} [\alpha_{i+1}] \dots u_n X_n [\alpha_n] u_{n+1},$$

$$– y = x_1 w_1 X_i [\alpha_i \beta] w_2 x_2,$$

oder

- ② – $x = x_1 A [i \beta] x_2$ mit $A \in N$, $i \in I$, $\beta \in I^*$, $x_1, x_2 \in (NI^* \cup T)^*$,
– $A [i] \rightarrow u_1 X_1 u_2 X_2 \dots u_{i-1} X_{i-1} u_i X_i u_{i+1} X_{i+1} \dots u_n X_n u_{n+1} \in P$,
– $y = x_1 u_1 X_1 u_2 X_2 \dots u_{i-1} X_{i-1} u_i X_i [\beta] u_{i+1} X_{i+1} \dots u_n X_n u_{n+1} x_2$.

Mit \Longrightarrow^* sei der reflexive und transitive Abschluss von \Longrightarrow bezeichnet.

iii) Die von einer linearen indizierten Grammatik $G = (N, T, I, P, S)$ erzeugte Sprache ist durch

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

definiert.

Falls in einer Regel auf der rechten Seite mindestens ein Nichtterminal existiert, so gibt es unter den Nichtterminalen ein ausgezeichnetes Nichtterminal. Dieses kennzeichnen wir durch Unterstreichen, falls noch ein anderes Nichtterminal vorkommt. Beim Anwenden einer Regel wird im Fall ① die Indexfolge und im Fall ② die um i gekürzte Indexfolge des Nichtterminals auf der linken Seite nur an dieses ausgezeichnete Nichtterminal übergeben.

Mit $\mathcal{L}(LI)$ bezeichnen wir die Menge der Sprachen, die von linearen indizierten Grammatiken erzeugt werden können.

Da die indizierten Grammatiken G_1 , G_5 und G_6 aus den Beispielen 1.22–1.24 auf der rechten Seite höchstens ein Nichtterminal haben, kann man sie als lineare indizierte Grammatiken auffassen, indem man das einzige Nichtterminal auf der rechten Seite jeweils auszeichnet. Der Ableitungsprozess ändert sich damit nicht, und wir erhalten daher, dass die nicht-kontextfreien Sprachen K_1 , K_2 und K_3 in $\mathcal{L}(LI)$ liegen.

Satz 1.32 $\mathcal{L}(CF) \subset \mathcal{L}(LI) \subset \mathcal{L}(I)$.

Beweis. Der Beweis für $\mathcal{L}(CF) \subseteq \mathcal{L}(LI)$ kann völlig analog zu dem für $\mathcal{L}(CF) \subseteq \mathcal{L}(I)$ geführt werden. Aus den Beispielen 1.22–1.24 und vorstehender Bemerkung folgt, dass die Inklusion sogar echt ist.

Es sei nun $L \in \mathcal{L}(LI)$ und $G = (N, T, I, P, S)$ eine lineare indizierte Grammatik, die $L(G) = L$ erfüllt. Dazu konstruieren wir die indizierte Grammatik

$$G' = (N \cup \{ A' \mid A \in N \}, T, I \cup \{ \# \}, P', S),$$

wobei P' durch

$$\begin{aligned} P' = & \{ A \rightarrow u_1 X_1 [\alpha_1 \#] \dots u_{i-1} X_{i-1} [\alpha_{i-1} \#] u_i X_i [\alpha_i] u_{i+1} X_{i+1} [\alpha_{i+1} \#] \dots u_n X_n [\alpha_n \#] u_{n+1} \mid \\ & A \rightarrow u_1 X_1 [\alpha_1] \dots u_{i-1} X_{i-1} [\alpha_{i-1}] u_i X_i [\alpha_i] u_{i+1} X_{i+1} [\alpha_{i+1}] \dots u_n X_n [\alpha_n] u_{n+1} \in P \} \\ \cup & \{ A[i] \rightarrow u_1 X'_1 u_2 X'_2 \dots u_{i-1} X'_{i-1} u_i X_i u_{i+1} X'_{i+1} \dots u_n X'_n u_{n+1} \mid \\ & A[i] \rightarrow u_1 X_1 u_2 X_2 \dots u_{i-1} X_{i-1} u_i X_i u_{i+1} X_{i+1} \dots u_n X_n u_{n+1} \in P \} \\ \cup & \{ A' \rightarrow A[\#] \mid A \in N \} \end{aligned}$$

definiert sei. Zwar erfolgt bei G' eine Übergabe der Indexfolge an alle Nichtterminale, aber entsprechend der Konstruktion steht bei den nicht ausgezeichneten Symbolen die übergebene Folge hinter einem $\#$. Da zum Abbau dieses zusätzlichen Indexelements keine Regel vorhanden ist, kann daher auf die übergebenen Indices nicht zugegriffen werden. Der Abbau der übergebenen Folge ist somit nur möglich, wenn ein terminales Wort erzeugt wird. Somit verlaufen die Ableitungen in G und G' gleichartig, nur dass bei G' überflüssige, hinter $\#$ stehende Indexfolgen auftauchen. Daher gilt $L(G') = L(G) = L$. Somit haben wir $L \in \mathcal{L}(I)$ und damit die Inklusion $\mathcal{L}(LI) \subseteq \mathcal{L}(I)$.

Wir wissen aus Beispiel 1.22, dass

$$L(G_3) = \{ a^n b^n c^n d^n e^n \mid n \geq 1 \} \in \mathcal{L}(I)$$

gilt. Wir geben hier die Idee des Beweises, dass $L(G_3) \notin \mathcal{L}(LI)$ ist, woraus folgt, dass die Inklusion $\mathcal{L}(LI) \subseteq \mathcal{L}(I)$ echt ist.

Wir haben zu sichern, dass an fünf Stellen die gleiche Anzahl von Buchstaben erzeugt wird. Mittels eines Nichtterminals können wir analog zur Erzeugung der linearen Sprache $\{ x^n y^n \mid n \geq 1 \}$ zwei Gleichheiten absichern. Durch Übergabe einer Indexfolge, die die Anzahl der erzeugten Buchstaben beschreibt, können wir auch für ein zweites Paar von Buchstaben Gleichheit der Anzahl sichern. Hierbei erfolgt ein Abbau der Indexfolge. Die Erzeugung des fünften Buchstabens muss von einem weiteren Nichtterminal geschehen, an das aber keine Indexfolge übergeben wird. Damit ist jedoch die geforderte Anzahl des fünften Buchstaben nicht zu realisieren. \square

Wir geben nun ein Normalformresultat für lineare indizierte Grammatiken an, wodurch wir ein Analogon zur Chomsky-Normalform für kontextfreie Grammatiken erreichen.

Satz 1.33 *Zu jeder linearen indizierten Grammatik $G = (N, T, I, P, S)$ gibt es eine lineare indizierte Grammatik $G' = (N', T, I, P', S)$ derart, dass*

- alle Regeln in P' von der Form

$$A \rightarrow BC[i], A \rightarrow \underline{B}[i]C, A[i] \rightarrow BC, A[i] \rightarrow \underline{BC} \text{ oder } A \rightarrow a$$

mit $A, B, C \in N'$, $i \in I$ und $a \in T$ sind und

- $L(G') = L(G)$ gilt.

Beweis. Wir beweisen nur eine schwächere Aussage. Wir zeigen, dass es eine Grammatik $G' = (N', T, I, P', S)$ gibt, bei der alle Regeln in P' von der Form

$$\begin{aligned} A \rightarrow BC[\alpha], A \rightarrow \underline{B}[\alpha]C, A \rightarrow B[\alpha], A[i] \rightarrow BC, A[i] \rightarrow \underline{BC}, A[i] \rightarrow B, \\ A \rightarrow \underline{BC}, A \rightarrow \underline{BC}, A \rightarrow a \end{aligned}$$

mit $A, B, C \in N'$, $i \in I$, $\alpha \in I^*$ und $a \in T$ sind (d. h. wir zeigen nicht, dass Kettenregeln und Regeln ohne Indexänderung entbehrlich sind, und geben auch keine Reduktion der Indexfolge α auf die Länge 1, die analog zum Beweis von Satz 1.27 durchgeführt werden kann).

Wir konstruieren zuerst eine Grammatik $G'' = (N'', T, I, P'', S)$. Dazu sei ein Homomorphismus $h : (N \cup T)^* \rightarrow (N \cup \{a' \mid a \in T\})^*$ durch $h(a) = a'$ für $a \in T$ und $h(A) = A$ für $A \in N$ gegeben. Dann nehmen wir anstelle einer jeden Regel $A \rightarrow w \in P$ oder $A[i] \rightarrow w \in P$ die Regel $A \rightarrow h(w)$ bzw. $A[i] \rightarrow h(w)$ in P'' auf, wobei die (vielleicht vorhandene) Auszeichnung eines Nichtterminals erhalten bleibt. Außerdem fügen wir zu den konstruierten Regeln noch die Regeln $a' \rightarrow a$ in P'' ein. Es ist leicht zu sehen, dass alle Regeln von P'' nun von der Form

$$\begin{aligned} A &\rightarrow C_1[\alpha_1]C_2[\alpha_2] \dots C_{i-1}[\alpha_{i-1}]C_i[\alpha_i]C_{i+1}[\alpha_{i+1}] \dots C_n[\alpha_n] \text{ mit } n \geq 1, \\ A[i] &\rightarrow C_1C_2 \dots C_{i-1}C_iC_{i+1} \dots C_n \text{ mit } n \geq 1, \\ a' &\rightarrow a \end{aligned} \quad (1.1)$$

sind und die erzeugte Sprache bei diesen Änderungen erhalten bleibt. Jede Regel der Form (1.1) ersetzen wir nun durch die Regeln

$$A \rightarrow C_1''C_2'' \dots C_{i-1}''C_i[\alpha_i]C_{i+1}'' \dots C_n'' \quad \text{und} \quad C_j'' \rightarrow C_j[\alpha_j] \text{ für } 1 \leq j \leq n, j \neq i,$$

wobei die C_j'' , $1 \leq j \leq n, j \neq i$, neue Nichtterminale sind. Diese Transformation verändert die erzeugte Sprache ebenfalls nicht. Wir haben damit G'' erhalten, wobei P'' nur Regeln der Form

$$A \rightarrow C_1C_2 \dots C_{i-1}C_i[\alpha]C_{i+1} \dots C_n \text{ mit } n \geq 1, \quad (1.2)$$

$$A[i] \rightarrow C_1C_2 \dots C_{i-1}C_iC_{i+1} \dots C_n \text{ mit } n \geq 1, \quad (1.3)$$

$$A \rightarrow C[\alpha],$$

$$a' \rightarrow a$$

enthält. Um zu G' überzugehen, ersetzen wir jede Regel der Form (1.2) durch die Regeln

$$\begin{aligned} A &\rightarrow C_1D_1, D_1 \rightarrow C_2D_2, D_2 \rightarrow C_3D_3, \dots, D_{i-2} \rightarrow C_{i-1}D_{i-1}, D_{i-1} \rightarrow C_i[\alpha]D_i, \\ D_i &\rightarrow C_{i+1}D_{i+1}, D_{i+1} \rightarrow C_{i+2}D_{i+2}, \dots, D_{n-3} \rightarrow C_{n-2}D_{n-2}, D_{n-2} \rightarrow C_{n-1}C_n \end{aligned}$$

mit neuen Nichtterminalen D_j , $1 \leq j \leq n-2$. Wenn wir beachten, dass an das Nichtterminal D_{i+1} und damit auch an die Nichtterminale D_j , $i+2 \leq j \leq n-2$ keine Indexfolge übergeben werden kann, ist leicht zu beweisen, dass diese Transformation die Sprache nicht verändert. Analog transformieren die Regeln der Form (1.3). Die so entstandene Grammatik G' hat dann nur Regeln der gewünschten Form und erfüllt $L(G') = L(G)$. \square

Die Chomsky-Normalform ist der Ausgangspunkt für den Cocke-Younger-Kasami-Algorithmus. In Satz 1.33 haben wir diese Normalform auf lineare indizierte Grammatiken verallgemeinert. Nun verallgemeinern wir entsprechend [26] den Cocke-Younger-Kasami-Algorithmus auf lineare indizierte Grammatiken.

Satz 1.34 *Das Mitgliedsproblem, ob ein Wort w von einer (festen) linearen indizierten Grammatik erzeugt wird, ist in der Zeit $O(|w|^7)$ lösbar.*

Beweis. Es seien eine lineare indizierte Grammatik $G = (N, T, I, P, S)$ und ein Wort $w = a_1 a_2 \dots a_n \in T^+$ der Länge n gegeben. Wegen Satz 1.33 können wir ohne Beschränkung der Allgemeinheit annehmen, dass in P alle Regeln von der Form

$$A \rightarrow \underline{BC}[\gamma], \quad A \rightarrow \underline{B}[\gamma]C, \quad A[\gamma] \rightarrow \underline{BC}, \quad A[\gamma] \rightarrow \underline{B}C \quad \text{oder} \quad A \rightarrow a$$

mit $A, B, C \in N$, $\gamma \in I$ und $a \in T$ sind.

Erneut konstruieren wir Mengen $L_{i,j}$, $1 \leq i \leq j \leq n$, deren Elemente für die Erzeugung von $a_i a_{i+1} \dots a_j$ verantwortlich sein sollen. Um aber auch die angehängten Indexfolgen berücksichtigen zu können, verwenden wir (anstelle der Nichtterminale im kontextfreien Fall) Sechstupel. Genauer gesagt, betrachten wir in $L_{i,j}$ die Tupel

- $(A, \gamma, A', \gamma', p, q)$ mit $A, A' \in N$, $\gamma, \gamma' \in I$ und $i \leq p \leq q \leq j$ und $(i, j) \neq (p, q)$ (wobei wir annehmen, dass Ableitungen

$$A[\gamma] \xRightarrow{*} a_i a_{i+1} \dots a_{p-1} \underline{A'} a_{q+1} a_{q+2} \dots a_j \quad \text{und} \quad A'[\gamma' \alpha] \xRightarrow{*} a_p a_{p+1} \dots a_q$$

für gewisse $\alpha \in I^*$ existieren, woraus auch

$$\begin{aligned} A[\gamma \gamma' \alpha] &\xRightarrow{*} a_i a_{i+1} \dots a_{p-1} \underline{A'}[\gamma' \alpha] a_{q+1} a_{q+2} \dots a_j \\ &\xRightarrow{*} a_i a_{i+1} \dots a_{p-1} a_p a_{p+1} \dots a_q a_{q+1} \dots a_j \end{aligned}$$

resultiert),

- $(A, \gamma, A', -, p, q)$ mit $A, A' \in N$, $\gamma \in I$ und $i \leq p \leq q \leq j$ und $(i, j) \neq (p, q)$ (wobei wir annehmen, dass Ableitungen

$$A[\gamma] \xRightarrow{*} a_i a_{i+1} \dots a_{p-1} \underline{A'} a_{q+1} a_{q+2} \dots a_j \quad \text{und} \quad A'[\alpha] \xRightarrow{*} a_p a_{p+1} \dots a_q$$

existieren, woraus sich erneut $A[\gamma \alpha] \xRightarrow{*} a_i a_{i+1} \dots a_j$ ergibt),

- $(A, -, -, -, -, -)$ mit $A \in N$ (wobei wir annehmen, dass eine Ableitung

$$A \xRightarrow{*} a_i a_{i+1} \dots a_j$$

existiert).

Dann wird $w \in L(G)$ oder äquivalent $S \xRightarrow{*} a_1 a_2 \dots a_n$ durch $(S, -, -, -, -, -) \in L_{1,n}$ beschrieben.

Wir wollen nun zeigen, wie die Mengen $L_{i,j}$ (unter Beachtung der Bedingungen an die Tupel) schrittweise konstruiert werden können, wobei wir mit den Mengen $L_{i,i}$ beginnen und dann bei jedem Schritt die Differenz zwischen den Indices von $L_{i,j}$ erhöhen.

Induktionsanfang: Für $1 \leq i \leq n$ setzen wir $L_{i,i} = \{(A, -, -, -, -, -)\}$, falls $A \rightarrow a_i$ in P liegt. Hierfür gilt dann – wie gewünscht – $A \xRightarrow{*} a_i$.

Induktionsschritt: Wir konstruieren $L_{i,j}$ aus $L_{i,k}$ und $L_{k+1,j}$, $i \leq k < j$ in Abhängigkeit von der angewendeten Regel.

a) Falls

$$A[\gamma] \rightarrow A_1 \underline{A_2}, (A_1, -, -, -, -, -) \in L_{i,k} \text{ und } (A_2, \gamma_2, A_3, \gamma_3, p, q) \in L_{k+1,j}$$

oder

$$A[\gamma] \rightarrow A_1 \underline{A_2}, (A_1, -, -, -, -, -) \in L_{i,k} \text{ und } (A_2, \gamma_2, A_3, -, p, q) \in L_{k+1,j}$$

gelten, so nehmen wir $(A, \gamma, A_2, \gamma_2, k+1, j)$ in $L_{i,j}$ auf. In diesem Fall haben wir nach Voraussetzung

$$A_1 \xRightarrow{*} a_i a_{i+1} \dots a_k, A_2[\gamma_2 \alpha] \xRightarrow{*} a_{k+1} a_{k+2} \dots a_j \text{ für ein } \alpha \in I^*$$

und erhalten daraus wie gewünscht

$$A[\gamma \gamma_2 \alpha] \Rightarrow A_1 \underline{A_2}[\gamma_2 \alpha] \xRightarrow{*} a_i a_{i+1} \dots a_k \underline{A_2}[\gamma_2 \alpha] \Rightarrow a_i a_{i+1} \dots a_k a_{k+1} \dots a_j.$$

Die Modifikation für Regeln der Form $A[\gamma] \rightarrow \underline{A_1} A_2$ überlassen wir dem Leser.

b) Falls

$$A \rightarrow A_1 \underline{A_2}[\gamma], (A_1, -, -, -, -, -) \in L_{i,k}, (A_2, \gamma, A_3, \gamma_3, p, q) \in L_{k+1,j} \\ \text{und } (A_3, \gamma_3, A_4, \gamma_4, r, s) \in L_{p,q} \text{ mit } \gamma_3 \neq -$$

gelten, so fügen wir $(A, \gamma_3, A_4, \gamma_4, r, s)$ zu $L_{i,j}$ hinzu. Diesmal haben wir die Relationen

$$\begin{aligned} A_1 &\xRightarrow{*} a_i a_{i+1} \dots a_k, \\ A_4[\gamma_4 \alpha] &\xRightarrow{*} a_r a_{r+1} \dots a_s, \\ A_3[\gamma_3 \gamma_4 \alpha] &\xRightarrow{*} a_p a_{p+1} \dots a_{r-1} A_4[\gamma_4 \alpha] a_{s+1} a_{s+2} \dots a_q \\ &\xRightarrow{*} a_p a_{p+1} \dots a_{r-1} a_r \dots a_s a_{s+1} \dots a_q, \\ A_2[\gamma \gamma_3 \gamma_4 \alpha] &\xRightarrow{*} a_{k+1} a_{k+2} \dots a_{p-1} A_3[\gamma_3 \gamma_4 \alpha] a_{q+1} a_{q+2} \dots a_j \\ &\xRightarrow{*} a_{k+1} a_{k+2} \dots a_{p-1} a_p \dots a_q a_{q+1} \dots a_j \end{aligned} \tag{1.4}$$

als Voraussetzung und erhalten daraus

$$\begin{aligned} A[\gamma_3 \gamma_4 \alpha] &\Rightarrow A_1 \underline{A_2}[\gamma \gamma_3 \gamma_4 \alpha] \\ &\xRightarrow{*} a_i a_{i+1} \dots a_k \underline{A_2}[\gamma \gamma_3 \gamma_4 \alpha] \\ &\xRightarrow{*} a_i a_{i+1} \dots a_k a_{k+1} \dots a_{p-1} A_3[\gamma_3 \gamma_4 \alpha] a_{q+1} a_{q+2} \dots a_j \\ &\xRightarrow{*} a_i a_{i+1} \dots a_k a_{k+1} \dots a_{p-1} a_p \dots a_r \dots a_{r-1} A_4[\gamma_4 \alpha] a_{s+1} a_{s+2} \dots a_q a_{q+1} \dots a_j. \end{aligned}$$

Zusammen mit (1.4) liefert dies gerade die durch $(A, \gamma_3, A_4, \gamma_4, r, s)$ beschriebene Situation.

Falls

$$A \rightarrow A_1 \underline{A_2}[\gamma], (A_1, -, -, -, -, -) \in L_{i,k}, (A_2, \gamma, A_3, -, p, q) \in L_{k+1,j} \\ \text{und } (A_3, -, -, -, -, -) \in L_{p,q}$$

gelten, so fügen wir $(A, -, -, -, -, -)$ zu $L_{i,j}$ hinzu. In diesem Fall haben wir nach Voraussetzung

$$\begin{aligned} A_1 &\xrightarrow{*} a_i a_{i+1} \dots a_k, \\ A_3 &\xrightarrow{*} a_p a_{p+1} \dots a_q, \\ A_2[\gamma] &\xrightarrow{*} a_{k+1} a_{k+2} \dots a_{p-1} A_3 a_{q+1} a_{q+2} \dots a_j \\ &\xrightarrow{*} a_{k+1} a_{k+2} \dots a_{p-1} a_p \dots a_q a_{q+1} \dots a_j \end{aligned}$$

woraus

$$A \Longrightarrow A_1 A_2[\gamma] \xrightarrow{*} a_i a_{i+1} \dots a_k A_2[\gamma] \xrightarrow{*} a_i a_{i+1} \dots a_k a_{k+1} \dots a_j$$

resultiert.

Bei $A \rightarrow \underline{A_1}[\gamma] A_2$ wird analog verfahren.

Wir untersuchen nun die Komplexität dieses Vorgehens. Offensichtlich benötigen wir $\frac{n(n+1)}{2}$ Mengen $L_{i,j}$, $1 \leq i \leq j \leq n$. Die Berechnung beim Induktionsanfang erfordert zur Konstruktion jeder Menge $L_{i,i}$ einen Schritt. Der Fall a) im Induktionsschritt erfordert die Berücksichtigung der Parameter k, p, q mit $1 \leq i \leq k \leq p \leq q \leq j$. Daher gibt es für jeden dieser Parameter höchstens n Möglichkeiten. Daraus ergibt sich höchstens der Aufwand $O(n^3)$. Im Fall b) sind noch zusätzlich die Parameter r und s zu erfassen, womit sich höchstens der Aufwand $O(n^5)$ ergibt. Damit ergibt sich für die Berechnung aller $L_{i,j}$ höchstens der Aufwand $O(n^7)$. \square

Durch eine etwas genauere Analyse des oben beschriebenen Algorithmus kann man zeigen, dass sogar die Zeitschranke $O(|w|^6)$ für das Mitgliedsproblem für lineare indizierte Grammatiken gilt. Wir geben uns mit der unschärferen Variante zufrieden, da wir für die schwache Kontextabhängigkeit nur die Existenz eines polynomialen Verfahrens benötigen.

Um nachzuweisen, dass die linearen indizierten Grammatiken eine Klasse schwach kontextabhängiger Grammatiken bilden, bleibt zu zeigen, dass jede Sprache aus $\mathcal{L}(LI)$ semi-linear ist. Wir werden dies im Abschnitt 1.6. nachtragen.

1.4. Baum einfügende Grammatiken

1.4.1. Definition, Beispiele und Eigenschaften

Entsprechend Definition 1.2 erhalten wir die kontextfreien Sprachen als die von links nach rechts gelesenen Markierungen der Blätter von Ableitungsbäumen, und die Ableitungsbäume erhalten wir durch Substitution eines Knotens durch den zu einer Regel gehörenden Baum. Wir verallgemeinern zuerst die Operation zur Substitution von Bäumen.

Definition 1.35 *Es seien \mathcal{B}_1 und \mathcal{B}_2 zwei Bäume, so dass ein Blatt b von \mathcal{B}_1 und die Wurzel von \mathcal{B}_2 die gleiche Markierung A besitzen. Ein Baum \mathcal{B}_3 entsteht aus \mathcal{B}_1 und \mathcal{B}_2 durch Substitution, indem wir das Blatt b von \mathcal{B}_1 und die Wurzel von \mathcal{B}_2 identifizieren.*

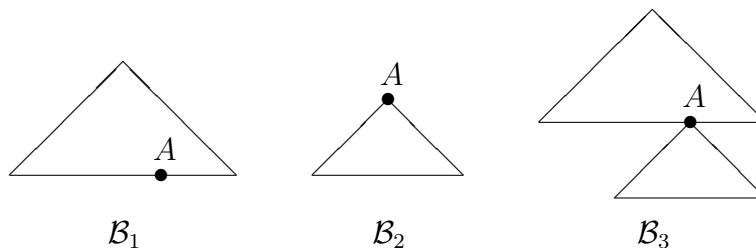


Abbildung 1.7: Substitution von Bäumen

Anschaulich ist die Substitution in Abbildung 1.7 gezeigt.

Im Beweis der Semi-Linearität kontextfreier Sprachen haben wir noch eine weitere Operation benutzt (siehe Abbildung 1.4). Wir wollen nun auch noch diese als Operation auf Bäumen einführen.

Definition 1.36 *Es seien \mathcal{B}_1 und \mathcal{B}_2 zwei Bäume, so dass ein innerer Knoten u von \mathcal{B}_1 , die Wurzel w von \mathcal{B}_2 und ein Blatt b von \mathcal{B}_2 die gleiche Markierung A haben. Dann entsteht ein Baum \mathcal{B}_3 aus \mathcal{B}_1 durch Einfügen von \mathcal{B}_2 , indem wir zuerst aus \mathcal{B}_1 jenen Teilbaum \mathcal{B} , dessen Wurzel u ist, ausschneiden und an seiner Stelle den Baum \mathcal{B}_2 einfügen (wir ersetzen also die Kante (x, u) aus \mathcal{B}_1 durch (x, w)) und dann b mit der Wurzel von \mathcal{B} identifizieren.*

Das Blatt b in \mathcal{B}_2 heißt Fuß oder Fußknoten von \mathcal{B}_2 ; um ihn schon an der Markierung erkennen zu können, werden wir ihn mit einem Stern $*$ kennzeichnen.

Anschaulich ist das Einfügen in Abbildung 1.8 gezeigt.

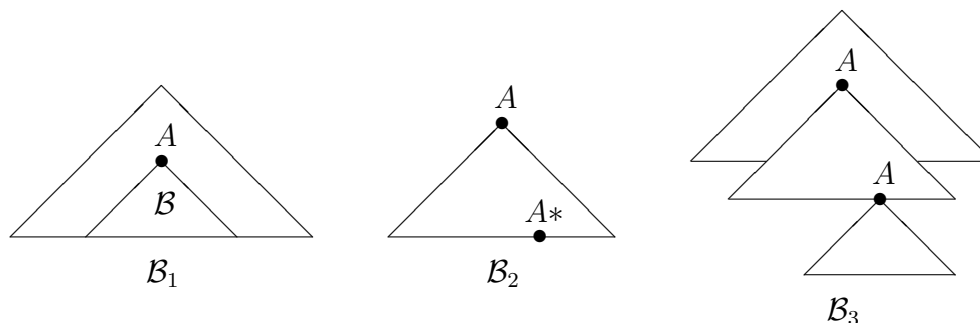


Abbildung 1.8: Einfügen von Bäumen

Wir wollen nun zeigen, dass wir unter Ausnutzung des Einfügens auch Sprachen erzeugen können, die nicht kontextfrei sind. Dazu verwenden wir die Bäume aus Abbildung 1.9.

Wir erhalten dann die in Abbildung 1.10 gezeigten Bäume \mathcal{T}_4 durch Einfügen von \mathcal{T}_2 in \mathcal{T}_1 , \mathcal{T}_5 durch Einfügen von \mathcal{T}_2 in \mathcal{T}_4 und \mathcal{T}_6 durch Einfügen von \mathcal{T}_3 in \mathcal{T}_4 .

Wenn wir – wie auch bei den Ableitungsbäumen kontextfreier Grammatiken – als erzeugte Wörter die von links nach rechts gelesenen Markierungen der Blätter nehmen, so erhalten wir aus \mathcal{T}_1 , \mathcal{T}_4 , \mathcal{T}_5 und \mathcal{T}_6 die Wörter abc , $aabbcc$, $aababbccc$ und $a^3b^3c^3$. Als erzeugte Sprache ergibt sich

$$L = \{wc^n \mid w \in \{a, b\}^+, \#_a(w) = \#_b(w) = n, \\ \#_a(u) \geq \#_b(u) \text{ für jedes Anfangsstück } u \text{ von } w\}.$$

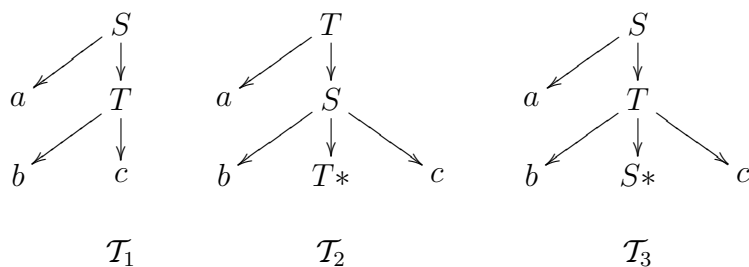


Abbildung 1.9: Bäume \mathcal{T}_1 , \mathcal{T}_2 und \mathcal{T}_3

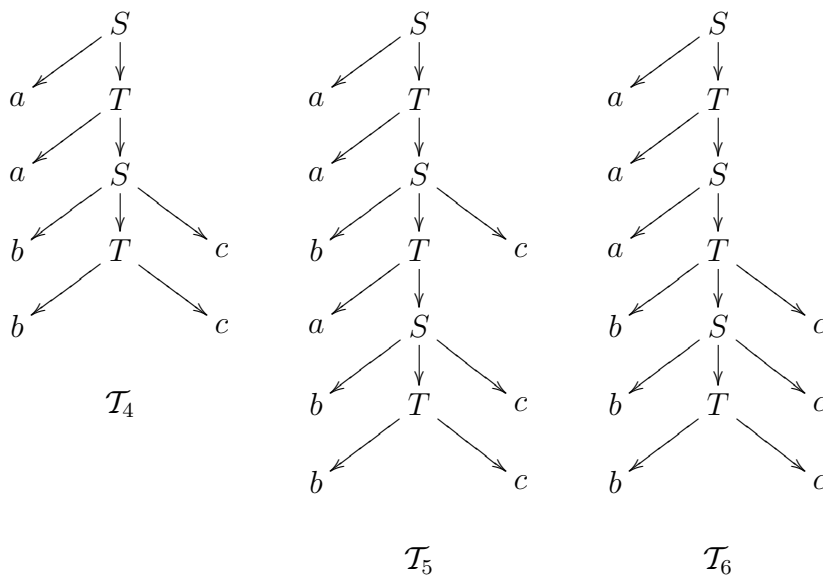


Abbildung 1.10: Erzeugung einiger Bäume aus \mathcal{T}_1 , \mathcal{T}_2 und \mathcal{T}_3 durch Einfügen

Dies folgt daraus, dass jeder Baum \mathcal{T}_1 , \mathcal{T}_2 und \mathcal{T}_3 genau ein Vorkommen von a , b und c als Blattmarkierung hat und sie auch in dieser Reihenfolge auftreten. Offenbar gilt

$$L \cap \{a\}^+ \{b\}^+ \{c\}^+ = \{a^n b^n c^n \mid n \geq 1\} = K_1.$$

Da der Durchschnitt von L mit einer regulären Sprache eine Sprache liefert, die nicht kontextfrei ist, ist L wegen Satz 1.18 auch keine kontextfreie Sprache.

Allerdings können wir K_1 nicht auf diese Weise erzeugen. Die Bäume, in die wir einfügen, müssen stets eine Blattfolge haben, die zu K_1 gehört. Gleiches muss für die unter Auslassung des Fußes gegebene Folge der Blattmarkierungen der eingefügten Bäume gelten. Nun kann eine Folge von Einfügungen konstruiert werden, die einen Baum ergibt, dessen Blattmarkierungen beim Lesen von links nach rechts einen Wechsel von b nach a haben; folglich gehört das hiervon induzierte Wort nicht zu K_1 .

Wir müssen also eine Einschränkung hinsichtlich der Anwendbarkeit des Baum-Einfügens vornehmen. Um dies zu erreichen, werden wir zusätzlich zu dem Nichtterminal in der Markierung noch eine Teilmenge von Bäumen angeben, die bei diesem Knoten eingefügt werden dürfen. Insbesondere können wir durch die Zusatzmarkierung mit der leeren Menge also erreichen, dass keine Baum-Einfügung bei diesem Knoten erfolgen darf. Außerdem werden wir noch eine weitere Komponente bei der Markierung verwenden,

durch die wir erreichen wollen, dass bei diesem Knoten eine Einfügung stattfinden muss; dafür verwenden wir den Wert 1, während der Wert 0 angibt, dass eine Einfügung erfolgen kann aber nicht muss.

Definition 1.37

- i) Eine Baum einfügende Grammatik ist ein 5-Tupel $G = (N, T, Init, Aux, S)$, wobei
- N und T Alphabete von Nichtterminalen bzw. Terminalen sind,
 - $S \in N$ gilt,
 - $Init$ eine endliche, nicht-leere Menge von Bäumen ist,
 - deren innere Knoten mit Tripeln (A, U, a) mit $A \in N$, $U \subseteq Aux$ und $a \in \{0, 1\}$ markiert sind und
 - deren Blätter mit Elementen aus $T \cup \{\lambda\}$ oder Paaren (A, \downarrow) für $A \in N$ markiert sind, und
 - Aux eine endliche Menge von Bäumen ist,
 - zu denen es $A \in N$, $U, U' \subseteq Aux$ sowie $a, a' \in \{0, 1\}$ und ein Blatt b derart gibt, dass die Wurzel mit (A, U, a) und das Blatt b mit (A, U', a') markiert sind (dieses Blatt ist der Fuß des Baumes),
 - deren weitere Blätter (die vom Fuß b verschieden sind) mit Elementen aus $T \cup \{\lambda\}$ oder Paaren (A, \downarrow) für $A \in N$ markiert sind, und
 - deren innere Knoten mit Tripeln (B, U, a) für $B \in N$, $U \subseteq Aux$ und $a \in \{0, 1\}$ markiert sind.
- ii) Die Menge $T(G)$ der von G erzeugten Bäume wird induktiv durch folgende Bedingungen definiert:
- Alle Bäume aus $Init$ gehören zu $T(G)$.
 - Sind \mathcal{B}_1 und \mathcal{B}_2 Bäume aus $T(G)$, ist (A, U, a) die Markierung der Wurzel von \mathcal{B}_2 und gibt es in \mathcal{B}_1 ein Blatt x mit der Markierung (A, \downarrow) , so gehört zu $T(G)$ auch jener Baum \mathcal{B}_3 , der durch Substitution von \mathcal{B}_2 in x entsteht, wobei x in \mathcal{B}_3 mit (A, U, a) markiert wird.
 - Zu einem Baum \mathcal{B}_1 aus $T(G)$, der einen Knoten y mit der Markierung (A, U, a) enthält, und einem Baum \mathcal{B}_2 aus U , dessen Wurzel mit (A, U', a') markiert ist, gehört jener Baum \mathcal{B}_3 zu $T(G)$, der aus \mathcal{B}_1 durch Baum-Einfügen von \mathcal{B}_2 im Knoten y entsteht, wobei y in \mathcal{B}_3 durch (A, U', a') markiert wird.
 - Zu jedem Baum \mathcal{B} in $T(G)$ gibt es eine natürliche Zahl n derart, dass \mathcal{B} durch n -maliges Substituieren oder Einfügen von Bäumen aus Aux in einen Baum aus $Init$ entsteht.
- iii) Die von G erzeugte Sprache $L(G)$ besteht aus allen Wörtern über T , die als von links nach rechts gelesene Folge der Blattmarkierungen von Bäumen entstehen, deren Wurzel mit einem Tripel $(S, U, 0)$ für eine gewisse Menge $U \subseteq Aux$ markiert ist und deren innere Knoten alle durch Elemente der Form $(A, U', 0)$, $A \in N$ und $U' \subseteq Aux$ markiert sind.

Eine Variante Baum einfügender Grammatiken wurde erstmals in [11] eingeführt. Die dortige Definition weicht jedoch von unserer Definition leicht ab, und in der Folgezeit wurden mehrere Modifikationen der Definition vorgenommen. Wir folgen hier der Definition aus [27]. Erste wichtige Resultate zu formalen Eigenschaften Baum einfügender Grammatiken wurden in [25] und [24] angegeben.

Mit $\mathcal{L}(TA)$ bezeichnen wir die Menge der Sprachen, die von Baum einfügenden Grammatiken erzeugt werden.¹

Wir betrachten drei Beispiele.

Beispiel 1.38 *Es sei die Baum einfügende Grammatik*

$$G_1 = (\{S, T\}, \{a, b, c\}, \text{Init}_1, \text{Aux}_1, S) \quad \text{mit} \quad \text{Init}_1 = \{\mathcal{B}_1\} \quad \text{und} \quad \text{Aux}_1 = \{\mathcal{B}_2, \mathcal{B}_3\}$$

gegeben, wobei die Bäume \mathcal{B}_1 , \mathcal{B}_2 und \mathcal{B}_3 aus Abbildung 1.11 zu ersehen sind. Wir erhalten

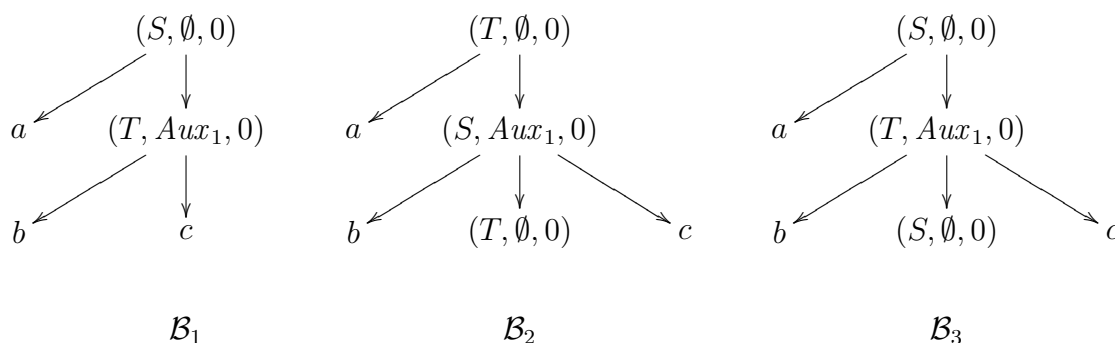


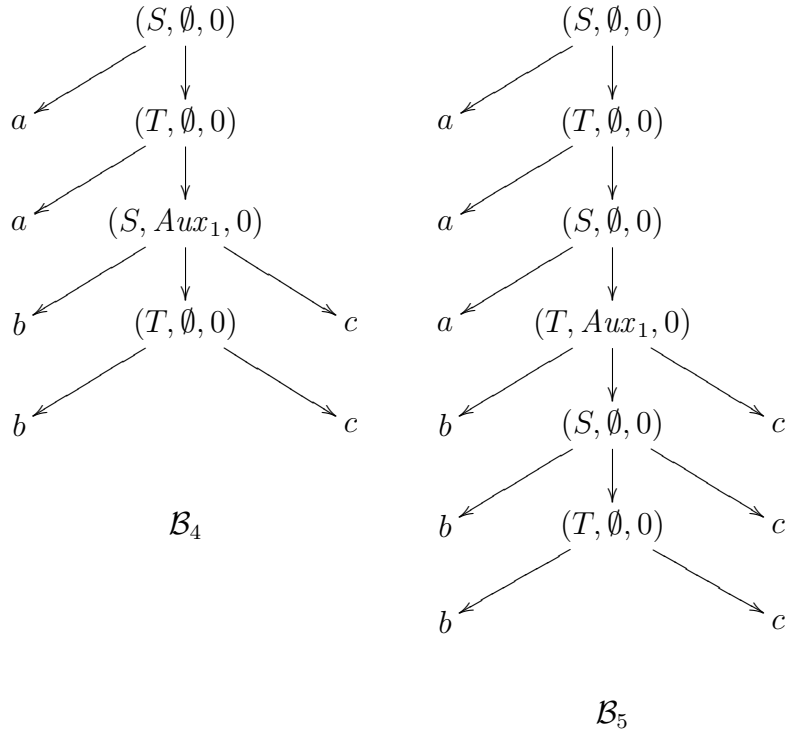
Abbildung 1.11: Bäume der Baum einfügenden Grammatik G_1 aus Beispiel 1.38

dann der Reihe nach die in Abbildung 1.12 gegebenen Bäume \mathcal{B}_4 aus \mathcal{B}_1 durch Einfügen von \mathcal{B}_2 und \mathcal{B}_5 durch Einfügen von \mathcal{B}_3 in \mathcal{B}_4 . Wir sehen, dass wir durch $\mathcal{B}_1 \in \text{Init}_1$ und die beiden eben konstruierten Bäume \mathcal{B}_4 und \mathcal{B}_5 nachgewiesen haben, dass abc , $a^2b^2c^2$ und $a^3b^3c^3$ Wörter der von G_1 erzeugten Sprache sind.

Wir zeigen nun durch vollständige Induktion, dass $L(G_1) = K_1$ gilt. Dazu behaupten wir, dass die erzeugten Bäume stets die folgenden drei Bedingungen erfüllen:

- Es gibt einen Pfad, der in der Wurzel $(S, \emptyset, 0)$ beginnt, dann abwechselnd die Knoten $(T, \emptyset, 0)$ und $(S, \emptyset, 0)$ hat und in einem Knoten $(A, \text{Aux}_1, 0)$ mit $A \in \{S, T\}$ endet. Es sei n die Anzahl der Kanten in diesem Pfad.
- An jedem Knoten des Pfades, der die leere Menge in der zweiten Komponente enthält, hängt ein Blatt a . Diese Blätter ergeben das Wort a^n .
- Die Blätter des Teilbaums mit der Wurzel $(A, \text{Aux}_1, 0)$ bilden das Wort $b^n c^n$. Alle von der Wurzel und den Blättern verschiedenen Knoten dieses Teilbaumes haben eine Markierung $(B, \emptyset, 0)$ mit $B \in \{S, T\}$.

¹Die Abkürzung entstammt der englischen Wortgruppe *tree adjoining* für *Baum einfügend*.

Abbildung 1.12: Bäume aus $T(G_1)$ für die Baum einfügende Grammatik G_1 aus Beispiel 1.38

Der Baum \mathcal{B}_1 aus $Init_1$ erfüllt diese Eigenschaften (mit $n = 1$).

Es sei ein Baum solcher Struktur gegeben. Wir können nun nur beim Knoten mit der Markierung $(A, Aux_1, 0)$ den Baum \mathcal{B}_2 oder \mathcal{B}_3 einfügen (in Abhängigkeit davon, ob $A = T$ oder $A = S$ gilt). Es ergibt sich dann erneut ein Baum dieser Struktur, jedoch wird durch \mathcal{B}_2 bzw. \mathcal{B}_3 jeweils ein weiteres Vorkommen von a , b und c hinzugefügt.

Beispiel 1.39 Wir betrachten die Baum einfügende Grammatik

$$G_2 = (\{S, T, U\}, \{a, b, c, d\}, Init_2, Aux_2, S) \text{ mit } Init_2 = \{\mathcal{B}_1\} \text{ und } Aux_2 = \{\mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4\}$$

mit den Bäumen \mathcal{B}_1 , \mathcal{B}_2 , \mathcal{B}_3 und \mathcal{B}_4 aus Abbildung 1.13. Man überzeugt sich leicht, dass durch wiederholtes Einfügen von \mathcal{B}_2 Bäume mit folgenden Eigenschaften entstehen:

- Es gibt einen (mittleren) Pfad $(S, \emptyset, 0)(T, \emptyset, 0)^m(T, Aux_2, 0)(T, \emptyset, 0)^m$ mit $m \geq 1$.
- An $(S, \emptyset, 0)$ hängt nach links ein Blatt a und nach rechts ein Blatt d .
- An den folgenden m Knoten $(T, \emptyset, 0)$ hängt jeweils nach links ein Blatt a .
- An $(T, Aux_2, 0)$ und den folgenden $m - 1$ Knoten $(T, \emptyset, 0)$ hängt jeweils ein Blatt c nach links.
- Am verbleibenden letzten Knoten $(T, \emptyset, 0)$ hängen nach links ein b und nach rechts ein c .

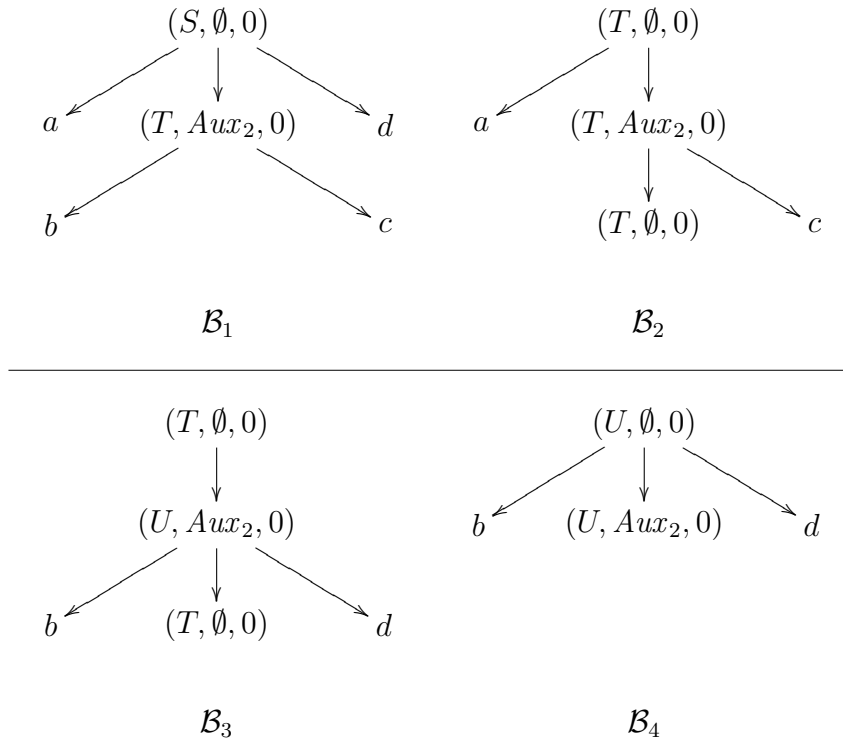


Abbildung 1.13: Bäume der Baum einfügenden Grammatik G_2 aus Beispiel 1.39

Die Blätter ergeben folglich ein Wort der Form $a^{m+1}bc^{m+1}d$.

Nach einer gewissen Anzahl von Schritten wird einmal \mathcal{B}_3 eingefügt und anschließend wird noch beliebig oft \mathcal{B}_4 eingefügt. Hierdurch werden jeweils ein Vorkommen von b hinter den a s und ein Vorkommen von c vor dem d aus dem initialen Baum \mathcal{B}_1 erzeugt.

Damit ergibt sich

$$L(G_2) = \{a^p b^q c^p d^q \mid p, q \geq 1\} = K_2.$$

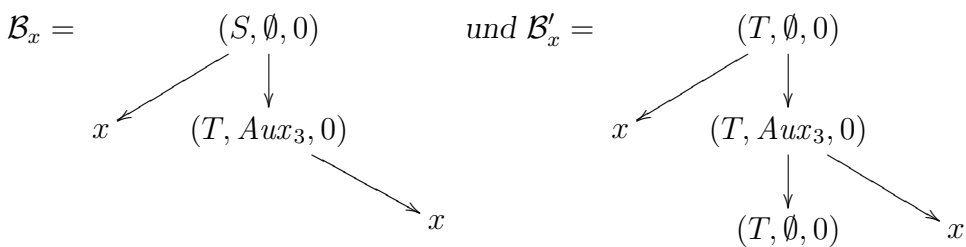
Beispiel 1.40 Die Baum einfügende Grammatik

$$G_3 = (\{S, T\}, \{a, b\}, \text{Init}_3, \text{Aux}_3, S)$$

mit

$$\text{Init}_3 = \{\mathcal{B}_x \mid x \in \{a, b\}\} \text{ und } \text{Aux}_3 = \{\mathcal{B}'_x \mid x \in \{a, b\}\}$$

sowie den Bäumen

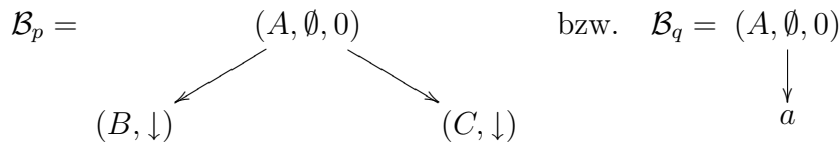


erzeugt die Sprache $L(G_3) = \{ww \mid w \in \{a, b\}^+\} = K_3$. Der Nachweis hierfür bleibt dem Leser überlassen.

Das nächste Resultat ist eine erste Aussage über die Erzeugungskraft Baum einfügender Grammatiken. Wir werden später noch eine Präzisierung vornehmen.

Satz 1.41 $\mathcal{L}(CF) \subset \mathcal{L}(TA)$.

Beweis. Es sei L eine kontextfreie Sprache. Dann gibt es eine kontextfreie Grammatik $G = (N, T, P, S)$ in Chomsky-Normalform (d. h. alle Regeln aus P sind von der Form $A \rightarrow BC$ oder $A \rightarrow a$ mit $A, B, C \in N$ und $a \in T$) mit $L(G) = L$. Wir konstruieren nun eine Baum einfügende Grammatik $G' = (N, T, Init, \emptyset, S)$, wobei die Menge $Init$ wie folgt erhalten wird: Für jede Regel $p = A \rightarrow BC \in P$ und $q = A \rightarrow a \in P$ nehmen wir in $Init$ den Baum



auf. Es ist offensichtlich, dass jeder Anwendung einer Regel p (bzw. q) eine Substitution des Baumes \mathcal{B}_p (bzw. \mathcal{B}_q) entspricht und umgekehrt. Folglich erzeugt G' ebenfalls die Sprache $L(G)$, womit bewiesen ist, dass $L \in \mathcal{L}(TA)$ gilt. Somit haben wir $\mathcal{L}(CF) \subseteq \mathcal{L}(TA)$.

Dass diese Inklusion echt ist, folgt aus den Beispielen 1.38 – 1.40. \square

In den Beispielen haben wir keine Substitutionen benutzt, um die Sprachen zu erzeugen. Wir wollen nun zeigen, dass dies auch allgemein gilt, d. h. es reicht, bei der Erzeugung nur Baum-Einfügungen vorzunehmen.

Satz 1.42 *Zu jeder Baum einfügenden Grammatik $G = (N, T, Init, Aux, S)$ gibt es eine Baum einfügende Grammatik $G' = (N', T, Init', Aux', S)$ derart, dass die Bäume in $Init'$ und Aux' keine Knoten mit einer Markierung (A, \downarrow) , $A \in N'$ enthalten (also keine Substitutionen möglich sind) und $L(G') = L(G)$ gilt.*

Beweis. Es sei die Baum einfügende Grammatik $G = (N, T, Init, Aux, S)$ gegeben. Wir definieren zuerst

$$N' = N \cup \{A' \mid A \in N\}.$$

Dann konstruieren wir zu jedem Baum \mathcal{B} aus $Init$ und Aux einen Baum \mathcal{B}' , indem wir jedes Blatt mit einer Markierung (A, \downarrow) durch einen Baum

$$\begin{array}{c} (A', Aux''', 1) \\ \downarrow \\ \lambda \end{array}$$

ersetzen. Falls \mathcal{B} kein solches Blatt enthält, gilt $\mathcal{B}' = \mathcal{B}$. Wir bemerken, dass die Blätter eines Baumes \mathcal{B}' von links nach rechts gelesen ein terminales Wort ergeben. Der Baum \mathcal{B}' trägt aber nur bei $\mathcal{B}' = \mathcal{B}$ zur erzeugten Sprache bei, da \mathcal{B}' sonst einen Knoten mit einer Markierung $(A', Aux''', 1)$ enthält. Des Weiteren ersetzen wir in jedem Baum \mathcal{B}' jeden in der zweiten Komponente einer Markierung auftretenden Baum \mathcal{T} durch \mathcal{T}' (falls in der

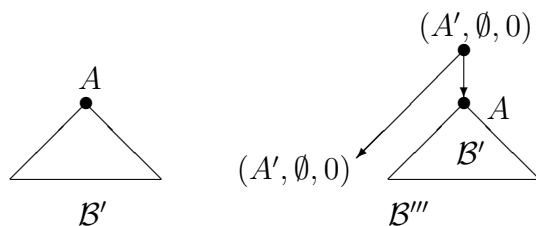


Abbildung 1.14: Übergang von $\mathcal{B}' \in \text{Init}'$ zu $\mathcal{B}''' \in \text{Aux}'''$

zweiten Komponente Aux steht, nehmen wir Aux''). Die so zu Init und Aux entstehenden Mengen seien Init' bzw. Aux'' .

Außerdem definieren wir zu jedem Baum $\mathcal{B}' \in \text{Init}'$ einen Baum $\mathcal{B}''' \in \text{Aux}'''$ entsprechend Abbildung 1.14. Offensichtlich wird die Substitution von $\mathcal{B}_2 \in \text{Init}$ in $\mathcal{B}_1 \in \text{Init}$ durch das Einfügen von $\mathcal{B}_2''' \in \text{Aux}'''$ in $\mathcal{B}_1' \in \text{Init}'$ simuliert (siehe dazu Abbildung 1.15), und für jede Substitution muss auch eine Einfügung erfolgen, da die Substitutionsmarkierung (A, \downarrow) durch die eine Einfügung erfordernde Markierung $(A', \text{Aux}''', 1)$ ersetzt wurde.

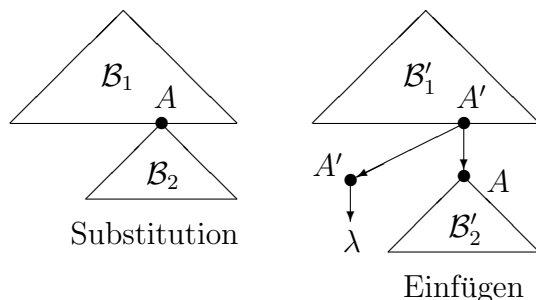


Abbildung 1.15: Simulation der Substitution durch Einfügen

Damit erzeugt die Baum einfügende Grammatik $G' = (N', T, \text{Init}', \text{Aux}'' \cup \text{Aux}''', S)$ die gleiche Sprache wie G , jedoch gibt es keine Markierungen mit \downarrow in zweiter Komponente, so dass keine Substitutionen möglich sind. \square

Satz 1.43 *Jede Sprache in $\mathcal{L}(TA)$ ist semi-linear.*

Beweis. Es sei $G = (N, T, \text{Init}, \text{Aux}, S)$ eine Baum einfügende Grammatik. Wir betrachten die Mengen T_1 und T_2 aller Bäume, die aus den Bäumen aus Init bzw. Aux durch iteriertes Einfügen von Bäumen aus Aux entstehen. Man beachte, dass alle Bäume aus T_2 genau ein Blatt haben, dass mit einem Nichtterminal markiert ist, und dass diese Markierung auch die der Wurzel des Baumes ist.

Wir können nun wie beim Beweis der Semi-Linearität kontextfreier Sprachen vorgehen, indem wir statt der Ableitungsbäume Bäume aus T_1 für H_1 bzw. aus T_2 für H_2 heranziehen. Der Beweis der Semi-Linearität für Sprachen aus $\mathcal{L}(TA)$ verläuft dann völlig analog. \square

Zum Nachweis der schwachen Kontextabhängigkeit der Klasse der Baum einfügenden Grammatiken, haben wir noch zu zeigen, dass ein polynomialer Algorithmus für das Mitgliedsproblem für Baum einfügende Grammatiken existiert. Wir werden die Existenz eines solchen Algorithmus in Abschnitt 1.6. nachtragen.

1.4.2. Lexikalische Grammatiken

In diesem Abschnitt wollen wir zeigen, dass durch Baum einfügende Grammatiken nicht nur die Erzeugungskraft gegenüber kontextfreien Grammatiken erhöht wird. Wir wollen nachweisen, dass auch in anderer Hinsicht Vorteile durch die Benutzung Baum einfügender Grammatiken entstehen. Wir werden zeigen, dass für jede kontextfreie Grammatik eine Baum einfügende Grammatik mit einer speziellen Eigenschaft gefunden werden kann, die die gleichen „Ableitungsbäume“ wie die gegebene kontextfreie Grammatik hat, während eine derartige Transformation innerhalb der Menge der kontextfreien Grammatiken unmöglich ist.

Wir geben dafür zwei Definitionen.

Definition 1.44 *Eine kontextfreie Grammatik heißt endlich mehrdeutig, falls es für jedes Wort der erzeugten Sprache nur endlich viele verschiedene Ableitungen bez. der Grammatik gibt.*

Die Grammatik $G = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow ab\}, S)$ ist endlich mehrdeutig, denn für jedes Wort $a^n b^n$ der erzeugten Sprache gibt es genau eine Ableitung. Andererseits ist jede Grammatik mit einer Ableitung $A \xRightarrow{*} A$ für ein Nichtterminal A , zu dem auch eine Ableitung $S \xRightarrow{*} xAy \xRightarrow{*} xzy$ in G existiert, nicht endlich mehrdeutig, denn das Wort xzy hat unendlich viele verschiedene Ableitungen der Form

$$S \xRightarrow{*} xAy \xRightarrow{*} xAy \xRightarrow{*} xAy \xRightarrow{*} \cdots \xRightarrow{*} xAy \xRightarrow{*} xzy.$$

Definition 1.45

- i) *Eine kontextfreie Grammatik $G = (N, T, P, S)$ heißt lexikalisch, wenn auf der rechten Seite einer jeden Regel aus P mindestens ein Terminal aus T vorkommt.*
- ii) *Eine Baum einfügende Grammatik $G = (N, T, Init, Aux, S)$ heißt lexikalisch, falls jeder Baum aus $Init \cup Aux$ mindestens ein Blatt enthält, das mit einem Terminal markiert ist.*

Eine kontextfreie Grammatik in Chomsky-Normalform ist nicht lexikalisch, da die Regeln der Form $A \rightarrow BC$ auf der rechten Seite nur Nichtterminale haben. In den Grundvorlesungen zur Theoretischen Informatik wurde die Greibach-Normalform für kontextfreie Sprachen behandelt, bei der jede Regel die Form $A \rightarrow aB_1B_2 \dots B_n$ mit $n \geq 0$, $A, B_1, B_2, \dots, B_n \in N$ und $a \in T$ hat. Daher gibt es für jede kontextfreie Sprache L eine lexikalische Grammatik G mit $L(G) = L$.

Die Baum einfügenden Grammatiken aus den Beispielen 1.38, 1.39 und 1.40 sind alle lexikalisch.

Satz 1.46 *Jede lexikalische kontextfreie und jede lexikalische Baum einfügende Grammatik ist endlich mehrdeutig.*

Beweis. Nach Definition wird bei jeder Anwendung einer Regel bzw. jedem Baum-Einfügen mindestens ein Terminal erzeugt. Damit ist die Länge einer jeden Ableitung eines terminalen Wortes w aus n Buchstaben durch n beschränkt. Da es aber nur endlich viele Ableitungen der Länge $k \leq n$ gibt, gibt es auch nur endlich viele Ableitungen für w . \square

Definition 1.47 Ein grammatikalischer Formalismus² F' heißt Lexikalisierung eines (anderen) grammatikalischen Formalismus F , wenn zu jeder endlich mehrdeutigen Grammatik G aus F eine Grammatik G' aus F' derart existiert, dass G' lexikalisch ist sowie G und G' die gleichen Mengen von Ableitungsbäumen haben.

Man beachte, dass aus der Gleichheit der Mengen der Ableitungsbäume selbstverständlich auch die Gleichheit der Sprachen $L(G)$ und $L(G')$ folgt.

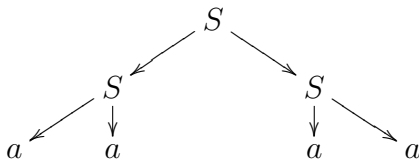
Im Sinne der Definition 1.47 ist die Menge F' der kontextfreien Grammatiken in Greibach-Normalform aber keine Lexikalisierung der Menge F der kontextfreien Grammatiken, weil wir die Gleichheit der Ableitungsbäume nicht sichern können. Dazu betrachten wir als Beispiel die kontextfreie Grammatik

$$G = (\{S\}, \{a\}, \{S \rightarrow SS, S \rightarrow a\}, S).$$

Offensichtlich gilt

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

Die Grammatik G ist nicht lexikalisch aber endlich mehrdeutig (was analog zum Beweis von Satz 1.46 gezeigt werden kann, da die Erzeugung von a^n genau $n - 1$ Anwendungen der Regel $S \rightarrow SS$ und genau n Anwendungen von $S \rightarrow a$ erfordert). Jede lexikalische kontextfreie Grammatik erzeugt mit jeder Anwendung einer Regel mindestens ein Terminal. Für den Ableitungsbaum bedeutet dies, dass im ersten Ableitungsschritt ein Terminal mit der Tiefe 1 erzeugt wird. Damit gibt es kein Äquivalent zum Ableitungsbaum



Es kann in gleicher Weise gezeigt werden, dass auch Baum einfügende Grammatiken, bei denen nur Substitutionen erlaubt sind, keine Lexikalisierung endlich mehrdeutiger kontextfreier Grammatiken gestatten, da auch hier in fester Tiefe k mindestens ein Terminal im Ableitungsbaum erscheinen muss, während dies bei G aber nicht der Fall ist, weil es für $n > k$ einen Ableitungsbaum von a^{2^n} gibt, dessen Terminale alle die Tiefe n haben.

In der Definition 1.47 wird davon Gebrauch gemacht, dass es zu einer Grammatik Ableitungsbäume gibt. Für kontextfreie Grammatiken ist ein solcher Baum definiert. Im Wesentlichen ist es der Baum, der zu einem terminalen Wort gehört. Dies kann auch auf Baum einfügende Grammatiken übertragen werden. Ein *Ableitungsbaum einer Baum einfügenden Grammatik* $G = (N, T, Init, Aux, S)$ ist ein Baum \mathcal{B} aus $T(G)$ mit folgenden Eigenschaften:

- die Wurzel von \mathcal{B} ist mit $(S, U, 0)$ für ein $U \subseteq Aux$ markiert,
- \mathcal{B} enthält keine Knoten mit einer Markierung $(X, U', 1)$, $X \in N$, $U' \subseteq Aux$, und
- die Blätter von \mathcal{B} sind alle mit Markierungen aus $T \cup \{\lambda\}$ versehen.

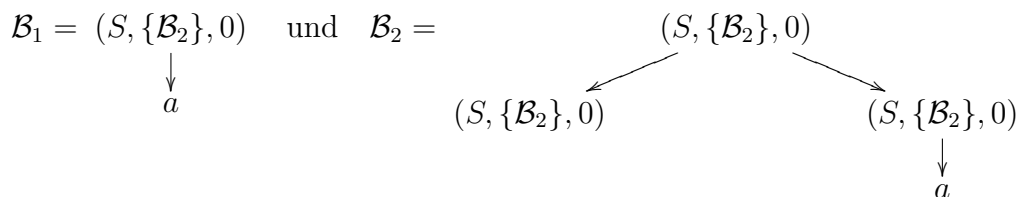
²eine Menge von Grammatiken mit gewissen Eigenschaften

Durch die Bedingungen an einen Ableitungsbaum wird abgesichert, dass Ableitungsbäume gerade die Bäume sind, deren Blätter als Wort gelesen zur erzeugten Sprache beitragen.

Wir zeigen nun, dass es eine Lexikalisierung für obige Grammatik G durch Baum einfügende Grammatiken (mit Substitutionen und Baum-Einfügungen) gibt. Dazu betrachten wir die Baum einfügende Grammatik

$$G' = (\{S\}, \{a\}, \{\mathcal{B}_1\}, \{\mathcal{B}_2\}, S)$$

mit



Daraus können durch sukzessives Baum-Einfügen u. a. die in Abbildung 1.16 gezeigten Bäume erzeugt werden, wobei wir der Einfachheit halber statt des Tripels nur die erste Komponente angeben. Wir überlassen es der Leserin / dem Leser zu beweisen, dass bei

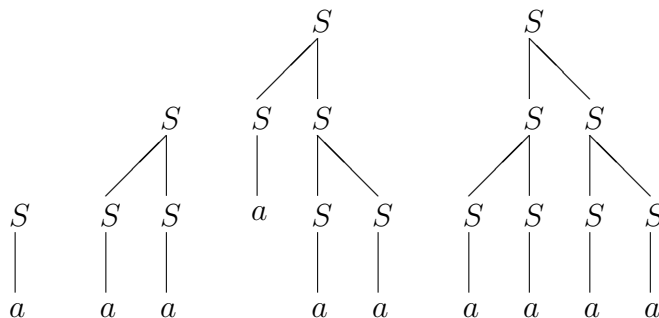


Abbildung 1.16: Erzeugte Ableitungsbäume

Beschränkung auf die erste Komponente der Tripel tatsächlich alle und nur die Ableitungsbäume der ursprünglichen Grammatik G erhalten werden.

Jetzt zeigen wir, dass dieser Sachverhalt nicht nur für die Beispielgrammatik G gilt. Vielmehr lässt sich zu jeder endlich mehrdeutigen kontextfreien Grammatik eine Baum einfügende Grammatik so konstruieren, dass die Ableitungsbäume beider Grammatiken übereinstimmen, d. h. Baum einfügende Grammatiken sind eine Lexikalisierung kontextfreier Grammatiken.

Satz 1.48 *Zu jeder endlich mehrdeutigen kontextfreien Grammatik G gibt es eine lexikalische Baum einfügende Grammatik $G' = (N, T, \text{Init}, \text{Aux}, S)$ derart, dass die Menge der Ableitungsbäume von G mit der Menge der Ableitungsbäume von G' bei Beschränkung auf die Nichtterminale in den mit Tripeln markierten Knoten übereinstimmt.*

Beweis. Eine endlich mehrdeutige kontextfreie Grammatik $G = (N, T, P, S)$ sei gegeben. Wir nennen ein Nichtterminal $A \in N$ rekursiv, falls es eine Ableitung $A \xRightarrow{*} uAv$ mit gewissen Wörtern $u, v \in (N \cup T)^*$ gibt. Eine Regel $A \rightarrow w \in P$ heißt nicht-rekursiv, falls

A nicht rekursiv ist. Es sei P' die Menge der nicht-rekursiven Regeln. (Um festzustellen, ob ein Nichtterminal rekursiv ist, kann man den gerichteten Graphen $H = (N, E)$ bilden, wobei $(A, B) \in E$ genau dann gilt, wenn es eine Regel $A \rightarrow xBy$ mit $x, y \in (N \cup T)^*$ gibt. Ein Nichtterminal A ist genau dann rekursiv, wenn es in H einen Kreis gibt, der A enthält. Die Graphentheorie stellt viele Methoden, z. B. Breitensuche oder Tiefensuche, zur Verfügung, mit denen man testen kann, ob A in einem Kreis liegt.)

Wir konstruieren nun zuerst die Menge aller Ableitungen von terminalen Wörtern, in denen nur nicht-rekursive Regeln benutzt werden. Diese Menge ist offenbar endlich. Wir definieren $Init$ als die Menge der Ableitungsbäume dieser Ableitungen. Da nach unserer Definition $\lambda \notin L(G)$ wegen des Fehlens löschender Regeln gilt, erzeugt jede der Ableitungen ein nichtleeres Wort und folglich enthält jeder Baum aus $Init$ mindestens ein Blatt, das mit einem Terminal markiert ist.

Mittels graphentheoretischer Methoden – angewandt auf den oben angegebenen Graphen H – ermitteln wir die minimalen Kreise in H , d. h. solche Kreise, die keinen echten Teilgraphen enthalten, der Kreis ist.

Wir konstruieren nun iterativ die Menge Aux . Zuerst setzen wir $Aux = \emptyset$. Dann durchlaufen wir den folgenden Zyklus so lange, bis sich keine Änderung von Aux mehr ergibt.

Für jeden minimalen Kreis c und jeden Knoten B aus c , der in einem Baum aus $Init \cup Aux$ vorkommt und für den $B \xRightarrow{*} xBy$ die entsprechend c vorhandene Ableitung ist, und alle Ableitungen $x \xRightarrow{*} x' \in T^*$ und $y \xRightarrow{*} y' \in T^*$, die nur nicht-rekursive Regeln verwenden, nehmen wir in Aux den Ableitungsbaum von $B \xRightarrow{*} xBy \xRightarrow{*} x'By'$ auf. Falls $x'y'$ das leere Wort ist, so gibt es in G die Ableitung $B \xRightarrow{*} B$, was der endlichen Mehrdeutigkeit von G widerspricht; somit gibt es im Ableitungsbaum von $B \xRightarrow{*} xBy \xRightarrow{*} x'By'$ mindestens ein Blatt, das mit einem Terminal markiert sind.

Folglich ist die Baum einfügende Grammatik $G' = (N, T, Init, Aux, S)$ lexikalisch. Die Aussage über die Gleichheit der Baumengen kann nun leicht mit vollständiger Induktion wie beim Nachweis der Semi-Linearität kontextfreier Sprachen erbracht werden. \square

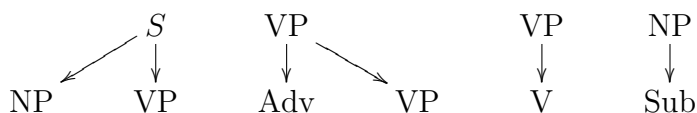
Wir erwähnen hier ohne Beweis den folgenden Satz, der das vorstehende Ergebnis auf Baum einfügende Grammatiken erweitert; für einen Beweis verweisen wir auf [21].

Satz 1.49 *Baum einfügende Grammatiken sind eine Lexikalisierung Baum einfügender Grammatiken.* \square

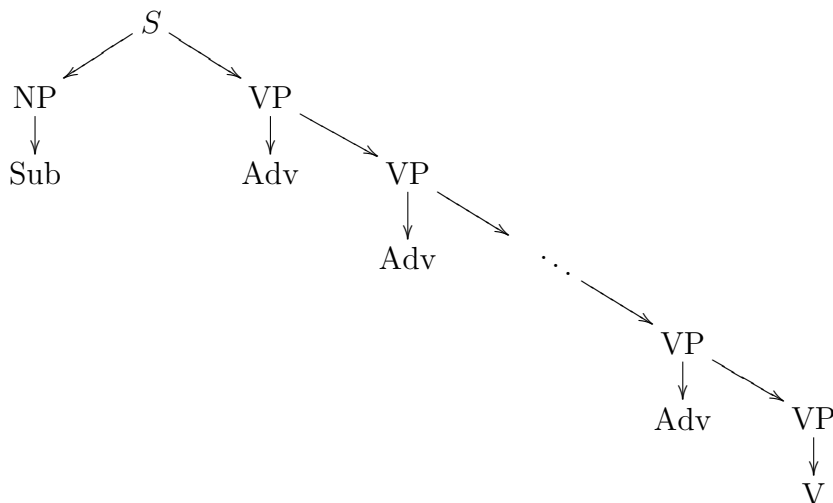
Wir wollen abschließend noch an einem Beispiel zeigen, dass das Einfügen von Bäumen auch einen linguistischen Gewinn geben kann, wenn eine Lexikalisierung ohne Baumeinfügungen möglich ist. Dazu betrachten wir die Grammatik

$$G = (\{S, NP, VP\}, \{Adv, V, Sub\}, P, S)$$

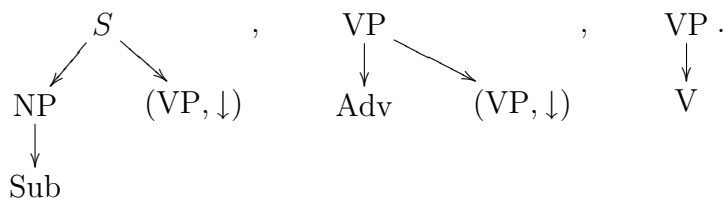
(wobei wieder NP für Nominalphrase, VP für Verbphrase, V für Verb, Adv für Adverb und Sub für Substantiv verwendet wird) mit den folgenden als Bäume gegebenen Regeln:



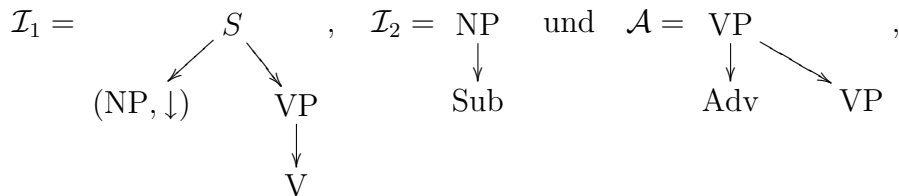
Die erzeugte Sprache besteht aus allen Wörtern der Form $\text{Sub Adv}^n \text{V}$; die zugehörigen Ableitungsbäume sind eindeutig bestimmt und haben die Form



Offensichtlich ist die gegebene Grammatik nicht lexikalisch, denn die erste Regel enthält auf der rechten Seite bzw. in den Blättern kein Terminal. Wir können durch eine leichte Modifikation aber eine lexikalische Baum einfügende Grammatik mit der gleichen Menge von Ableitungsbäumen angeben, z. B. unter Verwendung der Bäume



Jede Lexikalisierung von G durch eine „reine“ Baum anhängende Grammatik (wobei nur Substitutionen erlaubt sind) enthält einen Baum mit S in der Wurzel und kein Blatt ist mit V markiert. Wäre dies nicht der Fall, wäre also V Blatt eines jeden Baumes, der S in der Wurzel hat, so könnte nur noch Sub an die Bäume angefügt werden, womit die Anzahl der erhaltenen Bäume endlich wäre (Adv könnte nicht eingefügt werden). Damit hat V eine Sonderrolle. Dagegen hat die Grammatik G auch eine Lexikalisierung durch die Baum einfügende Grammatik $G' = (\{S, NP, VP\}, \{Adv, V, Sub\}, \{\mathcal{I}_1, \mathcal{I}_2\}, \{\mathcal{A}\}, S)$ mit



bei der die oben erwähnte Sonderrolle von V nicht vorliegt.

1.5. Kopf-Grammatiken

Unter einem Kopf-Wort über einem Alphabet V verstehen wir einen Wort $u \uparrow v$ mit $u \in V^*$ und $v \in V^*$. Diese Bezeichnung kommt von der Vorstellung, dass ein Lese- oder Schreib-

kopf über dem Wort an der Position steht, die durch das Symbol \uparrow angegeben wird.

Mit $K(V)$ bezeichnen wir die Menge aller Kopf-Wörter über V .

Wir erklären zuerst die Operationen

$$C_{i,n} : (K(V))^n \rightarrow K(V) \text{ für } n \geq 1, 1 \leq i \leq n, \text{ und}$$

$$W : (K(V))^2 \rightarrow K(V)$$

über $K(V)$ durch

$$C_{i,n}(u_1 \uparrow v_1, u_2 \uparrow v_2, \dots, u_n \uparrow v_n) = u_1 v_1 u_2 v_2 \dots u_{i-1} v_{i-1} u_i \uparrow v_i u_{i+1} v_{i+1} \dots u_n v_n \text{ und}$$

$$W(u_1 \uparrow v_1, u_2 \uparrow v_2) = u_1 u_2 \uparrow v_2 v_1.$$

Die Operationen $C_{i,n}$, $n \geq 1$, $1 \leq i \leq n$, sind der Konkatenation ähnlich; W heißt *Wickeloperation*. Ihre Idee entstammt der Situation, dass die kontextfreien Ableitungen $A \xRightarrow{*} u_1 B v_1$ und $B \xRightarrow{*} u_2 C v_2$ die Ableitung $A \xRightarrow{*} u_1 u_2 C v_2 v_1$ ergeben. Ersetzt man das Nichtterminal durch \uparrow , so ergibt sich gerade die Wickeloperation.

Wie man leicht nachrechnet, gelten die folgenden Gleichheiten

$$C_{i,n}(u_1 \uparrow v_1, \dots, u_n \uparrow v_n) = C_{1,2}(C_{i,n-1}(u_1 \uparrow v_1, \dots, u_{n-1} \uparrow v_{n-1}), C_{1,1}(u_n \uparrow v_n))$$

für $1 \leq i \leq n-1$,

$$C_{n,n}(u_1 \uparrow v_1, \dots, u_n \uparrow v_n) = C_{2,2}(C_{i,n-1}(u_1 \uparrow v_1, \dots, u_{n-1} \uparrow v_{n-1}), C_{1,1}(u_n \uparrow v_n))$$

für $1 \leq i \leq n-1$, (1.5)

$$C_{1,1}(u_1 \uparrow v_1) = C_{1,2}(u_1 \uparrow v_1, \lambda \uparrow \lambda),$$

$$C_{1,2}(u_1 \uparrow v_1, C_{1,2}(u_2 \uparrow v_2, u_3 \uparrow v_3)) = C_{1,2}(C_{1,2}(u_1 \uparrow v_1, u_2 \uparrow v_2), u_3 \uparrow v_3) = u_1 \uparrow v_1 u_2 v_2 u_3 v_3.$$

Durch wiederholtes Einsetzen kann jede Operation $C_{i,n}$, $n \geq 1$, $1 \leq i \leq n$, durch die Operationen $C_{1,2}$ und $C_{2,2}$ ausgedrückt werden.

Wir definieren sogenannte Kopf-Grammatiken, die von CARL POLLARD in seiner Dissertation ([17]) eingeführt wurden. Sie sind im Wesentlichen kontextfreie Grammatiken, bei denen anstelle der Konkatenation obige Operationen benutzt werden. Wesentliche, früh erzielte Resultate sind in [19] zu finden.

Definition 1.50

i) Eine Kopf-Grammatik ist ein Quadrupel $G = (N, T, P, S)$, wobei

- N , T und S wie bei einer Regelgrammatik definiert sind, und
- P eine endliche Menge von Regeln der Form

$$A \rightarrow f(A_1, \dots, A_n) \text{ mit } A \in N, n \geq 1, A_1, \dots, A_n \in N \cup K(T),$$

$$f \in \{W \mid n = 2\} \cup \{C_{i,n} \mid 1 \leq i \leq n\}$$

ist.

ii) Wir definieren die Relation \xRightarrow{k} induktiv wie folgt:

- Für alle $u, v \in T^*$ gilt $u \uparrow v \xRightarrow{0} u \uparrow v$.

- Falls $A \rightarrow f(A_1, A_2, \dots, A_n) \in P$ und $A_i \xrightarrow{k_i} u_i \uparrow v_i$ für gewisse $u_i, v_i \in T^*$ und $1 \leq i \leq n$, dann gilt

$$A \xrightarrow{k} f(u_1 \uparrow v_1, u_2 \uparrow v_2, \dots, u_n \uparrow v_n) \text{ mit } k = 1 + \sum_{i=1}^n k_i.$$

iii) Die von G erzeugte Sprache $L(G)$ ist durch

$$L(G) = \{w \mid w = uv, S \xrightarrow{k} u \uparrow v \text{ für ein } k \geq 0\}$$

definiert.

Mit $\mathcal{L}(H)$ bezeichnen wir die Menge aller von Kopf-Grammatiken erzeugten Sprachen.³

Man hätte auch eine Relation \Longrightarrow wie folgt definieren können: Es gelte $x \Longrightarrow y$ genau dann, wenn

- $x = uAv$ für gewisse $u, v \in (N \cup T \cup \{W, (,), \uparrow, \cdot\} \cup \{C_{i,n} \mid n \geq 1, 1 \leq i \leq n\})^*$ und $A \in N$,
- $A \rightarrow f(A_1, A_2, \dots, A_n) \in P$,
- $y = uf(A_1, A_2, \dots, A_n)v$

gelten (d. h. wir nehmen übliche kontextfreie Ersetzungen vor). Wie man durch vollständige Induktion über k leicht nachweist, gilt $A \xrightarrow{k} u \uparrow v$ mit $u, v \in T^*$ genau dann, wenn A durch k kontextfreie Ersetzungen in einen Term überführt wird, der durch Auswertung zu $u \uparrow v$ führt.

Bei der Kopf-Grammatik

$$G_0 = (\{A, A'\}, \{a, c\}, \{p_1, p_2, p_3\}, A)$$

mit

$$p_1 = A \rightarrow C_{2,2}(a \uparrow \lambda, A'), \quad p_2 = A' \rightarrow C_{1,1}(\lambda \uparrow c), \quad p_3 = A' \rightarrow W(A, \lambda \uparrow c)$$

erhalten wir unter anderem die folgende kontextfreie Ableitung:

$$\begin{aligned} A &\Longrightarrow C_{2,2}(a \uparrow \lambda, A') \Longrightarrow C_{2,2}(a \uparrow \lambda, W(A, \lambda \uparrow c)) \\ &\Longrightarrow C_{2,2}(a \uparrow \lambda, W(C_{2,2}(a \uparrow \lambda, A'), \lambda \uparrow c)) \\ &\Longrightarrow C_{2,2}(a \uparrow \lambda, W(C_{2,2}(a \uparrow \lambda, C_{1,1}(\lambda \uparrow c)), \lambda \uparrow c)). \end{aligned}$$

Der letzte erhaltene Ausdruck lässt sich nun auswerten, wodurch wir

$$\begin{aligned} A &\xrightarrow{4} C_{2,2}(a \uparrow \lambda, W(C_{2,2}(a \uparrow \lambda, C_{1,1}(\lambda \uparrow c)), \lambda \uparrow c)) \\ &= C_{2,2}(a \uparrow \lambda, W(C_{2,2}(a \uparrow \lambda, \lambda \uparrow c), \lambda \uparrow c)) \\ &= C_{2,2}(a \uparrow \lambda, W(a\lambda\lambda \uparrow c, \lambda \uparrow c)) = C_{2,2}(a \uparrow \lambda, W(a \uparrow c, \lambda \uparrow c)) \\ &= C_{2,2}(a \uparrow \lambda, a\lambda \uparrow cc) = C_{2,2}(a \uparrow \lambda, a \uparrow cc) \\ &= a\lambda a \uparrow cc = aa \uparrow cc \end{aligned}$$

³Die Abkürzung entstammt dem englischen Wort *head* für *Kopf*.

erhalten. Damit gehört das Wort a^2c^2 zur erzeugten Sprache.

Nach Definition 1.50 erhalten wir der Reihe nach

$$\begin{aligned} \lambda \uparrow c &\xrightarrow{0} \lambda \uparrow c, \\ A' &\xrightarrow{1} C_{1,1}(\lambda \uparrow c) = \lambda \uparrow c && \text{(mit } p_2), \\ A &\xrightarrow{2} C_{2,2}(a \uparrow \lambda, \lambda \uparrow c) = a \uparrow c && \text{(mit } p_1), \\ A' &\xrightarrow{3} W(a \uparrow c, \lambda \uparrow c) = a \uparrow cc && \text{(mit } p_3), \\ A &\xrightarrow{4} C_{2,2}(a \uparrow \lambda, a \uparrow cc) = aa \uparrow cc && \text{(mit } p_1), \end{aligned}$$

womit auch auf diese Weise $a^2c^2 \in L(G_0)$ nachgewiesen ist.

Bei Verwendung der kontextfreien Ableitungen kann die Auswertung im Wesentlichen erst erfolgen, wenn die Ableitung terminiert hat. Dabei entstehen unübersichtliche Ausdrücke, so dass es schwierig ist, die erzeugte Sprache zu bestimmen. Dies ist mittels \xrightarrow{k} aus Definition 1.50 meist einfacher.

Wir behandeln nun drei Beispiele, bei denen erneut die Sprachen K_1 , K_2 und K_3 erzeugt werden.

Beispiel 1.51 *Wir betrachten die Kopf-Grammatik*

$$G_1 = (\{S, S', A, A'\}, \{a, b, c, d\}, \{p_1, p_2, p_3, p_4, p_5, p_6\}, S)$$

mit

$$\begin{aligned} p_1 &= A \rightarrow C_{2,2}(a \uparrow \lambda, A'), \quad p_2 = A' \rightarrow C_{1,1}(\lambda \uparrow c), \quad p_3 = A' \rightarrow W(A, \lambda \uparrow c), \\ p_4 &= S' \rightarrow W(A, b \uparrow \lambda), \quad p_5 = S \rightarrow C_{1,2}(S', \lambda \uparrow d), \quad p_6 = S' \rightarrow W(S, b \uparrow \lambda). \end{aligned}$$

Die Regeln p_1 , p_2 und p_3 sind die gleichen Regeln wie in der oben betrachteten Kopf-Grammatik G_0 . Von den dortigen Betrachtungen kennen wir

$$A' \xrightarrow{1} \lambda \uparrow c, \quad A \xrightarrow{2} a \uparrow c, \quad A' \xrightarrow{3} a \uparrow c^2 \quad \text{und} \quad A \xrightarrow{4} a^2 \uparrow c^2.$$

Mittels vollständiger Induktion ist nun leicht zu zeigen, dass

$$A' \xrightarrow{2n-1} a^{n-1} \uparrow c^n \quad \text{und} \quad A \xrightarrow{2n} a^n \uparrow c^n$$

gelten, da p_3 ein c im zweiten Teil und p_1 ein a im ersten Teil des Kopf-Wortes einfügen. Damit erhalten wir

$$\begin{aligned} S' &\xrightarrow{2n+1} W(a^n \uparrow c^n, b \uparrow \lambda) = a^n b \uparrow c^n, \\ S &\xrightarrow{2n+2} C_{1,2}(a^n b \uparrow c^n, \lambda \uparrow d) = a^n b \uparrow c^n d, \\ S' &\xrightarrow{2n+3} W(a^n b \uparrow c^n d, b \uparrow \lambda) = a^n b^2 \uparrow c^n d, \\ S &\xrightarrow{2n+4} C_{1,2}(a^n b^2 \uparrow c^n d, \lambda \uparrow d) = a^n b^2 \uparrow c^n d^2 \end{aligned}$$

und allgemein

$$\begin{aligned} S' &\xrightarrow{2n+2m+1} W(a^n b^m \uparrow c^n d^m, b \uparrow \lambda) = a^n b^{m+1} \uparrow c^n d^m, \\ S &\xrightarrow{2n+2m+2} C_{1,2}(a^n b^{m+1} \uparrow c^n d^m, \lambda \uparrow d) = a^n b^{m+1} \uparrow c^n d^{m+1}. \end{aligned}$$

Hieraus resultiert

$$L(G_1) = \{a^n b^m c^n d^m \mid n, m \geq 1\} = K_2.$$

Beispiel 1.52 Es sei die Kopf-Grammatik

$$G_2 = (\{S, S_a, S_b\}, \{a, b\}, P_2, S)$$

mit

$$P_2 = \bigcup_{x \in \{a, b\}} \{S \rightarrow C_{1,1}(x \uparrow x), S_x \rightarrow W(S, x \uparrow \lambda), S \rightarrow C_{1,2}(S_x, \lambda \uparrow x)\}$$

gegeben. Wir zeigen mittels vollständiger Induktion

$$L(G_2) = \{ww \mid w \in \{a, b\}^+\} = K_3.$$

Aus $S \rightarrow C_{1,1}(x \uparrow x)$ folgt $S \xrightarrow{1} x \uparrow x$ für $x \in \{a, b\}$.

Es sei nun schon $S \xrightarrow{k} x_1 x_2 \dots x_m \uparrow x_1 x_2 \dots x_m$ für alle $x_i \in \{a, b\}$, $1 \leq i \leq m$, gezeigt. Dann erhalten wir

$$\begin{aligned} S_x &\xrightarrow{k+1} W(x_1 x_2 \dots x_m \uparrow x_1 x_2 \dots x_m, x \uparrow \lambda) = x_1 x_2 \dots x_m x \uparrow x_1 x_2 \dots x_m \text{ und} \\ S &\xrightarrow{k+2} C_{1,2}(x_1 x_2 \dots x_m x \uparrow x_1 x_2 \dots x_m, \lambda \uparrow x) = x_1 x_2 \dots x_m x \uparrow x_1 x_2 \dots x_m x, \end{aligned}$$

für $x \in \{a, b\}$, woraus die Behauptung resultiert.

Beispiel 1.53 Die Kopf-Grammatik

$$G_3 = (\{S, A\}, \{a, b, c\}, \{S \rightarrow C_{1,1}(ab \uparrow c), S \rightarrow C_{2,2}(a \uparrow \lambda, A), A \rightarrow W(S, b \uparrow c)\}, S)$$

erzeugt die Sprache

$$L(G_3) = \{a^n b^n c^n \mid n \geq 1\} = K_1,$$

wie man in Analogie zu den vorstehenden Beispielen leicht beweist.

Der nachfolgende Satz gibt ein erstes Resultat zur Erzeugungskraft von Kopf-Grammatiken an. Im folgenden Abschnitt werden wir dazu weitere Aussagen treffen.

Satz 1.54 $\mathcal{L}(CF) \subset \mathcal{L}(H)$.

Beweis. Es sei $L \in \mathcal{L}(CF)$. Dann gibt es eine kontextfreie Grammatik $G = (N, T, P, S)$ in Chomsky-Normalform, die L erzeugt und deren Regeln alle von der Form $A \rightarrow BC$ oder $A \rightarrow a$ mit $A, B, C \in N$ und $a \in T$ sind. Wir konstruieren die Kopf-Grammatik $G' = (N, T, P', S)$, wobei

$$P' = \{A \rightarrow C_{1,2}(B, C) \mid A \rightarrow BC \in P\} \cup \{A \rightarrow C_{1,1}(\lambda \uparrow a) \mid A \rightarrow a \in P\}$$

gesetzt wird. Dann ergibt sich durch kontextfreies Gebrauchen von Regeln aus P' ein Term, der nur die Operatoren $C_{1,2}$ und $C_{1,1}$ enthält und diese werden nur auf Terme

der Form $\lambda \uparrow a$ angewandt. Ferner ist die Reihenfolge der Buchstaben aus $N \cup T$ in einem mittels Regeln aus P' abgeleitetem Wort die gleiche wie bei dem Wort, das durch die entsprechenden Regeln in P abgeleitet wird. Insbesondere gilt $S \xrightarrow[G]{*} a_1 a_2 \dots a_m$ mit $a_i \in T$ genau dann, wenn $S \xrightarrow[G']{k} u_0 \lambda \uparrow u_1 a_1 u_2 a_2 \dots u_m a_m u_{m+1}$ für gewisse Wörter $u_i \in \{C_{1,2}, C_{1,1}, \uparrow, (,), , \}^*$, $0 \leq i \leq m+1$ gilt, wobei u_0 kein \uparrow enthält. Mittels der letzten Gleichheiten aus (1.5) ist leicht nachzuweisen, dass $S \xrightarrow[G]{*} a_1 a_2 \dots a_m$ genau dann gilt, wenn $S \xrightarrow[G']{*} \lambda \uparrow a_1 a_2 \dots a_m$ gültig ist. Damit erhalten wir $L(G') = L(G)$ und die Inklusion $\mathcal{L}(CF) \subseteq \mathcal{L}(H)$ ist gezeigt.

Aus den Beispielen 1.51 – 1.53 folgt, dass die Inklusion echt ist. \square

Wir geben nun ein Analogon zur Chomsky-Normalform für Kopf-Grammatiken an.

Satz 1.55 *Zu jeder Kopf-Grammatik $G = (N, T, P, S)$ gibt es eine Kopf-Grammatik $G' = (N', T, P', S)$ derart, dass*

– *P' nur Regeln der Form*

$$A \rightarrow C_{1,2}(B, C), A \rightarrow C_{2,2}(B, C), A \rightarrow W(B, C), A \rightarrow C_{1,1}(w_1 \uparrow w_2)$$

mit $A, B, C \in N', w_1, w_2 \in T^$*

enthält, und

– *$L(G') = L(G)$ gilt.*

Beweis. Zuerst sei $A \rightarrow f(u_1 \uparrow v_1, u_2 \uparrow v_1, \dots, u_n \uparrow v_n)$ eine terminierende Regel in G , d. h. $u_i, v_i \in T^*$ für $1 \leq i \leq n$, und es gelte $f(u_1 \uparrow v_1, u_2 \uparrow v_1, \dots, u_n \uparrow v_n) = w_1 \uparrow w_2$. Wegen

$$C_{1,1}(w_1 \uparrow w_2) = w_1 \uparrow w_2 = f(u_1 \uparrow v_1, u_2 \uparrow v_1, \dots, u_n \uparrow v_n)$$

können wir ohne Änderung der Sprache die Regel $A \rightarrow f(u_1 \uparrow v_1, u_2 \uparrow v_1, \dots, u_n \uparrow v_n)$ durch die Regel $A \rightarrow C_{1,1}(w_1 \uparrow w_2)$ ersetzen. Damit haben wir nur noch terminierende Regeln der gewünschten Form, die wir in die Menge P' aufnehmen.

Es sei nun $p = A \rightarrow f(\sigma_1, \sigma_2, \dots, \sigma_n) \in P$ mit $\sigma_i \in N \cup K(T)$ für $1 \leq i \leq n$ eine nicht-terminierende Regel. Wir führen dafür eine Regel $p' = A \rightarrow f(\sigma'_1, \sigma'_2, \dots, \sigma'_n)$ in P' ein, wobei für $1 \leq i \leq n$

$$\sigma'_i = \begin{cases} \sigma_i & \text{falls } \sigma_i \in N, \\ B_i & \text{falls } \sigma_i \in K(T) \end{cases}$$

mit neuen zusätzlichen Nichtterminalen B_i gesetzt wird. Für jedes neue Nichtterminal B_i führen wir noch in P' als einzige Regel $B_i \rightarrow C_{1,1}(\sigma_i)$ ein. Offensichtlich gilt dann nach Anwenden von p' und Ersetzen der B_i entsprechend ihrer Regeln $A \xrightarrow[G']{*} f(\sigma_1, \sigma_2, \dots, \sigma_n)$, d. h. wir können die Anwendung von p in G mittels G' simulieren. Damit sind alle Regeln in P' von der Form $A \rightarrow f(A_1, A_2, \dots, A_n)$ oder $A \rightarrow C_{1,1}(w_1 \uparrow w_2)$ mit $A_i \in N'$ und $w_1, w_2 \in T^*$.

Falls f eine n -stellige Operation $C_{i,n}$ mit $n \geq 3$ und $i \leq n-1$ ist, so ersetzen wir in P' die Regel $A \rightarrow f(A_1, A_2, \dots, A_n)$ durch zwei Regeln

$$A \rightarrow C_{1,2}(A', A_n) \text{ und } A' \rightarrow C_{i,n-1}(A_1, A_2, \dots, A_{n-1}),$$

wobei A' ein neues Nichtterminal ist. Es ergibt sich

$$A \Longrightarrow C_{1,2}(A', A_n) \Longrightarrow C_{1,2}(C_{i,n-1}(A_1, A_2, \dots, A_{n-1}), A_n).$$

Haben wir $A_i \xrightarrow{k_i} u_i \uparrow v_i$, so erhalten wir durch Anwenden von $A \rightarrow f(A_1, A_2, \dots, A_n)$ in G

$$A \xrightarrow{k} C_{i,n}(u_1 \uparrow v_1, u_2 \uparrow v_2, \dots, u_n \uparrow v_n) \quad (1.6)$$

und analog in G'

$$A \xrightarrow{k+1} C_{1,2}(C_{i,n-1}(u_1 \uparrow v_1, u_2 \uparrow v_2, \dots, u_{n-1} \uparrow v_{n-1}), u_n \uparrow v_n). \quad (1.7)$$

Nach Definition stimmen aber die entsprechend (1.6) und (1.7) von A erzeugten Kopf-Wörter überein. Daher liefert dieser Übergang keine Änderung der Sprache.

Falls f eine n -stellige Operation $C_{n,n}$ mit $n \geq 3$ ist, so ersetzen wir in P' die Regel $A \rightarrow f(A_1, A_2, \dots, A_n)$ durch zwei Regeln

$$A \rightarrow C_{2,2}(A_1, A') \text{ und } A' \rightarrow C_{n-1,n-1}(A_2, A_3, \dots, A_n),$$

wobei A' ein neues Nichtterminal ist. Es ergibt sich

$$A \Longrightarrow C_{2,2}(A_1, A') \Longrightarrow C_{2,2}(A_1, C_{n-1,n-1}(A_2, A_3, \dots, A_n)),$$

was erneut die Sprache nicht verändert.

Jede Regel $A \rightarrow C_{1,1}(B)$ wird durch zwei Regeln

$$A \rightarrow C_{1,2}(B, B') \text{ und } B' \rightarrow C_{1,1}(\lambda \uparrow \lambda)$$

ersetzt, wobei B' ein neues Nichtterminal ist. Es ergibt sich

$$A \Longrightarrow C_{1,2}(B, B') \Longrightarrow C_{1,2}(B, C_{1,1}(\lambda \uparrow \lambda)),$$

was ebenfalls die Sprache nicht verändert.

Wir verfahren so für jede Regel aus P' , die nicht die gewünschte Form hat. Da jede Regel mit einer n -stelligen Funktion ($n \geq 3$) durch eine Regel mit einer $(n-1)$ -stelligen Funktion und eine Regel mit einer zweistelligen Funktion ersetzt wird, führen wir schließlich alle Operationen $C_{i,n}$ auf $C_{1,2}$ und $C_{2,2}$ zurück. \square

1.6. Beziehungen zwischen den Grammatikklassen

In diesem Abschnitt diskutieren wir die Erzeugungskraft der in den vorhergehenden Abschnitten eingeführten drei Typen von Grammatiken genauer. Wir werden die zugehörigen Sprachfamilien miteinander vergleichen und zeigen, dass sie übereinstimmen. Dabei folgen wir im Wesentlichen [27]. Dieses Resultat gestattet uns dann auch, Ergebnisse, die bisher nur für einen Typ gezeigt wurden, einfach auf die anderen Typen zu übertragen.

Lemma 1.56 *Zu jeder linearen indizierten Grammatik G kann in polynomialer Zeit (in der Größe von G) eine Kopf-Grammatik G' so konstruiert werden, dass $L(G') = L(G)$ gilt und die Größe von G' polynomial in der Größe von G ist.*

Beweis. Ohne Beschränkung der Allgemeinheit können wir annehmen, dass die gegebene Grammatik $G = (N, T, I, P, S)$ in P nur Regeln der Form

- $A \rightarrow A_1 A_2 \dots A_{i-1} \underline{A_i} [l] A_{i+1} A_{i+2} \dots A_n,$
- $A[l] \rightarrow A_1 A_2 \dots A_{i-1} \underline{A_i} A_{i+1} \dots A_m,$
- $A \rightarrow w$

enthält (falls dies nicht der Fall ist, so überführen wir G in die Normalform entsprechend Satz 1.33, bei der alle Regeln die gewünschte Form haben).

Wir konstruieren nun eine Kopf-Grammatik $G' = (N', T, P', S)$, indem wir

$$N' = N \cup \{\langle A_1, A_2, \eta \rangle \mid A_1, A_2 \in N, \eta \in I \cup \{\lambda\}\}$$

setzen und die Regeln aus P' wie folgt ermitteln:

- Wenn $A \rightarrow A_1 A_2 \dots A_{i-1} \underline{A_i} [l] A_{i+1} A_{i+2} \dots A_n \in P$ ist, so nehmen wir in P' alle Regeln

$$\langle A, B, \lambda \rangle \rightarrow C_{i,n}(A_1, A_2, \dots, \langle A_i, B, l \rangle, A_{i+1}, A_{i+2}, \dots, A_n)$$

mit $B \in N$ auf.

- Wenn $A[l] \rightarrow A_1 A_2 \dots A_{i-1} \underline{A_i} A_{i+1} \dots A_m \in P$ ist, so nehmen wir in P' alle Regeln

$$\langle A, B, l \rangle \rightarrow C_{i,n}(A_1, A_2, \dots, \langle A_i, B, \lambda \rangle, A_{i+1}, A_{i+2}, \dots, A_n)$$

mit $B \in N$ auf.

- Wenn $A \rightarrow w \in P$ ist, so nehmen wir in P' die Regel $A \rightarrow C_{1,1}(\lambda \uparrow w)$ auf.

- Für alle $A, B, C \in N$ und $\eta \in I \cup \{\lambda\}$ nehmen wir in P' alle Regeln

$$\langle A, B, \eta \rangle \rightarrow W(\langle A, C, \lambda \rangle, \langle C, B, \eta \rangle), \langle A, A, \lambda \rangle \rightarrow C_{1,1}(\lambda \uparrow \lambda), A \rightarrow W(\langle A, B, \lambda \rangle, B)$$

auf.

Im Rahmen dieses Beweises werden wir einfach auch für die Kopf-Grammatik G' die Ableitungsrelation $\xrightarrow[G']{*}$ benutzen (und G' fortlassen, falls dies aus dem Kontext klar ist),

wobei $w \xrightarrow[G']{*} w_1 \uparrow w_2$ genau dann gilt, wenn $w \xrightarrow[G']{k} w_1 \uparrow w_2$ für ein k gilt. Für beide Typen von Grammatiken bezeichnen wir mit \xrightarrow{k} eine Ableitung der Länge k .

Wir beweisen nun mittels vollständiger Induktion über die Anzahl der Ableitungsschritte simultan die folgenden Aussagen:

- i) *Für alle $A, B \in N$, $\eta \in I \cup \{\lambda\}$ und $w_1, w_2 \in T^*$ gilt*

$$A[\eta] \xrightarrow[G']{*} w_1 B w_2 \text{ genau dann, wenn } \langle A, B, \eta \rangle \xrightarrow[G']{*} w_1 \uparrow w_2$$

gilt.

ii) Für alle $A \in N$ und $w \in T^*$ gilt $A \xrightarrow[G]{*} w$ genau dann, wenn $w = w_1 w_2$ für gewisse Wörter $w_1, w_2 \in T^*$ ist und $A \xrightarrow[G']{*} w_1 \uparrow w_2$ gilt.

Aus ii) folgt mit $A = S$ sofort $L(G) = L(G')$.

Induktionsanfang für i). Es sei $A[\eta] \xrightarrow[G]{*} w_1 B w_2$ eine Ableitung der Länge 0. Dann gelten $A = B$ und $w_1 = w_2 = \lambda$. In P' ist nach Konstruktion die Regel $\langle A, A, \lambda \rangle \rightarrow C_{1,1}(\lambda, \lambda)$ vorhanden, womit $\langle A, A, \lambda \rangle \xrightarrow[G']{1} \lambda \uparrow \lambda$ gezeigt ist.

Es sei nun umgekehrt eine Ableitung in G' der Länge 1 gegeben. Dann muss es eine Ableitung $\langle A, A, \lambda \rangle \xrightarrow[G']{\Rightarrow} \lambda \uparrow \lambda$ sein, bei der die Regel $\langle A, A, \lambda \rangle \rightarrow C_{1,1}(\lambda \uparrow \lambda)$ angewendet wird (alle anderen Regeln für ein Symbol $\langle A, B, \eta \rangle$ liefern kein terminales Wort). In G haben wir wegen der Reflexivität von $\xrightarrow[G]{*}$ aber auch die Ableitung $A \xrightarrow[G]{*} A$ der Länge 0.

Induktionsanfang für ii). Eine Ableitung $A \xrightarrow[G]{*} w$ der Länge 1 mit $w \in T^*$ ist genau dann möglich, wenn es die Regel $A \rightarrow w$ in P gibt. Damit gilt $A \xrightarrow[G]{*} w$ genau dann, wenn es die Regel $A \rightarrow C_{1,1}(\lambda \uparrow w)$ in P' gibt, was mit $A \xrightarrow[G']{\Rightarrow} \lambda \uparrow w$ gleichbedeutend ist.

Induktionsschritt für i). Es sei $A[\eta] \xrightarrow[G]{k} w_1 B w_2$ eine Ableitung der Länge k in G .

Fall 1. Als erste Regel wenden wir eine Regel der Form

$$A \rightarrow A_1 A_2 \dots A_{i-1} \underline{A_i} [l] A_{i+1} A_{i+2} \dots A_n$$

an. Dann gibt es eine Ableitung

$$\begin{aligned} A[\eta] &\xrightarrow[G]{\Rightarrow} A_1 A_2 \dots A_{i-1} \underline{A_i} [l\eta] A_{i+1} A_{i+2} \dots A_n \\ &\xrightarrow[G]{k_1} u_1 A_i [l\eta] v_1 \\ &\xrightarrow[G]{k_2} u_1 u_2 B' [\eta] v_2 v_1 \end{aligned} \tag{1.8}$$

$$\xrightarrow[G]{k_3} u_1 u_2 u_3 B v_3 v_2 v_1 \tag{1.9}$$

mit $k_1, k_2, k_3 < k$, terminierenden Ableitungen $A_j \xrightarrow[G]{*} z_j$ für $1 \leq j \leq n$, $j \neq i$, und

$$u_1 = z_1 z_2 \dots z_{i-1}, \quad v_1 = z_{i+1} z_{i+2} \dots z_n \quad \text{und} \quad w_1 = u_1 u_2 u_3, \quad w_2 = v_3 v_2 v_1. \tag{1.10}$$

Nach Induktionsannahme (von ii)) haben wir daher Ableitungen

$$A_j \xrightarrow[G']{*} p_j \uparrow q_j \quad \text{mit} \quad p_j q_j = z_j \quad \text{für} \quad 1 \leq j \leq n, \quad j \neq i. \tag{1.11}$$

Aus den Ableitungen aus (1.8) und (1.9) ergeben sich nach Induktionsvoraussetzung (von i)) die Ableitungen

$$\langle A_i, B', l \rangle \xrightarrow[G']{*} u_2 \uparrow v_2 \quad \text{und} \quad \langle B', B, \eta \rangle \xrightarrow[G']{*} u_3 \uparrow v_3 \tag{1.12}$$

Außerdem haben wir nach Konstruktion von G' die Regeln

$$\langle A, B', \lambda \rangle \rightarrow C_{i,n}(A_1, A_2, \dots, A_{i-1}, \langle A_i, B', l \rangle, A_{i+1}, A_{i+2}, \dots, A_n), \tag{1.13}$$

$$\langle A, B, \eta \rangle \rightarrow W(\langle A, B', \lambda \rangle, \langle B', B, \eta \rangle). \tag{1.14}$$

Damit ergeben sich die Ableitungen

$$\begin{aligned}
\langle A, B', \lambda \rangle &\xrightarrow[G']{*} C_{i,n}(p_1 \uparrow q_1, p_2 \uparrow q_2, \dots, p_{i-1} \uparrow q_{i-1}, u_2 \uparrow v_2, p_{i+1} \uparrow q_{i+1}, p_{i+2} \uparrow q_{i+2}, \dots, p_n \uparrow q_n) \\
&\quad (\text{wegen (1.13), (1.11) und (1.12)}) \\
&= p_1 q_1 p_2 q_2 \dots p_{i-1} q_{i-1} u_2 \uparrow v_2 p_{i+1} q_{i+1} p_{i+2} q_{i+2} \dots p_n q_n \\
&= z_1 z_2 \dots z_{i-1} u_2 \uparrow v_2 z_{i+1} z_{i+2} \dots z_n \\
&= u_1 u_2 \uparrow v_2 v_1 \quad (\text{wegen (1.10)})
\end{aligned}$$

und

$$\begin{aligned}
\langle A, B, \eta \rangle &\xrightarrow[G']{*} W(u_1 u_2 \uparrow v_2 v_1, u_3 \uparrow v_3) \quad (\text{wegen (1.14), vorstehender Ableitung und (1.12)}) \\
&= u_1 u_2 u_3 \uparrow v_3 v_2 v_1 \\
&= w_1 \uparrow w_2 \quad (\text{wegen (1.10)}).
\end{aligned}$$

Fall 2. Die erste angewendete Regel sei von der Form

$$A[l] \rightarrow A_1 A_2 \dots A_{i-1} \underline{A_i} A_{i+1} \dots A_m.$$

Dann gilt $\eta = l$, und unsere Ableitung ergibt sich als

$$\begin{aligned}
A[l] &\xrightarrow[G]{=} A_1 A_2 \dots A_{i-1} A_i A_{i+1} \dots A_m \\
&\xrightarrow[G]{k_1} u_1 A_i v_1 \\
&\xrightarrow[G]{k_2} u_1 u_2 B v_2 u_1
\end{aligned}$$

mit $k_1, k_2 < k$, terminierenden Ableitungen $A_j \xrightarrow[G]{*} z_j$ für $1 \leq j \leq n$, $j \neq i$, und

$$u_1 = z_1 z_2 \dots z_{i-1}, \quad v_1 = z_{i+1} z_{i+2} \dots z_n \quad \text{und} \quad w_1 = u_1 u_2, \quad w_2 = v_2 v_1.$$

Nach Induktionsvoraussetzung haben wir daher in G' die Ableitungen

$$\begin{aligned}
\langle A_i, B, \lambda \rangle &\xrightarrow[G']{*} u_2 \uparrow v_2 \quad \text{und} \\
A_j &\xrightarrow[G']{*} p_j \uparrow q_j \quad \text{mit} \quad p_j q_j = z_j \quad \text{für} \quad 1 \leq j \leq n, \quad j \neq i
\end{aligned}$$

und nach Konstruktion von G' die Regel

$$\langle A, B, l \rangle \rightarrow C_{i,n}(A_1, A_2, \dots, A_{i-1}, \langle A_i, B, \lambda \rangle, A_{i+1}, A_{i+2}, \dots, A_n),$$

woraus die Existenz der Ableitung

$$\begin{aligned}
\langle A, B, l \rangle &\xrightarrow[G']{*} C_{i,n}(p_1 \uparrow q_1, p_2 \uparrow q_2, \dots, p_{i-1} \uparrow q_{i-1}, u_2 \uparrow v_2, p_{i+1} \uparrow q_{i+1}, \dots, p_n \uparrow q_n) \\
&= p_1 q_1 p_2 q_2 \dots p_{i-1} q_{i-1} u_2 \uparrow v_2 p_{i+1} q_{i+1} p_{i+2} q_{i+2} \dots p_n q_n \\
&= z_1 z_2 \dots z_{i-1} u_2 \uparrow v_2 z_{i+1} z_{i+2} \dots z_n \\
&= u_1 u_2 \uparrow v_2 v_1 \\
&= w_1 \uparrow w_2
\end{aligned}$$

resultiert.

Es sei nun umgekehrt $(A, B, \eta) \xrightarrow[G']{k} w_1 \uparrow w_2$ eine Ableitung der Länge k in G' .

Fall 1. Es sei $\eta = \lambda$ und die Ableitung beginne mit der Anwendung einer Regel der Form

$$\langle A, B, \lambda \rangle \rightarrow C_{i,n}(A_1, A_2, \dots, A_{i-1}, \langle A_i, B, l \rangle, A_{i+1}, A_{i+2}, \dots, A_n)$$

und es gibt Ableitungen

$$\langle A_i, B, l \rangle \xrightarrow[G']{k_i} u \uparrow v \quad \text{und} \quad A_j \xrightarrow[G']{k_j} p_j \uparrow q_j \quad \text{für } 1 \leq j \leq n, j \neq i$$

mit $k_j < k$ für $1 \leq j \leq n$ so, dass

$$w_1 \uparrow w_2 = p_1 q_1 p_2 q_2 \dots p_{i-1} q_{i-1} u \uparrow v p_{i+1} q_{i+1} p_{i+2} q_{i+2} \dots p_n q_n$$

gilt. Nach Induktionsannahme haben wir daher in G die Ableitungen

$$A_i[l] \xrightarrow[G]{*} u B v \quad \text{und} \quad A_j \xrightarrow[G]{*} p_j q_j \quad \text{für } 1 \leq j \leq n, j \neq i$$

und aufgrund der Konstruktion von G' in P die Regel

$$A \rightarrow A_1 A_2 \dots A_{i-1} \underline{A_i[l]} A_{i+1} A_{i+2} \dots A_n,$$

womit es in G die gewünschte Ableitung

$$\begin{aligned} A &\xrightarrow[G]{*} A_1 A_2 \dots A_{i-1} \underline{A_i[l]} A_{i+1} A_{i+2} \dots A_n \\ &\xrightarrow[G]{*} p_1 q_1 p_2 q_2 \dots p_{i-1} q_{i-1} u B v p_{i+1} q_{i+1} p_{i+2} q_{i+2} \dots p_n q_n \\ &= w_1 B w_2 \end{aligned}$$

gibt.

Fall 2. Es sei $\eta \in I$ und die Ableitung beginne mit der Anwendung einer Regel der Form

$$\langle A, B, \eta \rangle \rightarrow C_{i,n}(A_1, A_2, \dots, A_{i-1}, \langle A_i, B, \lambda \rangle, A_{i+1}, A_{i+2}, \dots, A_n).$$

Dann gibt es Ableitungen

$$\langle A_i, B, \lambda \rangle \xrightarrow[G']{k_i} u \uparrow v \quad \text{und} \quad A_j \xrightarrow[G']{k_j} p_j \uparrow q_j \quad \text{für } 1 \leq j \leq n, j \neq i$$

mit $k_j < k$ für $1 \leq j \leq n$ so, dass

$$w_1 \uparrow w_2 = p_1 q_1 p_2 q_2 \dots p_{i-1} q_{i-1} u \uparrow v p_{i+1} q_{i+1} p_{i+2} q_{i+2} \dots p_n q_n$$

gilt. Nach Induktionsannahme haben wir daher in G die Ableitungen

$$A_i \xrightarrow[G]{*} u B v \quad \text{und} \quad A_j \xrightarrow[G]{*} p_j q_j \quad \text{für } 1 \leq j \leq n, j \neq i$$

und aufgrund der Konstruktion von G' in P die Regel

$$A[l] \rightarrow A_1 A_2 \dots A_{i-1} \underline{A_i} A_{i+1} A_{i+2} \dots A_n,$$

womit es in G die gewünschte Ableitung

$$\begin{aligned} A[l] &\xrightarrow{G} A_1 A_2 \dots A_{i-1} A_i A_{i+1} A_{i+2} \dots A_n \\ &\xrightarrow{G}^* p_1 q_1 p_2 q_2 \dots p_{i-1} q_{i-1} u B v p_{i+1} q_{i+1} p_{i+2} q_{i+2} \dots p_n q_n \\ &= w_1 B w_2 \end{aligned}$$

gibt.

Fall 3. Es sei $\eta \in I \cup \{\lambda\}$ und die Ableitung beginne mit der Anwendung einer Regel der Form

$$\langle A, B, \eta \rangle \rightarrow W(\langle A, C, \lambda \rangle, \langle C, B, \eta \rangle).$$

Dann gibt es Ableitungen

$$\langle A, C, \lambda \rangle \xrightarrow{G'}^{k_1} u_1 \uparrow v_1 \quad \text{und} \quad \langle C, B, \eta \rangle \xrightarrow{G'}^{k_2} u_2 \uparrow v_2$$

mit $k_1, k_2 < k$ so, dass

$$w_1 \uparrow w_2 = u_1 u_2 \uparrow v_2 v_1$$

gilt. Nach Induktionsannahme haben wir daher in G die Ableitungen

$$A \xrightarrow{G}^* u_1 C v_1 \quad \text{und} \quad C[\eta] \xrightarrow{G}^* u_2 B v_2,$$

woraus die Ableitung

$$A[\eta] \xrightarrow{G}^* u_1 C[\eta] v_1 \xrightarrow{G}^* u_1 u_2 B v_2 v_1 = w_1 B w_2$$

resultiert.

Induktionsschritt für ii). Es sei $A \xrightarrow{G}^* w$ eine Ableitung der Länge k . Diese Ableitung kann in Teilschritte

$$\begin{aligned} A &\xrightarrow{G} A_1 A_2 \dots A_{i-1} A_i[l] A_{i+1} A_{i+2} \dots A_n \\ &\xrightarrow{G}^{k_1} u_1 A_i[l] v_1 \\ &\xrightarrow{G}^{k_2} u_1 u_2 B v_2 v_1 \end{aligned} \tag{1.15}$$

$$\xrightarrow{G}^{k_3} u_1 u_2 u v_2 v_1 \tag{1.16}$$

mit $k_1, k_2, k_3 < k$, terminierenden Ableitungen $A_j \xrightarrow{G}^* z_j$ für $1 \leq j \leq n$, $j \neq i$, und

$$u_1 = z_1 z_2 \dots z_{i-1}, \quad v_1 = z_{i+1} z_{i+2} \dots z_n \quad \text{und} \quad w = u_1 u_2 u v_2 v_1$$

zerlegt werden. Nach Induktionsvoraussetzung haben wir in G' die Ableitungen

$$\begin{aligned} A_j &\xrightarrow{G'}^* p_j \uparrow q_j \quad \text{mit} \quad p_j q_j = z_j && \text{für } 1 \leq j \leq n, \quad j \neq i, \\ \langle A_i, B, l \rangle &\xrightarrow{G'}^* u_2 \uparrow v_2 && \text{(wegen (1.15)),} \\ B &\xrightarrow{G'}^* u_3 \uparrow v_3 \quad \text{mit} \quad u_3 v_3 = u && \text{(wegen (1.16))} \end{aligned}$$

und in P' haben wir nach Konstruktion die Regeln

$$\begin{aligned} \langle A, B, \lambda \rangle &\rightarrow C_{i,n}(A_1, A_2, \dots, A_{i-1}, \langle A_i, B, l \rangle, A_{i+1}, A_{i+2}, \dots, A_n), \\ A &\rightarrow W(\langle A, B, \lambda \rangle, B). \end{aligned}$$

Damit ergeben sich die Ableitungen

$$\begin{aligned} \langle A, B, \lambda \rangle &\xrightarrow[G']{*} C_{i,n}(p_1 \uparrow q_1, p_2 \uparrow q_2, \dots, p_{i-1} \uparrow q_{i-1}, u_2 \uparrow v_2, p_{i+1} \uparrow q_{i+1}, p_{i+2} \uparrow q_{i+2}, \dots, p_n \uparrow q_n) \\ &= p_1 q_1 p_2 q_2 \dots p_{i-1} q_{i-1} u_2 \uparrow v_2 p_{i+1} q_{i+1} p_{i+2} q_{i+2} \dots p_n q_n \\ &= z_1 z_2 \dots z_{i-1} u_2 \uparrow v_2 z_{i+1} z_{i+2} \dots z_n \\ &= u_1 u_2 \uparrow v_2 v_1 \end{aligned}$$

und

$$A \xrightarrow[G']{*} W(u_1 u_2 \uparrow v_2 v_1, u_3 \uparrow v_3) = u_1 u_2 u_3 \uparrow v_3 v_2 v_1,$$

womit wegen $u_1 u_2 u_3 v_3 v_2 v_1 = u_1 u_2 u v_2 v_1 = w$ die Behauptung gezeigt ist.

Ist $A \xrightarrow[G']{k} w_1 \uparrow w_2$ eine Ableitung der Länge $k \geq 2$ in G' , so beginnt diese Ableitung mit der Anwendung der Regel $A \rightarrow W(\langle A, B, \lambda \rangle, B)$ für ein $B \in N$ (da dies die einzigen nicht terminierenden Regeln für ein $A \in N$ sind). Daher gibt es auch noch Ableitungen

$$\langle A, B, \lambda \rangle \xrightarrow[G']{k_1} u_1 \uparrow v_1 \quad \text{und} \quad B \xrightarrow[G']{k_2} u_2 \uparrow v_2$$

mit $k_1, k_2 < k$, $w_1 = u_1 u_2$ und $w_2 = v_2 v_1$. Nach Induktionsvoraussetzung gibt es daher in G die Ableitungen $A \xrightarrow[G]{*} u_1 B v_1$ und $B \xrightarrow[G]{*} u_2 v_2$, womit auch die Ableitung

$$A \xrightarrow[G]{*} u_1 B v_1 \xrightarrow[G]{*} u_1 u_2 v_2 v_1 = w_1 w_2$$

besteht, die nachzuweisen war.

Die Aussage zur Komplexität der Konstruktionen ergibt sich sofort. \square

Folgerung 1.57 $\mathcal{L}(LI) \subseteq \mathcal{L}(H)$.

Beweis. Es sei $L \in \mathcal{L}(LI)$. Folglich gibt es eine lineare indizierte Grammatik G mit $L = L(G)$. Nach Lemma 1.56 gibt es dann eine Kopf-Grammatik G' mit $L(G') = L(G) = L$. Damit gilt $L \in \mathcal{L}(H)$. \square

Lemma 1.58 *Zu jeder Kopf-Grammatik G kann man in polynomialer Zeit (in der Größe von G) eine Baum einfügende Grammatik G' so konstruieren, dass $L(G') = L(G)$ gilt und die Größe von G' polynomial in der Größe von G ist.*

Beweis. Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $G = (N, T, P, S)$ in der Normalform aus Satz 1.55 vorliegt, d. h. P enthält nur Regeln der Form

$$A \rightarrow C_{1,2}(B, C), A \rightarrow C_{2,2}(B, C), A \rightarrow W(B, C), A \rightarrow C_{1,1}(w_1 \uparrow w_2)$$

mit $A, B, C \in N$, $w_1, w_2 \in T^*$. Wir konstruieren nun eine Baum einfügende Grammatik $G' = (N, T, \{\mathcal{B}\}, Aux, S)$ mit

$$\mathcal{B} = (S, Aux, 1) \\ \downarrow \\ \lambda$$

als initialem Baum. Die Menge Aux entsteht aus Bäumen, die in Abbildung 1.17 gezeigt sind, indem wir für eine Regel

- $p = A \rightarrow C_{1,2}(B, C)$ den Baum \mathcal{B}_p ,
- $q = A \rightarrow C_{2,2}(B, C)$ den Baum \mathcal{B}_q ,
- $r = A \rightarrow W(B, C)$ den Baum \mathcal{B}_r und
- $s = A \rightarrow C_{1,1}(w_1 \uparrow w_2)$ den Baum \mathcal{B}_s

nehmen.

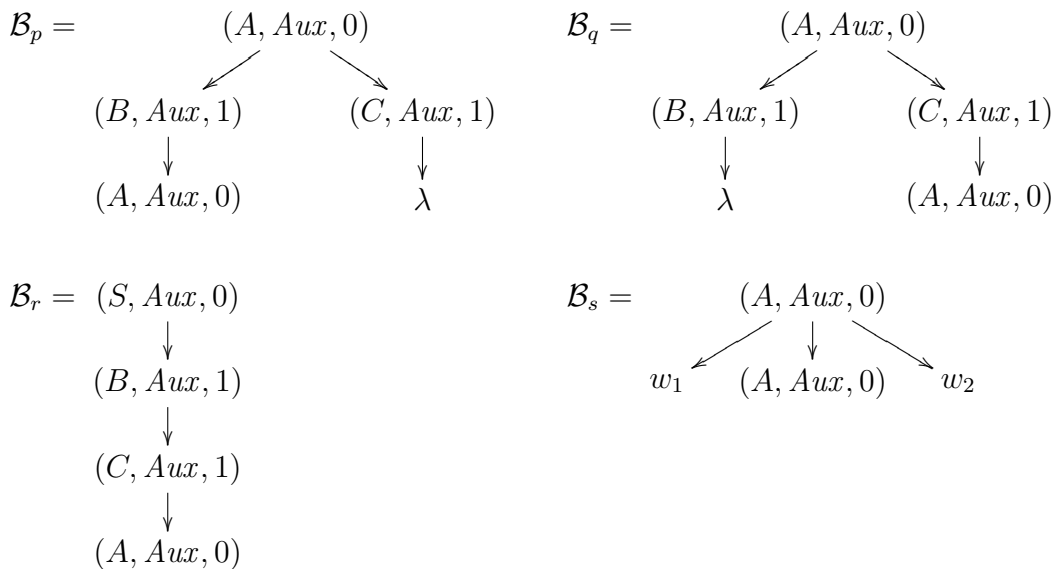


Abbildung 1.17: Bäume aus Aux

Wir zeigen nun zuerst mittels vollständiger Induktion über die Anzahl der Schritte beim Ableiten bzw. Baum-Einfügen, dass *für alle $A \in N$ und $w_1, w_2 \in T^*$ die Ableitung $A \xrightarrow[G]{*} w_1 \uparrow w_2$ genau dann besteht, wenn Bäume \mathcal{B}_1 und \mathcal{B}_2 so existieren, dass folgende Bedingungen erfüllt sind:*

- \mathcal{B}_1 liegt in Aux und die Wurzel von \mathcal{B}_1 ist mit A markiert,
- \mathcal{B}_2 entsteht aus \mathcal{B}_1 durch iteriertes Baum-Einfügen,
- \mathcal{B}_2 hat keinen Knoten, der noch ersetzt werden muss (kein Knoten hat eine Markierung $(X, U, 1)$), und die Blätter von \mathcal{B}_2 ergeben von links nach rechts gelesen das Wort $w_1 A w_2$.

Induktionsanfang: Wird bei $A \Rightarrow w_1 \uparrow w_2$ nur ein Ableitungsschritt ausgeführt, so ist $s = A \rightarrow C_{1,1}(w_1 \uparrow w_2)$ eine Regel in P . Somit gibt es in Aux den zugehörigen Baum \mathcal{B}_s . Mit der Festlegung $\mathcal{B}_1 = \mathcal{B}_2 = \mathcal{B}_s$ sind alle Aussagen erfüllt.

Entsteht umgekehrt \mathcal{B}_2 aus \mathcal{B}_1 durch 0 Baum-Einfügungen, so ist \mathcal{B}_1 ein Baum der Form \mathcal{B}_s und die Anwendung der zugehörigen Regel s liefert $A \xrightarrow[G]{} w_1 \uparrow w_2$.

Induktionsschritt: Es sei $A \xrightarrow[G]{k} w_1 \uparrow w_2$ eine Ableitung der Länge $k \geq 2$. Wir diskutieren nun die drei Möglichkeiten für die zuerst angewendete Regel (der vierte Typ entfällt, da ihm eine terminierende Regel zugrunde liegt).

Fall 1. Es sei $p = A \rightarrow C_{1,2}(B, C)$ die zuerst angewendete Regel. Dann gibt es Ableitungen

$$B \xrightarrow[G]{k_1} u_1 \uparrow u_2 \quad \text{und} \quad C \xrightarrow[G]{k_2} v_1 \uparrow v_2$$

mit $k_1, k_2 < k$ derart, dass

$$C_{1,2}(u_1 \uparrow u_2, v_1 \uparrow v_2) = u_1 \uparrow u_2 v_1 v_2 = w_1 \uparrow w_2$$

gilt. Damit existieren nach Induktionsvoraussetzung Bäume \mathcal{B}_1 , \mathcal{B}_2 , \mathcal{C}_1 und \mathcal{C}_2 mit folgenden Eigenschaften:

- \mathcal{B}_1 und \mathcal{C}_1 liegen in Aux und die Wurzeln sind mit B bzw. C markiert,
- \mathcal{B}_2 entsteht aus \mathcal{B}_1 und \mathcal{C}_2 entsteht aus \mathcal{C}_1 jeweils durch iteriertes Baum-Einfügen,
- \mathcal{B}_2 und \mathcal{C}_2 haben keine Knoten mit einer Markierung $(X, U, 1)$, und die Blätter von \mathcal{B}_2 bzw. \mathcal{C}_2 ergeben von links nach rechts gelesen die Wörter $u_1 B u_2$ bzw. $v_1 C v_2$.

Folglich haben \mathcal{B}_2 und \mathcal{C}_2 auch die Wurzeln B bzw. C . Wir fügen nun \mathcal{B}_2 und \mathcal{C}_2 in den Baum \mathcal{B}_p ein, der zur zuerst angewendeten Regel gehört. Dies liefert den Baum \mathcal{B}' aus Abbildung 1.18, der die Wurzel A hat, der keine mit $(X, U, 1)$ markierten Knoten enthält und dessen Blätter von links nach rechts gelesen das Wort $u_1 A u_2 v_1 v_2 = w_1 A w_2$ ergeben. Folglich erfüllen \mathcal{B}_p und \mathcal{B}' die geforderten Bedingungen.

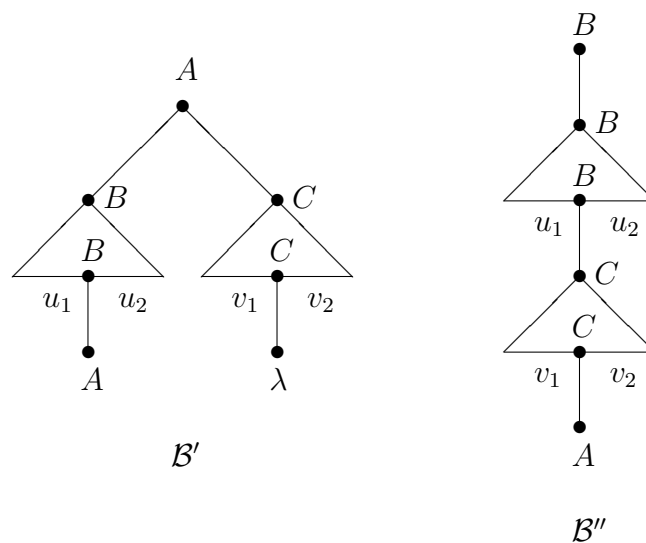


Abbildung 1.18: Bäume \mathcal{B}' und \mathcal{B}'' aus dem Beweis von Lemma 1.58

Fall 2. Die erste angewendete Regel sei $q = A \rightarrow C_{2,2}(B, C)$. Dann können wir eine analoge Konstruktion zum Fall 1 geben.

Fall 3. Die erste angewendete Regel sei $r = A \rightarrow W(B, C)$. Dann gibt es Ableitungen

$$B \xrightarrow[G]{k_1} u_1 \uparrow u_2 \quad \text{und} \quad C \xrightarrow[G]{k_2} v_1 \uparrow v_2$$

mit $k_1, k_2 < k$ derart, dass

$$W(u_1 \uparrow u_2, v_1 \uparrow v_2) = u_1 v_1 \uparrow v_2 u_2 = w_1 \uparrow w_2$$

gilt. Damit existieren nach Induktionsvoraussetzung Bäume $\mathcal{B}_1, \mathcal{B}_2, \mathcal{C}_1$ und \mathcal{C}_2 mit den in Fall 1 genannten Eigenschaften. Analog zu Fall 1 erhalten wir unter Verwendung von \mathcal{B}_r den Baum \mathcal{B}'' aus Abbildung 1.18, die beide den gewünschten Bedingungen genügen.

Für die Umkehrung der Aussage sei ein Baum \mathcal{B}_2 aus einem Baum \mathcal{B}_1 durch k -maliges Baum-Einfügen entstanden und die Bäume \mathcal{B}_1 und \mathcal{B}_2 mögen die in der Aussage genannten Bedingungen erfüllen. Wir diskutieren die Möglichkeiten für \mathcal{B}_1 .

Fall 1. Der Baum \mathcal{B}_1 habe die Form \mathcal{B}_p für eine Regel $p = A \rightarrow C_{1,2}(B, C)$. Da im Baum \mathcal{B}_2 keine Knoten mehr zu ersetzen sind, ist \mathcal{B}_2 aus \mathcal{B}_1 entstanden, indem an die Knoten mit den Markierungen B und C zwei Bäume \mathcal{B} und \mathcal{C} eingefügt wurden. Damit entspricht der Baum \mathcal{B}_2 dem Baum \mathcal{B}' aus Abbildung 1.18. Zu den eingefügten Bäumen \mathcal{B} und \mathcal{C} gehören nach Induktionsvoraussetzung zwei Ableitungen $B \xrightarrow[G]{*} u_1 \uparrow u_2$ und $C \xrightarrow[G]{*} v_1 \uparrow v_2$. Dies liefert die Existenz der gewünschten Ableitung

$$A \xrightarrow[G]{*} C_{1,2}(u_1 \uparrow u_2, v_1 \uparrow v_2) = u_1 \uparrow u_2 v_1 v_2.$$

Fall 2. Der Baum \mathcal{B}_1 habe die Form \mathcal{B}_q für eine Regel $q = A \rightarrow C_{2,2}(B, C)$. Durch analoge Betrachtungen wie in Fall 1 kann der Beweis erbracht werden.

Fall 3. Der Baum \mathcal{B}_1 habe die Form \mathcal{B}_r für eine Regel $r = A \rightarrow W(B, C)$. Auch hier kann die Aussage analog zu Fall 1 bewiesen werden.

Es sei $w \in L(G)$. Dann gibt es in der Kopf-Grammatik G eine Ableitung $S \xrightarrow[G]{*} w_1 \uparrow w_2$ mit $w = w_1 w_2$. Nach der gerade bewiesenen Aussage gibt es einen Baum \mathcal{B} , dessen Wurzel mit S markiert ist, der aus Bäumen in Aux durch Einfügen entstanden ist, der keinen Knoten mit einer Markierung $(X, U, 1)$ enthält und dessen Blätter von links nach rechts gelesen $w_1 S w_2$ ergeben. Fügen wir diesen Baum \mathcal{B} in den einzigen Baum in $Init$ ein, so erhalten wir einen Baum mit Wurzel S , ohne Knoten mit einer Markierung $(X, U, 1)$ und dessen Blätter von links nach rechts gelesen $w_1 w_2$ ergeben. Damit gilt $w \in L(G')$ und folglich $L(G) \subseteq L(G')$.

Die umgekehrte Inklusion kann durch Umkehren der Schlüsse bewiesen werden. \square

Die folgende Aussage können wir analog zur Folgerung 1.57 beweisen.

Folgerung 1.59 $\mathcal{L}(H) \subseteq \mathcal{L}(TA)$. \square

Lemma 1.60 *Zu jeder Baum einfügenden Grammatik G kann in polynomialer Zeit (in der Größe von G) eine lineare indizierte Grammatik G' so konstruiert werden, dass $L(G') = L(G)$ gilt und die Größe von G' polynomial in der Größe von G ist.*

Beweis. Es sei $G = (N, T, Init, Aux, S)$ eine Baum einfügende Grammatik. Ohne Beschränkung der Allgemeinheit haben die Bäume in $Init$ und Aux die folgenden Eigenschaften:

1. Es gibt keine Substitutionsknoten (Knoten mit einer Markierung (A, \downarrow)). Dies ist nach Satz 1.42 möglich.
2. Alle Blätter – mit Ausnahme der Fußknoten von den Bäumen in Aux – sind mit einem Terminal oder λ markiert. Dies folgt aus der Definition 1.37 unter der Bedingung, dass es keine Substitutionsknoten gibt.
3. Jede Wurzel hat die Markierung $(S, \emptyset, 0)$. Dies kann gewährleistet werden, indem ein Baum mit einer Wurzelmarkierung (S, U, a) eine neue Wurzel „vorgehängt“ bekommt, die die Markierung $(S, \emptyset, 0)$ erhält. Der gesamte ursprüngliche Baum hängt danach an diesem Knoten. Diese Änderung hat keinen Einfluss auf die Terminalsymbole des betreffenden Baumes und verändert somit die erzeugte Sprache nicht. Da Substitutionen nicht möglich sind, tragen nur Bäume zur erzeugten Sprache bei, in deren Wurzelmarkierung S steht.
4. Jeder Fuß eines Baumes in Aux hat eine Markierung $(A, \emptyset, 0)$. Dies kann analog zur Wurzel erreicht werden, indem an einen Fußknoten mit einer Markierung (A, U, a) ein Knoten mit der Markierung $(A, \emptyset, 0)$ angehängt wird. Der angehängte Knoten ist der neue Fuß; der ehemalige Fuß wird zu einem inneren Knoten. Auch diese Modifikation verändert die erzeugte Sprache nicht.

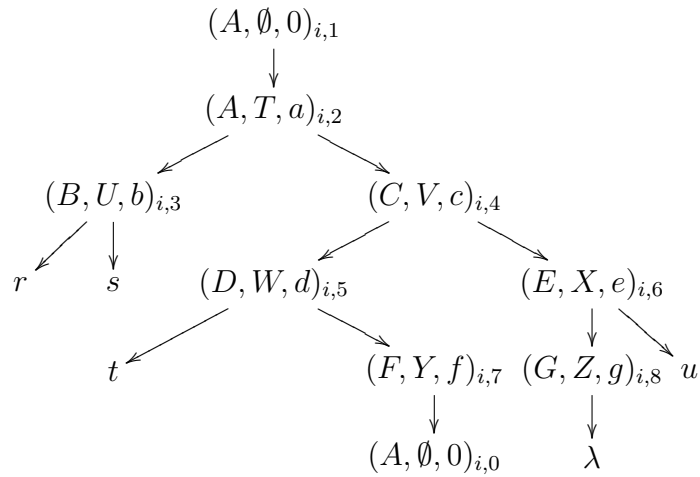
Es seien B_1, B_2, \dots, B_n die Bäume aus $Init \cup Aux$. In jedem Baum B_i , $1 \leq i \leq n$, versehen wir die Nichtterminal-Knoten (und deren Markierungen) mit dem Baumindex i und einer bezüglich des Baumes fortlaufenden Nummer j . Dabei wird $j = 1$ an die Wurzel vergeben und $j = 0$ an den Fuß, falls vorhanden; in einem Baum aus $Init$ tritt $j = 0$ nicht auf. In einem Baum B_i sei n_i die Anzahl der Nichtterminalknoten ohne den Fußknoten; die Knoten seien $K_{i,0}, K_{i,1}, \dots, K_{i,n_i}$, deren Markierungen seien $M_{i,0}, M_{i,1}, \dots, M_{i,n_i}$. Mit $N(i, j)$ für $1 \leq i \leq n$ und $1 \leq j \leq n_i$ sei jenes Wort bezeichnet, das die Markierungen der Nachfolger des Knotens $K_{i,j}$ (von links nach rechts gelesen) ergeben (für Fußknoten $K_{i,0}$ ist $N(i, 0)$ nicht definiert, da es keine Nachfolger gibt). Darüber hinaus gelte für jedes Wort $N(i, j)$ folgendes: Falls B_i ein Baum aus Aux ist und eine der in $N(i, j)$ auftretenden Nichtterminalmarkierungen auf dem Pfad von der Wurzel zum Fuß liegt, so betrachten wir diese Markierung in $N(i, j)$ als unterstrichen. Die Abbildung 1.19 zeigt ein Beispiel für einen Baum $B_i \in Aux$. Darin gilt u. a. $N(i, 5) = t(F, Y, f)_{i,7}$ und $N(i, 6) = (G, Z, g)_{i,8u}$.

Wir werden nun eine lineare indizierte Grammatik $\overline{G'} = (N', T, I, P, S)$ konstruieren, deren Ableitungsbäume in gewisser Weise den Bäumen entsprechen, die die Grammatik G erzeugt. Als Nichtterminale nehmen wir alle vorkommenden Knotenmarkierungen sowie das Symbol S :

$$N' = \{S\} \cup \{M_{i,j} \mid 1 \leq i \leq n, B_i \in Init, 1 \leq j \leq n_i\} \\ \cup \{M_{i,j} \mid 1 \leq i \leq n, B_i \in Aux, 0 \leq j \leq n_i\}.$$

Die Indices werden die Paare (i, j) , die als Knotenindices auftreten:

$$I = \{(i, j) \mid 1 \leq i \leq n, 1 \leq j \leq n_i, M_{i,j} \in N'\}.$$

Abbildung 1.19: Beispiel für einen Baum $B_i \in Aux$

Außerdem nehmen wir die folgenden Regeln auf:

- Startregeln

$$S \rightarrow M_{i,1}$$

für $1 \leq i \leq n$ und $B_i \in Init$. Jedes Wort, das von der Baum einfügenden Grammatik G erzeugt wird, ergibt sich aus den Blattmarkierungen eines Baumes, der aus einem initialen Baum B_i durch Einfügen von Bäumen aus Aux entsteht.

- Regeln, die einen Baum „durchlaufen“

$$M_{i,j} \rightarrow N(i, j)$$

für $1 \leq i \leq n$, $1 \leq j \leq n_i$, wobei der Knoten $K_{i,j}$ nicht ersetzt werden muss (die Markierung ist $M_{i,j} = (A, U, 0)_{i,j}$ für ein Nichtterminal $A \in N$ und eine Menge $U \subseteq Aux$). Durch diese Regeln entstehen Ableitungsbäume, die auch von der Grammatik G erzeugt werden.

- Regeln, um zu einem einzufügenden Baum „zu springen“

$$M_{i,j} \rightarrow M_{k,1}[(i, j)]$$

für $1 \leq i \leq n$, $2 \leq j \leq n_i$, $1 \leq k \leq n$, wobei $M_{i,j} = (A, U, a)_{i,j}$ für ein Nichtterminal $A \in N$, eine Menge $U \subseteq Aux$ und $a \in \{0, 1\}$ gilt sowie B_k in U liegt. Durch diese Regeln wird simuliert, dass ein Baum B_k an einem Knoten $K_{i,j}$ in den Baum B_i eingefügt wird. Das Paar (i, j) wird als Index gespeichert, damit an den Fuß von B_k die Nachfolger des Knotens $K_{i,j}$ „angehängt“ werden (und nicht die Nachfolger irgend eines anderen Knotens). Für $j = 0$ oder $j = 1$ gibt es solche Regeln nicht, da aufgrund der Eigenschaften der Bäume an den Wurzeln und Füßen keine Bäume eingefügt werden können.

- Regeln, um von einem eingefügten Baum „zurück zu springen“

$$M_{k,0}[(i, j)] \rightarrow N(i, j)$$

für $1 \leq i \leq n$, $2 \leq j \leq n_i$ und $1 \leq k \leq n$. Wegen $j \neq 0$ ist $N(i, j)$ stets definiert. Diese Regeln „beenden“ das Einfügen des Baumes B_k in den Baum B_i (an den Knoten $K_{i,j}$), indem an den Fußknoten $K_{k,0}$ die Nachfolger des Knotens $K_{i,j}$ „angehängt“ werden. Für $j = 0$ oder $j = 1$ gibt es solche Regeln nicht, da es dafür auch keine Index aufbauenden Regeln gibt (da an den Wurzeln und Fußknoten keine Bäume eingefügt werden).

Alle diese genannten Regeln fassen wir zur Menge P zusammen. Wir werden nun zeigen, dass die lineare indizierte Grammatik $G' = (N', T, I, P, S)$ die gleiche Sprache erzeugt wie die Baum einfügende Grammatik G . Dazu beweisen wir die folgenden beiden Behauptungen mittels vollständiger Induktion über die „Schachtelungstiefe“:

1. Für alle natürlichen Zahlen i mit $1 \leq i \leq n$ und $B_i \in Aux$ sowie Wörter $w_1, w_2 \in T^*$ gilt $M_{i,1} \xrightarrow[G']{*} w_1 M_{i,0} w_2$ genau dann, wenn es einen Baum T mit folgenden Eigenschaften gibt:
 - T entsteht aus dem Baum $B_i \in Aux$ durch Einfügen von Bäumen aus Aux ,
 - T enthält keinen Knoten der ersetzt werden muss (bei keiner Markierung ist die dritte Komponente gleich Eins) und
 - die Blätter von T ergeben von links nach rechts gelesen das Wort $w_1 M_{i,0} w_2$.
2. Für alle natürlichen Zahlen i mit $1 \leq i \leq n$ und $B_i \in Init$ sowie Wörter $w \in T^*$ gilt $M_{i,1} \xrightarrow[G']{*} w$ genau dann, wenn es einen Baum T mit folgenden Eigenschaften gibt:
 - T entsteht aus dem Baum $B_i \in Init$ durch Einfügen von Bäumen aus Aux ,
 - T enthält keinen Knoten der ersetzt werden muss (bei keiner Markierung ist die dritte Komponente gleich Eins) und
 - die Blätter von T ergeben von links nach rechts gelesen das Wort w .

Zu 1. *Induktionsanfang*: Es sei $M_{i,1} \xrightarrow[G']{*} w_1 M_{i,0} w_2$ eine Ableitung bezüglich G' , bei der zu keiner Zeit Indices auftreten. Damit sind nur Baum durchlaufende Regeln $M_{i,j} \rightarrow N(i, j)$ angewendet worden. Dann ist jedes in der Ableitung auftretende Nichtterminal von der Form $(A, U, 0)_{i,j}$ (es ist Markierung im Baum B_i ; die dritte Komponente ist 0, da anderweitig eine Index aufbauende Regel angewendet werden müsste, um das Nichtterminal abzuleiten). Aufgrund der vorhandenen Nichtterminale und der Regeln der Grammatik G' hat die Ableitung als Ableitungsbaum B_i . Der Baum B_i erfüllt die genannten Bedingungen für T .

Es sei nun $B_i \in Aux$ ein Baum, dessen Blätter von links nach rechts gelesen ein Wort $w_1 M_{i,0} w_2$ ergeben und dessen Knoten nicht ersetzt werden müssen. Wegen der Konstruktion von G' ist B_i Ableitungsbaum für die Ableitung $M_{i,1} \xrightarrow[G']{*} w_1 M_{i,0} w_2$, bei der nur Baum durchlaufende Regeln $M_{i,j} \rightarrow N(i, j)$ angewendet werden.

Induktionsschritt: Es sei $D = (M_{i,1} \xrightarrow[G']{*} w_1 M_{i,0} w_2)$ eine Ableitung bezüglich G' , bei der zu jeder Zeit an jedem Nichtterminal höchstens $n > 0$ Indices vorkommen. Damit sind nicht nur Baum durchlaufende Regeln $M_{i,j} \rightarrow N(i, j)$ angewendet worden. Folglich wurde eine Index aufbauende Regel $p_k = (M_{i,k} \rightarrow M_{m,1}[(i, k)])$ verwendet. In der Ableitung D tritt kein Nichtterminal erneut auf, wenn es einmal mittels einer Regel ersetzt worden

ist. Also wird jede Regel höchstens einmal verwendet. Da der Index (i, k) auch wieder abgebaut werden muss, gibt es eine „Unterableitung“ $M_{m,1} \xrightarrow[G']{*} v_1 M_{m,0} v_2$, in der zu jeder Zeit an jedem Nichtterminal höchstens $n - 1$ Indices auftreten. Nach Induktionsvoraussetzung gibt es einen Baum T_k , der aus dem Baum $B_m \in Aux$ entsteht, dessen Knoten nicht ersetzt werden müssen und dessen Blätter von links nach rechts gelesen das Wort $v_1 M_{m,0} v_2$ ergeben. Es sei T' ein Baum, der aus dem Baum $B_i \in Aux$ entsteht, in dem an jedem Knoten $K_{i,k}$ der Baum T_k eingefügt wird, falls auf die Markierung $M_{i,k}$ die Index aufbauende Regel p_k angewendet wird. Da auf eine Markierung mit einer 1 in der dritten Komponente nur eine Index aufbauende Regel angewendet werden kann, wird jeder Knoten mit einer 1 in der Markierung durch einen Baum ersetzt. Folglich enthält der Baum T' keine zu ersetzenden Knoten mehr (bei jeder Markierung steht in der dritten Komponente 0). Der Baum T' entspricht im Wesentlichen dem Ableitungsbaum zu der Ableitung D mit dem Unterschied, dass in T' die Nichtterminale nicht mit Indexfolgen versehen sind und die Knoten, an denen Bäume eingefügt wurden, nicht mehr vorhanden sind, wogegen im Ableitungsbaum die betreffenden Nichtterminale noch auftreten. Dies hat jedoch keinen Einfluss auf die Blätter, so dass die Blätter in T' von links nach rechts gelesen das Wort $w_1 M_{i,0} w_2$ ergeben. Der Baum T' erfüllt also die in der Behauptung genannten Bedingungen.

Es sei nun T ein Baum, der aus einem Baum $B_i \in Aux$ durch Einfügen von Bäumen aus Aux entsteht, der keine zu ersetzenden Knoten enthält und dessen Blätter von links nach rechts gelesen ein Wort $w_1 M_{i,0} w_2$ ergeben. Aufgrund der Konstruktion von G' gibt es die Ableitung $M_{i,1} \xrightarrow[G']{*} w_1 M_{i,0} w_2$. Ausgehend von der Wurzelmarkierung $M_{i,1}$ im Baum T werden zunächst Baum durchlaufende Regeln der Form $M_{i,j} \rightarrow N(i, j)$ angewendet. Wurde an einem Knoten $K_{i,j}$ ein Baum T' eingefügt, der wiederum aus einem Baum $B_m \in Aux$ durch Einfügen entstand, der keine zu ersetzenden Knoten enthält und dessen Blätter ein Wort $v_1 M_{m,1} v_2$ ergeben, so wird das Nichtterminal $M_{i,j}$ in mehreren Schritten zu $v_1 N(i, j) v_2$ abgeleitet. Dies ist möglich, da es wegen der vorhandenen Regeln und nach Induktionsvoraussetzung folgende Ableitung gibt:

$$M_{i,j} \xrightarrow[G']{\Rightarrow} M_{m,1}[(i, j)] \xrightarrow[G']{*} v_1 M_{m,1}[(i, j)] v_2 \xrightarrow[G']{\Rightarrow} v_1 N(i, j) v_2.$$

Zu 2. Diese Behauptung kann analog zur ersten bewiesen werden. Aus der zweiten Behauptung folgt wegen der vorhandenen Startregeln $S \rightarrow M_{i,1}$, dass eine Ableitung $S \xrightarrow[G']{\Rightarrow} M_{i,1} \xrightarrow[G']{*} w$ genau dann existiert, wenn w von der Baum einfügenden Grammatik G erzeugt wird. Damit gilt $L(G') = L(G)$. Die Größe der Grammatik G sei k . Dann gilt

$$k = |N| + |T| + |\{ K \mid K \text{ ist Knoten eines Baumes } B_i \in Init \cup Aux \}| \\ + |\{ E \mid E \text{ ist Kante eines Baumes } B_i \in Init \cup Aux \}|.$$

Für die Größe k' der linearen indizierten Grammatik G' gilt $k' = |N'| + |T'| + |I'| + |P'|$ mit $|N'| + |T'| \leq k$, $|I'| \leq k$ und $|P'| \leq 2k^2 + 2k$ und folglich $k' \leq 2k^2 + 4k$. Somit erzeugt die konstruierte lineare indizierte Grammatik G' nicht nur die gleiche Sprache wie die Baum einfügende Grammatik G , sondern ihre Größe ist auch polynomial in der Größe von G . \square

Erneut folgt aus Lemma 1.60 die zugehörige Inklusion.

Folgerung 1.61 $\mathcal{L}(TA) \subseteq \mathcal{L}(LI)$. □

Durch Kombination der vorstehenden Folgerungen 1.57, 1.59 und 1.61 erhalten wir sofort die folgende Aussage.

Satz 1.62 $\mathcal{L}(LI) = \mathcal{L}(H) = \mathcal{L}(TA)$. □

Da die Sprachfamilien übereinstimmen, lassen sich die Ergebnisse, die wir bisher nur für eine Familie haben, auf die anderen Sprachfamilien übertragen.

Folgerung 1.63

- i) *Jede Sprache in $\mathcal{L}(LI)$ ist semi-linear.*
- ii) *Jede Sprache in $\mathcal{L}(H)$ ist semi-linear.*

Beweis. Nach Satz 1.43 ist jede Sprache in $\mathcal{L}(TA)$ semi-linear. Folglich gilt das wegen Satz 1.62 auch für die Sprachen, die von linearen indizierten Grammatiken bzw. Kopf-Grammatiken erzeugt werden. □

Folgerung 1.64

- i) *Das Mitgliedsproblem für Baum einfügende Grammatiken ist in polynomialer Zeit entscheidbar.*
- ii) *Das Mitgliedsproblem für Kopf-Grammatiken ist in polynomialer Zeit entscheidbar.*

Beweis.

- i) Es seien eine (feste) Baum einfügende Grammatik G und ein Wort w gegeben. Wir konstruieren zuerst entsprechend Lemma 1.60 eine lineare indizierte Grammatik G' mit $L(G') = L(G)$. Folglich gilt $w \in L(G)$ genau dann, wenn $w \in L(G')$ gilt. Letztere Aussage ist nach Satz 1.34 in polynomialer Zeit entscheidbar. Folglich ist auch $w \in L(G)$ in polynomialer Zeit entscheidbar. (Man beachte, dass der Aufwand für Transformation von G zu G' und die Größe von G' polynomial nur von der Größe von G abhängen; bei einer festen Grammatiken also durch eine Konstante beschränkt sind.)
- ii) Die Aussage beweist man analog unter Verwendung von i) und Lemma 1.58. □

Folgerung 1.65 *Die Mengen der linearen indizierten Grammatiken, der Baum einfügenden Grammatiken und der Kopf-Grammatiken sind jeweils schwach kontextabhängig.*

Beweis. Wir haben für alle drei Grammatikklassen gezeigt, dass die Sprachen K_1 , K_2 und K_3 erzeugbar sind, alle kontextfreien Sprachen erzeugt werden, jede erzeugte Sprache semi-linear ist und das Mitgliedsproblem in polynomialer Zeit entscheidbar ist.

Die Tabelle 1.1 gibt eine Übersicht.

Somit haben wir alle Eigenschaften, die eine schwach kontextabhängige Grammatikklasse haben soll, für die Mengen der linearen indizierten Grammatiken, der Baum einfügenden Grammatiken und der Kopf-Grammatiken jeweils nachgewiesen. □

Damit haben wir unser Ziel, die Klasse der kontextfreien Grammatiken so zu erweitern, dass eine schwach kontextabhängige Grammatikklasse entsteht, in drei Fällen erreicht. Allerdings liefern die drei schwach kontextabhängigen Grammatikklassen nur eine Sprachfamilie, da die Erzeugungskraft gleich ist.

	K_1, K_2, K_3	$\mathcal{L}(CF)$	semi-linear	Mitgliedsproblem
Lineare indizierte G.	Beispiele 1.22–1.24	Satz 1.32	Folg. 1.63	Satz 1.34
Baum einfügende G.	Beispiele 1.38–1.40	Satz 1.41	Satz 1.43	Folg. 1.64
Kopf-G.	Beispiele 1.51–1.53	Satz 1.54	Folg. 1.63	Folg. 1.64

Tabelle 1.1: Übersicht über den Nachweis der schwachen Kontextabhängigkeit

Kapitel 2

Einfügende Grammatiken und streichende Automaten

2.1. Kontextuale Grammatiken

2.1.1. Allgemeine kontextuale Grammatiken

Wir erinnern zuerst an die Idee, aus Sätzen neue Sätze zu konstruieren, indem Wörter eingefügt werden. Als Ausgangspunkt wählen wir den Satz

Der Mann geht.

der deutschen Sprache. Hieraus gewinnen wir durch Einschoben des Adjektivs *alt* vor das Substantiv den Satz

Der alte Mann geht.

Hieraus können wir durch Anfügen des Adverbs *langsam* den Satz

Der alte Mann geht langsam.

generieren.

Wir geben nun die formale Definition einiger grundlegender Typen von kontextualen Grammatiken.

Definition 2.1

- i) Eine kontextuale Grammatik ist ein Tripel $G = (V, B, C)$, wobei
 - V ein (endliches) Alphabet ist,
 - B eine endliche Teilmenge von V^* ist,
 - C eine endliche Menge von Paaren (u, v) mit $u, v \in V^*$ und $uv \neq \lambda$ ist.
- ii) Für zwei Wörter $x \in V^*$ und $y \in V^*$ sagen wir, dass y durch äußeres Ableiten aus x entsteht, falls $y = uxv$ für ein Paar $(u, v) \in C$ gilt. Wir schreiben dies als $x \xrightarrow{\text{ex}} y$. Wir sagen, dass y durch inneres Ableiten aus x entsteht, und schreiben $x \xrightarrow{\text{in}} y$, falls $x = x_1x_2x_3$ und $y = x_1ux_2vx_3$ für gewisse Wörter $x_1, x_2, x_3 \in V^*$ und ein Paar $(u, v) \in C$ gelten.
Mit $\xrightarrow{\text{ex}}^*$ und $\xrightarrow{\text{in}}^*$ bezeichnen wir den reflexiven und transitiven Abschluss der Relation $\xrightarrow{\text{ex}}$ bzw. $\xrightarrow{\text{in}}$.

iii) Die von einer kontextualen Grammatik G erzeugten Sprachen definieren wir als

$$L_{\text{ex}}(G) = \{ w \mid z \xrightarrow[\text{ex}]{}^* w \text{ für ein } z \in B \} \text{ und}$$

$$L_{\text{in}}(G) = \{ w \mid z \xrightarrow[\text{in}]{}^* w \text{ für ein } z \in B \}.$$

Die Menge B heißt Basis von G , und die Elemente von C werden Kontexte genannt.

Beim äußeren Ableiten werden die Kontexte jeweils um das ganze Wort geschrieben, während sie beim inneren Ableiten um ein (beliebiges) Teilwort geschrieben werden. Wir werden auch abkürzend von einer äußeren oder inneren kontextualen Grammatik sprechen, wenn wir eine kontextuale Grammatik im äußeren bzw. inneren Ableitungsmodus arbeitend meinen.

Wir geben nun einige Beispiele.

Beispiel 2.2 Für jede kontextuale Grammatik $G_1 = (V, B, \emptyset)$ gilt offenbar

$$L_{\text{ex}}(G_1) = L_{\text{in}}(G_1) = B,$$

da wegen der fehlenden Kontexte keine Ableitungsschritte durchgeführt werden können, und daher wegen der Reflexivität von $\xrightarrow[\text{ex}]{}^*$ und $\xrightarrow[\text{in}]{}^*$ nur (und alle) Wörter aus B in der erzeugten Sprache liegen.

Analog erhält man für eine beliebige kontextuale Grammatik $G = (V, B, C)$ die Beziehungen

$$B \subseteq L_{\text{ex}}(G) \quad \text{und} \quad B \subseteq L_{\text{in}}(G).$$

Beispiel 2.3 Wir betrachten die kontextuale Grammatik

$$G_2 = (\{a, b\}, \{ab\}, \{(a, b)\}).$$

Im äußeren Ableitungsmodus ergibt sich die eindeutige Ableitung

$$ab \xrightarrow[\text{ex}]{} aabb \xrightarrow[\text{ex}]{} aaabbb \xrightarrow[\text{ex}]{} \dots \xrightarrow[\text{ex}]{} a^n b^n \xrightarrow[\text{ex}]{} \dots,$$

woraus

$$L_{\text{ex}}(G_2) = \{ a^n b^n \mid n \geq 1 \}$$

resultiert.

Durch inneres Ableiten hingegen ergibt sich unter anderem die folgende Ableitung:

$$ab \xrightarrow[\text{in}]{} aabb \xrightarrow[\text{in}]{} aababb \xrightarrow[\text{in}]{} aabaabbb \xrightarrow[\text{in}]{} aaabababb,$$

wobei wir als Wort, um das der Kontext gesetzt wird, der Reihe nach b , das zweite b , das dritte a und das erste Vorkommen von ba verwendet haben. Im inneren Ableitungsmodus wird die Dyck-Sprache $D_{a,b}$ erzeugt, d. h. die Menge aller korrekt geklammerten Ausdrücke über $\{a, b\}$ (wobei a für die öffnende und b für die schließende Klammer steht). Dies ist wie folgt zu sehen.

$D_{a,b} \subseteq L_{\text{in}}(G_2)$. Wir beweisen diese Aussage durch vollständige Induktion über die Länge n der Wörter aus $D_{a,b}$. Für den Induktionsanfang $n = 2$ ist ab das einzige Wort aus $D_{a,b}$. Wegen $ab \in B$ gilt auch $ab \in L_{\text{in}}(G_2)$. Es sei nun y ein korrekt geklammertes Wort der Länge $n \geq 4$. Dann beginnt y mit einem a und es gibt korrekt geklammerte Wörter x_2 und x_3 derart, dass $y = ax_2bx_3$ gilt. Damit ist auch $x = x_2x_3$ korrekt geklammert. Nach Induktionsannahme gilt damit $z \xrightarrow[\text{in}]^* x$ für ein $z \in B$. Da aber $x = x_2x_3 \xrightarrow[\text{in}]^* ax_2bx_3 = y$ (mit $x_1 = \lambda$) gilt, erhalten wir auch $z \xrightarrow[\text{in}]^* y$, womit $y \in L_{\text{in}}(G_2)$ gezeigt ist.

$L_{\text{in}}(G_2) \subseteq D_{a,b}$. Offensichtlich ist das einzige Wort in B (das durch 0 Ableitungsschritte erzeugt wird) korrekt geklammert. Mittels vollständiger Induktion über die Anzahl der Ableitungsschritte kann die gewünschte Aussage nun gezeigt werden (wir verzichten auf einen detaillierten Beweis, der eine umfangreiche Fallunterscheidung erfordert).

Beispiel 2.4 Wir betrachten die kontextuale Grammatik

$$G_3 = (\{a, b, c\}, \{ \alpha\beta\gamma \mid \{\alpha, \beta, \gamma\} = \{a, b, c\} \}, \{ (\alpha\beta, \gamma) \mid \{\alpha, \beta, \gamma\} = \{a, b, c\} \}).$$

Es sei

$$H = \{ w \mid w \in \{a, b, c\}^+, \#_a(w) = \#_b(w) = \#_c(w) \}.$$

Wir zeigen, dass $L_{\text{in}}(G_3) = H$ gilt.

Da in jedem Wort aus der Basismenge $\{ \alpha\beta\gamma \mid \{\alpha, \beta, \gamma\} = \{a, b, c\} \}$ jeder Buchstabe genau einmal vorkommt und durch jeden Kontext genau ein weiteres Vorkommen eines jeden Buchstaben zu einem Wort hinzugefügt wird, ergibt sich durch vollständige Induktion über die Anzahl der Ableitungsschritte sofort, dass jedes Wort $w \in L_{\text{in}}(G_3)$ die Bedingungen $w \in \{a, b, c\}^+$ und $\#_a(w) = \#_b(w) = \#_c(w)$ erfüllt, womit die Inklusion $L_{\text{in}}(G_3) \subseteq H$ gezeigt ist.

Wir zeigen nun $H \subseteq L_{\text{in}}(G_3)$ durch vollständige Induktion über die Wortlänge. Die Wörter der Länge 3 aus H liegen alle in B und folglich in $L_{\text{in}}(G_3)$. Es sei nun $w \in H$ ein Wort der Länge $n \geq 6$. Der erste Buchstabe von w sei α . Dann gilt $w = \alpha^k\beta w'$ für ein $k \geq 1$, ein $\beta \in \{a, b, c\}$ mit $\beta \neq \alpha$ und ein Wort $w' \in \{a, b, c\}^*$. Außerdem gibt es in w' ein Vorkommen von $\gamma \in \{a, b, c\}$ mit $\gamma \neq \alpha$ und $\gamma \neq \beta$, d. h. $w = \alpha^{k-1}\alpha\beta w_1\gamma w_2$ für gewisse Wörter w_1 und w_2 . Offensichtlich liegt $x = \alpha^{k-1}w_1w_2$ in H und damit nach Induktionsannahme in $L_{\text{in}}(G_3)$. Somit gibt es eine Ableitung $z \xrightarrow[\text{in}]^* x$ mit z aus der Basis der Grammatik. Da offensichtlich durch Setzen des Kontextes $(\alpha\beta, \gamma)$ um w_1 das Wort w entsteht, haben wir die Ableitung $z \xrightarrow[\text{in}]^* x \xrightarrow[\text{in}]^* w$, womit $w \in L_{\text{in}}(G_3)$ gezeigt ist.

Beispiel 2.5 Die (äußere) kontextuale Grammatik $G_4 = (\{a, b\}, \{aa, bb\}, \{(a, a), (b, b)\})$ erzeugt die Sprache

$$L_{\text{ex}}(G_4) = \{ ww^R \mid w \in \{a, b\}^+ \},$$

wobei w^R das Spiegelwort von w ist ($(a_1a_2 \dots a_n)^R = a_na_{n-1} \dots a_1$), denn es gibt nur die Ableitungen

$$x_1x_1 \xrightarrow[\text{ex}]{} x_2x_1x_1x_2 \xrightarrow[\text{ex}]{} x_3x_2x_1x_1x_2x_3 \xrightarrow[\text{ex}]{} \dots \xrightarrow[\text{ex}]{} x_nx_{n-1} \dots x_2x_1x_1x_2 \dots x_{n-1}x_n$$

mit $x_i \in \{a, b\}$ für $1 \leq i \leq n$.

Beispiel 2.6 Die Grammatik $G_5 = (\{a, b\}, \{bb\}, \{(a, \lambda), (\lambda, a)\})$ erzeugt die Sprachen

$$L_{\text{in}}(G_5) = \{ a^n b a^m b a^k \mid n, m, k \geq 0 \} \text{ und } L_{\text{ex}}(G_5) = \{ a^n b b a^m \mid n, m \geq 0 \},$$

da an jeder Stelle bzw. nur am Anfang oder Ende Vorkommen von a hinzugefügt werden.

Diese Grammatiken sind zum Erzeugen natürlicher Sprachen jedoch kaum geeignet, da die Kontexte relativ beliebig eingefügt werden dürfen, also nicht wirklich ein Kontext vom gesamten Wort im Fall äußerer Grammatiken bzw. von einem Teilwort bei inneren Grammatiken sind. So können wir im einleitenden Beispiel die Einfügung des Adjektivs durch den Kontext (λ, alte) realisieren, aber dies erlaubt auch die Ableitungen

$$\text{Der Mann geht.} \xrightarrow{\text{in}} \text{Der Mann alte geht.}$$

und

$$\text{Der Mann geht.} \xrightarrow{\text{in}} \text{Der Mann geht alte.}$$

Beide sind grammatikalisch keine korrekten Sätze der deutschen Sprache. Wir müssen also fordern, dass die Kontexte nur an gewissen Stellen eingefügt werden dürfen. Im Beispiel wäre zu fordern, dass der gegebene Kontext nur um einen Artikel gesetzt werden darf. Dieser Gedanke wird durch die folgende Definition realisiert.

Definition 2.7

- i) Eine kontextuale Grammatik mit Auswahl ist ein Quadrupel $G = (V, B, C, \varphi)$, wobei (V, B, C) eine kontextuale Grammatik darstellt und $\varphi : V^* \rightarrow 2^C$ eine berechenbare Funktion von der Menge der Wörter über V in die Potenzmenge von C ist.
- ii) Für zwei Wörter $x \in V^*$ und $y \in V^*$ sagen wir, dass y in einem Schritt durch äußeres Ableiten aus x entsteht, falls $y = uxv$ für ein Paar $(u, v) \in \varphi(x)$ gilt. Wir schreiben dies als $x \xrightarrow{\text{ex},c} y$. Wir sagen, dass y durch inneres Ableiten aus x entsteht, und schreiben $x \xrightarrow{\text{in},c} y$, falls $x = x_1 x_2 x_3$ und $y = x_1 u x_2 v x_3$ für gewisse Wörter $x_1, x_2, x_3 \in V^*$ und ein Paar $(u, v) \in \varphi(x_2)$ gelten.

Mit $\xrightarrow{\text{ex},c}^*$ und $\xrightarrow{\text{in},c}^*$ bezeichnen wir den reflexiven und transitiven Abschluss der Relation $\xrightarrow{\text{ex},c}$ bzw. $\xrightarrow{\text{in},c}$.

- iii) Die von einer kontextualen Grammatik G mit Auswahl erzeugten Sprachen definieren wir als

$$L_{\text{ex},c}(G) = \{ w \mid z \xrightarrow{\text{ex},c}^* w \text{ für ein } z \in B \} \text{ und}$$

$$L_{\text{in},c}(G) = \{ w \mid z \xrightarrow{\text{in},c}^* w \text{ für ein } z \in B \}.$$

Definition 2.8

- i) Eine totale kontextuale Grammatik ist ein Quadrupel $G = (V, B, C, \varphi)$, wobei (V, B, C) eine kontextuale Grammatik darstellt und φ eine berechenbare Funktion von $V^* \times V^* \times V^*$ in 2^C ist.

- ii) Für zwei Wörter $x \in V^*$ und $y \in V^*$ sagen wir, dass y aus x abgeleitet wird (in Zeichen $x \xrightarrow[t]{\Rightarrow} y$), falls $x = x_1x_2x_3$ und $y = x_1ux_2vx_3$ für gewisse Wörter $x_1, x_2, x_3 \in V^*$ und ein Paar $(u, v) \in \varphi(x_1, x_2, x_3)$ gelten. Mit $\xrightarrow[t]{\Rightarrow^*}$ bezeichnen wir den reflexiven und transitiven Abschluss von $\xrightarrow[t]{\Rightarrow}$.
- iii) Die von einer totalen kontextualen Grammatik G erzeugte Sprache $L_t(G)$ definieren wir als

$$L_t(G) = \{ w \mid z \xrightarrow[t]{\Rightarrow^*} w \text{ für ein } z \in B \}.$$

Beispiel 2.9 Wir betrachten die kontextuale Grammatik

$$G_6 = (\{a\}, \{a^3\}, \{(\lambda, a), (\lambda, a^2)\}, \varphi_6)$$

mit Auswahl und

$$\varphi_6(a^i) = \begin{cases} \{(\lambda, a)\} & \text{falls } i+1 \neq 2^k \text{ für alle } k \geq 0, \\ \{(\lambda, a^2)\} & \text{sonst.} \end{cases}$$

Im Falle des äußeren Ableitens haben wir

$$a^3 \xrightarrow[\text{ex,c}]{\Rightarrow} a^5 \xrightarrow[\text{ex}]{\Rightarrow} a^6 \xrightarrow[\text{ex,c}]{\Rightarrow} a^7 \xrightarrow[\text{ex}]{\Rightarrow} a^9 \xrightarrow[\text{ex,c}]{\Rightarrow} a^{10} \xrightarrow[\text{ex,c}]{\Rightarrow} \dots,$$

und offensichtlich gilt allgemein

$$a^i \xrightarrow[\text{ex,c}]{\Rightarrow} \begin{cases} a^{i+1} & \text{falls } i+1 \neq 2^k \text{ für alle } k \geq 0, \\ a^{i+2} & \text{sonst,} \end{cases}$$

woraus

$$L_{\text{ex,c}}(G_6) = \{ a^n \mid n \geq 3, n \neq 2^k \text{ für alle } k \geq 0 \}$$

resultiert.

Betrachten wir dagegen das innere Ableiten, so ergibt sich

$$a^i = a^2 a^{i-2} \xrightarrow[\text{in,c}]{\Rightarrow} a^2 a a^{i-2} = a^{i+1}$$

durch Einfügen des Kontextes (λ, a) um a^2 . Folglich lassen sich alle a^i mit $i \geq 3$ erzeugen, womit wir

$$L_{\text{in,c}}(G_6) = \{ a^n \mid n \geq 3 \}$$

erhalten.

Beispiel 2.10 Für die totale Grammatik $G_7 = (\{a, b, c\}, \{abc\}, \{(ab, c)\}, \varphi_7)$ mit

$$\varphi_7(x_1, x_2, x_3) = \begin{cases} \{(ab, c)\} & \text{falls } x_1 = a^n, x_2 = b^m, x_3 = c^k \text{ für gewisse } n, m, k \geq 1, \\ \emptyset & \text{sonst} \end{cases}$$

ergibt sich

$$L_t(G_7) = K_1 = \{ a^n b^n c^n \mid n \geq 1 \},$$

da

$$abc \xrightarrow[t]{\implies} aabbcc = a^2 b^2 c^2 \xrightarrow[t]{\implies} a^2 abb^2 cc^2 = a^3 b^3 c^3 \xrightarrow[t]{\implies} a^3 abb^3 cc^3 = a^4 b^4 c^4 \xrightarrow[t]{\implies} \dots$$

die einzige Ableitung ist.

Wir wollen nun zeigen, dass sich K_1 nicht durch äußere oder innere kontextuale Grammatiken mit Auswahl erzeugen lässt.

Angenommen, K_1 wird von einer kontextualen Grammatik $G = (\{a, b, c\}, B, C, \varphi)$ erzeugt. Wir betrachten ein Wort $y = a^n b^n c^n$, wobei $n > \max\{|z| \mid z \in B\}$ gelte. Da y aufgrund seiner Länge nicht zur Menge B gehört, gibt es eine Ableitung $z \xrightarrow[m,c]{*} x \xrightarrow[m,c]{} y$ für ein Wort $z \in B$ und einen Modus $m \in \{\text{ex}, \text{in}\}$. Offensichtlich gilt auch $x \in L_{m,c}(G)$. Folglich haben wir $x = a^p b^p c^p$ für eine natürliche Zahl $p < n$. Jedoch gilt für jeden Kontext $(u, v) \in \varphi(x)$ sicher $a^n b^n c^n = y \neq u x v = u a^p b^p c^p v$. Damit kann y also nicht durch äußeres Ableiten entstanden sein. Die innere Ableitung $a^p b^p c^p \xrightarrow[\text{in},c]{\implies} a^n b^n c^n$ ist nur möglich, wenn $(a^{n-p} b^{n-p}, c^{n-p}) \in \varphi_2(b^p)$ oder $(a^{n-p}, b^{n-p} c^{n-p}) \in \varphi_2(b^p)$ gilt. Das Wort $a^{p+1} b^{p+1} c^{p+1}$ liegt in K_1 und wegen unserer Annahme auch in $L_{\text{in},c}(G)$. Daher gibt es auch die Ableitung

$$z' \xrightarrow[\text{in},c]{*} a^{p+1} b^{p+1} c^{p+1} = a^{p+1} b b^p c^{p+1} \xrightarrow[\text{in},c]{\implies} a^{p+1} b a^{n-p} b^n c^{n-p} c^{p+1} = a^{p+1} b a^{n-p} b^n c^{n+1}$$

mit $z' \in B$, die zu einem Wort führt, das nicht in K_1 liegt. Somit kann y auch nicht durch inneres Ableiten entstanden sein. Die Sprache K_1 wird also weder durch äußeres noch durch inneres Ableiten von einer kontextualen Grammatik erzeugt.

Beispiel 2.11 Es sei $G_8 = (\{a, b, c, d\}, \{abcd\}, \{(a, c), (b, d)\}, \varphi_8)$ eine innere kontextuale Grammatik mit Auswahl, wobei φ_8 durch

$$\varphi_8(w) = \begin{cases} \{(a, c)\} & \text{falls } w = ab^i c \text{ für ein } i \in \mathbb{N}, \\ \{(b, d)\} & \text{falls } w = bc^i d \text{ für ein } i \in \mathbb{N}, \\ \emptyset & \text{sonst} \end{cases}$$

definiert sei. Aus einem Wort $a^n b^m c^n d^m$ entsteht durch Einfügen des Kontextes (a, c) um das Teilwort $ab^m c$ das Wort $a^{n+1} b^m c^{n+1} d^m$ und durch Einfügen des Kontextes (b, d) um das Teilwort $bc^n d$ das Wort $a^n b^{m+1} c^n d^{m+1}$. Somit ist die erzeugte Sprache $L_{\text{in},c}(G_8) = K_2$.

Ändern wir die Auswahlfunktion zu

$$\varphi'_8(x_1, x_2, x_3) = \begin{cases} \{(a, c)\} & \text{falls } x_1 \in \{a\}^*, x_2 = ab^i c \text{ für ein } i \in \mathbb{N}, x_3 \in \{c\}^* \{d^*\}, \\ \{(b, d)\} & \text{falls } x_1 \in \{a\}^* \{b\}^*, x_2 = bc^i d \text{ für ein } i \in \mathbb{N}, x_3 \in \{d^*\}, \\ \emptyset & \text{sonst} \end{cases}$$

ab, so erhalten wir eine totale kontextuale Grammatik G'_8 , die ebenfalls die Sprache K_2 erzeugt.

Mittels innerer Ableitung ohne Auswahl ist die Sprache K_2 nicht erzeugbar. Angenommen, sie wäre es. Dann müssen in einem Kontext (u, v) mindestens ein a und ein c auftreten. Fügen wir so einen Kontext (u, v) das Wort $abbbcbddd$ ein, erhalten wir beispielsweise das Wort $abubvcbddd$, was jedoch nicht zur Sprache K_2 gehört.

Beispiel 2.12 Es sei $G_9 = (\{a, b\}, \{aa, bb\}, \{(a, a), (b, b)\}, \varphi_9)$ eine totale kontextuale Grammatik, wobei φ_9 durch

$$\varphi_9(x_1, x_2, x_3) = \begin{cases} \{(a, a), (b, b)\} & \text{falls } x_1 = \lambda \text{ und } x_2 = x_3, \\ \emptyset & \text{sonst} \end{cases}$$

definiert sei. Wie man mittels vollständiger Induktion leicht nachweist, wird die Sprache $K_3 = \{ww \mid w \in \{a, b\}^+\}$ von dieser Grammatik erzeugt.

In den letzten drei Beispielen haben wir gezeigt, dass die Sprachen K_1 , K_2 und K_3 von totalen kontextualen Grammatiken erzeugbar sind.

Mit $\mathcal{L}(EC)$, $\mathcal{L}(ECC)$, $\mathcal{L}(IC)$, $\mathcal{L}(ICC)$ und $\mathcal{L}(TC)$ bezeichnen wir die Mengen von Sprachen, die von äußeren kontextualen Grammatiken ohne bzw. mit Auswahl, inneren kontextualen Grammatiken ohne bzw. mit Auswahl und totalen kontextualen Grammatiken erzeugt werden.

Wir geben nun einige Eigenschaften kontextualer Sprachen an.

Satz 2.13 Für jede Sprache L aus $\mathcal{L}(X)$ mit $X \in \{EC, ECC, IC, ICC, TC\}$ gibt es Konstanten k und k' derart, dass zu jedem Wort $w \in L$ mit $|w| > k$ ein Wort $w' \in L$ existiert, so dass $0 < |w| - |w'| < k'$ gilt.

Beweis. Es seien $X \in \{EC, ECC, IC, ICC, TC\}$ und $L \in \mathcal{L}(X)$. Dann gibt es eine kontextuale Grammatik $G = (V, B, C, \varphi)$ (möglicherweise mit Auswahl oder total), die die Sprache L erzeugt. Wir setzen

$$k = 1 + \max\{|z| \mid z \in B\} \quad \text{und} \quad k' = 1 + \max\{|uv| \mid (u, v) \in C\}.$$

Es sei w ein Wort in L mit $|w| > k$. Nach Definition von k liegt w nicht in B . Folglich gibt es eine Ableitung $z \xrightarrow[G]{*} x_1x_2x_3 \xrightarrow[G]{*} x_1ux_2vx_3 = w$ für Wörter $z \in B$, $x_1, x_2, x_3 \in V^*$ und einen Kontext $(u, v) \in C$. Offenbar gilt auch $x_1x_2x_3 \in L$. Außerdem haben wir noch

$$0 < |w| - |x_1x_2x_3| = (|x_1x_2x_3| + |uv|) - |x_1x_2x_3| = |uv| < k'$$

wegen $uv \neq \lambda$ und der Definition von k' . □

Damit ist die Forderung von JOSHI hinsichtlich der „Wortabstände“ für die schwache Kontextabhängigkeit einer Familie von Sprachen für jede der von uns bisher eingeführten Klassen kontextualer Sprachen erfüllt. Wir diskutieren nun die stärkere Forderung nach der Semi-Linearität der Sprachen.

Satz 2.14 Jede Sprache aus $\mathcal{L}(X)$ mit $X \in \{EC, IC\}$ ist semi-linear.

Beweis. Wir geben den Beweis nur für $\mathcal{L}(EC)$; der Beweis für $\mathcal{L}(IC)$ kann analog geführt werden.

Es sei $L \in \mathcal{L}(EC)$. Dann gibt es eine äußere kontextuale Grammatik

$$G = (V, \{w_1, w_2, \dots, w_r\}, \{(u_1, v_1), (u_2, v_2), \dots, (u_s, v_s)\})$$

mit $L = L_{\text{ex}}(G)$. Wir setzen

$$\begin{aligned} \underline{p}_i &= \pi_V(w_i) \quad \text{für } 1 \leq i \leq r, \\ \underline{q}_j &= \pi_V(u_j v_j) \quad \text{für } 1 \leq j \leq s, \\ H &= \left\{ \underline{p}_i + \sum_{j=1}^s \alpha_j \underline{q}_j \mid 1 \leq i \leq r, \alpha_j \in \mathbb{N}, 1 \leq j \leq s \right\}. \end{aligned}$$

Wir beweisen $\pi_V(L) = H$, womit dann die Behauptung gezeigt ist.

$H \subseteq \pi_V(L)$. Wir beweisen diese Inklusion durch vollständige Induktion über $t = \sum_{j=1}^s \alpha_j$. Ist $t = 0$, so haben wir nur \underline{p}_i für ein i , $1 \leq i \leq r$. Wegen $\underline{p}_i = \pi_V(w_i)$ und $w_i \in B \subseteq L_{\text{ex}}(G) = L$ gilt $\underline{p}_i \in \pi_V(L)$. Es sei nun

$$\underline{x} = \underline{p}_i + \sum_{j=1}^s \alpha_j \underline{q}_j \quad \text{mit} \quad \sum_{j=1}^s \alpha_j > 0$$

ein Element aus H . Wegen $\sum_{j=1}^s \alpha_j > 0$ gibt es ein k mit $\alpha_k > 0$, $1 \leq k \leq s$. Wir betrachten

$$\underline{x}' = \underline{p}_i + \sum_{j=1}^{k-1} \alpha_j \underline{q}_j + (\alpha_k - 1) \underline{q}_k + \sum_{j=k+1}^s \alpha_j \underline{q}_j \in H.$$

Da offenbar

$$\sum_{j=1}^{k-1} \alpha_j + (\alpha_k - 1) + \sum_{j=k+1}^s \alpha_j < \sum_{j=1}^s \alpha_j$$

gilt, ist nach Induktionsannahme \underline{x}' auch in $\pi_V(L)$. Somit gibt es ein Wort $w \in L$ mit $\pi_V(w) = \underline{x}'$. Wegen $L = L_{\text{ex}}(G)$ gibt es eine Ableitung $z \xrightarrow[\text{ex}]{*} w$ in G . Damit haben wir auch die Ableitung $z \xrightarrow[\text{ex}]{*} w \xrightarrow[\text{ex}]{} u_k w v_k$, womit $u_k w v_k \in L_{\text{ex}}(G) = L$ gilt. Wegen

$$\begin{aligned} \pi_V(u_k w v_k) &= \pi_V(w) + \pi_V(u_k v_k) \\ &= \underline{p}_i + \sum_{j=1}^{k-1} \alpha_j \underline{q}_j + (\alpha_k - 1) \underline{q}_k + \sum_{j=k+1}^s \alpha_j \underline{q}_j + \underline{q}_k \\ &= \underline{p}_i + \sum_{j=1}^s \alpha_j \underline{q}_j \\ &= \underline{x} \end{aligned}$$

ist $\underline{x} \in \pi_V(L)$ gezeigt.

$\pi_V(L) \subseteq H$. Wir beweisen diese Inklusion durch vollständige Induktion über die Anzahl t der Ableitungsschritte. Für $t = 0$ erhalten wir nur die Wörter w_i , $1 \leq i \leq r$, aus B . Wegen $\pi_V(w_i) = \underline{p}_i$ ergibt sich $\pi_V(w_i) \in H$. Es sei nun $z \xrightarrow[\text{ex}]{*} w \xrightarrow[\text{ex}]{} u_k w v_k$ mit $z \in B$ eine Ableitung in $t + 1$ Schritten. Dann haben wir

$$\pi_V(u_k w v_k) = \pi_V(w) + \pi_V(u_k v_k) = \pi_V(w) + \underline{q}_k.$$

Da nach Induktionsannahme $\pi_V(w)$ in H liegt, da w durch t Schritte erzeugt wird, ergibt sich auch $\pi_V(u_k w v_k) \in H$. \square

Äußere und innere kontextuale Grammatiken mit Auswahl sowie totale kontextuale Grammatiken hingegen erzeugen auch Sprachen, die nicht semi-linear sind.

Beispiel 2.15 Wir betrachten die äußere kontextuale Grammatik

$$G_{10} = (\{a, b\}, \{a\}, \{(\lambda, a), (\lambda, b)\}, \varphi_{10})$$

mit Auswahl, wobei φ_{10} durch

$$\varphi_{10}(w) = \begin{cases} \{(\lambda, a)\} & \text{falls } w = a^i, i \neq 2^k \text{ für alle } k \geq 0, \\ \{(\lambda, a), (\lambda, b)\} & \text{falls } w = a^{2^k} \text{ für ein } k \geq 0, \\ \emptyset & \text{sonst} \end{cases}$$

definiert sei. An ein Wort a^n darf stets ein a angefügt werden, während ein b nur hinzugefügt werden darf, wenn n eine Potenz von 2 ist. Nach dem Hinzufügen von b bricht der Ableitungsprozess ab. Daher ergibt sich

$$L_{\text{ex,c}}(G_{10}) = \{a^n \mid n \geq 1\} \cup \{a^{2^k} b \mid k \geq 0\}.$$

Wir betrachten wir die Sprache

$$L_{\text{ex,c}}(G_{10}) = \{a^n \mid n \geq 1\} \cup \{a^{2^k} b \mid k \geq 0\}$$

aus Beispiel 2.15. Offensichtlich ist

$$\pi(L_{\text{ex,c}}(G_{10})) = \{(n, 0) \mid n \geq 1\} \cup \{(2^k, 1) \mid k \geq 0\}.$$

Wir schneiden nun $\pi(L_{\text{ex,c}}(G_{10}))$ mit der semi-linearen Menge

$$H = \{(n, 1) \mid n \geq 1\} = \{(1, 1) + (n-1)(1, 0) \mid n \geq 1\}$$

und erhalten

$$\pi(L_{\text{ex,c}}(G_{10})) \cap H = \{(2^k, 1) \mid k \geq 0\}.$$

Da semi-lineare Mengen unter Durchschnitt abgeschlossen sind, aber $\{(2^k, 1) \mid k \geq 0\}$ offensichtlich keine semi-lineare Menge ist, ist auch $\pi(L_{\text{ex,c}}(G_{10}))$ nicht semi-linear.

Für den Beweis, dass es auch eine innere kontextuale Grammatik mit Auswahl gibt, die keine semi-lineare Sprache erzeugt, verweisen wir auf [16].

Wir geben nun zwei Pumping-Lemmata für kontextuale Sprachen.

Satz 2.16 Für jede Sprache $L \in \mathcal{L}(ICC)$ gibt es Konstanten k und k' derart, dass sich jedes Wort $z \in L$ mit $|z| > k$ als $z = x_1ux_2vx_3$ mit $0 < |uv| < k'$ schreiben lässt und $x_1u^ix_2v^ix_3 \in L$ für jedes $i \geq 0$ gilt.

Beweis. Es sei $L \in \mathcal{L}(ICC)$. Dann gibt es eine kontextuale Grammatik $G = (V, B, C, \varphi)$ mit Auswahl derart, dass $L_{\text{in},c}(G) = L$ gilt. Wir setzen

$$k = 1 + \max\{|z| \mid z \in B\} \quad \text{und} \quad k' = 1 + \max\{|uv| \mid (u, v) \in C\}.$$

Es sei w ein Wort in L mit $|w| > k$. Nach Definition von k , kann dann w nicht in B liegen. Folglich gibt es eine Ableitung $z \xrightarrow[\text{in},c]{*} x_1x_2x_3 \xrightarrow[\text{in},c]{=} x_1ux_2vx_3 = w$ für einen Kontext $(u, v) \in \varphi(x_2)$. Offenbar gilt auch $x_1x_2x_3 \in L$. Dann gibt es aber auch die Ableitung

$$z \xrightarrow[\text{in},c]{*} x_1x_2x_3 \xrightarrow[\text{in},c]{=} x_1ux_2vx_3 \xrightarrow[\text{in},c]{=} x_1uux_2vux_3 \xrightarrow[\text{in},c]{=} x_1u^3x_2v^3x_3 \xrightarrow[\text{in},c]{=} \cdots \xrightarrow[\text{in},c]{=} x_1u^ix_2v^ix_3,$$

womit $x_1u^ix_2v^ix_3 \in L$ für alle $i \geq 0$ gezeigt ist. Außerdem gilt $|uv| < k'$ nach Definition von k' . \square

Die folgende Aussage wird analog bewiesen.

Satz 2.17 Für jede Sprache $L \in \mathcal{L}(EC)$ gibt es Konstanten k und k' derart, dass sich jedes Wort $z \in L$ mit $|z| > k$ als $z = uxv$ mit $0 < |uv| < k'$ schreiben lässt und $u^ixv^i \in L$ für jedes $i \geq 0$ gilt. \square

Aus Satz 2.16 erhalten wir $K_1 \notin \mathcal{L}(ICC)$, jedoch weder die Aussage $K_2 \notin \mathcal{L}(ICC)$ noch $K_3 \notin \mathcal{L}(ICC)$, da im Gegensatz zum Pumpinglemma für kontextfreie Sprachen die Länge von x hier nicht beschränkt ist. Insbesondere haben wir im Beispiel 2.11 nachgewiesen, dass $K_2 \in \mathcal{L}(ICC)$ gilt.

Aus Satz 2.17 folgt

$$K_1 \notin \mathcal{L}(EC), \quad K_2 \notin \mathcal{L}(EC) \quad \text{und} \quad K_3 \notin \mathcal{L}(EC).$$

Man beachte, dass wir bereits $K_1 \notin \mathcal{L}(ECC)$ gezeigt haben (siehe Beispiel 2.10).

Wir vergleichen nun die Erzeugungskraft der verschiedenen Typen kontextualer Grammatiken untereinander und mit der der Grammatiken der Chomsky-Hierarchie.

Satz 2.18 Es gilt das Diagramm aus Abbildung 2.1. Dabei bedeutet die Existenz eines gerichteten Pfades von X nach Y , dass $X \subset Y$ gilt. Sind zwei Mengen nicht durch einen gerichteten Pfad verbunden, so sind sie unvergleichbar.

Beweis.

- a) Die (echten) Inklusionen zwischen den Familien der Chomsky-Hierarchie sind bekannt (siehe Satz 1.9). Die Inklusionen zwischen den Familien der kontextualen Sprachen ergeben sich wie folgt.

$$\mathcal{L}(IC) \subseteq \mathcal{L}(ICC).$$

Es sei $L \in \mathcal{L}(IC)$. Dann gibt es eine kontextuale Grammatik $G = (V, B, C)$ mit $L = L_{\text{in}}(G)$. Die innere kontextuale Grammatik $G' = (V, B, C, \varphi)$ mit Auswahl und $\varphi(w) = C$ für alle $w \in V^*$ erzeugt ebenfalls die Sprache L .

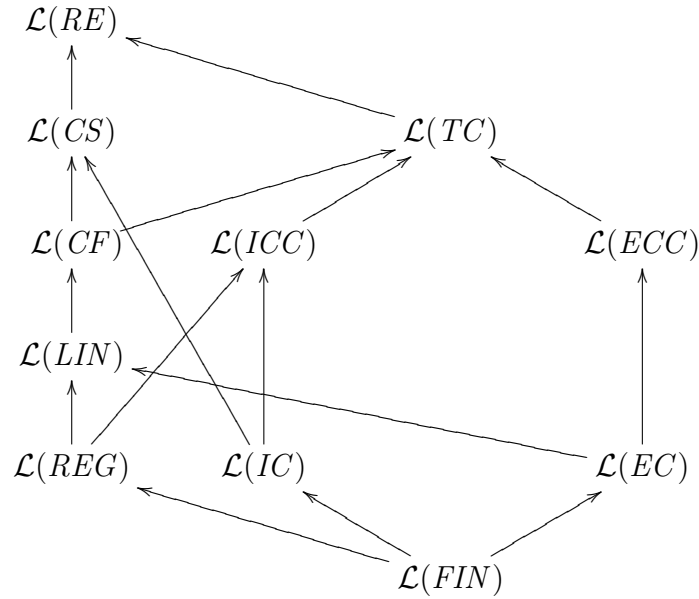


Abbildung 2.1: Hierarchie der Familien kontextueller Sprachen

$\mathcal{L}(EC) \subseteq \mathcal{L}(ECC)$ kann analog bewiesen werden.

$\mathcal{L}(ICC) \subseteq \mathcal{L}(TC)$.

Es sei $L \in \mathcal{L}(ICC)$. Dann gibt es eine kontextuale Grammatik $G = (V, B, C, \varphi)$ mit Auswahl, die $L = L_{\text{in,c}}(G)$ erfüllt. Offenbar erzeugt die totale kontextuale Grammatik $G' = (V, B, C, \varphi')$ mit $\varphi'(x_1, x_2, x_3) = \varphi(x_2)$ für alle $x_1, x_2, x_3 \in V^*$ ebenfalls die Sprache L .

$\mathcal{L}(ECC) \subseteq \mathcal{L}(TC)$.

Es sei $L \in \mathcal{L}(ECC)$. Dann gibt es eine kontextuale Grammatik $G = (V, B, C, \varphi)$ mit Auswahl und $L = L_{\text{ex,c}}(G)$. Die totale kontextuale Grammatik $G' = (V, B, C, \varphi')$ mit $\varphi'(\lambda, x_2, \lambda) = \varphi(x_2)$ für alle $x_2 \in V^*$ und $\varphi'(x_1, x_2, x_3) = \emptyset$ für $x_1 x_3 \neq \lambda$ erzeugt ebenfalls die Sprache L .

- b) $\mathcal{L}(ICC)$ ist unvergleichbar mit $\mathcal{L}(LIN)$, $\mathcal{L}(CF)$ und $\mathcal{L}(CS)$.

Hierzu reicht es, Sprachen L_1 und L_2 anzugeben, für die

$$L_1 \in \mathcal{L}(LIN) \text{ und } L_1 \notin \mathcal{L}(ICC) \quad \text{bzw.} \quad L_2 \in \mathcal{L}(ICC) \text{ und } L_2 \notin \mathcal{L}(CS)$$

gelten.

Wir betrachten die Sprache

$$L_1 = \{xx^R \mid x \in \{a, b\}^+\}.$$

Die Sprache L_1 wird von der linearen Grammatik

$$H_1 = (\{S\}, \{a, b\}, \{S \rightarrow aSa, S \rightarrow bSb, S \rightarrow aa, S \rightarrow bb\}, S)$$

erzeugt, womit $L_1 \in \mathcal{L}(LIN)$ gezeigt ist.

Wir nehmen nun an, dass $L_1 \in \mathcal{L}(ICC)$ gilt. Dann gibt es eine innere kontextuale Grammatik $H_2 = (\{a, b\}, B, C, \varphi)$ mit Auswahl, für die $L_1 = L_{\text{in,c}}(H_2)$ erfüllt ist. Es sei $(u, v) \in \varphi(x)$ für ein $x \in \{a, b\}^*$. Wir unterscheiden die Möglichkeiten für (u, v) .

Fall 1. $\#_a(v) > 0$.

Dann betrachten wir ein Wort $w = xb^k b^k x^R$ mit $k > 2|uv|$. Da $w \in L_1$ gilt, liegt w auch in $L_{\text{in},c}(H_2)$ und daher gibt es eine Ableitung $z \xrightarrow[\text{in},c]^* w$ mit $z \in B$. Wir fügen nun um x zweimal den Kontext $(u, v) \in \varphi(x)$ hinzu. Dies liefert die Ableitung

$$z \xrightarrow[\text{in},c]^* w = xb^k b^k x^R \xrightarrow[\text{in},c]{} uxb^k b^k x^R \xrightarrow[\text{in},c]{} uuxv b^k b^k x^R = w'.$$

Wegen $w' \in L_{\text{in},c}(H_2)$ ist auch $w' \in L_1$. Da

$$|w'| = 2|u| + |x| + 2|v| + 2k + |x| = 2(|uvx| + k)$$

gilt, ist $w' = yy^R$ für ein Wort y der Länge $|uvx| + k$. Damit ergibt sich

$$y = uuxv b^{k-|uv|} \quad \text{und} \quad y^R = b^{|uv|+k} x^R$$

und daraus

$$y = uuxv b^{k-|uv|} \quad \text{und} \quad y = x b^{|uv|+k}.$$

Hieraus folgt wegen der Wahl von k , dass v nur den Buchstaben b enthält, was der Annahme dieses Falles $\#_a(v) > 0$ widerspricht.

Fall 2. $\#_b(v) > 0$.

Wir erhalten analog einen Widerspruch, wenn wir statt des Teilwortes b^k das Teilwort a^k verwenden.

Fall 3. $v = \lambda$ und $\#_a(u) > 0$.

Wir betrachten diesmal ein Wort $w = x^R b^k b^k x$ mit $k > 2|u|$ und erhalten die Ableitung

$$z \xrightarrow[\text{in},c]^* w = x^R b^k b^k x \xrightarrow[\text{in},c]{} x^R b^k b^k u x \xrightarrow[\text{in},c]{} x^R b^k b^k u u x = w'$$

für ein $z \in B$ und weisen analog zu Fall 1 nach, dass u nur den Buchstaben b enthält, womit erneut ein Widerspruch vorliegt.

Fall 4. $v = \lambda$ und $\#_b(u) > 0$.

Erneut weisen wir einen Widerspruch nach – analog zum Fall 3.

Da die Fallunterscheidung alle Möglichkeiten für den Kontext (u, v) erfasst, aber jeweils ein Widerspruch hergeleitet wurde, ist die Annahme $L_1 \in \mathcal{L}(ICC)$ zu verwerfen. Folglich gilt $L_1 \in \mathcal{L}(LIN) \setminus \mathcal{L}(ICC)$.

Es sei $L \subseteq V^+$ eine beliebige rekursive, aber nicht kontextabhängige Sprache. Ferner seien c und d zwei Buchstaben, die nicht in V enthalten sind. Die innere kontextuale Grammatik

$$G_L = (V \cup \{c, d\}, \{cc\}, \{(\lambda, a) \mid a \in V\} \cup \{(d, d)\}, \varphi)$$

mit

$$\varphi(w) = \begin{cases} \{(\lambda, a) \mid a \in V\} & \text{falls } w = cc \\ \{(d, d)\} & \text{falls } w \in \{c\}L \\ \emptyset & \text{sonst} \end{cases}$$

erzeugt die Sprache

$$L_{\text{in,c}}(G_L) = \{cc\}V^* \cup \{cd^{k_n+k_{n-1}+\dots+k_1}cx_1d^{k_1}x_2d^{k_2}\dots x_{n-1}d^{k_{n-2}}x_nd^{k_n} \mid x_1x_2\dots x_i \in L, k_i \geq 1, 1 \leq i \leq n\}.$$

Wir setzen $L_2 = L_{\text{in,c}}(G_L)$.

Somit gilt nach Konstruktion $L_2 \in \mathcal{L}(ICC)$.

Angenommen, $L_2 \in \mathcal{L}(CS)$. Dann gelten wegen der Abgeschlossenheit von $\mathcal{L}(CS)$ unter Durchschnitt und unter den Operationen D_w^r und D_w^l , die für ein Wort w durch

$$D_w^r(L) = \{x \mid xw \in L\} \quad \text{und} \quad D_w^l(L) = \{x \mid wx \in L\}$$

definiert sind,

$$\begin{aligned} L_2 \cap \{cdc\}V^*\{d\} &= \{cdc\}L\{d\} \in \mathcal{L}(CS), \\ D_{cdc}^l(L_2 \cap \{cdc\}V^*\{d\}) &= L\{d\} \in \mathcal{L}(CS), \\ D_d^r(D_{cdc}^l(L_2 \cap \{cdc\}V^*\{d\})) &= L \in \mathcal{L}(CS), \end{aligned}$$

was im Widerspruch zur Wahl von L steht. Damit gilt $L_2 \notin \mathcal{L}(CS)$.

c) $\mathcal{L}(TC)$ und $\mathcal{L}(CS)$ sind unvergleichbar.

Die Sprache $\{a^{2^n} \mid n \geq 0\}$ liegt in $\mathcal{L}(CS)$, sie liegt aber wegen Satz 2.13 nicht in der Menge $\mathcal{L}(TC)$.

Die Sprache L_2 aus dem vorhergehenden Teil b) dieses Beweises liegt nicht in $\mathcal{L}(CS)$, aber in $\mathcal{L}(ICC)$ also nach Teil a) auch in $\mathcal{L}(TC)$.

d) $\mathcal{L}(IC)$ ist unvergleichbar mit $\mathcal{L}(REG)$, $\mathcal{L}(LIN)$ und $\mathcal{L}(CF)$.

Es reicht, Sprachen U_1 und U_2 anzugeben, für die

$$U_1 \in \mathcal{L}(REG) \text{ und } U_1 \notin \mathcal{L}(IC) \quad \text{bzw.} \quad U_2 \in \mathcal{L}(IC) \text{ und } U_2 \notin \mathcal{L}(CF)$$

gelten.

Wir wählen als U_1 die Sprache $\{a\}^+ \cup \{b\}^+$. Da U_1 durch einen regulären Ausdruck definiert ist, ist U_1 eine reguläre Sprache. Angenommen, U_1 liegt in $\mathcal{L}(IC)$. Dann gibt es eine innere kontextuale Grammatik $G = (\{a, b\}, B, C)$ mit $U_1 = L_{\text{in}}(G)$. Für die Erzeugung von a^k mit $k > |z|$ für alle $z \in B$ benötigen wir einen Kontext $(u, v) \in C$ mit $\#_a(uv) > 0$. Damit gibt es wegen $b^k \in L = L_{\text{in}}(G)$ für ein $z \in B$ die Ableitung $z \xrightarrow[\text{in}]{*} b^k \xrightarrow[\text{in}]{*} ubvb^{k-1}$. Folglich liegt $ubvb^{k-1}$ in $L_{\text{in}}(G)$. Wegen $\#_a(uv) > 0$ enthält $ubvb^{k-1}$ aber sowohl mindestens ein Vorkommen von a als auch Vorkommen von b . Damit liegt $ubvb^{k-1}$ nicht in U_1 im Widerspruch zu $U_1 = L_{\text{in}}(G)$. Daher ist unsere obige Annahme falsch, also gilt $U_1 \notin \mathcal{L}(IC)$.

Für U_2 verwenden wir die in Beispiel 2.4 betrachtete Sprache $L_{\text{in}}(G_3)$. Damit gilt $U_2 \in \mathcal{L}(IC)$. Da aber

$$U_2 \cap \{a^n b^m c^k \mid n \geq 1, m \geq 1, k \geq 1\} = \{a^n b^n c^n \mid n \geq 1\}$$

ist und $\mathcal{L}(CF)$ unter Durchschnitt mit regulären Sprachen abgeschlossen ist, erhalten wir auch $U_2 \notin \mathcal{L}(CF)$.

e) $\mathcal{L}(ECC)$ ist unvergleichbar mit $\mathcal{L}(REG)$, $\mathcal{L}(LIN)$, $\mathcal{L}(CF)$ und $\mathcal{L}(CS)$.

Es reicht, Sprachen W_1 und W_2 anzugeben, für die

$$W_1 \in \mathcal{L}(REG) \text{ und } W_1 \notin \mathcal{L}(ECC) \quad \text{bzw.} \quad W_2 \in \mathcal{L}(ECC) \text{ und } W_2 \notin \mathcal{L}(CS)$$

gelten.

Die Menge $W_1 = \{a\}^*\{b\}\{a\}^*\{b\}\{a\}^*$ ist regulär.

Zu einer Menge L definieren wir die Menge $ps(L)$ als die Menge aller Wörter aus L , deren echte Teilwörter nicht in L liegen. Dann gilt $ps(W_1) = \{b\}\{a\}^*\{b\}$, da einerseits die echten Teilwörter eines Wortes dieser Menge höchstens ein b enthalten und damit nicht in W_1 liegen, und andererseits jedes Wort aus der Restmenge $\{a\}^+\{b\}\{a\}^*\{b\}\{a\}^* \cup \{a\}^*\{b\}\{a\}^*\{b\}\{a\}^+$ mindestens ein echtes Teilwort enthält, das in W_1 liegt. Die Menge $ps(W_1)$ ist also unendlich.

Es sei nun $G = (V, B, C, \varphi)$ eine äußere kontextuale Grammatik mit Auswahl. Dann haben alle Ableitungen die Form

$$\begin{aligned} z &\xrightarrow{\text{ex,c}} u_1 z v_1 \xrightarrow{\text{ex,c}} u_2 u_1 z v_1 v_2 \xrightarrow{\text{ex,c}} u_3 u_2 u_1 z v_1 v_2 v_3 \xrightarrow{\text{ex,c}} \dots \\ &\xrightarrow{\text{ex,c}} u_n u_{n-1} \dots u_2 u_1 z v_1 v_2 \dots v_{n-1} v_n = w \end{aligned}$$

mit

$$z \in B \text{ und } (u_i, v_i) \in \varphi(u_{i-1} u_{i-2} \dots u_2 u_1 z v_1 v_2 \dots v_{i-2} v_{i-1}) \text{ für } 1 \leq i \leq n.$$

Daher gibt es zu jedem Wort $w \in L_{\text{ex,c}}(G)$ ein Wort $z \in B$, das Teilwort von w ist. Da außerdem $u_1 v_1 \neq \lambda$ ist und jedes Wort aus B auch in $L_{\text{ex,c}}(G)$ liegt, enthält jedes Wort $w \in L_{\text{ex,c}}(G)$ mit $w \notin B$ ein echtes Teilwort aus $L_{\text{ex,c}}(G)$. Daher gilt $ps(L_{\text{ex,c}}(G)) \subseteq B$.

Damit ist für jede Sprache $L \in \mathcal{L}(ECC)$ die Menge $ps(L)$ endlich. Folglich kann W_1 nicht in $\mathcal{L}(ECC)$ liegen.

Es seien nun $L \subseteq V^*$ eine beliebige rekursive Sprache mit $L \notin \mathcal{L}(CS)$ und c ein Symbol mit $c \notin V$. Dazu konstruieren wir die äußere kontextuale Grammatik

$$H_L = (V \cup \{c\}, \{\lambda\}, \{(\lambda, a) \mid a \in V \cup \{c\}\}, \varphi)$$

mit Auswahl, bei der φ durch

$$\varphi(w) = \begin{cases} \{(\lambda, a) \mid a \in V\} & \text{falls } w \in V^* \setminus L, \\ \{(\lambda, a) \mid a \in V \cup \{c\}\} & \text{falls } w \in L, \\ \emptyset & \text{sonst} \end{cases}$$

gegeben sei. Es ist leicht zu sehen, dass

$$L_{\text{ex,c}}(H_L) = V^* \cup L\{c\}$$

gilt. Wegen der Abgeschlossenheit von $\mathcal{L}(CS)$ unter Durchschnitt und D_c^r und der Wahl von L als nicht kontextabhängiger Sprache folgt aus

$$D_c^r(L_{\text{ex,c}}(H_L) \cap V^*\{c\}) = D_c^r(L\{c\}) = L,$$

dass $L_{\text{ex,c}}(H_L)$ keine kontextabhängige Sprache ist. Damit erfüllt $L_{\text{ex,c}}(H_L)$ als Sprache W_2 die oben geforderten Bedingungen.

f) $\mathcal{L}(EC)$ ist unvergleichbar mit $\mathcal{L}(REG)$.

Nach e) liegt die reguläre Sprache $W_1 = \{a\}^*\{b\}\{a\}^*\{b\}\{a\}^*$ nicht in $\mathcal{L}(ECC)$. Daher ist nach a) auch $W_1 \notin \mathcal{L}(EC)$ gegeben.

Andererseits liegt die Sprache $L_{\text{ex}}(G_4) = \{ww^R \mid w \in \{a, b\}^+\}$ aus Beispiel 2.5 in der Menge $\mathcal{L}(EC)$ aber nicht in $\mathcal{L}(REG)$, wie mittels Satz 1.7 a) einfach zu beweisen ist.

g) $\mathcal{L}(ECC)$ und $\mathcal{L}(EC)$ sind unvergleichbar mit $\mathcal{L}(ICC)$ und $\mathcal{L}(IC)$.

Es ist ausreichend, Sprachen V_1 und V_2 mit

$$V_1 \in \mathcal{L}(IC) \text{ und } V_1 \notin \mathcal{L}(ECC) \quad \text{bzw.} \quad V_2 \in \mathcal{L}(EC) \text{ und } V_2 \notin \mathcal{L}(ICC)$$

anzugeben.

Wir wählen $V_1 = W_1 = \{a\}^*\{b\}\{a\}^*\{b\}\{a\}^*$. Nach e) gilt $V_1 \notin \mathcal{L}(ECC)$. Andererseits gilt $V_1 = L_{\text{in}}(G_5)$ nach Beispiel 2.6, womit $V_1 \in \mathcal{L}(IC)$ nachgewiesen ist.

Wir wählen $V_2 = L_1 = \{ww^R \mid x \in \{a, b\}^+\}$. In b) haben wir $V_2 \notin \mathcal{L}(ICC)$ gezeigt. Andererseits gilt $V_1 = L_{\text{ex}}(G_4) \in \mathcal{L}(EC)$ nach Beispiel 2.5.

h) $\mathcal{L}(IC) \subset \mathcal{L}(CS)$.

Es sei $L \in \mathcal{L}(IC)$. Dann gibt es eine innere kontextuale Grammatik $G = (V, B, C)$ mit $L = L_{\text{in}}(G)$. Wir konstruieren einen linear beschränkten Automaten \mathcal{A} , der $L_{\text{in}}(G)$ und damit L akzeptiert. Wir verzichten auf eine detaillierte Angabe von \mathcal{A} und geben nur eine nicht-formale Beschreibung. Der Automat arbeitet folgenden Zyklus iteriert ab. Zuerst liest \mathcal{A} das Wort w auf dem Band und testet, ob $w \in B$ gilt. Ist dies der Fall, akzeptiert \mathcal{A} . Anderenfalls zerlegt \mathcal{A} das Wort w nichtdeterministisch in fünf Teilwörter $w = x_1ux_2vx_3$ (z. B. indem es die Teile u und v durch gestrichene Buchstaben ersetzt) und überprüft, ob (u, v) in C liegt. Wenn diese Bedingung erfüllt ist, streicht \mathcal{A} die Teilwörter u und v und durchläuft den Zyklus erneut; anderenfalls stoppt \mathcal{A} , ohne zu akzeptieren. Offenbar wird in jedem Zyklus ein Kontext gestrichen und akzeptiert, falls nach mehrfachem Streichen von Kontexten ein Wort aus B vorliegt; der Automat \mathcal{A} simuliert also in umgekehrter Reihenfolge eine Ableitung in G . Da nur Streichungen erfolgen, wird das Wort auf dem Band stets kürzer, und daher ist \mathcal{A} linear beschränkt.

Die Inklusion ist echt, da die Sprache $\{a^{2^n} \mid n \geq 0\}$ in $\mathcal{L}(CS)$ liegt, aber wegen Satz 2.13 nicht in $\mathcal{L}(IC)$ liegt.

i) $\mathcal{L}(TC) \subset \mathcal{L}(RE)$.

Die Inklusion kann analog zu $\mathcal{L}(IC) \subseteq \mathcal{L}(CS)$ bewiesen werden, wobei man nur innerhalb des Zyklus anstelle von $(u, v) \in C$ zu überprüfen hat, ob die Bedingung $(u, v) \in \varphi(x_1, x_2, x_3)$ erfüllt ist. Die Echtheit der Inklusion kann ebenfalls wie unter h) gezeigt werden.

j) $\mathcal{L}(EC) \subset \mathcal{L}(LIN)$.

Es sei die äußere kontextuale Grammatik $G = (V, B, C)$ gegeben. Die zugehörigen Ableitungen haben alle die Form

$$z \xrightarrow{\text{ex}} u_1zv_1 \xrightarrow{\text{ex}} u_2u_1zv_1v_2 \xrightarrow{\text{ex}} \cdots \xrightarrow{\text{ex}} u_nu_{n-1} \dots u_2u_1zv_1v_2 \dots v_{n-1}v_n$$

mit

$$n \geq 0, z \in B, (u_i, v_i) \in C \text{ für } 1 \leq i \leq n.$$

Daher gilt

$$L_{\text{ex}}(G) = \{ u_n u_{n-1} \dots u_1 z v_1 v_2 \dots v_n \mid n \geq 0, z \in B, (u_i, v_i) \in C \text{ für } 1 \leq i \leq n \}.$$

Wir konstruieren nun die lineare Grammatik

$$G' = (\{S\}, V, \{S \rightarrow uSv \mid (u, v) \in C\} \cup \{S \rightarrow z \mid z \in B\}, S).$$

Da die terminierenden Ableitungen bez. G' alle die Form

$$\begin{aligned} S &\Longrightarrow u_1 S v_1 \Longrightarrow u_1 u_2 S v_2 v_1 \Longrightarrow \dots \Longrightarrow u_1 u_2 \dots u_n S v_n \dots v_2 v_1 \\ &\Longrightarrow u_1 u_2 \dots u_n z v_n \dots v_2 v_1 \end{aligned}$$

mit

$$n \geq 0, z \in B, (u_i, v_i) \in C \text{ für } 1 \leq i \leq n$$

haben, erhalten wir

$$L(G') = \{ u_1 u_2 \dots u_n z v_n \dots v_2 v_1 \mid n \geq 0, z \in B, (u_i, v_i) \in C \text{ für } 1 \leq i \leq n \}.$$

Hieraus folgt unmittelbar $L_{\text{ex}}(G) = L(G')$.

(Entsprechend dem Beweis reicht es, für die Erzeugung von Sprachen aus $\mathcal{L}(EC)$ lineare Grammatiken mit genau einem Nichtterminal zu verwenden. Man kann umgekehrt leicht analog zeigen, dass jede lineare Sprache, die von einer linearen Grammatik mit genau einem Nichtterminal erzeugt wird, in $\mathcal{L}(EC)$ liegt. Dadurch erhält man eine Charakterisierung einer kontextualen Sprachfamilie durch eine Einschränkung einer Familie der Chomsky-Hierarchie.)

Die Sprache $K = \{ a^n b^m c^n \mid n \geq 1, m \geq 1 \}$ ist linear, denn sie wird von der Grammatik $G_K = (\{S, S'\}, \{a, b, c\}, \{S \rightarrow aSc, S \rightarrow S', S' \rightarrow bS', S' \rightarrow b\}, S)$ erzeugt. Angenommen, es gelte $K \in \mathcal{L}(EC)$. Dann gibt es eine kontextuale Grammatik $G'_K = (\{a, b, c\}, B, C)$ mit $K = L_{\text{ex}}(G'_K)$. Jedes Wort aus der Basis B hat die Form $a^n b^m c^n$ mit $n \geq 1$ und $m \geq 1$. Folglich hat jeder Kontext die Form (a^k, c^k) für $k \geq 1$. Damit gibt es aber nicht zu jeder beliebigen Zahl m (sondern nur für endlich viele) das Wort $ab^m c$ in der Sprache $L_{\text{ex}}(G'_K)$. Also gilt $K \notin \mathcal{L}(EC)$. Damit ist die Echtheit der Inklusion nachgewiesen.

k) $\mathcal{L}(REG) \subset \mathcal{L}(ICC)$.

Es seien $L \subseteq V^*$ eine reguläre Sprache und $\mathcal{A} = (V, Z, z_0, F, \delta)$ ein deterministischer endlicher Automat, der die Sprache L akzeptiert. Zu jedem Wort $w \in V^*$ definieren wir eine Abbildung $\varrho_w : Z \rightarrow Z$ durch $\varrho_w(z) = \delta^*(z, w)$, wobei δ^* die Erweiterung von δ auf V^* ist. Da jede derartige Abbildung die endliche Menge Z in die endliche Menge Z abbildet, gibt es nur endlich viele verschiedene Funktionen ϱ_w . Es sei n_0 die Anzahl dieser (verschiedenen) Abbildungen.

Falls für vier Wörter $x_1, x_3, y_1, y_2 \in V^*$ die Bedingungen $\varrho_{y_1} = \varrho_{y_2}$ und $x_1y_1x_3 \in L$ gelten, so haben wir auch $x_1y_2x_3 \in L$, denn es gilt

$$\begin{aligned} \delta^*(z_0, x_1y_2x_3) &= \delta^*(\delta^*(\delta^*(z_0, x_1), y_2), x_3) = \delta^*(\varrho_{y_2}(\delta^*(z_0, x_1)), x_3) \\ &= \delta^*(\varrho_{y_1}(\delta^*(z_0, x_1)), x_3) = \delta^*(\delta^*(\delta^*(z_0, x_1), y_1), x_3) \\ &= \delta^*(z_0, x_1y_1x_3) \in F. \end{aligned}$$

Wir konstruieren nun die innere kontextuale Grammatik $G = (V, B, C, \varphi)$ mit

$$\begin{aligned} B &= \{ w \mid w \in L, |w| \leq n_0 - 1 \}, \\ C &= \{ (u, v) \mid 1 \leq |uv| \leq n_0 \}, \\ \varphi(w) &= \begin{cases} \{ (u, v) \mid \varrho_w = \varrho_{uvw} \} & \text{für } |w| \leq n_0 - 1, \\ \emptyset & \text{sonst.} \end{cases} \end{aligned}$$

Nach Definition von G haben wir $B \subseteq L$. Es sei nun schon $x \in L$, und es sei $x = x_1x_2x_3 \xrightarrow{\text{in}} x_1ux_2vx_3 = y$ eine Ableitung bezüglich G . Dann gilt $\varrho_{x_2} = \varrho_{ux_2v}$ nach Konstruktion von G , und nach Obigem erhalten wir aus $x \in L$ auch $y \in L$. Daher folgt $L_{\text{in},c}(G) \subseteq L$.

Wir nehmen nun an, dass die umgekehrte Inklusion nicht gilt. Dann gibt es ein Wort $x \in L \setminus L_{\text{in},c}(G)$ mit minimaler Länge unter allen Wörtern in $L \setminus L_{\text{in},c}(G)$. Da $x \in B$ nicht möglich ist, hat x mindestens die Länge n_0 . Es sei $x = x'x''$ mit $x' = x_1x_2 \dots x_{n_0}$, $x_i \in V$ für $1 \leq i \leq n_0$. Da x' genau $n_0 + 1$ verschiedene Präfixe $\lambda, x_1, x_1x_2, \dots, x_1x_2 \dots x_{n_0}$ hat, gibt es zwei Präfixe u_1 und u_2 mit

$$u_2 = u_1u', \quad u' \neq \lambda \text{ und } \varrho_{u_1} = \varrho_{u_2}.$$

Es sei $x' = u_1u'u'' = u_2u''$ und damit $x = u_1u'u''x'' = u_2u''x''$. Dann gilt

$$\begin{aligned} \delta^*(z_0, u_1u''x'') &= \delta^*(\delta^*(z_0, u_1), u''x'') = \delta^*(\varrho_{u_1}(z_0), u''x'') \\ &= \delta^*(\varrho_{u_2}(z_0), u''x'') = \delta^*(\delta^*(z_0, u_2), u''x'') \\ &= \delta^*(z_0, u_2u''x'') = \delta^*(z_0, x) \in F \end{aligned}$$

wegen $x \in L$. Damit gilt auch $u_1u''x'' \in L$. Da x aber mit minimaler Länge in der Menge $L \setminus L_{\text{in},c}(G)$ gewählt ist, liegt $u_1u''x''$ in $L_{\text{in},c}(G)$. Wegen $\varrho_{u_1} = \varrho_{u_2} = \varrho_{u_1u'}$ und $|u'| \leq n_0$, ist $(\lambda, u') \in \varphi(u_1)$. Damit liegt $\lambda u_1u'u''x'' = x$ in $L_{\text{in},c}(G)$ im Gegensatz zur Wahl von x . Dieser Widerspruch beweist $L \subseteq L_{\text{in},c}(G)$.

Folglich haben wir $L = L_{\text{in},c}(G)$. Somit gehört jede reguläre Sprache auch zur Menge $\mathcal{L}(ICC)$. Die Echtheit der Inklusion folgt daraus, dass die Sprache K_2 in der Menge $\mathcal{L}(ICC)$ liegt (Beispiel 2.11), aber nicht regulär ist.

1) $\mathcal{L}(CF) \subset \mathcal{L}(TC)$.

Der entscheidende Schritt im vorhergehenden Beweis ist, dass mit $u_1u'u''x'' \in L$ auch $u_1u''x'' \in L$ gilt. Dies entspricht dem Pumping-Lemma für reguläre Sprachen (siehe Satz 1.7). Unter Verwendung des Pumping-Lemmas für kontextfreie Sprachen kann man daher einen analogen Beweis geben. Man hat als Kontexte die Paare (v, x) zu wählen, zu denen ein Nichtterminal A mit $A \xrightarrow{*} vAx$ und $|uv| \leq k$ existiert, und

durch φ zu testen, dass zu einem hinreichend langen Wort $uvwxy$ ein Nichtterminal A so existiert, dass $S \xRightarrow{*} uAy$ und $A \xRightarrow{*} w$ gelten. Dies kann eine totale kontextuale Grammatik leisten. Die Echtheit der Inklusion zeigt wiederum die Sprache K_2 , die in der Menge $\mathcal{L}(TC)$ liegt, aber nicht kontextfrei ist.

m) $\mathcal{L}(ICC) \subset \mathcal{L}(TC)$ und $\mathcal{L}(ECC) \subset \mathcal{L}(TC)$.

Wegen a) haben wir nur die Echtheit der Inklusionen zu zeigen. Nach b) gibt es eine kontextfreie (sogar lineare) Sprache L_1 , die nicht in $\mathcal{L}(ICC)$ liegt. Wegen l) gehört L_1 aber zu $\mathcal{L}(TC)$. Unter Verwendung von e) kann die zweite Echtheit analog bewiesen werden.

n) $\mathcal{L}(IC) \subset \mathcal{L}(ICC)$.

Wegen a) haben wir nur die Echtheit der Inklusion zu zeigen. Nach d) gibt es eine reguläre Sprache U_1 mit $U_1 \notin \mathcal{L}(IC)$. Wegen k) gilt aber $U_1 \in \mathcal{L}(ICC)$.

o) $\mathcal{L}(EC) \subset \mathcal{L}(ECC)$.

Wegen a) haben wir nur die Echtheit der Inklusion zu zeigen. Nach e) enthält $\mathcal{L}(ECC)$ eine Sprache W_2 , die nicht kontextabhängig und damit auch nicht linear ist. Folglich liegt W_2 wegen j) auch nicht in $\mathcal{L}(EC)$.

p) $\mathcal{L}(FIN) \subset \mathcal{L}(IC)$ und $\mathcal{L}(FIN) \subset \mathcal{L}(EC)$.

Die Inklusionen folgen aus Beispiel 2.2, wo eine beliebige endliche Sprache B als Element von $\mathcal{L}(IC)$ und $\mathcal{L}(EC)$ nachgewiesen wurde. Die Echtheit der Inklusionen folgt aus Beispiel 2.3, wo sowohl im äußeren als auch inneren Ableitungsmodus unendliche Sprachen erzeugt werden. \square

Abschließend diskutieren wir Entscheidbarkeitsprobleme bei kontextualen Sprachen.

Satz 2.19 *Das Mitgliedsproblem ist für äußere und innere kontextuale Grammatiken mit und ohne Auswahl und totale kontextuale Grammatiken entscheidbar.*

Beweis. Es seien $G = (V, B, C)$ bzw. $G = (V, B, C, \varphi)$ eine kontextuale Grammatik und m ein Ableitungsmodus. Wir setzen

$$L_m^0(G) = B \quad \text{und} \quad L_m^i(G) = \{ u \mid v \xRightarrow[m]{*} u \text{ für ein } v \in L_m^{i-1}(G) \} \text{ für } i \geq 1.$$

Offensichtlich sind alle Mengen $L_m^i(G)$ endlich und lassen sich der Reihe nach konstruieren. Da bei jedem Ableitungsschritt eine Verlängerung des Wortes um mindestens einen Buchstaben gesichert ist, gilt $w \in L_m(G)$ genau dann, wenn es eine Stufe $i \leq |w|$ so gibt, dass w in $L_m^i(G)$ liegt. Da letztere Bedingung algorithmisch entscheidbar ist, ist auch $w \in L_m(G)$ algorithmisch entscheidbar. \square

Wir merken aber an, dass die Komplexität des angegebenen Algorithmus für die Mitgliedsprobleme sehr hoch, insbesondere exponentiell sein kann. Das liegt zum einen daran, dass die Vereinigung der Mengen $L_m^i(G)$ mit $i \leq |w|$ alle (oder zumindest sehr viele) Wörter aus V^* der Länge höchstens $|w|$ und damit in der Länge von w exponentiell viele Wörter enthalten kann, und zum anderen erfordert die Gewinnung von u mit $v \xRightarrow[m]{*} u$ aus v die Berechnung von $\varphi(x)$ für die Teilwörter x von v , was großen Aufwand erfordern kann.

Satz 2.20 *Das Leerheitsproblem ist für äußere und innere kontextuale Grammatiken mit und ohne Auswahl und totale kontextuale Grammatiken entscheidbar.*

Beweis. Es sei $G = (V, B, C)$ bzw. $G = (V, B, C, \varphi)$ eine kontextuale Grammatik. Für jeden Ableitungsmodus m haben wir $B \subseteq L_m(G)$ und $L_m(G) = \emptyset$ für $B = \emptyset$. Damit gilt offenbar $L_m(G) = \emptyset$ genau dann, wenn $B = \emptyset$ ist. Da letztere Bedingung trivialerweise entscheidbar ist, ist auch $L_m(G) = \emptyset$ entscheidbar. \square

Ohne Beweis geben wir Resultate zur Entscheidbarkeit des Endlichkeits- und Äquivalenzproblems; Beweise sind in [4] zu finden.

Satz 2.21

- i) *Das Endlichkeitsproblem ist für äußere und innere kontextuale Grammatiken ohne Auswahl und innere kontextuale Grammatiken mit Auswahl entscheidbar; für äußere kontextuale Grammatiken mit Auswahl und totale kontextuale Grammatiken ist es unentscheidbar.*
- ii) *Das Äquivalenzproblem ist für innere und äußere kontextuale Grammatiken mit Auswahl und totale kontextuale Grammatiken unentscheidbar.* \square

Wir merken an, dass der Entscheidbarkeitsstatus des Äquivalenzproblems für innere und äußere kontextuale Grammatiken (ohne Auswahl) noch offen ist.

2.1.2. Lokale und maximal lokale kontextuale Grammatiken

Wir haben im vorhergehenden Abschnitt gesehen, dass die inneren und äußeren Grammatiken mit und ohne Auswahl die Sprache $K_1 = \{ a^n b^n c^n \mid n \geq 1 \}$ nicht erzeugen können. Gleiches gilt in ähnlicher Weise auch für die beiden anderen Sprachen K_2 und K_3 aus der Definition schwach kontextsensitiver Grammatikklassen. Die Semi-Linearität ist nur für die beiden hinsichtlich der Erzeugungskraft schwächsten Typen von kontextualen Grammatiken gegeben. Weiterhin kann die Komplexität des (entscheidbaren) Mitgliedsproblems für kontextuale Grammatiken mit Auswahl in Abhängigkeit von der Komplexität zur Berechnung der Auswahlfunktion φ zu hoch werden.

In diesem Abschnitt behandeln wir einen speziellen Typ innerer kontextualer Grammatiken, der in [8] von LUCIAN ILIE eingeführt wurde. Wir wollen zeigen, dass durch derartige Grammatiken die drei Sprachen K_1 , K_2 und K_3 erzeugt werden können, dass hinsichtlich der Länge keine beliebig großen Lücken in den erzeugten Sprachen auftreten und dass das zugehörige Mitgliedsproblem in polynomialer Zeit lösbar ist. Auch hierbei folgen wir weitgehend [8].

Wir beginnen mit einem Grammatiktyp innerer kontextualer Grammatiken mit Auswahl, bei dem die Auswahlfunktion im Wesentlichen von der Zugehörigkeit zu gewissen regulären Mengen abhängt. Es handelt sich daher eigentlich mehr um eine Vereinfachung der Schreibweise für gewisse Auswahlfunktionen, denn um einen wirklich neuen Grammatiktyp.

Definition 2.22

- i) *Eine innere kontextuale Grammatik mit regulärer Auswahl ist ein Tupel*

$$G = (V, B, (C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)),$$

wobei

- $n \geq 1$ eine natürliche Zahl ist,
 - V und B wie bei einer inneren kontextualen Grammatik definiert sind,
 - C_i für $1 \leq i \leq n$ eine endliche Teilmenge von $V^* \times V^*$ (eine Menge von Kontexten) ist und
 - R_i für $1 \leq i \leq n$ eine reguläre Menge über V ist.
- ii) Für zwei Wörter $x, y \in V^*$ schreiben wir $x \xRightarrow[r]{} y$, falls es eine Zahl i , $1 \leq i \leq n$, so gibt, dass

$$x = x_1x_2x_3, \quad y = x_1ux_2vx_3, \quad (u, v) \in C_i \text{ und } x_2 \in R_i$$

für gewisse Wörter x_1, x_2, x_3, u und v gelten.

Wir merken an, dass wir die Ableitungsrelation auch mittels einer Auswahlfunktion φ beschreiben können, bei der $(u, v) \in \varphi(x_2)$ genau dann gilt, wenn es eine Zahl i , $1 \leq i \leq n$, so gibt, dass $(u, v) \in C_i$ und $x_2 \in R_i$ gelten.

Definition 2.23 Es sei $z = z_1z_2z_3 \xRightarrow[r]{} z_1uz_2vz_3 = x$ ein Ableitungsschritt bez. einer inneren kontextualen Grammatik $G = (V, B, (C_1, R_1), (C_2, R_2), \dots, (C_n, R_n))$ mit regulärer Auswahl. Ein Ableitungsschritt $x \xRightarrow[r]{} y$ heißt lokal bezüglich $z \xRightarrow[r]{} x$, falls

$$x = x_1x_2x_3, \quad y = x_1sx_2tx_3, \quad (s, t) \in C_i, \quad x_2 \in R_i, \quad x_1 = z_1u', \quad x_2 = u''z_2v', \quad x_3 = v''z_3$$

für eine Zahl i , $1 \leq i \leq n$, und gewisse Wörter u', u'', v', v'' erfüllt sind.

Die Lokalität eines Ableitungsschrittes bedeutet, dass das Wort x_2 , um das der Kontext (s, t) geschrieben wird, ein Teilwort des Wortes uz_2v sein muss, dass durch Hinzufügen eines Kontextes um z_2 im vorhergehenden Schritt entstanden ist. Die Abbildung 2.2 veranschaulicht die Situation.

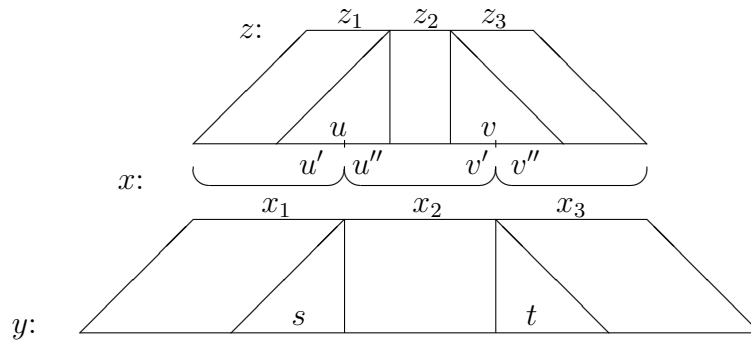


Abbildung 2.2: Situation bei einem lokalen Ableitungsschritt

Wir schreiben zukünftig $x \xRightarrow[1]{} y$, falls es sich um einen lokalen Ableitungsschritt handelt. Ferner schreiben wir $x \xRightarrow[1]{*} y$, falls es eine Ableitung

$$x = x_0 \xRightarrow[r]{} x_1 \xRightarrow[1]{} x_2 \xRightarrow[1]{} x_3 \xRightarrow[1]{} \dots \xRightarrow[1]{} x_m = y$$

gibt, bei der – mit Ausnahme des ersten Schrittes – jeder Ableitungsschritt lokal bez. des vorhergehenden Schrittes ist.

Definition 2.24 Es sei $z = z_1 z_2 z_3 \xRightarrow[r]{\quad} z_1 u z_2 v z_3 = x$ ein Ableitungsschritt bezüglich einer inneren kontextualen Grammatik $G = (V, B, (C_1, R_1), (C_2, R_2), \dots, (C_n, R_n))$ mit regulärer Auswahl. Ferner sei $x = x_1 x_2 x_3 \xRightarrow[1]{\quad} x_1 s x_2 t x_3 = y$ ein lokaler Ableitungsschritt bez. $z \xRightarrow[r]{\quad} x$ mit $x_2 \in R_i$. Wir sagen, dass $x \xRightarrow[1]{\quad} y$ maximal lokal bez. $z \xRightarrow[r]{\quad} x$ ist, falls es keine bez. $z \xRightarrow[r]{\quad} x$ lokale Ableitung $x = x'_1 x'_2 x'_3 \xRightarrow[1]{\quad} x'_1 s' x'_2 t' x_3$ mit

$$|x'_1| \leq |x_1|, |x'_2| > |x_2|, |x'_3| \leq |x_3| \text{ und } x'_2 \in R_i$$

gibt.

Intuitiv bedeutet die Maximalität, dass wir $x_2 \in R_i$ als ein Teilwort von uz_2v nur auswählen dürfen, wenn es nicht Teilwort eines anderen Wortes $x'_2 \in R_i$ ist, das ebenfalls als Teilwort von uz_2v für eine Auswahl zulässig ist.

Wir schreiben zukünftig $x \xRightarrow[ml]{\quad} y$, falls es sich um einen maximal lokalen Ableitungsschritt handelt. Ferner schreiben wir $x \xRightarrow[*]{ml} y$, falls es eine Ableitung

$$x = x_0 \xRightarrow[r]{\quad} x_1 \xRightarrow[ml]{\quad} x_2 \xRightarrow[ml]{\quad} x_3 \dots \xRightarrow[ml]{\quad} x_m = y$$

gibt, bei der – mit Ausnahme des ersten Schrittes – jeder Ableitungsschritt maximal lokal bez. des vorhergehenden Schrittes ist, wobei auch im ersten Schritt ein maximales Teilwort in dem Sinne gewählt werden soll, dass es nicht echtes Teilwort eines anderen zulässigen Wortes der gleichen Auswahl ist.

Definition 2.25

i) Wir nennen eine innere kontextuale Grammatik

$$G = (V, B, (C_1, R_1), (C_2, R_2), \dots, (C_n, R_n))$$

mit regulärer Auswahl lokal oder maximal lokal, wenn in ihrem Ableitungsprozess – mit Ausnahme des ersten Schrittes – jeder Ableitungsschritt lokal bzw. maximal lokal bez. des vorhergehenden Schrittes ist.

ii) Für eine lokale bzw. maximal lokale Grammatik G ist die erzeugte Sprache $L_1(G)$ bzw. $L_{ml}(G)$ durch

$$L_1(G) = \{ w \mid z \xRightarrow[1]{*} w \text{ für ein } z \in B \} \quad \text{bzw.}$$

$$L_{ml}(G) = \{ w \mid z \xRightarrow[ml]{*} w \text{ für ein } z \in B \}$$

definiert.

Beispiel 2.26 Wir betrachten die maximal lokale Grammatik

$$H_1 = (\{a, b, c\}, \{abc\}, (\{(ab, c)\}, \{b\}^+)).$$

Jede Ableitung in dieser Grammatik hat die Form

$$abc \xRightarrow[r]{\quad} aabbcc \xRightarrow[ml]{\quad} aaabbbccc \xRightarrow[ml]{\quad} a^4 b^4 c^4 \xRightarrow[ml]{\quad} \dots \xRightarrow[ml]{\quad} a^n b^n c^n,$$

da b^m das Teilwort von $ab^m c$ mit maximaler Länge ist, das in $\{b\}^+$ liegt. Folglich erhalten wir

$$L_{\text{ml}} = \{a^n b^n c^n \mid n \geq 1\} = K_1.$$

Wir bemerken, dass H_1 als lokale Grammatik die Sprache

$$L_1(H_1) = \{abc\} \cup \{aa^{i_1} b a^{i_2} b \dots a^{i_k} b^l c^n \mid n \geq 2, 2 \leq l \leq n, k = n - l + 1, \\ i_j \geq 1 \text{ für } 1 \leq j \leq k, \sum_{j=1}^k i_j = n - 1\}$$

erzeugt.

Beispiel 2.27 Es sei die maximal lokale Grammatik

$$H_2 = (\{a, b, c, d\}, \{abcd\}, (\{(b, d)\}, \{b^m c \mid m \geq 1\}), (\{(a, c)\}, \{b^m c^n \mid m \geq 1, n \geq 1\}))$$

gegeben. Die Ableitungen sind dann von der Form

$$abcd \xrightarrow[r]{\implies} abbcdd \xrightarrow[\text{ml}]{\implies} ab^3 cd^3 \xrightarrow[\text{ml}]{\overset{*}{\implies}} ab^m cd^m \xrightarrow[\text{ml}]{\implies} aab^m ccd^m \xrightarrow[\text{ml}]{\implies} a^3 b^m c^3 d^m \xrightarrow[\text{ml}]{\overset{*}{\implies}} a^n b^m c^n d^m,$$

woraus

$$L_{\text{ml}}(H_2) = \{a^n b^m c^n d^m \mid n \geq 1, m \geq 1\} = K_2$$

folgt.

Beispiel 2.28 Die Grammatik

$$H_3 = (\{a, b, c\}, \{aca, bcb\}, (\{(a, a), (b, b)\}, \{c\}\{a, b\}^+))$$

erzeugt maximal lokal die Sprache

$$L_{\text{ml}}(H_3) = \{w c w \mid w \in \{a, b\}^+\}$$

und lokal die Sprache

$$L_1(H_3) = \{x w c x v \mid x \in \{a, b\}, w, v \in \{a, b\}^*, \#_a(w) = \#_a(v), \#_b(w) = \#_b(v)\}.$$

Damit haben wir gezeigt, dass die drei Sprachen aus der Definition schwach kontextabhängiger Grammatikklassen im Wesentlichen durch maximal lokale Grammatiken erzeugt werden können, wobei das „im Wesentlichen“ darin begründet ist, dass wir statt $K_3 = \{w w \mid w \in \{a, b\}^+\}$ die Sprache $\{w c w \mid w \in \{a, b\}^+\}$ erzeugt haben.

Wir erhalten mit einem analogen Beweis auch für lokale und maximal lokale Grammatiken die Aussage, die in Satz 2.13 für kontextuale Grammatiken mit und ohne Auswahl formuliert ist.

Satz 2.29 Für jede Sprache L , die von einer lokalen oder maximal lokalen Grammatik G erzeugt wird, gibt es Konstanten k und k' derart, dass zu jedem Wort $w \in L$ mit $|w| > k$ ein Wort $w' \in L$ so existiert, dass $0 < |w| - |w'| < k'$ gilt. \square

Damit ist die schwache Forderung von JOSHI für die schwache Kontextabhängigkeit erfüllt; es ist aber noch ein offenes Problem, ob die Sprachen, die von lokalen und maximal lokalen Grammatiken erzeugt werden, semi-linear sind.

Satz 2.30 *Das Mitgliedsproblem für lokale und maximal lokale Grammatiken ist in polynomialer Zeit entscheidbar.*

Beweis. Im Beweis benutzen wir die folgende Aussage der Komplexitätstheorie:

Jede Sprache, die durch eine nichtdeterministische, die Eingabe nicht verändernde Mehrband-Turing-Maschine mit logarithmischem Platzbedarf auf den Arbeitsbändern akzeptiert wird, ist auch durch eine deterministische Turing-Maschine mit polynomialem Zeitaufwand akzeptierbar.

Wir verweisen für einen Beweis auf [18]. Aufgrund dieser Aussage reicht es zu beweisen, dass wir für eine gegebene lokale bzw. maximal lokale Grammatik G die erzeugte Sprache durch eine nichtdeterministische, die Eingabe nicht verändernde Mehrband-Turing-Maschine mit logarithmischem Platzbedarf akzeptieren können. Wir beweisen diese Aussage nur für lokale Grammatiken; die einfachen Modifikationen für den maximal lokalen Fall bleiben der Leserin / dem Leser überlassen.

Es seien eine lokale Grammatik $G = (V, B, (C_1, R_1), (C_2, R_2), \dots, (C_n, R_n))$ und ein Wort $w = w_1 w_2 \dots w_m \in V^*$ der Länge m mit $w_i \in V$ für $1 \leq i \leq m$ gegeben. Wir konstruieren eine Mehrband-Turing-Maschine \mathcal{M} , auf deren Eingabeband das Eingabewort w steht (das während der Bearbeitung nicht verändert wird), und die acht Arbeitsbänder hat, auf denen acht natürliche Zahlen $p_1, p_2, r_1, r_2, s_1, s_2, t_1, t_2$ (in Binärdarstellung) gespeichert werden, die Positionen im Wort w angeben und folglich nicht größer als m sind. Daher ist der Platzbedarf durch $8(\log_2(m) + 1)$ beschränkt.

Die Idee des Arbeitens der Maschine besteht darin, rückwärts eine Ableitung bez. der Grammatik G zu simulieren, indem nacheinander Kontexte gestrichen werden. Durch die Zahlen werden sich dabei die Stellen gemerkt, die den Teil des Eingabewortes w angeben, der noch nicht gestrichen wurde. Die Lokalität des Ableitungsprozesses sichert, dass die acht Zahlen $p_1, p_2, r_1, r_2, s_1, s_2, t_1, t_2$ dafür ausreichend sind.

Bevor wir mit dem Zyklus anfangen, nehmen wir eine Initialisierung vor. Wir testen zuerst, ob w in B liegt. Ist dies der Fall, so akzeptieren wir und die Maschine stoppt ihre Arbeit. Im anderen Fall wählen wir nichtdeterministisch Zahlen p_1, r_2, s_1, t_2 so, dass

$$0 \leq p_1 \leq r_2 \leq s_1 \leq t_2 \leq m$$

gilt. Dadurch erhalten wir eine Zerlegung des Wortes w in $w = x_1 u x_2 v x_3$ mit

$$\begin{aligned} x_1 &= w_1 w_2 \dots w_{p_1}, \\ u &= w_{p_1+1} w_{p_1+2} \dots w_{r_2}, \\ x_2 &= w_{r_2+1} w_{r_2+2} \dots w_{s_1}, \\ v &= w_{s_1+1} w_{s_1+2} \dots w_{t_2}, \\ x_3 &= w_{t_2+1} w_{t_2+2} \dots w_m. \end{aligned}$$

Wir wählen nun nichtdeterministisch eine Zahl h , $1 \leq h \leq n$, und testen, ob $(u, v) \in C_h$ und $x_2 \in R_h$ erfüllt sind. Ist dies nicht der Fall, so stoppt die Maschine, ohne zu akzeptieren. Im anderen Fall ist es möglich, dass im letzten Ableitungsschritt der Kontext

(u, v) um das Wort x_2 hinzugefügt worden ist. Daher erhalten wir durch „Streichen“ dieses Kontextes das Wort $w^{(1)} = x_1x_2x_3$; jedoch streichen wir nicht wirklich, da das Eingabewort nicht verändert werden darf, sondern wissen, dass unser Wort $w^{(1)}$ Konkatination der Teilwörter bestehend aus den ersten p_1 Buchstaben, aus dem $(r_2 + 1)$ -ten Buchstaben bis zum s_1 -ten Buchstaben und aus den letzten $m - t_2$ Buchstaben ist: $w^{(1)} = w_1w_2 \dots w_{p_1}w_{r_2+1}w_{r_2+2} \dots w_{s_1}w_{t_2+1}w_{t_2+2} \dots w_m$. Dann überprüfen wir, ob $w^{(1)}$ in B liegt. Ist dies der Fall, so akzeptieren wir und die Maschine stoppt. Ist dies nicht der Fall, läuft der folgende Zyklus (iteriert) ab, der jeweils eine Rückwärtssimulation eines Ableitungsschrittes vornimmt.

Wir setzen $p_2 = p_1$, $r_1 = r_2$, $s_2 = s_1$ und $t_1 = t_2$. Die Wörter $w_{p_2+1}w_{p_2+2} \dots w_{p_{r_1}}$ und $w_{s_2+1}w_{s_2+2} \dots w_{p_{t_1}}$ sind die beim Rückwärtssimulieren bereits gestrichenen Teilwörter. Das verbliebene Wort ist $w_1w_2 \dots w_{p_2}w_{r_1+1}w_{r_1+2} \dots w_{s_2}w_{t_1+1}w_{t_1+2} \dots w_m$ und sei mit $w^{(i)}$ bezeichnet. Nun werden nichtdeterministisch wieder Zahlen p_1, r_2, s_1, t_2 gewählt, wobei

$$0 \leq p_1 \leq p_2 \leq r_1 \leq r_2 \leq s_1 \leq s_2 \leq t_1 \leq t_2 \leq m$$

gelten soll. Damit wird eine Zerlegung von $w^{(i)}$ wie folgt festgelegt: $w^{(i)} = x_1^{(i)}u^{(i)}x_2^{(i)}v^{(i)}x_3^{(i)}$ mit

$$\begin{aligned} x_1^{(i)} &= w_1w_2 \dots w_{p_1}, \\ u^{(i)} &= w_{p_1+1}w_{p_1+2} \dots w_{p_2}w_{r_1+1}w_{r_1+2} \dots w_{r_2}, \\ x_2^{(i)} &= w_{r_2+1}w_{r_2+2} \dots w_{s_1}, \\ v^{(i)} &= w_{s_1+1}w_{s_1+2} \dots w_{s_2}w_{t_1+1}w_{t_1+2} \dots w_{t_2}, \\ x_3^{(i)} &= w_{t_2+1}w_{t_2+2} \dots w_m. \end{aligned}$$

Die Abbildung 2.3 veranschaulicht die Situation.

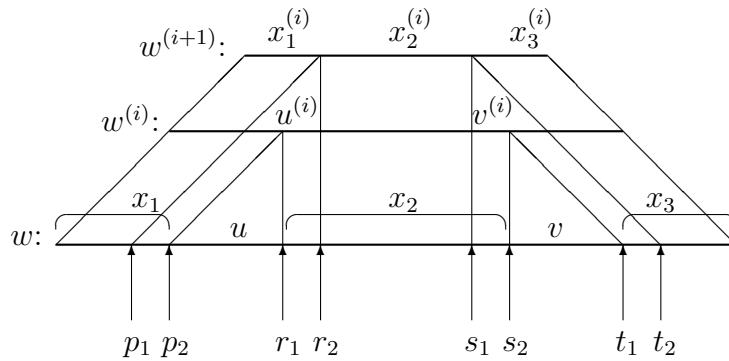


Abbildung 2.3: Von der Maschine nichtdeterministisch geratene Ableitung

Wir wählen erneut nichtdeterministisch eine Zahl h , $1 \leq h \leq n$, und testen, ob $(u^{(i)}, v^{(i)}) \in C_h$ und $x_2^{(i)} \in R_h$ gelten. Ist dies nicht der Fall, stoppt die Maschine, ohne zu akzeptieren. Im anderen Fall ist die Ableitung $x_1^{(i)}x_2^{(i)}x_3^{(i)} \xRightarrow[r]{\quad} w^{(i)}$ möglich und sogar so, dass die zuletzt simulierte Ableitung $w^{(i)} \xRightarrow[r]{\quad} w^{(i-1)}$ lokal ist. Wir untersuchen nun, ob das Wort $w^{(i+1)} = x_1^{(i)}x_2^{(i)}x_3^{(i)}$ in B liegt. Ist dies der Fall, so akzeptieren wir und die Maschine stoppt. Sollte $w^{(i+1)}$ das Leerwort sein, aber nicht in B liegen, stoppt die

Maschine, ohne zu akzeptieren, da nun kein Kontext mehr gestrichen werden kann. Im anderen Fall beginnen wir den nächsten Zyklus. \square

Wir merken an, dass unser Beweis keine Aussage über den Grad des Polynoms macht, das als Schranke für die Komplexität wirkt. In [6] haben RADU GRAMATOVICI und FLORIN MANEA einen Algorithmus für das Mitgliedsproblem angegeben, dessen Komplexität durch ein Polynom des Grades 4 in der Länge des Eingabewortes beschränkt ist.

Insgesamt haben wir damit nachgewiesen, dass maximal lokale Grammatiken im Wesentlichen eine schwach kontextabhängige Klasse bilden (die zweite, dritte und vierte Forderung aus der Definition 1.20 sind bis auf die Modifikation bei K_3 erfüllt, ob auch die erste Forderung erfüllt ist – alle kontextfreien Sprachen erzeugbar sind, ist noch offen).

2.2. Automaten mit Neustart

In diesem Abschnitt wollen wir ein Verfahren zur Überprüfung von Sätzen auf syntaktische Korrektheit formalisieren. Es handelt sich hierbei also um eine Analyse, bei der syntaktisch richtige Sätze zu akzeptieren sind. Daher verwenden wir in diesem Abschnitt Automaten anstelle von generierenden Grammatiken.

Das Verfahren besteht im Wesentlichen darin, in einem gegebenen Satz sukzessive Phrasen zu streichen, wobei die Korrektheit oder Inkorrektheit erhalten bleiben. Dieses Verfahren bietet sich insbesondere für solche Sprachen an, bei denen die Position einer Phrase innerhalb des Satzes von untergeordneter Bedeutung ist. Dazu gehören u. a. mehrere slawische Sprachen. Bei derartigen Sprachen bietet sich der generative Ansatz von NOAM CHOMSKY nicht an, da die Regeln der Grammatik vielfach modellieren, dass gewisse Phrasen benachbart oder in relativer Nähe zueinander stehen.

Bei der Analyse durch Reduktion spielt nicht die Position einer Phrase im Satz sondern ihre Beziehung zu anderen Satzteilen eine Rolle. Man unterscheidet hier drei Fälle:

- *Unabhängigkeit von Phrasen:* Zwei Phrasen sind unabhängig, wenn man jede Phrase streichen oder stehen lassen kann, ohne das sich die syntaktische Korrektheit ändert.

Als ein Beispiel erwähnen wir den Satz

Der Mann aß heute sein Frühstück langsam.

Durch Streichen einer oder beider Phrasen *sein Frühstück* und *langsam* erhalten wir die ebenfalls syntaktisch richtigen Sätze

Der Mann aß heute sein Frühstück.

Der Mann aß heute langsam.

Der Mann aß heute.

- *Vertikale Abhängigkeit von zwei Phrasen:* Eine Phrase A ist vertikal abhängig von einer Phrase B , wenn das Streichen von A sowie das Streichen von A und B jeweils zu einem korrekten Satz führt, während das alleinige Streichen der Phrase B zu einem syntaktisch falschen Satz führt.

Als Beispiel betrachten wir den Satz

Die Fahrt begann mitten im Januar.

aus dem durch Streichen der Phrasen *mitten* und *im Januar* die Sätze

Die Fahrt begann im Januar.

Die Fahrt begann mitten.

Die Fahrt begann.

entstehen, von denen der zweite nicht korrekt ist. Die Phrase *mitten* ist also vertikal abhängig von der Phrase *im Januar*.

- *Horizontale Abhängigkeit von Phrasen* liegt vor, wenn die syntaktische Korrektheit nur erhalten bleibt, wenn die Phrasen gleichzeitig gestrichen werden.

Wir erwähnen hier nur, dass bei dieser Analyse durch Reduktion auch ein Beitrag zur Vermeidung von Mehrdeutigkeiten gegeben wird. So kann das Wort *sein* sowohl ein Verb als auch ein Possessivpronomen sein. Durch die Vereinfachung der Sätze durch Fortlassen von Phrasen wird vielfach eine Struktur erreicht, bei der die Mehrdeutigkeit einfach zu klären/handhaben ist.

Wir wollen nun eine Formalisierung der Analyse durch Reduktion mittels Automaten geben.

2.2.1. Definitionen und Beispiele

Intuitiv formalisieren wir die Methode wie folgt. Wir betrachten einen Automaten, der von links nach rechts ein gegebenes Wort (das einem Satz der Sprache entspricht) liest und zu einem gewissen Zeitpunkt einen Teil des Wortes (der einer Phrase entspricht) streicht. Da der Automat ganze Phrasen streichen soll, liest er nicht nur einen Buchstaben des Wortes, sondern den gesamten Inhalt eines Fensters, das von links nach rechts über das Wort bewegt wird. Hat der Automat eine Phrase gestrichen, so kehrt er in seinen Anfangszustand und mit dem Fenster an den Anfang des neuen Satzes zurück, läuft wieder von links nach rechts und nimmt die nächste Streichung vor. Jedoch ist dieses Vorgehen nur bei Unabhängigkeit der Phrase wirklich anwendbar. Anderenfalls muss er unter Umständen noch eine Markierung vornehmen, welcher Teil ebenfalls zu streichen ist. Der in der folgenden Definition gegebene Automat mit Neustart und seine Arbeitsweise widerspiegeln diese intuitive Idee.

Die Eingabe des Automaten wollen wir auf ein Band schreiben. Zusätzlich verwenden wir zur Markierung des Wortanfangs und Wortendes zwei Markierungen $\$$ bzw. $\$$.

Um den Inhalt eines Fensters zu beschreiben, definieren wir folgende Mengen für ein gegebenes Alphabet V , wobei wir zu beachten haben, dass das Fenster weniger Buchstaben enthalten darf, als seine Größe zulässt, da das Eingabewort kürzer sein kann oder beim Verschieben nach rechts über die rechte Markierung hinaus nicht mehr alle Zellen des

Bandes gefüllt sein müssen:

$$\begin{aligned}
 V^{\leq n} &= \bigcup_{i=0}^n V^i \text{ für } n \geq 0, \\
 PC(V, 1) &= \{\clubsuit, \$\} \cup V, \\
 PC(V, n) &= \clubsuit V^{n-1} \cup V^n \cup V^{\leq n-1} \$ \cup \clubsuit V^{\leq n-2} \$ \text{ für } n \geq 2, \\
 PC(V, \leq n) &= \bigcup_{i=1}^n PC(V, i) \text{ für } n \geq 1.
 \end{aligned}$$

Die Potenzmenge einer Menge M bezeichnen wir mit $\mathcal{P}(M)$.

Definition 2.31 *Ein Automat mit Neustart (oder kurz RRN-Automat) ist ein 8-Tupel*

$$M = (Q, \Sigma, \Gamma, \clubsuit, \$, q_0, k, \delta),$$

wobei

- Q eine endliche Menge von Zuständen ist,
- Σ eine endliche Menge von Eingabesymbolen ist,
- Γ ein endliches Bandalphabet mit $\Sigma \subseteq \Gamma$ ist,
- \clubsuit und $\$$ Markierungen für den linken und rechten Rand des beschriebenen Teils des Bandes sind, die beide nicht in Γ liegen,
- $q_0 \in Q$ der Anfangszustand des Automaten ist,
- $k \geq 1$ eine natürliche Zahl ist, die die Größe des Lese-/Schreibfensters angibt, und
- $\delta : Q \times PC(\Gamma, k) \rightarrow \mathcal{P}((Q \times (\{MVR\} \cup PC(\Gamma, \leq k-1))) \cup \{RESTART, ACCEPT\})$ die Überföhrungsfunktion des Automaten ist, für die außerdem gefordert wird, dass bei $(q'_1, v_1) \in \delta(q_1, \clubsuit u_1)$ und $(q'_2, v_2) \in \delta(q_2, u_2 \$)$ mit $v_1, v_2 \in PC(\Gamma, \leq k-1)$ auch $v_1 = \clubsuit v'_1$ bzw. $v_2 = v'_2 \$$ gelten sowie bei $(q', x) \in \delta(q, \$)$ nie $x = MVR$ gilt.

Falls x das einzige Element aus $\delta(q, u)$ ist, schreiben wir $\delta(q, u) = x$ anstatt $\delta(q, u) = \{x\}$.

Bei der Überföhrungsfunktion gibt es folgende vier Möglichkeiten für ein Element aus $\delta(q, u)$ mit $q \in Q$ und $u \in PC(\Gamma, k)$:

- (q', MVR) beschreibt eine Bewegung des Fensters um eine Zelle nach rechts und eine gleichzeitige Zustandsänderung,
- (q', v) beschreibt einen Ersetzungsschritt, wobei der Fensterinhalt u durch einen kürzeren Fensterinhalt v ersetzt wird (die angegebene spezielle Forderung sichert ab, dass die Markierungen an den Wortenden erhalten bleiben), der nachfolgende Teil an v heran geschoben wird (damit keine leeren Felder auf dem Band entstehen) und das Fenster mit dem ersten Buchstaben hinter v beginnt (falls v als letzten Buchstaben $\$$ hat, beginnt das Fenster mit $\$$),

- *RESTART* beschreibt den Neustart des Automaten (er wird in den Anfangszustand und an den Wortanfang gesetzt),
- *ACCEPT* beschreibt das Ende der Arbeit des Automaten und Akzeptanz des Eingabewortes. Nichtakzeptanz wird dadurch realisiert, dass δ nur eine partielle Funktion ist, also nicht in allen denkbaren Fällen definiert ist.

Wir formalisieren diese Ideen, indem wir Konfigurationen und die Wirkung der Befehle auf eine Konfiguration definieren.

Definition 2.32 *Es sei ein Automat M mit Neustart wie in Definition 2.31 gegeben. Eine Konfiguration von M ist ein Wort der Form*

$$q\wp w\$ \quad \text{mit } q \in Q \text{ und } w \in \Gamma^*$$

oder

$$\wp w_1 q w_2 \$ \quad \text{mit } q \in Q \text{ und } w_1, w_2 \in \Gamma^*$$

oder

ACCEPT.

Die damit beschriebene Situation ist, dass sich der Automat im Zustand q befindet, $\wp w \$$ bzw. $\wp w_1 w_2 \$$ auf dem Band steht und das Fenster mit dem gelesenen Inhalt bei \wp bzw. dem ersten Buchstaben von $w_2 \neq \lambda$ beginnt und bei $w_2 = \lambda$ nur $\$$ enthält.

Definition 2.33 *Es seien M ein Automat mit Neustart wie in Definition 2.31 sowie K_1 und K_2 zwei Konfigurationen von M wie in Definition 2.32. Wir definieren den Übergang von K_1 nach K_2 (geschrieben als $K_1 \vdash K_2$) wie folgt:*

- Für $(q', MVR) \in \delta(q, u)$ gilt

$$K_1 = w_1 q a u' w_2 \$, \quad w_1 \in \wp \Gamma^* \cup \{\lambda\}, \quad u = a u', \quad a \in \Gamma \cup \{\wp\}, \quad u' \in \Gamma^{k-1}, \quad w_2 \in \Gamma^* \text{ und} \\ K_2 = w_1 a q' u' w_2 \$$$

oder

$$K_1 = w_1 q a u' \$, \quad w_1 \in \wp \Gamma^* \cup \{\lambda\}, \quad u = a u' \$, \quad a \in \Gamma \cup \{\wp\}, \quad u' \in \Gamma^{\leq k-2} \text{ und} \\ K_2 = w_1 a q' u' \$.$$

- Für $(q', v) \in \delta(q, u)$ gilt

$$K_1 = w_1 q u w_2 \$, \quad w_1 \in \wp \Gamma^* \cup \{\lambda\}, \quad w_2 \in \Gamma^* \text{ und} \\ K_2 = w_1 v q' w_2 \$$$

oder

$$K_1 = w_1 q u' \$, \quad w_1 \in \wp \Gamma^* \cup \{\lambda\}, \quad u = u' \$, \quad u' \in \wp \Gamma^{\leq k-2} \cup \Gamma^{\leq k-1} \text{ und} \\ K_2 = w_1 v' q' \$, \quad v = v' \$.$$

- Für $RESTART \in \delta(q, u)$ gilt

$$K_1 = w_1qw_2, \quad w_1 \in \mathbb{C}\Gamma^* \cup \{\lambda\}, \quad w_2 \in \Gamma^*\$ \cup \mathbb{C}\Gamma^*\$, \quad w_2 = uu' \text{ und} \\ K_2 = q_0w_1w_2.$$

- Für $ACCEPT \in \delta(q, u)$ gilt

$$K_1 = w_1qw_2, \quad w_1 \in \mathbb{C}\Gamma^* \cup \{\lambda\}, \quad w_2 \in \Gamma^*\$ \cup \mathbb{C}\Gamma^*\$, \quad w_2 = uu' \text{ und} \\ K_2 = ACCEPT.$$

Den reflexiven und transitiven Abschluss von \vdash bezeichnen wir mit \vdash^* .

Wir definieren nun die von einem Automaten mit Neustart akzeptierte Sprache.

Definition 2.34 Die von einem Automaten M mit Neustart wie in Definition 2.31 akzeptierte Sprache ist durch

$$T(M) = \{w \mid w \in \Sigma^*, \mathbb{C}w\$ \vdash^* ACCEPT\}$$

definiert.

Wir geben einige Beispiele.

Beispiel 2.35 Wir betrachten den Automaten

$$M_1 = (\{q_0, q_c, q_d, q_e\}, \{a, b, c, d\}, \{a, b, c, d\}, \mathbb{C}, \$, q_0, 3, \delta)$$

mit Neustart, bei dem die Überföhrungsfunktion δ durch

$$\begin{aligned} \delta(q_0, \mathbb{C}c\$) &= \delta(q_0, \mathbb{C}d\$) = ACCEPT, \\ \delta(q_0, \mathbb{C}ab) &= \delta(q_0, \mathbb{C}aa) = \delta(q_0, aaa) = \delta(q_0, aab) = (q_0, MVR), \\ \delta(q_0, abb) &= \{(q_d, \lambda), (q_c, b)\}, \\ \delta(q_c, bbb) &= \delta(q_c, bbc) = \delta(q_c, bc\$) = (q_c, MVR), \\ \delta(q_d, bbb) &= \delta(q_d, bbd) = \delta(q_d, bd\$) = (q_d, MVR), \\ \delta(q_c, c\$) &= \delta(q_d, d\$) = \delta(q_e, \$) = RESTART, \\ \delta(q_0, abc) &= (q_e, c) \end{aligned}$$

gegeben ist. Zuerst geben wir die möglichen Berechnungen von M_1 auf der Eingabe $aabbc$ an (wobei wir stets den Inhalt des Fensters unterstreichen):

$$\begin{aligned} q_0\mathbb{C}a\underline{abbc}\$ \vdash \mathbb{C}q_0\underline{a}abbc\$ \vdash \mathbb{C}a\underline{q_0}abbc\$ \vdash \mathbb{C}ab\underline{q_c}c\$ \\ \vdash q_0\mathbb{C}a\underline{bc}\$ \vdash \mathbb{C}q_0\underline{abc}\$ \vdash \mathbb{C}c\underline{q_e}\$ \vdash q_0\mathbb{C}c\underline{\$} \vdash ACCEPT \end{aligned}$$

und M akzeptiert $aabbc$ oder

$$q_0\mathbb{C}a\underline{abbc}\$ \vdash \mathbb{C}q_0\underline{a}abbc\$ \vdash \mathbb{C}a\underline{q_0}abbc\$ \vdash \mathbb{C}a\underline{q_d}c\$$$

und die Berechnung stoppt, da keine Nachfolgekongfiguration erklärt ist, ohne zu akzeptieren.

Der Automat arbeitet wie folgt: Zuerst läuft er über das Wort nach rechts, bis sich abb im Fenster befindet. Er entscheidet sich nun nichtdeterministisch, ob er ab oder abb streicht, läuft weiter nach rechts und führt einen Neustart nur dann aus, wenn er am Wortende den zum Streichen passenden Buchstaben c bzw. d findet. Dies wird solange durchgeführt, bis ein so kurzes Wort erreicht wird, dass von ihm sofort oder in zwei Schritten entschieden werden kann, ob das Wort zu akzeptieren ist. Damit ergibt sich

$$T(M_1) = L_1 = \{a^n b^n c \mid n \geq 0\} \cup \{a^n b^{2n} d \mid n \geq 0\}.$$

Wir werden im Folgenden nur solche Automaten betrachten, bei denen zwischen Beginn der Arbeit und dem ersten Neustart sowie in jedem Zyklus zwischen zwei Neustarts jeweils genau ein Ersetzungsschritt und nach dem letzten Neustart höchstens ein Ersetzungsschritt erfolgt.

Beispiel 2.36 Wir wollen einen Automaten M_2 mit Neustart konstruieren, der die Menge

$$K'_3 = \{w\#w \mid w \in \{a, b\}^*\}$$

akzeptiert (K'_3 unterscheidet sich von der in den vorhergehenden Abschnitten behandelten Sprache K_3 nur dadurch, dass die Mitte des Wortes durch die Markierung $\#$ genau gekennzeichnet ist). Die naheliegende Idee ist, bei einem Wort $w_1\#w_2$ auf dem Band das Teilwort w_1 vor dem $\#$ mit dem Teilwort w_2 nach $\#$ buchstabenweise zu vergleichen. Wir wollen diesen Vergleich mit den jeweils letzten beiden Buchstaben der beiden Teilwörter vornehmen. Daher müssen wir zuerst immer nach rechts laufen, bis sich ein Wort der Form $cd\#$ (mit $c, d \in \{a, b\}$) im Fenster befindet. Wir dürfen cd nicht sofort löschen, da wir dann nach einem Neustart ein Wort $\#z\#w_2\#$ mit $w_1 = zcd$ auf dem Band erhalten. Kann nun die Berechnung erfolgreich fortgesetzt werden, d. h. $\#z\#w_2\# \vdash^* ACCEPT$, so gilt $z = w_2$. Dies liefert aber auch die erfolgreiche Rechnung $\#w_1\#w_2\# \vdash \#z\#w_2\# \vdash^* ACCEPT$, die das nicht in K'_3 liegende Wort $w_1\#w_2 = zcd\#z$ akzeptiert. Daher ersetzen wir $cd\#$ durch das kürzere Wort $[c, d]\#$, wobei $[c, d]$ ein Buchstabe ist, der zu Γ gehören muss, und merken uns im Zustand z_{cd} das Buchstabenpaar cd . Danach laufen wir weiter nach rechts und kontrollieren, ob am Ende des Wortes auf dem Band $cd\#$ steht. Ist dies nicht der Fall, liegt ein nicht zu akzeptierendes Wort vor, d. h. wir beenden die Arbeit von M , indem wir keine Nachfolgekonfiguration zulassen, also $\delta(q_{cd}, xy\#)$ für $xy \neq cd$ undefiniert lassen. Ist der Vergleich erfolgreich, führen wir einen Neustart durch. Wir laufen jetzt erneut nach rechts, bis wir $[c, d]\#$ finden (wir können erneut $[c, d]$ nicht sofort löschen). Wir merken uns nun cd im Zustand z'_{cd} und ersetzen am Ende $cd\#$ durch $[c, d]\#$. Danach laufen wir erneut über das Wort und ersetzen $[c, d]\#$ durch $\#$. Dies ist jetzt möglich, da am Wortende auf dem Band $[c, d]\#$ steht (was wir überprüfen). Beim nächsten Lauf über das Wort ersetzen wir $[c, d]\#$ durch $\#$. Damit haben wir das Wort der Länge 2 am Ende von w_1 und w_2 gestrichen und können diesen Prozess erneut starten. Hat das Wort w_1 ungerade bzw. gerade Länge, so erhalten wir auf dem Band $\#x\#x\#$ oder $\#x\#$ und in diesen beiden Situationen akzeptieren wir. Formal erhalten wir daher den Automaten M_2

$$M_2 = (Q, \{a, b, \#\}, \{a, b, \#\} \cup \{[c, d] \mid c, d \in \{a, b\}\}, \#, \$, q_0, 3, \delta)$$

mit

$$Q = \{q_0, q_1\} \cup \bigcup_{c \in \{a, b\}} (\{q_c\} \cup \bigcup_{d \in \{a, b\}} \{q_{cd}, q'_{cd}, q''_{cd}\})$$

und δ definiert durch

$$\begin{aligned}
\delta(q_0, \clubsuit xy) &= \delta(q_0, xyz) = \delta(q_0, x\#y) = \delta(q_0, \#xy) = (q_0, MVR) \text{ f\"ur } x, y, z \in \{a, b\}, \\
\delta(q_0, cd\#) &= \{(q_{cd}, [c, d]\#), (q_0, MVR)\} \text{ f\"ur } c, d \in \{a, b\}, \\
\delta(q_0, [c, d]\$) &= (q_1, \$) \text{ f\"ur } c, d \in \{a, b\}, \\
\delta(q_0, x[c, d]\#) &= \{(q'_{cd}, MVR), (q''_{cd}, \#)\} \text{ f\"ur } c, d \in \{a, b\}, \\
\delta(q_{cd}, xyz) &= (q_{cd}, MVR) \text{ f\"ur } x, y, z, c, d \in \{a, b\}, \\
\delta(q_{cd}, cd\$) &= RESTART \text{ f\"ur } c, d \in \{a, b\}, \\
\delta(q'_{cd}, [c, d]\#x) &= \delta(q'_{cd}, \#xy) = \delta(q'_{cd}, xyz) = (q'_{cd}, MVR) \text{ f\"ur } x, y, z, c, d \in \{a, b\}, \\
\delta(q'_{cd}, cd\$) &= (q_1, [c, d]\$) \text{ f\"ur } c, d \in \{a, b\}, \\
\delta(q''_{cd}, xyz) &= (q''_{cd}, MVR) \text{ f\"ur } x, y, z, c, d \in \{a, b\}, \\
\delta(q''_{cd}, x[c, d]\$) &= RESTART \text{ f\"ur } x, c, d \in \{a, b\}, \\
\delta(q_1, \$) &= RESTART, \\
\delta(q_0, \clubsuit c\#) &= (q_c, \clubsuit) \text{ und } \delta(q_c, c\$) = ACCEPT \text{ f\"ur } c \in \{a, b\}, \\
\delta(q_0, \clubsuit\#\$) &= ACCEPT.
\end{aligned}$$

Wir erw\u00e4hnen ohne Angabe von zugeh\u00f6rigen Automaten, dass die Sprachen

$$K_1 = \{a^n b^n c^n \mid n \geq 0\}, \quad K_2 = \{a^n b^m c^n d^m \mid n, m \geq 1\} \text{ und } K_3 = \{ww \mid w \in \{a, b\}^*\}$$

von Automaten mit Neustart akzeptiert werden k\u00f6nnen. In den ersten beiden F\u00e4llen kann eine Konstruktion analog zu Beispiel 2.36 gegeben werden, da auch bei diesen Sprachen gewissen Teile miteinander „verglichen“ werden m\u00fcssen (ab mit c bzw. a mit c und dann b mit d). F\u00fcr K_3 wird nichtdeterministisch durch Ersetzen von pqr an einer Stelle durch $[p, q]\#[r, s]'$ die Mitte geraten, und dann analog zu obigem Beispiel vorgegangen. Damit sind die drei bei der Definition schwach kontextabh\u00e4ngiger Sprachfamilien genannten Sprachen in der Familie der von Automaten mit Neustart akzeptierten Sprachen enthalten. Diese Familie ist aber nicht schwach kontextabh\u00e4ngig, da auch Sprachen darin liegen, die nicht semi-linear sind, wie das folgende Beispiel zeigt.

Beispiel 2.37 Wir geben einen Automaten mit Neustart an, der die Sprache

$$L_2 = \{a^{2^n} \mid n \geq 0\}$$

akzeptiert. Die Idee dabei ist ganz einfach. Wir ersetzen der Reihe nach, am Wortende beginnend, jedes Teilwort aa durch b . Hierdurch halbieren wir die Wortl\u00e4nge. Wenn dies geschehen ist, ersetzen wir wieder von hinten beginnend jedes bb durch a , womit erneut eine Halbierung verbunden ist. Wir akzeptieren nur, wenn alle Halbierungen erfolgreich waren, d. h., wenn am Ende nur a oder b neben den Endmarkierungen auf dem Band steht. Formal ergibt sich der Automat

$$M_3 = (\{q_0, q_1\}, \{a\}, \{a, b\}, \clubsuit, \$, q_0, 3, \delta)$$

mit

$$\begin{aligned}
\delta(q_0, \clubsuit aa) &= \delta(q_0, \clubsuit bb) = \delta(q_0, \clubsuit aaa) = \delta(q_0, \clubsuit bbb) = (q_0, MVR), \\
\delta(q_0, \clubsuit aa\$) &= (q_1, \clubsuit b\$), \\
\delta(q_0, \clubsuit aab) &= (q_1, \clubsuit bb), \\
\delta(q_0, \clubsuit bb\$) &= (q_1, \clubsuit a\$), \\
\delta(q_0, \clubsuit bba) &= (q_1, \clubsuit aa), \\
\delta(q_1, xyz) &= \text{RESTART für } xyz \in \{a, b\}^3 \cup \{a, b\}^{\leq 2}\$, \\
\delta(q_0, \clubsuit a\$) &= \delta(q_0, \clubsuit b\$) = \text{ACCEPT}.
\end{aligned}$$

Es sei M ein Automat mit Neustart wie in Definition 2.31. Offensichtlich verläuft die Arbeit des Automaten in Zyklen, die aus mehreren Bewegungen nach rechts, einem Ersetzungsschritt, erneut Bewegungen nach rechts und einem *RESTART*-Schritt bestehen.

Es ist leicht zu erkennen, dass ein Ersetzungsschritt $\delta(q, u) = (q', v)$ nur möglich ist, wenn die Zustandsänderung in den vorhergehenden Verschiebungen nach rechts in den Zustand q führt. Da die Verschiebungen im Wesentlichen wie bei einem endlichen Automaten erfolgen (zuerst sieht man im Fenster die ersten Buchstaben des Wortes, danach immer genau einen zusätzlichen Buchstaben), gehört das Wort, das vor u auf dem Band steht zu einer regulären Menge R . Analog gehört das Wort hinter u zu einer regulären Menge R' , da bis zum *RESTART*-Befehl wieder nur gelesen wird (und der nicht gelesene Teil beliebig sein kann). Daher lässt sich ein Zyklus verkürzt dadurch beschreiben, dass wir angeben, welche Ersetzung $u \rightarrow v$ vorgenommen wird und zu welchen regulären Sprachen die vor bzw. nach u stehenden Wörter gehören müssen.

Definition 2.38 *Es sei M ein Automat mit Neustart wie in Definition 2.31.*

- i) *Eine Metaregel von M ist ein Tripel $(R, u \rightarrow v, R')$, wobei u und v Wörter mit $|u| > |v|$ sowie R und R' reguläre Mengen sind, oder ein Paar (R, ACCEPT) mit einer regulären Menge R .*
- ii) *Die Anwendung einer Metaregel $p = (R, u \rightarrow v, R')$ ist nur auf einen Bandinhalt $\clubsuit u_1 u u_2 \$$ möglich, bei dem $\clubsuit u_1 \in R$ und $u_2 \$ \in R'$ gelten, und liefert $\clubsuit u_1 v u_2 \$$; wir schreiben dann $\clubsuit u_1 u u_2 \$ \models \clubsuit u_1 v u_2 \$$. Eine Metaregel der Form (R, ACCEPT) ist nur anwendbar, wenn sich der Bandinhalt $\clubsuit w \$$ als $w_1 w_2$ darstellen lässt und w_1 zu R gehört. Die Anwendung einer solchen Metaregel liefert die Akzeptanz; wir schreiben dann $\clubsuit w \$ \models \text{ACCEPT}$.*

Eine Berechnung ist dann offensichtlich eine Abfolge von Anwendungen von Metaregeln.

Unter Verwendung von Metaregeln können wir M_1 einfach durch

$$\begin{aligned}
(\clubsuit a^*, ab \rightarrow \lambda, b^* c\$) &\quad (\text{wir haben die Ersetzung } abb \rightarrow b \text{ verkürzt}), \\
(\clubsuit a^*, abb \rightarrow \lambda, b^* d\$), \\
(\clubsuit c \$, \text{ACCEPT}), \\
(\clubsuit d \$, \text{ACCEPT})
\end{aligned}$$

und M_2 durch

- $(\mathfrak{c}\{a, b\}^*, cd\# \rightarrow [c, d]\#, \{a, b\}^*cd\$)$ für $c, d \in \{a, b\}$,
- $(\mathfrak{c}\{a, b\}^*[c, d]\#\{a, b\}^*, cd \rightarrow [c, d], \$)$ für $c, d \in \{a, b\}$,
- $(\mathfrak{c}\{a, b\}^*, [c, d]\# \rightarrow \#, \{a, b\}^*[c, d]\$)$ für $c, d \in \{a, b\}$,
- $(\mathfrak{c}\{a, b\}^*\#\{a, b\}^*, [c, d] \rightarrow \lambda, \$)$ für $c, d \in \{a, b\}$,
- $(\mathfrak{c}\#c\$, ACCEPT)$ für $c \in \{a, b\}$,
- $(\mathfrak{c}\#\$, ACCEPT)$

beschreiben.

In der Folge werden wir die Automaten mit Neustart stets nur durch Metaregeln angeben.

Wir definieren nun einige spezielle Klassen von Automaten mit Neustart und die zugehörigen Sprachklassen.

Definition 2.39

- i) Ein RR-Automat ist ein RRN-Automat $M = (Q, \Sigma, \Gamma, \mathfrak{c}, \$, q_0, k, \delta)$ mit $\Sigma = \Gamma$.
- ii) Ein RN-Automat ist ein RRN-Automat $M = (Q, \Sigma, \Gamma, \mathfrak{c}, \$, q_0, k, \delta)$, bei dem auf jeden Ersetzungsschritt (es wird $(q', v) \in \delta(q, u)$ angewendet) ein RESTART-Schritt erfolgt (also für alle Fensterinhalte u' die Beziehung $\delta(q', u') = RESTART$ gilt).
- iii) Ein R-Automat ist ein RN-Automat $M = (Q, \Sigma, \Gamma, \mathfrak{c}, \$, q_0, k, \delta)$ mit $\Sigma = \Gamma$.

Bei einem RN- und R-Automaten fallen in gewisser Weise der Ersetzungsschritt (engl. Rewrite) und der Neustart (engl. Restart) zusammen, weshalb wir nur einmal R anstelle von zweimal R verwenden. Der Buchstabe N steht für die Verwendung von Nichtterminalen, da das Bandalphabet Γ Buchstaben enthalten darf, die nicht im Eingabealphabet Σ vorkommen.

Der Automat M_1 ist ein RR-Automat, M_2 ist ein RRN-Automat, und M_3 ist ein RN-Automat.

Bei RN- und R-Automaten ist bei einer Metaregel $(R, u \rightarrow v, R')$ die reguläre Menge R' nicht von Bedeutung, da in jedem Fall auf den Ersetzungsschritt ein Neustart erfolgt. Daher werden wir R' bei den Metaregeln von RN- und R-Automaten nicht angeben.

Mit $\mathcal{L}(RRN)$, $\mathcal{L}(RN)$, $\mathcal{L}(RR)$ und $\mathcal{L}(R)$ bezeichnen wir die Mengen aller Sprachen, die von RRN-, RN-, RR- bzw. R-Automaten akzeptiert werden.

Ein Automat $M = (Q, \Sigma, \Gamma, \mathfrak{c}, \$, q_0, k, \delta)$ mit Neustart heißt *deterministisch*, wenn jede Menge $\delta(q, u)$ einelementig ist. Der Automat M_3 ist deterministisch.

Mit $\mathcal{L}(DX)$, $X \in \{RRN, RN, RR, R\}$, bezeichnen wir die Menge aller Sprachen, die von deterministischen X -Automaten erzeugt werden.

Damit haben wir die folgenden Relationen:

$$L_1 \in \mathcal{L}(RR), K'_3 \in \mathcal{L}(RRN) \text{ und } L_2 \in \mathcal{L}(DRN). \quad (2.1)$$

Wir wollen nun untersuchen, ob tatsächlich für diese Sprachen die Trennung von Ersetzung und Neustart bzw. der Gebrauch von Nichtterminalen erforderlich ist.

Wir zeigen zuerst, dass

$$L_1 \in \mathcal{L}(RN) \quad \text{und} \quad L_1 \notin \mathcal{L}(R) \quad (2.2)$$

gelten, d. h. wir benötigen Nichtterminale oder die Trennung von Ersetzung und Neustart, um L_1 zu akzeptieren.

Die erste Aussage folgt einfach daraus, dass $L_1 = T(M_4)$ für den RN-Automaten

$$M_4 = (\{q_0\}, \{a, b, c, d\}, \{a, b, c, d, C, D\}, \mathbb{C}, \$, q_0, 4, \delta)$$

mit den Metaregeln

$$\begin{array}{ll} (\mathbb{C}a^*, ab \rightarrow C), & (\mathbb{C}a^*, aCb \rightarrow C), \\ (\mathbb{C}a^*, abb \rightarrow D), & (\mathbb{C}a^*, aDbb \rightarrow D), \\ (\mathbb{C}Cc\$, ACCEPT), & (\mathbb{C}Dd\$, ACCEPT) \end{array}$$

gilt (beim ersten Lauf über das Wort entscheidet sich der Automat nichtdeterministisch, ob er ab oder abb streichen will; im ersten Fall führt er C ein, im zweiten wird D erzeugt; in den nachfolgenden Zyklen wird jeweils wieder ein zu C bzw. D passendes ab bzw. abb gestrichen; eine Akzeptanz erfolgt nur, wenn am Ende Cc bzw. Dd auf dem Band steht, d. h. alle Streichungen korrekt waren).

Zum Nachweis der zweiten Aussage von (2.2) nehmen wir an, dass es einen R-Automaten $M = (Q, \{a, b, c, d\}, \{a, b, c, d\}, \mathbb{C}, \$, q_0, k, \delta)$ mit $T(M) = L_1$ gibt. Wir betrachten für eine hinreichend große Zahl n das Wort $a^n b^{2n} d \in L_1$. Dann gibt es eine Berechnung

$$\mathbb{C}a^n b^{2n} d\$ \models \mathbb{C}w\$ \models^* ACCEPT.$$

Wir diskutieren die Möglichkeiten der Ersetzung im ersten Schritt. Angenommen, die angewendete Metaregel ist von der Form $(\mathbb{C}a^*, a^r \rightarrow a^s)$ mit $r > s$ (man beachte, dass bei einem R-Automaten nach der Ersetzung ein Neustart erfolgt, das Ende des Wortes folglich keinen Einfluss hat). Dann erhalten wir $w = a^{n-r+s} b^{2n} d$. Wegen $\mathbb{C}w\$ \models^* ACCEPT$ wird $a^{n-r+s} b^{2n} d$ von M akzeptiert. Andererseits liegt $a^{n-r+s} b^{2n} d$ nicht in L_1 , da $n-r+s < n$ ist. Dieser Widerspruch zeigt, dass eine andere Metaregel verwendet worden sein muss. Durch analoge Schlüsse weist man nach, dass nur eine Metaregel der Form $(\mathbb{C}a^*, a^r b^{2r} \rightarrow \lambda)$ anwendbar ist (diese Metaregel ist im Wesentlichen äquivalent zu einer Metaregel mit $a^{r+s} b^{2r+2s} \rightarrow a^s b^{2s}$, weshalb es reicht, $a^r b^{2r} \rightarrow \lambda$ zu betrachten). Dann gibt es aber auch die Berechnung

$$\mathbb{C}a^{n+r} b^{n+2r} c\$ \models \mathbb{C}a^n b^n c\$ \models^* ACCEPT,$$

da $a^n b^n c \in L_1$ liegt. Damit haben wir erneut einen Widerspruch mit $a^{n+r} b^{n+2r} c \in T(M)$ und $a^{n+r} b^{n+2r} c \notin L_1$. Folglich existiert kein R-Automat, der die Sprache L_1 akzeptiert.

Wir zeigen nun $K'_3 \in \mathcal{L}(RN)$, d. h. die Trennung von Ersetzung und Neustart wie in Beispiel 2.36 ist nicht erforderlich. Wir betrachten dafür den RN-Automaten

$$M'_2 = (Q, \{a, b, \#\}, \{a, b, \#, \#'\} \cup \{[c, d, i] \mid c, d \in \{a, b\}, i \in \{0, 1\}\}, \mathbb{C}, \$, q_0, 4, \delta)$$

mit den Metaregeln

$$\begin{aligned} & (\clubsuit, cd \rightarrow [c, d, 0]) \text{ f\"ur } c, d \in \{a, b\}, \\ & (\clubsuit(\Gamma \setminus \Sigma)^*[e, f, S], cd \rightarrow [c, d, \bar{S}]) \text{ f\"ur } c, d, e, f \in \{a, b\}, \bar{S} = 1 - S \end{aligned}$$

(durch diese Regeln k\u00f6nnen von links nach rechts Teilw\u00f6rter $cd \in \{a, b\}^2$ der L\u00e4nge 2 durch Symbole $[c, d, S]$ ersetzt werden, wobei mit 0 in der dritten Komponente begonnen wird und dann 0 und 1 alternieren),

$$\begin{aligned} & (\clubsuit[c, d, 0]\{a, b\}^*\#, cd \rightarrow [c, d, 0]) \text{ f\"ur } c, d \in \{a, b\}, \\ & (\clubsuit(\Gamma \setminus \Sigma)^*[c, d, S]\{a, b\}^*\#(\Gamma \setminus \Sigma)^*[e, f, \bar{S}], cd \rightarrow [c, d, S]) \text{ f\"ur} \\ & \hspace{15em} c, d, e, f \in \{a, b\}, \bar{S} = 1 - S \end{aligned}$$

(zu einem Symbol $[c, d, S]$ im ersten Teil darf auch im zweiten Teilwort eine Ersetzung vorgenommen werden, die $[c, d, S]$ liefert; auch diese Ersetzungen erfolgen von links nach rechts; nach einem Ersetzungsschritt im zweiten Wort muss ein Ersetzungsschritt im ersten Teil erfolgen, da die dritten Komponenten der im ersten und zweiten Teil stehenden letzten Symbole \u00fcbereinstimmen; durch Anwenden der bisherigen Regeln entsteht daher ein Wort der Form $\clubsuit w_1 w'_1 \# w_2 w'_2 \$$ mit

- $w_1 = [c_1, d_1, 0][c_2, d_2, 1][c_3, d_3, 0][c_4, d_4, 1] \dots [c_k, d_k, i]$ und $w'_1 \in \{a, b\}^*$,
 - $w_2 = [c_1, d_1, 0][e_1, f_1, 1][e_2, f_2, 0][e_3, f_3, 1] \dots [e_j, f_j, i']$ und $w'_2 \in \{a, b\}^*$,
 - zu jedem $[e_p, f_p, r]$ gibt es ein q mit $[c_q, d_q, r] = [e_p, f_p, r]$,
- daher gilt $c_1 d_1 c_2 d_2 \dots c_k d_k = c_1 d_1 u_1 e_1 f_1 u_2 e_2 f_2 \dots u_j e_j f_j u_{j+1}$ f\u00fcr gewisse $u_t \in \{a, b\}^*$, d. h. $e_1 f_1 e_2 f_2 \dots e_j f_j$ ist ein verstreutes Teilwort von $c_2 d_2 c_3 d_3 \dots c_k d_k$,

$$\begin{aligned} & (\clubsuit(\Gamma \setminus \Sigma)^*, [c, d, S]X[e, f, S'] \rightarrow \#') \text{ f\"ur } c, d, e, f \in \{a, b\}, S, S' \in \{0, 1\}, X \in \{\#, \#'\}, \\ & (\clubsuit(\Gamma \setminus \Sigma)^*, [c, d, S]gX[e, f, S'] \rightarrow g\#') \text{ f\"ur} \\ & \hspace{15em} c, d, e, f, g \in \{a, b\}, S, S' \in \{0, 1\}, X \in \{\#, \#'\} \end{aligned}$$

(wenn alle m\u00f6glichen Teilw\u00f6rter cd zu $[c, d, i]$ umgewandelt sind (es kann bei ungerader L\u00e4nge des ersten Teils des Eingabewortes ein Buchstabe $g \in \{a, b\}$ stehen bleiben), streichen wir jeweils eines der Symbole $[c, d, S]$ im ersten Teil und eines der Symbole $[e, f, S']$ im zweiten Teil und ersetzen (falls noch nicht geschehen die Markierung $\#$ durch $\#'$, wodurch abgesichert, dass nach dem Anfang dieses Streichens im zweiten Teil keine Ersetzungen $xy \rightarrow [x, y, S'']$ erfolgen k\u00f6nnen),

$$(\clubsuit\#'\$, ACCEPT), (\clubsuit g\#'\$, ACCEPT), (\clubsuit\#\$, ACCEPT), (\clubsuit g\#\$, ACCEPT)$$

f\u00fcr $g \in \{a, b\}$ (eine Akzeptanz erfolgt nur, wenn durch die Streichungen ein Wort ohne $[c, d, S]$ entstanden ist, womit abgesichert ist, dass die Teile vor $\#$ und hinter $\#$ gleiche L\u00e4nge haben, in obiger Notation also $w_1 w'_1 = w_2 w'_2$ gilt, bzw. die Eingabe $\#$ bzw. $g\#$ ist).

Aufgrund der Erkl\u00e4rungen nach den Regeln ist nun leicht zu sehen, dass $T(M'_2) = K'_3$ gilt.

Wir geben jetzt einige einfache Eigenschaften von Automaten mit Neustart an.

Satz 2.40 *Es seien $M = (Q, \Sigma, \Gamma, \phi, \$, q_0, k, \delta)$ ein RRN-Automat sowie u und v zwei Wörter über Σ . Falls $\phi u\$ \models \phi v\$$ und $u \notin T(M)$ gelten, so haben wir auch $v \notin T(M)$.*

Beweis. Angenommen, es gilt $v \in T(M)$. Dann gibt es eine Berechnung $\phi v\$ \models^* ACCEPT$ von M . Damit gibt es dann aber auch die Berechnung $\phi u\$ \models \phi v\$ \models^* ACCEPT$, woraus $u \in T(M)$ im Widerspruch zur Voraussetzung folgt. \square

Satz 2.41 *Es seien $M = (Q, \Sigma, \Gamma, \phi, \$, q_0, k, \delta)$ ein deterministischer RRN-Automat sowie u und v zwei Wörter über Σ . Falls $\phi u\$ \models \phi v\$$ und $u \in T(M)$ gelten, so haben wir auch $v \in T(M)$.*

Beweis. Da $u \in T(M)$ gilt, gibt es eine Berechnung $\phi u\$ \models \phi w\$ \models^* ACCEPT$. Nach Voraussetzung haben wir auch $\phi u\$ \models \phi v\$$. Wegen der Determiniertheit des Automaten gilt $\phi w\$ = \phi v\$$. Damit gilt $\phi v\$ \models^* ACCEPT$. Somit haben wir $v \in T(M)$. \square

Bei deterministischen Automaten mit Neustart, bei denen $\phi u\$ \models \phi v\$$ für gewisse Wörter u und v über dem Eingabealphabet gilt, sind damit $u \in T(M)$ und $v \in T(M)$ gleichwertig.

Satz 2.42 *Zu jedem RRN-Automaten $M = (Q, \Sigma, \Gamma, \phi, \$, q_0, k, \delta)$ gibt es eine Konstante p derart, dass für Wörter u, v, v' und w mit $\phi uvw\$ \models \phi uv'w\$$ und $|u| > p$ (oder $|w| > p$) eine Zerlegung $u = u_1u_2u_3$ mit $u_2 \neq \lambda$ (bzw. eine Zerlegung $w = w_1w_2w_3$ mit $w_2 \neq \lambda$) derart existiert, dass $\phi u_1u_2^i u_3vv\$ \models \phi u_1u_2^i u_3v'w\$$ (bzw. $\phi uvw_1w_2^i w_3\$ \models \phi uv'w_1w_2^i w_3\$$) für alle $i \geq 0$ gilt.*

Beweis. Die Berechnung $\phi uvw\$ \models \phi uv'w\$$ basiert auf der Anwendung einer Metaregel $(R, v \rightarrow v', R')$. Für die reguläre Menge R gibt es ein Pumping-Lemma, d. h. es gibt eine Konstante p_R derart, dass für jedes Wort $\phi u \in R$ mit $|u| > p_R$ eine Zerlegung $u = u_1u_2u_3$ mit $u_2 \neq \lambda$ so existiert, dass $\phi u_1u_2^i u_3 \in R$ für alle $i \geq 0$ gilt. Damit ist die Metaregel auch auf $\phi u_1u_2^i u_3vv\$$ anwendbar, woraus $\phi u_1u_2^i u_3vv\$ \models \phi u_1u_2^i u_3v'w\$$ resultiert. Analog kann man die Aussage für den w -Teil beweisen. Um die Konstante p unabhängig von den Metaregeln zu erhalten, wählt man einfach das Maximum der p_R über alle regulären Mengen in den Metaregeln. \square

Satz 2.43 $\mathcal{L}(RRN) \subseteq \mathbf{NP}$ und $\mathcal{L}(DRRN) \subseteq \mathbf{P}$.

Beweis. Es sei ein RRN-Automat $M = (Q, \Sigma, \Gamma, \phi, \$, q_0, k, \delta)$ gegeben. Wir konstruieren nun eine nichtdeterministische Turing-Maschine M' , die wie folgt arbeitet. Auf dem Band stehe ein Wort $x = x'ux''$ und der Lese-/Schreibkopf befinde sich über dem ersten Buchstaben von x im Anfangszustand. Die Maschine M' bewegt ihren Lese-/Schreibkopf solange nach rechts, wie M das Fenster nach rechts bewegt. Wenn M eine Ersetzung $u \rightarrow v$ ausführt, so ersetzt auch M' das Wort u durch das kürzere Wort v und schiebt anschließend den hinter v stehenden Teil x'' an das Ende von v heran, wodurch leere Zellen zwischen Buchstaben auf dem Band vermieden werden. Danach verhält sich M' wieder wie M , d. h. M' bewegt sich solange nach rechts, bis ein Neustart erfolgt, den M' dadurch simuliert, dass an den Wortanfang zurückgekehrt wird und in den Anfangszustand gewechselt

wird. Für solch einen Zyklus benötigt M' höchstens $2|x| + (2(|u| - |v|) + 1)|x''|$ Schritte, wobei der erste Summand aus den Bewegungen nach rechts, der Ersetzung und der Rückkehr an den Wortanfang resultiert und der zweite Summand für das Heranschieben von x'' an v erforderlich ist.

Es sei die Eingabe w gegeben. Dann kann M höchstens $|w|$ Zyklen durchlaufen, da das Wort auf dem Band in jedem Zyklus um mindestens 1 verkürzt wird. Ferner haben wir noch $|x''| < |x| \leq |w|$ und $|u| - |v| < |u| \leq k$. Daher werden in jedem Zyklus höchstens $(2k + 3)|w|$ Schritte ausgeführt, woraus ein Gesamtaufwand von höchstens $(2k + 3)|w|^2$ resultiert. Damit erfolgt die Akzeptanz von w nach einer in $|w|$ polynomialen Anzahl von Schritten.

Falls M deterministisch ist, so kann auch M' als deterministische Turing-Maschine konstruiert werden. \square

2.2.2. Hierarchieresultate

In diesem Abschnitt untersuchen wir Beziehungen der von Automaten mit Neustart akzeptierten Sprachklassen zu den Sprachklassen der Chomsky-Hierarchie und vergleichen sie untereinander.

Wir beginnen mit Beziehungen zu den kontextabhängigen und kontextfreien Sprachen.

Satz 2.44 $\mathcal{L}(RRN) \subseteq \mathcal{L}(CS)$.

Beweis. Der Beweis erfolgt durch Konstruktion einer Turing-Maschine wie im Beweis von Satz 2.43, denn die konstruierte Turing-Maschine ist linear beschränkt, da sie nie über die Zellen hinausläuft, in denen die Markierungen ¢ und § zu Beginn standen. \square

Satz 2.45 $\mathcal{L}(CF) \subseteq \mathcal{L}(RN)$.

Beweis. Es sei L eine kontextfreie Grammatik. Dann gibt es eine kontextfreie Grammatik $G = (N, T, P, S)$ in Chomsky-Normalform mit $L(G) = L$.

Wir konstruieren den RN-Automaten $M = \{Q, T, N \cup T, \text{¢}, \text{§}, q_0, 2, \delta\}$ mit den Metaregeln

$$\begin{aligned} &(\text{¢}(N \cup T)^*, bc \rightarrow A) \text{ für } A \rightarrow BC, B \rightarrow b, C \rightarrow c \in P, \\ &(\text{¢}(N \cup T)^*, bC \rightarrow A) \text{ für } A \rightarrow BC, B \rightarrow b \in P, \\ &(\text{¢}(N \cup T)^*, Bc \rightarrow A) \text{ für } A \rightarrow BC, C \rightarrow c \in P, \\ &(\text{¢}(N \cup T)^*, BC \rightarrow A) \text{ für } A \rightarrow BC \in P, \\ &(\text{¢}S\text{§}, ACCEPT). \end{aligned}$$

Offensichtlich gelten

$$\begin{aligned} \text{¢}x\text{b}c\text{x}'\text{§} \models \text{¢}x\text{A}x'\text{§} &\text{ genau dann, wenn} \\ &x\text{A}x' \implies x\text{B}C\text{x}' \implies x\text{b}C\text{x}' \implies x\text{b}c\text{x}' \text{ eine Ableitung in } G \text{ ist,} \\ \text{¢}x\text{B}c\text{x}'\text{§} \models \text{¢}x\text{A}x'\text{§} &\text{ genau dann, wenn} \\ &x\text{A}x' \implies x\text{B}C\text{x}' \implies x\text{B}c\text{x}' \text{ eine Ableitung in } G \text{ ist,} \end{aligned}$$

$\wp xbCx'\$ \models \wp xAx'\$$ genau dann, wenn
 $xAx' \implies xBCx' \implies xbCcx'$ eine Ableitung in G ist,
 $\wp xBCx'\$ \models \wp xAx'\$$ genau dann, wenn
 $xAx' \implies xBCx'$ eine Ableitung in G ist.

Daraus folgt, dass es eine Berechnung $\wp w\$ \models^* \wp S\$ \models ACCEPT$ genau dann gibt, wenn in G eine Ableitung $S \xRightarrow{*} w$ existiert. Somit gelten $L = L(G) = T(M)$ und $L \in \mathcal{L}(RN)$. \square

Als nächstes geben wir eine Beziehung zwischen XN -Sprachen und X -Sprachen an, die besagt, dass der zusätzliche Gebrauch von Nichtterminalen die Akzeptanz von Durchschnitten von X -Sprachen mit regulären Sprachen gestattet.

Satz 2.46 *Eine Sprache L wird genau dann von einem (deterministischen) RRN-Automaten M akzeptiert, wenn es einen (deterministischen) RR-Automaten M' und eine reguläre Sprache R derart gibt, dass $L = T(M') \cap R$ gilt.*

Beweis. i) Es sei $M = (Q, \Sigma, \Gamma, \wp, \$, q_0, k, \delta)$ ein RRN-Automat. Dann gilt $w \in T(M)$ genau dann, wenn $w \in \Sigma^*$ und $\wp w\$ \models_M^* ACCEPT$ erfüllt sind.

Wir betrachten nun den RR-Automaten $M' = (Q, \Gamma, \Gamma, \wp, \$, q_0, k, \delta)$. Für diesen Automaten gilt $w \in T(M')$ genau dann, wenn $w \in \Gamma^*$ und $\wp w\$ \models_{M'}^* ACCEPT$ erfüllt sind. Damit folgt $w \in T(M)$ genau dann, wenn $w \in T(M')$ und $w \in \Sigma^*$ gelten. Daher gilt $T(M) = T(M') \cap \Sigma^*$.

ii) Es seien nun ein RR-Automat $M' = (Q, \Sigma, \Sigma, \wp, \$, q_0, k, \delta)$ und eine reguläre Menge R gegeben. Dann existiert ein deterministischer endlicher Automat $\mathcal{A} = (Z, \Sigma, z_0, F, \delta_{\mathcal{A}})$ mit $T(\mathcal{A}) = R$. Wir haben einen RRN-Automaten $M = (Q', \Sigma, \Gamma, \wp, \$, q'_0, k', \delta')$ zu konstruieren, für den $T(M) = T(M') \cap R$ gilt. Die Idee der Konstruktion besteht in Folgendem: Als Menge Q' wählen wir $Q \times Z \times \{0, 1, 2\}$ und als Anfangszustand $(q_0, z_0, 0)$. Wir lassen M auf der ersten Komponente wie M' arbeiten, und lesen während der Bewegungen nach rechts die Buchstaben der Eingabe und verarbeiten diese wie \mathcal{A} . Wenn M' einen Neustart ausführt, wechselt M in der dritten Komponente zu 1 und liest die restlichen Buchstaben der Eingabe durch Bewegungen nach rechts. Ist die gesamte Eingabe gelesen, so sieht M in der zweiten Komponente nach, ob ein akzeptierender Zustand aus F erreicht wurde. Ist dies der Fall wird ein Neustart ausgeführt, anderenfalls wird nicht akzeptiert. Natürlich darf M bei allen folgenden Zyklen nicht mehr testen, ob das Wort auf dem Band zu R gehört, da dies nur für die Eingabe selbst gelten muss. Daher verwenden wir ein Fenster der Länge $k' = k + 2$, womit bei der gleichen Ersetzung immer ein Buchstabe im Fenster erhalten bleibt. Diesen markieren wir, indem wir $(x, 1)$ statt x dafür ersetzen, d. h. statt $u \rightarrow v$ wird $xu \rightarrow (x, 1)v$ oder $ux \rightarrow v(x, 1)$ benutzt. (Es wird $k' = k + 2$ genommen, damit die Sonderfälle mit dem Erhalt von \wp und $\$$ auch berücksichtigt werden.) Durch diesen markierten Buchstaben erkennt M , dass es nicht mehr der erste Zyklus ist, ändert die dritte Komponente zu 2, und führt den Neustart an der gleichen Stelle wie M' aus. Aus diesen Erklärungen ist leicht zu sehen, dass M schließlich nur akzeptiert, wenn auch M' akzeptiert und außerdem im ersten Zyklus $w \in R$ festgestellt wurde. Somit gilt $w \in T(M)$ genau dann, wenn $w \in T(M')$ und $w \in R$ erfüllt sind, woraus $T(M) = T(M') \cap R$ resultiert. Die formale Definition von δ' und der anderen Komponenten von M' überlassen wir dem Leser. Wir merken aber an, dass bei der Konstruktion die zusätzlichen Symbole $(x, 1)$ benutzt werden, so dass ein RRN-Automat entsteht.

Bei beiden Konstruktionen bleibt Determinismus erhalten, so dass die Aussage auch für determinische Automaten mit Neustart gilt. \square

Satz 2.47 *Eine Sprache L wird genau dann von einem (deterministischen) RN-Automaten M akzeptiert, wenn es einen (deterministischen) R-Automaten M' und eine reguläre Sprache R derart gibt, dass $L = T(M') \cap R$ gilt.*

Beweis. i) Es kann wie beim Beweis von Satz 2.46 gezeigt werden, dass $T(M) = T(M') \cap \Sigma^*$ gilt.

ii) Der Beweis gestaltet sich schwieriger als der von Satz 2.46, da der RN-Automat beim ersten Zyklus nach der Ersetzung nicht mehr weiter lesen kann. Dies wird dadurch überwunden, dass man sich den letzten bei einem Zyklus gelesenen Buchstaben und den erreichten Zustand von \mathcal{A} durch eine Markierung merkt, wobei die Größe des Fensters noch einmal um Eins erhöht wird. Dadurch gelingt es dem Automaten, schließlich das ganze Wort zu lesen und zu entscheiden, ob die Eingabe in R liegt. \square

Als nächstes geben wir einige Sprachen an, die wir im Folgenden zum Trennen der Sprachklassen, die von verschiedenen Typen von Automaten mit Neustart erzeugt werden, verwenden.

Wir setzen

$$\begin{aligned} L_3 &= \{a^n b^n \mid n \geq 0\} \cup \{a^n b^m \mid m > 2n \geq 0\}, \\ L_{4,1} &= \{(ab)^{2^n - i} c (ab)^i \mid n \geq 0, 0 \leq i \leq 2^n\}, \\ L_{4,2} &= \{(ab)^{2^{n+1} - 2i} (abb)^i \mid n \geq 0, 0 \leq i \leq 2^n\}, \\ L_{4,3} &= \{(abb)^{2^n - i} (ab)^i \mid n \geq 0, 0 \leq i \leq 2^n\}, \\ L_4 &= L_{4,1} \cup L_{4,2} \cup L_{4,3}. \end{aligned}$$

Lemma 2.48 $L_3 \in \mathcal{L}(CF)$, $L_3 \in \mathcal{L}(DRN)$ und $L_3 \notin \mathcal{L}(RR)$.

Beweis. i) Wir zeigen zuerst $L_3 \in \mathcal{L}(CF)$. Offensichtlich erzeugt die kontextfreie Grammatik

$$G = (\{S, A, B, C\}, \{a, b\}, P, S)$$

mit der Regelmenge

$$P = \{S \rightarrow \lambda, S \rightarrow A, A \rightarrow aAb, A \rightarrow ab, S \rightarrow B, B \rightarrow aBbb, B \rightarrow C, C \rightarrow bC, C \rightarrow b\}$$

die Sprache L_3 .

ii) Wir konstruieren nun einen deterministischen RN-Automaten M , der L_3 akzeptiert. Dabei geben wir nicht die vollständige Beschreibung von M sondern nur die Grundideen; die formale komplette Definition von M bleibt dem Leser überlassen. Die Idee besteht darin, zuerst aus einem Wort $\$a^n b^m\$$ ein Wort $\$x_1 x_2 \dots x_r y'_s y'_{s-1} \dots y'_1\$$ zu erzeugen, wobei die x_i , $1 \leq i \leq r$, und y_j , $1 \leq j \leq s$ aus $\{0, 1\}$ sind (man beachte, wir verwenden im zweiten Teil gestrichelte Varianten von 0 und 1) sowie $x_r x_{r-1} \dots x_1$ und $y'_s y'_{s-1} \dots y'_1$ die

Dualdarstellungen von n und m sind. Dies kann z. B. durch folgende Metaregeln erreicht werden:

$$\begin{array}{ll}
(\mathfrak{c}\{0, 1\}^*, xxxb \rightarrow 11b), & (\mathfrak{c}\{0, 1\}^*, yyy\epsilon \rightarrow 1'1'\epsilon), \\
(\mathfrak{c}\{0, 1\}^*, xxxxb \rightarrow 001b), & (\mathfrak{c}\{0, 1\}^*, yyyyy\epsilon \rightarrow 1'0'0'\epsilon), \\
(\mathfrak{c}\{0, 1\}^*x^+, xxxxb \rightarrow xx\bar{x}b), & (\mathfrak{c}\{0, 1\}^*, yyyyyy \rightarrow \bar{y}yyy), \\
(\mathfrak{c}\{0, 1\}^*, xxx\bar{x}b \rightarrow 101b), & (\mathfrak{c}\{0, 1\}^*, \bar{y}yyy\epsilon \rightarrow 1'0'1'\epsilon), \\
(\mathfrak{c}\{0, 1\}^*, xxx\bar{x}\bar{x} \rightarrow 1\bar{x}\bar{x}\bar{x}), & (\mathfrak{c}\{0, 1\}^*\bar{y}^+, yyyyy\epsilon \rightarrow \bar{y}\bar{y}0'\epsilon), \\
(\mathfrak{c}\{0, 1\}^*, xxx\bar{x}\bar{x} \rightarrow 0\bar{x}\bar{x}\bar{x}), & (\mathfrak{c}\{0, 1\}^*\bar{y}^+, yyyyyy \rightarrow \bar{y}yyy), \\
(\mathfrak{c}\{0, 1\}^*x^+, xxx\bar{x}\bar{x} \rightarrow xx\bar{x}\bar{x}), & (\mathfrak{c}\{0, 1\}^*\bar{y}^+, \bar{y}yyy\epsilon \rightarrow \bar{y}\bar{y}1'\epsilon),
\end{array}$$

sowohl für $x = a$ und $\bar{x} = a'$ als auch für $x = a'$ und $\bar{x} = a$ bzw. sowohl für $y = b$ und $\bar{y} = b'$ als auch für $y = b'$ und $\bar{y} = b$ sowie $\epsilon \in \{\$, 0', 1'\}$ (es erfolgt jeweils eine Division der as oder $a's$ bzw. bs oder $b's$ durch 2 und am Wortanfang bzw. Wortende wird die entsprechende Binärzahl eingefügt). Nachdem die Binärzahlen erzeugt wurden, wird ein Vergleich der beiden Binärzahlen durchgeführt, wobei folgende Metaregeln benutzt werden:

$$\begin{array}{ll}
(\mathfrak{c}\{0, 1\}^*, 00' \rightarrow \lambda), & (\mathfrak{c}\{0, 1\}^*, 11' \rightarrow \lambda), \\
(\mathfrak{c}\{0, 1\}^*, 01' \rightarrow \#), & (\mathfrak{c}\{0, 1\}^*, 10' \rightarrow \#'), \\
(\mathfrak{c}\{0, 1\}^*, x\#y' \rightarrow \#), & (\mathfrak{c}\{0, 1\}^*, x\#'y' \rightarrow \#') \text{ für } x, y \in \{0, 1\}
\end{array}$$

(wenn ein Unterschied auftritt, so geben $\#$ und $\#'$ an, dass die Dualziffer vom a -Teil kleiner bzw. größer als die vom b -Teil ist). Die Akzeptanz erfolgt nun in folgenden Fällen:

$$\begin{array}{l}
(\mathfrak{c}\$, ACCEPT) \text{ (es gilt } n = m), \\
(\mathfrak{c}\#x', ACCEPT), (\mathfrak{c}x', ACCEPT), (\mathfrak{c}\#x'y', ACCEPT) \text{ für } x, y \in \{0, 1\} \text{ (} m > 2n).
\end{array}$$

Mittels der genannten Metaregeln werden nur die Wörter „richtig“ behandelt, bei denen die Anzahl der as mindestens Drei ist. Um auch Wörter mit weniger as akzeptieren zu können, nehmen wir noch die Metaregel $(R, ACCEPT)$ mit

$$R = \{ \mathfrak{c}ab\$, \mathfrak{c}aabb\$ \} \cup \{ \mathfrak{c}a^n b^m \$ \mid n \in \{0, 1, 2\}, m > 2n \}$$

hinzu.

iii) Wir nehmen jetzt an, dass es einen RR-Automaten $M' = (Q, \Sigma, \Sigma, \mathfrak{c}, \$, q_0, k, \delta)$ mit $T(M') = L_3$ gibt. Zu einer Metaregel $m = (R, u \rightarrow v, R')$ des RR-Automaten M' sei $\mathcal{A}_m = (Z_m, \{a, b\}, z_{0,m}, F_m, \delta_m)$ ein deterministischer endlicher Automat, der R' akzeptiert. Wir definieren p als das Maximum der Zahlen $\#(Z_m)$ über alle Metaregeln. Es sei nun n ein Vielfaches von $p! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot p$, und es gelte $n > 3k + 1$.

Wir betrachten das Wort $a^n b^n \in L_3$. Dann gibt es eine Berechnung

$$\mathfrak{c}a^n b^n \$ \models \mathfrak{c}w \$ \models^* ACCEPT.$$

Damit gilt auch $w \in L_3$. Wir zeigen zuerst, dass $w = a^{n-r} b^{n-r}$ mit einer gewissen Zahl $r \geq 1$ gilt. Dies folgt daraus, dass die maximale Änderung zugunsten der b dadurch entsteht, dass $k - 1$ Vorkommen von a durch Vorkommen von b ersetzt werden, wodurch $\mathfrak{c}a^{n-k+1} b^{n+k-1} \$$ entsteht. Wegen

$$2(n - k + 1) = n + n - 2k + 2 > n + (3k + 1) - 2k + 2 = n + k + 3 > n + k - 1$$

kann also nicht erreicht werden, dass die Anzahl der bs größer als das Doppelte der Anzahl der as ist. Damit muss in w_1 die Anzahl der as gleich der Anzahl der bs sein. Da mindestens ein Buchstabe beim Übergang zu w gestrichen wird, gilt $w = a^{n-r}b^{n-r}$ mit einer gewissen Zahl $r \geq 1$.

Wir wählen nun n mit obigen Eigenschaften so groß, dass Satz 2.42 anwendbar ist. Dann gibt es also Wörter w_1, w_2, w_3 mit $w_2 \neq \lambda$ derart, dass $a^n b^n = a^n b^r w_1 w_2 w_3$ und $\mathfrak{C}a^n b^r w_1 w_2^i w_3 \$ \models \mathfrak{C}a^{n-r} w_1 w_2^i w_3 \$$ für alle $i \geq 0$ gelten. Dies lässt sich dann mit $b^r w_1 = b^s$, $w_2 = b^q$ und $w_3 = b^{n-s-q}$ als

$$\mathfrak{C}a^n b^{n+(i-1)q} \$ \models \mathfrak{C}a^{n-r} b^{n-r+(i-1)q} \$ \quad (2.3)$$

schreiben. Entsprechend dem Beweis von Satz 2.42 ist $q \leq p$. Damit ist q ein Teiler von n .

Wir wählen $i = \frac{n}{q} + 1$. Dann erhalten wir

$$n - r + (i - 1)q = n - r + \left(\frac{n}{q} + 1 - 1\right)q = 2n - r > 2(n - r).$$

Folglich ist $a^{n-r} b^{n-r+(i-1)q} \in L_3$ und wir erhalten aus (2.3) sofort.

$$\mathfrak{C}a^n b^{n+(i-1)q} \$ \models \mathfrak{C}a^{n-r} b^{n-r+(i-1)q} \$ \models^* ACCEPT.$$

Damit ist auch $a^n b^{n+(i-1)q} \in T(M')$. Dies widerspricht aber $T(M') = L_3$, da $a^n b^{n+(i-1)q}$ wegen

$$n + (i - 1)q = n + \left(\frac{n}{q} + 1 - 1\right)q = 2n$$

nicht in L_3 liegt. □

Lemma 2.49 $L_4 \in \mathcal{L}(DRR)$ und $L_4 \notin \mathcal{L}(R)$.

Beweis. i) Wir betrachten einen deterministischen RR-Automaten M mit den Metaregeln

$$(\mathfrak{C}\{ab\}^*, c \rightarrow \lambda, \{ab\}^* \$), \quad (2.4)$$

$$(\mathfrak{C}\{ab\}^*, abab \rightarrow abb, \{abb\}^* \$), \quad (2.5)$$

$$(\mathfrak{C}\{abb\}^*, abb \rightarrow ab, \{ab\}^* \$), \quad (2.6)$$

$$(\mathfrak{C}ab \$, ACCEPT).$$

Eine typische Ableitung ist

$$\begin{aligned} \mathfrak{C}(ab)^{2^n-i} c (ab)^i \$ &\models \mathfrak{C}(ab)^{2^n} \$ = \mathfrak{C}(ab)^{2^n-2} abab \$ \\ &\models \mathfrak{C}(ab)^{2^n-2} abb \$ \models (ab)^{2^n-4} (abb)^2 \\ &\models^* \mathfrak{C}(abb)^{2^{n-1}} \$ = \mathfrak{C}(abb)^{2^{n-1}-1} abb \$ \\ &\models \mathfrak{C}(abb)^{2^{n-1}-1} ab \$ \models \mathfrak{C}(abb)^{2^{n-1}-2} (ab)^2 \$ \\ &\models^* \mathfrak{C}(ab)^{2^{n-1}} \$ \\ &\models^* \mathfrak{C}(ab)^{2^{n-2}} \$ \models^* \mathfrak{C}ab \$ \models ACCEPT. \end{aligned}$$

Es ist einfach zu sehen, dass jede Ableitung im Wesentlichen so aussieht.

- Auf ein Wort w mit einem c ist aufgrund der regulären Mengen der anderen Regeln nur (2.4) anwendbar; außerdem hat w dann die Form $\mathfrak{c}(ab)^p c(ab)^q \mathfrak{\$}$ und es entsteht $\mathfrak{c}(ab)^{p+q} \mathfrak{\$}$.
- Es kann ansonsten nur die Regel (2.5) auf Wörter der Form $\mathfrak{c}(ab)^p (abb)^q \mathfrak{\$}$ angewendet werden, wodurch letzten Endes $\mathfrak{c}(abb)^{p/2+q} \mathfrak{\$}$ entsteht (wenn p gerade ist) oder eine Konfiguration $\mathfrak{c}ab(abb)^r \mathfrak{\$}$ erzeugt wird, die keine Fortsetzung erlaubt, oder es kann die Regel (2.6) nur auf Wörter der Form $\mathfrak{c}(abb)^p (ab)^q \mathfrak{\$}$ angewendet werden, wodurch $\mathfrak{c}(ab)^{p+q} \mathfrak{\$}$ entsteht.
- Ein Wort $(ab)^p$ wird nur akzeptiert, wenn p eine Potenz von 2 ist, da durch die Regel (2.5) stets eine Division durch 2 bewirkt wird und für Akzeptanz $\mathfrak{c}ab \mathfrak{\$}$ benötigt wird.
- Damit liegen in der Sprache von M nur die Wörter
 - $(ab)^p c(ab)^q$ mit $p + q = 2^n$ (also $p = 2^n - q$),
 - $(ab)^p (abb)^q$ mit $p/2 + q = 2^n$ (also $p = 2^{n+1} - 2q$),
 - $(abb)^p (ab)^q$ mit $p + q = 2^n$ (also $p = 2^n - q$).

Somit akzeptiert M gerade L_4 . Damit ist $L_4 \in \mathcal{L}(DRR)$ gezeigt.

ii) Angenommen, es gibt einen R-Automaten $M' = (Q, \Sigma, \Sigma, \mathfrak{c}, \mathfrak{\$}, q_0, k, \delta)$ mit der akzeptierten Sprache $T(M') = L_4$. Dann gibt es wegen

$$(ab)^{2^n} c(ab)^{2^n} = (ab)^{2^{n+1}-2^n} c(ab)^{2^n} \in L_4$$

eine Berechnung

$$\mathfrak{c}(ab)^{2^n} c(ab)^{2^n} \mathfrak{\$} \models \mathfrak{c}w \mathfrak{\$} \models^* ACCEPT.$$

Folglich liegt w auch in L_4 . Wenn ein Teilwort von $(ab)^{2^n} c(ab)^{2^n}$ durch ein Wort ersetzt wird, in dem abb vorkommt, so sind in w sowohl abb als auch c enthalten oder es steht sowohl vor als auch hinter abb noch ab ; beides widerspricht $w \in L_4$. Somit kann nur eine Ersetzung von $(ab)^r c(ab)^s$ durch c oder λ vorgenommen werden. Im ersten Fall entsteht $\mathfrak{c}(ab)^{2^n-r} c(ab)^{2^n-s} \mathfrak{\$}$, was $2^n - r + 2^n - s = 2^{n+1} - (r + s) = 2^t$ für ein gewisses t impliziert und für hinreichend großes n nur für $r = s = 0$ möglich ist, womit die angewendete Metaregel $c \rightarrow c$ enthalten würde, was nicht zulässig ist. Im zweiten Fall erhalten wir analog $r = s = 0$ (wir streichen also nur c) und erhalten $\mathfrak{c}(ab)^{2^n} c(ab)^{2^n} \mathfrak{\$} \models \mathfrak{c}(ab)^{2^{n+1}} \mathfrak{\$}$. Wegen $(ab)^{2^{n+1}} c \in L_4$ (man wähle $i = 0$) erhalten wir damit die Berechnung

$$\mathfrak{c}(ab)^{2^n} c(ab)^{2^n} c \mathfrak{\$} \models \mathfrak{c}(ab)^{2^{n+1}} c \mathfrak{\$} \models^* ACCEPT$$

und damit $(ab)^{2^n} c(ab)^{2^n} c \in T(M')$ im Widerspruch zu $T(M') = L_4$, da $(ab)^{2^n} c(ab)^{2^n} c \notin L_4$ gilt. \square

Lemma 2.50 *Es gibt eine Sprache in $\mathcal{L}(DR)$, die nicht kontextfrei ist.*

Beweis. Aus Beispiel 2.37 wissen wir, dass die Sprache $L_2 = \{ a^{2^n} \mid n \geq 0 \}$ in der Familie $\mathcal{L}(DRN)$ liegt. Nach Satz 2.47 gibt es dann eine Sprache $K \in \mathcal{L}(DR)$ und eine reguläre Sprache R mit $K \cap R = L_2$. Wenn K kontextfrei wäre, so wäre nach den Abschlusseigenschaften von $\mathcal{L}(CF)$ auch L_2 kontextfrei, was aber unmöglich ist, da L_2 nicht semi-linear ist. \square

Die folgenden beiden Lemmata geben wir ohne Beweis. Für die (nicht einfachen) Beweise, die weitere Sprachklassen und deren Eigenschaften benutzen, verweisen wir auf [15]

Lemma 2.51 *Es gibt eine Sprache in $\mathcal{L}(R)$, die nicht in $\mathcal{L}(DRRN)$ liegt.* \square

Lemma 2.52 $\mathcal{L}(DRN) = \mathcal{L}(DRRN)$. \square

Zusammenfassend erhalten wir den folgenden Satz.

Satz 2.53

- i) *Es gelten die Inklusionen entsprechend dem Graphen in Abbildung 2.4. (Der Graph wird wie folgt interpretiert: Sind zwei Mengen A und B durch eine Kante verbunden, und ist A weiter oben als B angeordnet, so gilt $B \subseteq A$. Ist die Kante ein Pfeil, so ist die Inklusion sogar echt, d. h. $B \rightarrow A$ steht für $B \subset A$.)*
- ii) *Die Familie $\mathcal{L}(CF)$ ist hinsichtlich der Inklusion unvergleichbar mit den Sprachfamilien $\mathcal{L}(RR)$, $\mathcal{L}(DRR)$, $\mathcal{L}(R)$ und $\mathcal{L}(DR)$.*
- iii) *Die Familien $\mathcal{L}(DRR)$ und $\mathcal{L}(DRN)$ sind unvergleichbar mit $\mathcal{L}(R)$. Die Sprachfamilie $\mathcal{L}(DRN)$ ist unvergleichbar mit $\mathcal{L}(RR)$.* \square

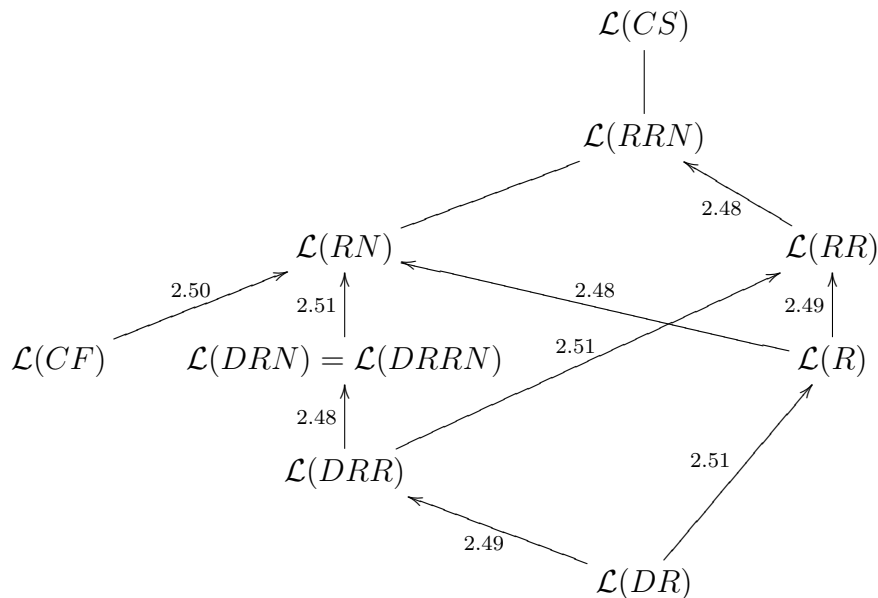


Abbildung 2.4: Hierarchie der Sprachklassen, die von Automaten mit Neustart erzeugt werden (Die an die Kanten geschriebenen Markierungen verweisen jeweils auf das Lemma, aus dem die Echtheit der Inklusion resultiert.)

Als wesentliche offene Probleme verbleiben die Fragen, ob die Inklusionen

$$\mathcal{L}(RRN) \subseteq \mathcal{L}(CS) \text{ und } \mathcal{L}(RN) \subseteq \mathcal{L}(RRN)$$

echt sind.

Abschließend geben wir noch eine Beziehung zwischen Automaten mit Neustart und kontextualen Grammatiken an. Dazu definieren wir zuerst zwei spezielle Klassen dieser Automaten bzw. Grammatiken.

Definition 2.54 Ein R-Automat $M = (Q, \sigma, \Sigma, \phi, \$, q_o, k, \delta)$ heißt normal, wenn zu jedem Element $(q', v) \in \delta(q, u)$ mit $v \in \Sigma^*$ Wörter $x_1, x_2, x_3, x_4, x_5 \in \Sigma^*$ so existieren, dass $u = x_1x_2x_3x_4x_5$ und $v = x_1x_3x_5$ gelten.

Bei einem normalen R-Automaten werden in jedem Ersetzungsschritt zwei Teilwörter gleichzeitig gestrichen.

Definition 2.55 Wir sagen, dass eine totale kontextuale Grammatik $G = (V, B, C, \varphi)$ einen regulären Präfix, beschränkten Infix und Grad $k, k \geq 0$, hat (wofür wir kurz CRPBI-Grammatik vom Grad k sagen), wenn folgende Bedingungen erfüllt sind:

- Für jedes Paar (y, z) von Wörtern über V ist die Menge

$$R(y, z) = \{x \mid \varphi(x, y, z) \neq \emptyset\}$$

regulär.

- Die Menge $\{y \mid \varphi(x, y, z) \neq \emptyset\}$ ist endlich.
- Für jedes Quadrupel (x, y, z_1, z_2) von Wörtern über V , bei denen die Präfixe der Länge k von z_1 und z_2 übereinstimmen, gilt $\varphi(x, y, z_1) = \varphi(x, y, z_2)$.

Damit hängt der auszuwählende Kontext $(u, v) \in \varphi(x, y, z)$ nur von den ersten k Buchstaben von z (und nicht vom ganzen Wort z) ab. Folglich gilt auch $R(y, zz') = R(y, zz'')$ für beliebige Wörter y, z, z', z'' mit $|z| = k$. Wir geben daher als zweites Argument bei $R(y, z)$ nur noch Wörter z mit $|z| \leq k$ an. Beachten wir nun noch die zweite Forderung aus Definition 2.55, so ergibt sich, dass es nur endlich viele reguläre Mengen $R(y, z)$ gibt.

Satz 2.56 Für eine Sprache L sind die beiden folgenden Aussagen äquivalent:

- L wird von einer kontextualen Grammatik mit regulärem Präfix, beschränktem Infix und Grad $k, k \geq 0$, erzeugt.
- L wird von einem normalen R-Automaten akzeptiert.

Beweis. i) \implies ii). Es sei eine CRPBI-Grammatik $G = (V, B, C, \varphi)$ vom Grad k gegeben. Wir setzen

$$\begin{aligned} k_1 &= \max\{|uyv| \mid (u, v) \in \varphi(x, y, z) \text{ für gewisse } x, z \in V^*\}, \\ k_2 &= \max\{|b| \mid b \in B\}, \\ k_3 &= \max\{k_2 + 1, k_1 + k\} \end{aligned}$$

und betrachten den normalen R-Automaten

$$M = (Q, V, V, \mathfrak{c}, \$, q_0, k_3, \delta),$$

bei dem Q , q_0 und δ so definiert sind, dass folgende Metaregeln gelten:

$$\begin{aligned} & (\mathfrak{c}R(y, z), uylvz \rightarrow yz) \text{ f\u00fcr } (u, v) \in \varphi(x, y, z), x \in R(y, z), |z| \leq k, \\ & (\mathfrak{c}u\$, ACCEPT) \text{ f\u00fcr } |u| < k_3, u \in L(G). \end{aligned}$$

F\u00fcr W\u00f6rter einer L\u00e4nge kleiner k_3 erfolgt sofort eine Akzeptanz oder Ablehnung. F\u00fcr l\u00e4ngere W\u00f6rter erhalten wir nach Definition von M

$$\mathfrak{c}xuyvzz''\$ \models_M \mathfrak{c}xyzz''\$ \text{ genau dann, wenn } xyz z'' \xrightarrow[G]{=} xuyvzz'',$$

woraus

$$\mathfrak{c}w\$ \models_M^* \mathfrak{c}u\$ \models_M ACCEPT \text{ genau dann, wenn } u \xrightarrow[G]{*} w$$

f\u00fcr ein $u \in L(G)$ mit $|u| < k_3$ folgt. Hieraus erh\u00e4lt man direkt $L(G) = T(M)$.

ii) \implies i) Es sei ein normaler R-Automat

$$M' = (Q, \Sigma, \Sigma, \mathfrak{c}, \$, q_0, k', \delta)$$

gegeben. Wir konstruieren die totale kontextuale Grammatik $G' = (\Sigma, B', C', \varphi')$ mit

$$\begin{aligned} & (u, v) \in \varphi'(xx_1, y, zz_1) \text{ f\u00fcr eine Metaregel } (\mathfrak{c}R, x_1uyvz \rightarrow x_1yz) \text{ mit } x \in R \text{ und} \\ & B = \{ b \mid (\mathfrak{c}b\$, ACCEPT) \text{ ist eine Metaregel} \}. \end{aligned}$$

Wie im ersten Teil dieses Beweises kann nun $L(G') = T(M')$ gezeigt werden. Es bleibt zu zeigen, dass G' eine CRPBI-Grammatik mit einem gewissen Grad l ist. Dazu bemerken wir zuerst, dass die L\u00e4nge von y mit $\varphi'(xx_1, y, zz_1) \neq \emptyset$ wegen der zugeh\u00f6rigen Metaregel $(\mathfrak{c}R, x_1uyvz \rightarrow x_1yz)$ durch die Fensterl\u00e4nge k' beschr\u00e4nkt ist. Damit ist die Menge dieser y endlich und die zweite Forderung aus Definition 2.55 erf\u00fcllt. Weiterhin gilt $\varphi'(xx_1, y, zz_1) \neq \emptyset$ genau dann, wenn $x \in R$ und $|x_1yz| = k'$ gelten sowie $(\mathfrak{c}R, x_1uyvz \rightarrow x_1yz)$ eine Metaregel ist. Damit ergibt sich

$$R(y, zz_1) = R\{x_1 \mid |x_1yz| = k' \text{ und } (\mathfrak{c}R, x_1uyvz \rightarrow x_1yz) \text{ ist eine Metaregel}\}.$$

Folglich ist $R(x, y)$ das Produkt einer regul\u00e4ren Menge mit einer endlichen Menge, also regul\u00e4r nach den Abschlusseigenschaften von $\mathcal{L}(REG)$. Somit ist auch die erste Bedingung aus Definition 2.55 nachgewiesen. Die dritte Bedingung folgt einfach daraus, dass $\varphi'(xx_1, y, zz_1)$ nicht von z' abh\u00e4ngt und die L\u00e4nge von z durch k' beschr\u00e4nkt ist. \square

F\u00fcr weitere Resultate \u00fcber Automaten mit Neustart verweisen wir auf [15] und [9] und die dort angegebene Literatur.

Literaturverzeichnis

- [1] A. AHO Indexed grammars – an extension of context-free grammars. *Journal ACM* **15** (1968), 647–671.
- [2] N. CHOMSKY, *Aspects of the Theory of Syntax*. MIT Press, Cambridge, 1965.
- [3] N. CHOMSKY, Three models for the description of language. *IRE Transactions on Information Theory* **2** (1956), 113–124.
- [4] A. EHRENFEUCHT, GH. PĂUN und G. ROZENBERG, Contextual grammars and formal languages. In: [20], 237–293.
- [5] G. GAZDAR, Applicability of indexed grammars to natural languages. In: U. REYLE und C. ROHRER (Eds.), *Natural Language Parsing and Linguistic Theories*, Reidel, Dordrecht, Holland, 1988, 69–94.
- [6] R. GRAMATOVICI and F. MANEA, A CYK-based parser for local internal contextual grammars with context-free choice. In: Z. ESIK and Z. FÜLÖP, *Proc. 11th Internat. Conference Automata and Formal Languages*, Dobogókő, Ungarn, 2005, 122–130.
- [7] J. E. HOPCROFT und J. D. ULLMAN, *Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie*. 2. Aufl., Addison-Wesley, 1990.
- [8] L. ILIE On the computational complexity of contextual languages. *Theor. Comp. Sci.* **183** (1997), 33–44.
- [9] P. JANČAR, F. MRÁZ, M. PLÁTEK, M. PROCHÁZKA und J. VOGEL, Restarting automata, Marcus grammars and context-free languages. In: J. DASSOW, G. ROZENBERG und A. SALOMAA (Hrsg.), *Developments of Language Theory II*. World Scientific Publ. Co., Singapore, 1996, 102–111.
- [10] A. K. JOSHI, How much context-sensitivity is necessary for characterizing structural descriptions: Tree adjoining grammars. In: D. DOWTY, L. KARTTUNEN und A. ZWICKY (Eds.) *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*. Cambridge University Press, New York, 1985.
- [11] A. K. JOSHI, L. S. LEVY und M. TAKAHASHI, Tree adjunct grammars. *J. Comput. Syst. Sci.* **10** (1975), 136–163.
- [12] M. KUDLEK, C. MARTIN-VIDE, A. MATEESCU und V. MITRANA, Contexts and the concept of mild context-sensitivity. *Linguistics and Philosophy* **26** (2003), 703–725.

- [13] S. MARCUS, Contextual grammars. *Revue Roum. Math. Pures Appl.* **14** (1969), 1473–1482.
- [14] S. MARCUS, Contextual grammars and natural languages. In: [20], Vol. II, 215–235.
- [15] F. OTTO, *Formale Sprachen und Automaten*. Kapitel 7. Vorlesungsskript, Universität Kassel, Fachbereich Mathematik/Informatik, 2007.
- [16] GH. PĂUN, *Marcus Contextual Grammars*. Kluwer, 1997.
- [17] C. POLLARD, *Generalized Phrase Structure Grammars, Head Grammars and Natural Language*. PhD Thesis, Stanford University, 1984.
- [18] K. R. REISCHUK, *Einführung in die Komplexitätstheorie*. Teubner-Verlag, Stuttgart, 1990.
- [19] K. ROACH, Formal properties of head grammars. In: A. MANASTER-RAMER (Ed.), *Mathematics of Language*. John Benjamins, Amsterdam, 1987.
- [20] G. ROZENBERG und A. SALOMAA (Eds.), *Handbook of Formal Languages*, Vol. I – Vol. III. Springer-Verlag, Berlin, 1997.
- [21] Y. SCHABES, *Mathematical and Computational Aspects of Lexicalized Grammars*. PhD Thesis, University of Pennsylvania, Philadelphia, 1990 (auch Technical Report MS-CIS-90-48, LINC LAB179 des Department of Computer Science der Universität).
- [22] U. SCHÖNING, *Theoretische Informatik kurz gefaßt*. B. I.-Wissenschaftsverlag, Mannheim, 1992.
- [23] M. TOMITA (Ed.) *Current Issues in Parsing Technology*. Kluwer, 1991.
- [24] K. VIJAY-SHANKAR, *A Study of Tree Adjoining Grammars*. PhD Thesis, University of Pennsylvania, Philadelphia, 1987.
- [25] K. VIJAY-SHANKAR and A. K. JOSHI, Some computational properties of tree adjoining grammars. In: *23rd Meeting Assoc. Comput. Ling.*, Chicago, Illinois, 1985, 82–93.
- [26] K. VIJAY-SHANKAR und D. J. WEIR, Polynomial parsing of extensions of context-free grammars. In: [23], 191–206.
- [27] K. VIJAY-SHANKAR und D. J. WEIR, The equivalence of four extensions of context-free grammars. *Mathematical System Theory* **87** (1994), 511–646.
- [28] K. WAGNER, *Einführung in die Theoretische Informatik*. Springer-Verlag, Berlin, 2003.
- [29] I. WEGENER, *Theoretische Informatik*. Teubner-Verlag, Stuttgart, 1993.