

Seminar
Graph-Grammatiken

Lehrstuhl für Informatik III
RWTH Aachen
Sommersemester 2004

Hyperedge Replacement Graph Grammars

Thomas Lettow und Thorsten Hermes

Inhaltsverzeichnis

1	Einführung	1
2	Formale Definitionen	2
2.1	Hyperkanten und Hypergraphen	3
2.1.1	Spezialfälle von Hypergraphen	3
2.2	Hyperkanten-Ersetzung	3
2.2.1	Sequentialisierbarkeit und Parallelisierbarkeit	4
2.2.2	Konfluenz	4
2.2.3	Assoziativität	4
2.3	Hyperkanten-Ersetzungs-Grammatiken	4
3	Eigenschaften von Hyperkanten-Ersetzungs-Grammatiken	6
3.1	Kontext-Freiheit	8
3.1.1	Ableitungsbäume	9
3.2	Fixpunkt-Theorem	11
3.3	Pumping Lemma	12
3.4	Parikh's Theorem	15
4	Generative Mächtigkeit, Hierarchie-Theoreme	15
4.1	Graphen	16
4.2	Zeichenketten-Graphen	17
5	Entscheidungsprobleme	22
5.1	Kompatible Eigenschaften	22
5.2	Kompatible Funktionen	24
6	Das Wort-Problem	25
6.1	NP-Vollständigkeit	25
6.2	Einschränkungen und Polynomielle Algorithmen	26
7	Zusammenfassung und Ausblick	26

1 Einführung

Eine *Hyperkante* stellt, wie der Name bereits nahe legt, eine Erweiterung der klassischen Kanten zwischen zwei Knoten eines Graphen dar. Eine Hyperkante besitzt eine feste Anzahl Tentakeln; diese Zahl wird der Typ der Hyperkante genannt. Jede dieser Tentakeln ist mit einem Knoten des umgebenden Graphen verbunden, den sogenannten *Anknüpfungs-Knoten*. Die Tentakeln werden in einer geordneten Menge verwaltet, können also anhand ihrer Reihenfolge unterschieden werden. Gewöhnliche gerichtete Kanten kann man also als Hyperkanten vom Typ 2 interpretieren. Eine Ansammlung von Knoten und Hyperkanten, deren Tentakeln alle mit Knoten verbunden sind, bezeichnet man als *Hypergraph*.

Hypergraphen bieten sich aufgrund dieser Erweiterung und der damit verbundenen Möglichkeiten noch stärker als gewöhnliche Graphen als überaus elegante Mittel zur Darstellung verschiedenster Daten an. Darüber hinaus teilen die im Folgenden vorgestellten Hyperkanten-Ersetzungs-Grammatiken viele der bereits bekannten Charakteristika kontext-freier Grammatiken auf Zeichenketten, was ihre Anwendung relativ einfach macht.

Beispiel. Als einführendes Beispiel wollen wir in Abbildung 1 die Repräsentation einer mathematischen Formel durch einen Hypergraphen betrachten, und bereits die auch in späteren Grafiken verwendeten Symbole definieren (Abbildung 2).

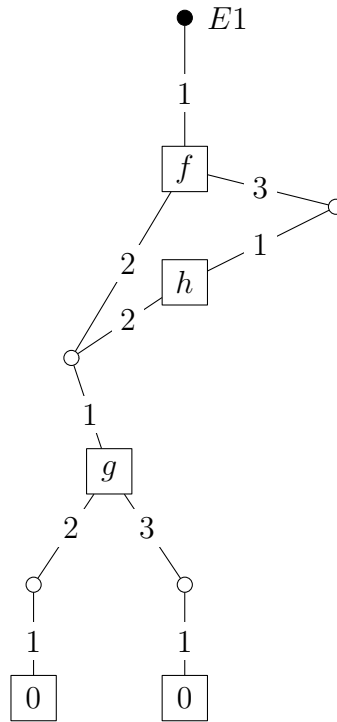


Abbildung 1: Beispiel: Hypergraph zum Ausdruck $f(g(0,0), h(g(0,0)))$

Alle Hyperkanten in diesem Beispiel verfügen über mindestens eine Tentakel (jeweils mit 1 bezeichnet), welche hier als die Stelle zum Auslesen des Wertes des durch die Hyperkante modellierten Termes angesehen werden kann, und darüber hinaus

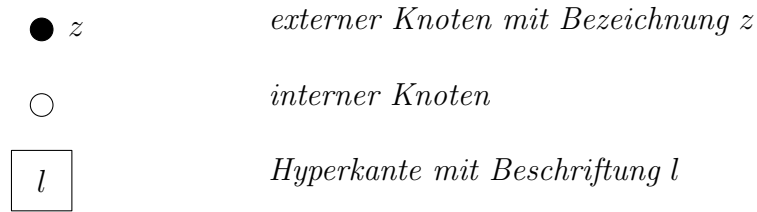


Abbildung 2: Legende zur visuellen Darstellung von Hypergraphen

optionale weitere Tentakeln zur Anbindung von Argumenten in Form weiterer Hyperkanten.

Die Konstanten 0 zum Beispiel werden als Hyperkanten mit dem Label 0 modelliert, und besitzen, da sie keine weiteren Parameter erhalten, auch keine weiteren Tentakeln. Dagegen hat f mit den beiden Argumenten $g(0, 0)$ und $h(g(0, 0))$ insgesamt drei Tentakeln. Die Tentakel 1 von f als dem äußersten Ausdruck in der Klammerung von $f(g(0, 0), h(g(0, 0)))$ stellt den Wert des gesamten Termes nach außen hin zur Verfügung, und ist deshalb mit einem externen Knoten $E1$ verbunden.

Der gesamte dargestellte Graph könnte nun im Zuge einer Ersetzung in einem anderen Hypergraph an Stelle eine Hyperkante mit einer Tentakel eingefügt werden.

Bei der Ersetzung einer Hyperkante durch eine andere Struktur werden die sogenannten *externen Knoten* dieser Struktur, deren Anzahl dem Typ der Hyperkante entsprechen muss, mit den Anknüpfungs-Knoten des umgebenden Graphen verschmolzen.

Derartige Ersetzungen können beliebig iteriert werden, so lange sich im übrigen Graphen oder der eingesetzten Struktur noch nichtterminale Hyperkanten befinden. Durch Beschriftung der Hyperkanten ist es möglich, diese Ersetzung in Form von *Produktionen* (Ableitungsregeln) zu definieren. Durch eine solche Menge von Produktionen mit disjunkter Unterscheidung der Beschriftungen von Hyperkanten nach Terminal- und Nichtterminal-Symbolen, sowie Angabe einer Startstruktur kann eine Hyperkanten-Ersetzungs-Grammatik definiert werden, welche eine Hypergraph-Sprache erzeugt.

Von einer solchen Ersetzung ist der Kontext offensichtlich nicht betroffen, Hyperkanten-Ersetzungs-Grammatiken sind also, wie die schon in Kapitel 1 in [rozenberg97] vorgestellten C-edNCE Graph Grammatiken, kontextfreie Grammatiken, in dem Sinne dass bei einer Hyperkantenersetzung die Einbettung unverändert bleibt. Der Begriff “kontextfrei” hat also in unterschiedlichen Kalkülen eine unterschiedliche Bedeutung.

2 Formale Definitionen

Dieses Kapitel liefert die Grundlage für die nachfolgenden Betrachtungen, indem es die oben beschriebenen Hypergraphen und darauf mögliche Operationen auf eine formale Basis stellt. Ausgehend von der Beschreibung der statischen Struktur wird zunächst Form und Semantik von Ersetzungen festgelegt und schließlich deren Verwendung in Hyperkanten-Ersetzungs-Grammatiken formalisiert, mit denen sich

Hypergraph-Sprachen erzeugen lassen.

2.1 Hyperkanten und Hypergraphen

C sei eine beliebige, festgelegte Menge von Beschriftungen und $type : C \rightarrow \mathbb{N}$ eine Typisierungs-Funktion für C . Ein *Hypergraph* H ist ein Tupel (V, E, att, lab, ext) . Dabei ist

- V eine endliche Menge von Knoten,
- E eine endliche Menge von Hyperkanten,
- $att : E \rightarrow V^*$ eine Zuordnung der Anknüpfungs-Knoten zu jeder Hyperkante,
- $lab : E \rightarrow C$ die Zuordnung der Beschriftungen (wobei $type(lab(e)) = |att(e)|$ gelten muss), und
- $ext \subseteq V^*$ die Auflistung der (paarweise verschiedenen) externen Knoten.

Eine m -Kante $e \in E$ für ein $m \in \mathbb{N}$ bezeichnet eine Kante vom Typ m , wenn $type(lab(e)) = m$ gilt. Die Klasse aller Hypergraphen über C bezeichnen wir als $\mathcal{H}_C$. Unter einem n -Hypergraphen $H \in \mathcal{H}_C$ verstehen wir einen Hypergraphen mit $|ext_H| = n$, wobei $n \in \mathbb{N}$ der Typ des Hypergraphen, kurz $type(H)$, genannt wird.

Die *Größe* eines Hypergraphen ist definiert durch die Anzahl der in ihm enthaltenen Hyperkanten addiert mit der Anzahl der Knoten, und wird geschrieben als $|H| = |V| + |E|$.

Ein n -Hypergraph H mit n Knoten (was bedeutet, dass alle Knoten extern sind), und einer einzelnen Hyperkante e heißt *Henkel*, wenn $att_H(e) = ext_H$. Sei nun die Beschriftung dieser Hyperkante $lab_H(e) = A$, so nennt man H den durch A induzierten Henkel, was wir als $A^\bullet$ schreiben.

E_H bezeichnet die Menge aller Hyperkanten in einem Hypergraph H , E_H^X die Menge $\{e \in E_H \mid lab_H(e) \in X\}$ von Hyperkanten H mit Beschriftungen aus $X \subseteq C$.

2.1.1 Spezialfälle von Hypergraphen

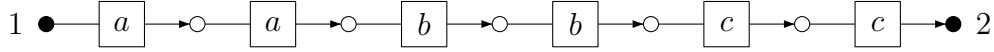
Wie bereits erwähnt, kann man gewöhnliche Kanten als Hyperkanten vom Typ 2 auffassen. Demnach sind Graphen spezielle Hypergraphen, die ausschliesslich Hyperkanten vom Typ 2 beinhalten, also bei denen die Funktion $att : E \rightarrow V^2$ jeder Kante genau zwei Verknüpfungs-Knoten zuordnet, zwischen denen dann je nach Gerichtetheit des Graphen zwischen Quelle und Ziel der Kante unterschieden werden kann.

Ein weiterer Spezialfall von Hypergraphen sind die sogenannten Zeichenketten-Graphen. Für ein Alphabet Σ kann man zu jedem Wort $w \in \Sigma^*$ den zugehörigen Zeichenketten-Graph konstruieren, indem man von einem Startknoten aus $|w|$ weitere Knoten der Reihe nach verbindet und die $|w|$ so aneinandergereihten Kanten in der richtigen Reihenfolge mit den Buchstaben aus w beschriftet. Betrachten wir derartige Graphen als Hypergraphen und ist w nicht das leere Wort, so besteht der aus w resultierende Zeichenketten-Hypergraph $w^\bullet$ immer aus zwei externen Knoten und $|w| - 1$ internen Knoten.

Beispiel. Abbildung 3 zeigt den Hypergraphen zur Zeichenkette $w = aabbcc$ aus der Sprache $L = \Sigma^*$ mit $\Sigma = \{a, b, c\}$.

2.2 Hyperkanten-Ersetzung

Sei $H \in \mathcal{H}_C$ ein Hypergraph, $B \subseteq E_H$ eine Menge zu ersetzender Hyperkanten in H und $repl : B \rightarrow \mathcal{H}_C$ eine Abbildung mit $type(repl(e)) = type(e)$ für alle $e \in B$. Die Ersetzung von B in H ergibt den Hypergraphen $H[repl]$, welchen man erhält, indem man B aus H entfernt, für jedes $e \in B$ jeweils die Knoten und Hyperkanten

Abbildung 3: Der Zeichenketten-Graph $w^\bullet$ zum Wort $w = aabcc$

aus $repl(e)$ hinzufügt, und schließlich für alle $i = 1, \dots, type(e)$ den i -ten externen Knoten von $repl(e)$ mit dem i -ten Anknüpfungs-Knoten von e verschmilzt. Die externen Knoten von $H[repl]$ sind genau die externen Knoten von H . Eine alternative Schreibweise für Ersetzungen lautet $H[e_1/R_1, \dots, e_n/R_n]$, wobei $repl(e_i) = R_i$ für alle $i = 1, \dots, n$ impliziert wird.

Aus dieser Definition lassen sich bereits einige grundlegende Eigenschaften ableiten:

2.2.1 Sequentialisierbarkeit und Parallelisierbarkeit

Das Ergebnis einer Ersetzung mehrerer Hyperkanten ist unabhängig davon, ob diese Ersetzung gleichzeitig oder nacheinander geschieht. Sei H ein Hypergraph mit paarweise verschiedenen Hyperkanten $e_1, \dots, e_n \in E_H$, und sei für alle $i = 1, \dots, n$ H_i ein Hypergraph mit $type(H_i) = type(e_i)$. Dann gilt:

$$H[e_1/H_1, \dots, e_n/H_n] = H[e_1/H_1] \dots [e_n/H_n]$$

2.2.2 Konfluenz

Dieses Teilergebnis wird eigentlich bereits durch das vorhergehende impliziert, da bei einer parallelen Ersetzung keine feste Reihenfolge definiert ist, dennoch formulieren wir es hier zur Verdeutlichung noch einmal explizit. Sei H ein Hypergraph mit verschiedenen Hyperkanten $e_1, e_2 \in E_H$, und sei für alle $i \in \{1, 2\}$ H_i ein Hypergraph mit $type(H_i) = type(e_i)$. Dann gilt:

$$H[e_1/H_1][e_2/H_2] = H[e_2/H_2][e_1/H_1]$$

2.2.3 Assoziativität

Wird eine Hyperkante durch einen Hypergraphen ersetzt, und wird in diesem im nächsten Schritt wiederum eine Ersetzung vorgenommen, so ist das Ergebnis das selbe, wenn man zuerst diese Ersetzung vornimmt und dann die ursprüngliche Hyperkante durch den so entstandenen Hypergraphen ersetzt. Seien H, H_1, H_2 Hypergraphen mit Hyperkanten $e_1 \in E_H$, $e_2 \in E_{H_1}$, und sei $type(e_1) = type(H_1)$ sowie $type(e_2) = type(H_2)$. Dann gilt:

$$H[e_1/H_1][e_2/H_2] = H[e_1/H_1[e_2/H_2]]$$

2.3 Hyperkanten-Ersetzungs-Grammatiken

Sei $N \subseteq C$ eine Menge von *Nichtterminalen*. Eine *Produktion* über N ist ein geordnetes Paar $p = (A, R)$ mit $A \in N$, $R \in \mathcal{H}_C$ und $type(A) = type(R)$, welches die Ersetzung von A durch R beschreibt. A wird hierbei als die linke Seite der Produktion, formal $lhs(p)$, R als die rechte Seite, $rhs(p)$, bezeichnet.

Existiert zu einem Nichtterminal $n \in N$ sowohl eine Produktion $(n, R) \in P$ als auch eine Hyperkante $e \in E_H$ mit Markierung n in einem Hypergraphen H , so ist $H' = H[e/R]$ eine *direkte Ableitung* von H . Wir schreiben $H \Rightarrow_P H'$, oder kurz $H \Rightarrow H'$, wenn P aus dem Kontext eindeutig ersichtlich ist.

Eine Folge direkter Ableitungen, $H_0 \Rightarrow \cdots \Rightarrow H_k$, bezeichnen wir als eine *Ableitung der Länge k* von H_0 nach H_k , und notieren sie analog obiger Regelung als $H_0 \Rightarrow_P^* H_k$ oder kurz $H_0 \Rightarrow^* H_k$, bzw., falls die Länge der Ableitung von Interesse ist, $H_0 \Rightarrow^k H_k$. Bei Identität (unter Isomorphie) sprechen wir auch von einer Ableitung der Länge 0.

Eine *Hyperkanten-Ersetzungs-Grammatik* ist ein Tupel $HRG = (N, T, P, S)$, wobei $N \subseteq C$ wie oben eine Menge von Nichtterminalen, $T \subseteq C$ eine Menge von Terminalen mit $T \cap N = \emptyset$ und $T \cup N = C$, P eine endliche Menge von Produktionen über N und $S \in N$ das *Startsymbol* darstellt.

Eine Sprache $L_A(HRG)$ mit $A \in N$ umfasst alle Hypergraphen in $\mathcal{H}_T$ (der Menge aller Hypergraphen, welche nur noch Terminalsymbole als Beschriftungen enthalten), welche von $A^\bullet$ durch Produktionen in P ableitbar sind:

$$L_A(HRG) = \{H \in \mathcal{H}_T \mid A^\bullet \Rightarrow_P^* H\}$$

Die von HRG generierte *Hypergraph-Sprache* $L(HRG)$ ist demzufolge $L_S(HRG)$.

Die Klasse aller Hyperkanten-Ersetzungs-Grammatiken bezeichnen wir mit $\mathcal{HRG}$, die aller Hyperkanten-Ersetzungs-Sprachen mit $\mathcal{HRL}$.

Eine Hyperkanten-Ersetzungs-Grammatik ist von r -ter Ordnung (für ein $r \in \mathbb{N}$), wenn für alle zugehörigen Produktionen $(A, R) \in P$ gilt: $type(R) \leq r$. Da der Typ der linken Seite einer Produktion immer gleich dem der rechten Seite ist, impliziert dies gleichzeitig $type(A^\bullet) \leq r$. Eine Hyperkanten-Ersetzungs-Sprache L ist von r -ter Ordnung, wenn eine Hyperkanten-Ersetzungs-Grammatik r -ter Ordnung mit $L(HRG) = L$ existiert. Die so definierten Klassen schreiben wir als $\mathcal{HRG}_r$ bzw. $\mathcal{HRL}_r$.

Stellen alle rechten Seiten der Produktionsregeln einer Grammatik gewöhnliche Graphen dar, so sprechen wir von einer *Kanten-Ersetzungs-Grammatik*, deren Klasse wir mit $\mathcal{ERG}$ bezeichnen. Ohne Beschränkung der Allgemeinheit lässt sich für diese Klasse offensichtlich annehmen, dass alle Nichtterminale mit Ausnahme des Startsymbols vom Typ 2 sind.

Beispiel. Abbildung 4 zeigt ein Beispiel für eine Hyperkanten-Ersetzungs-Grammatik $A^n B^n C^n = (N, T, P, S)$, wobei

- die vier Produktionen aus P im Bild dargestellt sind,
- die Menge der Nichtterminal-Symbole N gegeben ist durch die linken Seiten der Produktionen, also $N = \{S, A\}$,
- die Menge der Terminal-Symbole T gegeben ist durch die Beschriftungen der Hyperkanten auf den rechten Seiten der Produktionen, die nicht Nichtterminal-Symbole sind, also $T = \{a, b, c\}$,
- das Startsymbol durch S festgelegt ist.

Die Grammatik ist demnach eine Hyperkanten-Ersetzungs-Grammatik 4-ter Ordnung, denn die Hypergraphen, die auf den rechten Seiten der Produktionen vorkommen, sind höchstens vom Typ 4.

Abbildung 5 zeigt eine Folge von Ableitungen, wie sie mit den Produktionen aus P (siehe Abbildung 4), ausgehend von Startsymbol S durchgeführt werden können.

Man erkennt, dass die Sprache, die von $A^n B^n C^n$ erzeugt wird $L_S(A^n B^n C^n) = \{(a^n b^n c^n)^\bullet \mid n \geq 1\}$ ist. $A^n B^n C^n$ ist somit ein Beispiel für eine Hyperkanten-Ersetzungs-Grammatik, die Zeichenketten-Graphen erzeugt.

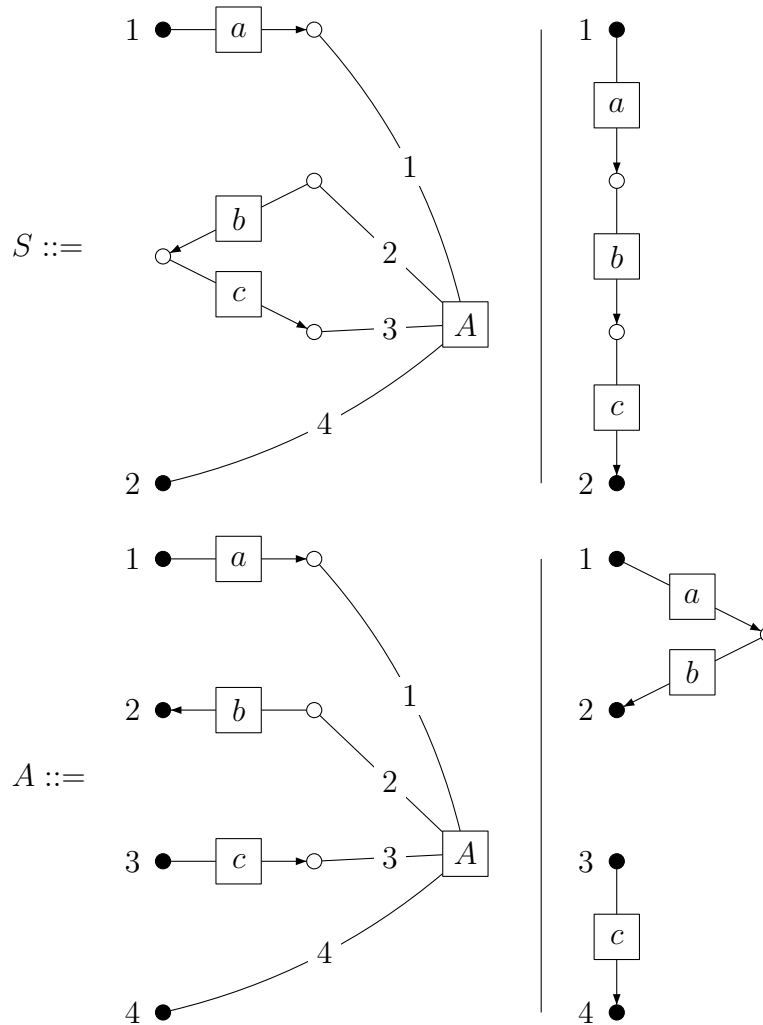


Abbildung 4: Produktionen zur Erzeugung von $(a^n b^n c^n)^\bullet$

An diesem Beispiel kann man erkennen, dass die Mächtigkeit von Hyperkanten-Ersetzungs-Grammatiken bei der Erzeugung von Zeichenketten sehr viel größer ist als die gewöhnlicher kontextfreier Zeichenketten-Grammatiken, mit denen man Sprachen der Familie $Z_m = \{a_1^n, \dots, a_m^n \mid n \geq 0\}$ nur für $m \leq 2$ erzeugen kann. Dies ist begründet durch die zusätzliche Dimension durch die Verwendung von Graphen, die nicht reine Ketten von Kanten darstellen. In Kapitel 4 wird gezeigt, dass man für jedes m eine Hyperkanten-Ersetzungs-Grammatik angeben kann, die Z_m erzeugt.

3 Eigenschaften von Hyperkanten-Ersetzungs-Grammatiken

Im folgenden Teil übertragen wir einige der bereits für Zeichenketten-Grammatiken bekannten Theorien auf Hyperkanten-Ersetzungs-Grammatiken. Wir

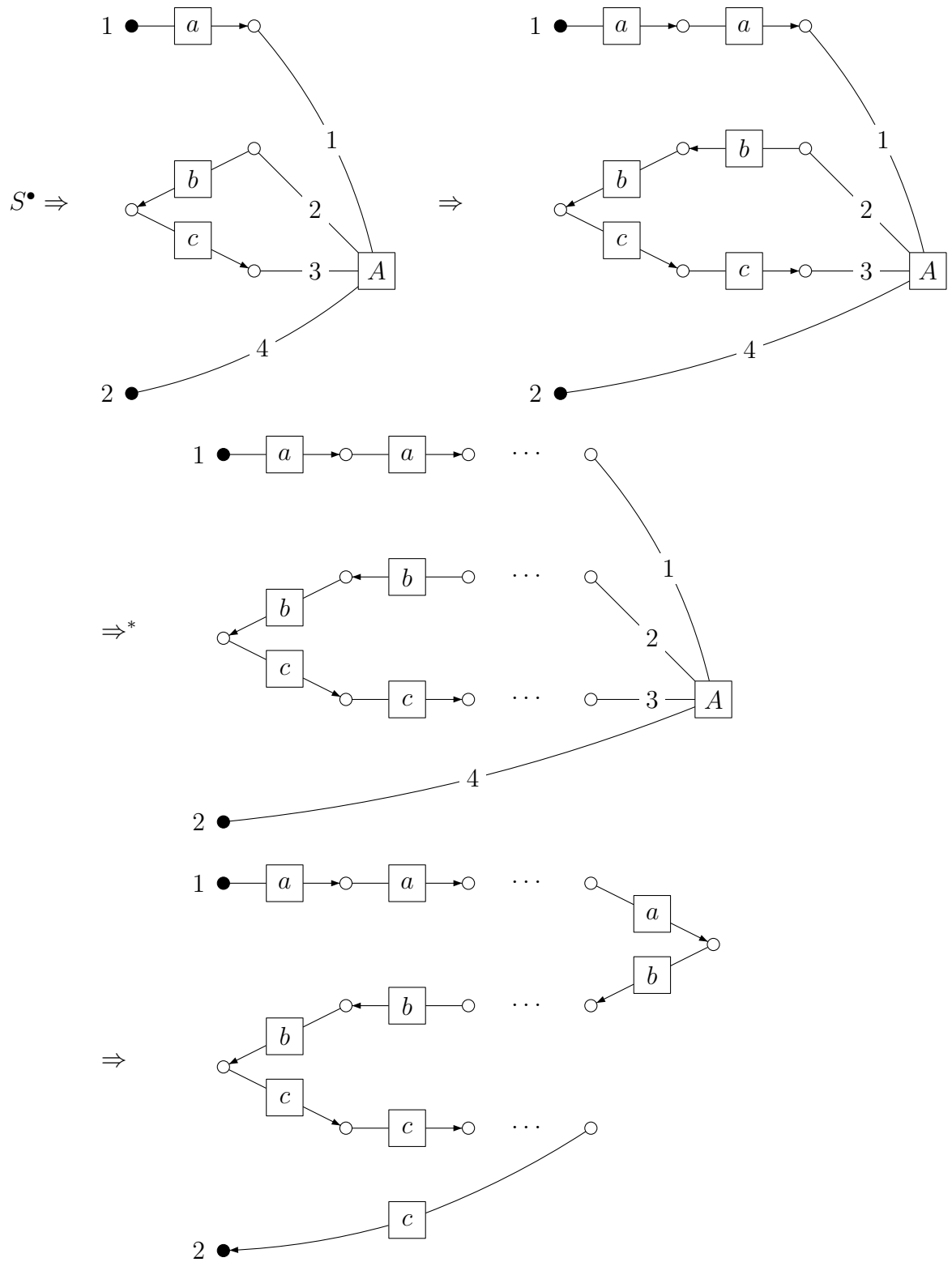


Abbildung 5: Eine Ableitung mit den Produktionen aus Abbildung 4

stellen fest, dass es auch hierbei eine kontext-freie Ersetzungs-Semantik vorliegt, welche wir als Grundlage für den Beweis eines Fixpunkt-Theorems, Pumping Lemmas und Parikh's Theorems verwenden werden.

3.1 Kontext-Freiheit

Die Kontext-Freiheit stellt die Grundlage für viele der folgenden Beweise dar.

Zum besseren Verständnis zuerst eine intuitive Erläuterung: Sei R der ursprüngliche und H der abgeleitete Hypergraph. Im Zuge der Ableitung werden alle Nichtterminal-Hyperkanten in R zu Unter-Hypergraphen von H abgeleitet. Wichtig ist hierbei, dass für die Durchführbarkeit einer solchen Ableitung allein die Existenz des Nichtterminals entscheidend ist, nicht jedoch der Kontext, weshalb wir unsere Betrachtung auf den zugehörigen Henkel $lab_R(e)^\bullet$ einer jeden Hyperkante $e \in E_R^N$ beschränken können, wodurch wir Teil-Ableitungen der Form $lab_R(e)^\bullet \Rightarrow^* repl(e)$ erhalten, so dass $R[repl] = H$ gilt.

Umgekehrt gilt auch, dass, wenn die Teil-Hypergraphen $repl(e)$ mit Ableitungen $lab_R(e)^\bullet \Rightarrow^* repl(e)$ für alle $e \in E_R^N$ gegeben sind, sich $R[repl]$ ($= H$) offensichtlich aus R ableiten lässt.

Im Folgenden verfolgen wir einen rekursiven Ansatz, da sich dieser gut für weitere Beweise verwenden lässt: Eine von einem Henkel $A^\bullet$ ausgehende Ableitung muss die Form $A^\bullet \Rightarrow R \Rightarrow^* H$ besitzen, wobei (A, R) eine Produktion ist. Der Rest $R \Rightarrow^* H$ kann nun wie beschrieben zerlegt werden.

Theorem 3.1.1 (Kontext-Freiheits-Lemma). Seien $N, T \subset C$ mit $N \cap T = \emptyset$ zwei disjunkte Teilmengen des Alphabets C . Sei P eine Menge von Produktionen über N . Sei $H \in \mathcal{H}_C$, $A \in N$ und $k \in \mathbb{N}$. Dann existiert eine Ableitung $A^\bullet \Rightarrow^{k+1} H$ genau dann, wenn eine Produktion $(A, R) \in P$ existiert und eine Abbildung $repl : E_R^N \rightarrow \mathcal{H}_C$ mit $H = R[repl]$, so dass $lab_R(e)^\bullet \Rightarrow^{k(e)} repl(e)$ für alle $e \in E_R^N$ und $\sum_{e \in E_R^N} k(e) = k$.

Beweis. Gemäß der Definition von Ableitungen muss, wenn eine Ableitung $A^\bullet \Rightarrow^{k+1} H$ existiert, diese die Form $A^\bullet \Rightarrow R \Rightarrow^k H$ für eine Produktion $(A, R) \in P$ haben. Zu zeigen verbleibt also:

Seien G, G' Hypergraphen. Dann gilt $G \Rightarrow^k G'$ genau dann, wenn eine Abbildung $repl : E_G^N \rightarrow \mathcal{H}_C$ mit $G' = G[repl]$ existiert, so dass $lab_G(e)^\bullet \Rightarrow^{k(e)} repl(e)$ für alle $e \in E_G^N$ und $\sum_{e \in E_G^N} k(e) = k$.

Wir beweisen dies per Induktion über k . Für den Fall $k = 0$ sind beide Richtungen trivialerweise erfüllt. Nehmen wir als $k > 0$ an.

Hin-Richtung: Es muss $G \Rightarrow G_0 \Rightarrow^{k-1} G'$ gelten, wobei $G_0 = G[e_0/R_0]$ für ein $e_0 \in E_G^N$ und ein $(lab_G(e_0), R_0) \in P$. Die Induktionsannahme liefert eine Abbildung $repl_0 : E_{G_0}^N \rightarrow \mathcal{H}_C$ mit $G' = G_0[repl_0]$, so dass $lab_{G_0}(e)^\bullet \Rightarrow^{k_0(e)} repl_0(e)$ für alle $e \in E_{G_0}^N$, wobei die Summe der $k_0(e)$ gleich $k - 1$ ist.

Sei nun $repl(e) = repl_0(e)$ für $e \in E_G^N \setminus \{e_0\}$ und $repl(e_0) = R_0[repl'_0]$, wobei $repl'_0$ die Einschränkung von $repl_0$ auf $E_{R_0}^N$ ist. Wenn nun $repl''_0$ die Einschränkung von $repl_0$ auf $E_G \setminus \{e_0\}$ ist, erhalten wir $G' = G_0[repl_0] = G[e_0/R_0][repl_0] = G[repl'_0][e_0/R_0][repl'_0] = G[repl]$, wie gefordert.

Rück-Richtung: Sei $G' = G[repl]$ mit $lab_G(e)^\bullet \Rightarrow^{k(e)} repl(e)$ für alle $e \in E_G^N$ und $\sum_{e \in E_G^N} k(e) = k$. O.B.d.A. können wir annehmen, dass $k(e) > 0$ für alle $e \in E_G^N$ (denn andernfalls beschränken wir $repl$ auf diese Hyperkanten). Sei $E_G^N = \{e_1, \dots, e_n\}$. Dann existieren Hypergraphen $G_1, \dots, G_n$, so dass $lab_G(e_i)^\bullet \Rightarrow$

$G_i \Rightarrow^{k(e_i)-1} \text{repl}(e_i)$ für $i = 1, \dots, n$.

Unter Verwendung der bereits bewiesenen Richtung existieren $\text{repl}_i : E_{G_i}^N \rightarrow \mathcal{H}_C$ für $i = 1, \dots, n$, so dass $G_i[\text{repl}_i] = \text{repl}(e_i)$, $\text{lab}_{G_i}(e)^\bullet \Rightarrow^{k_i(e)} \text{repl}_i(e)$ für alle $e \in E_{G_i}^N$, und $\sum_{e \in E_{G_i}^N} k_i(e) = k(e_i) - 1$.

Sei $G_0 = G[e_1/G_1, \dots, e_n/G_n]$ und $\text{repl}_0(e) = \text{repl}_i(e)$ für alle $e \in E_{G_i}^N$ ($i = 1, \dots, n$). Dann erhalten wir $G' = G[\text{repl}] = G[e_1/G_1[\text{repl}_1], \dots, e_n/G_n[\text{repl}_n]] = G[e_1/G_1, \dots, e_n/G_n][\text{repl}_0] = G_0[\text{repl}_0]$. Nach Induktionsannahme liefert dies eine Ableitung $G_0 \Rightarrow^{k-n} G'$, und durch die Konstruktion von G_0 erhalten wir $G \Rightarrow^n G_0$, womit der Beweis beendet ist. $\square$

3.1.1 Ableitungsbäume

Aufgrund der soeben formulierten Erkenntnis lassen sich Ableitungen durch Hyperkanten-Ersetzung auch durch Ableitungsbäume repräsentieren.

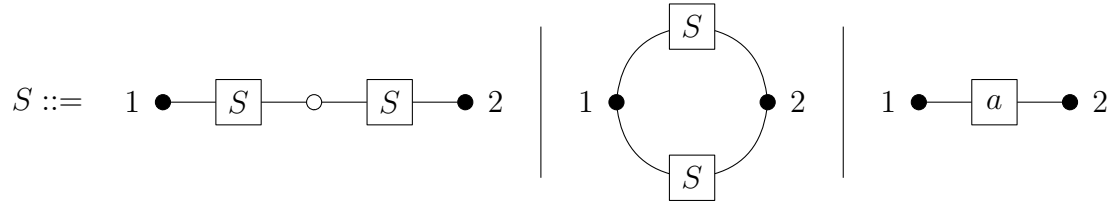
Für eine Ableitung $A^\bullet \Rightarrow^k H$ im trivialen Fall $k = 0$ gilt $A^\bullet = H$, und der Baum besteht nur aus dem Wurzelknoten mit der Beschriftung A . Ist hingegen $k > 0$, so lässt sich die Ableitung gemäß des obigen Ergebnisses zerlegen in $A^\bullet \Rightarrow R \Rightarrow^{k-1} H$, wobei (A, R) eine Produktion aus der Menge P der Grammatik, und zugleich Wurzelknoten des hier betrachteten Baumes ist. Der verbleibende Teil der Ableitung kann dann rekursiv weiter aufgelöst werden, wodurch Teilbäume des Ableitungsbaums aufgebaut werden.

Das Ergebnis eines Ableitungsbaumes ergibt sich aus der Anwendung der kumulierten Ersetzungen auf die ursprüngliche Hyperkante. Im Gegensatz zu Ableitungsbäumen bei Zeichenketten-Grammatiken kann man das Ergebnis der Ableitung im Allgemeinen nicht an der Front des Baumes ablesen, da hierbei die (Hyper-)Graphstruktur verloren gehen würde.

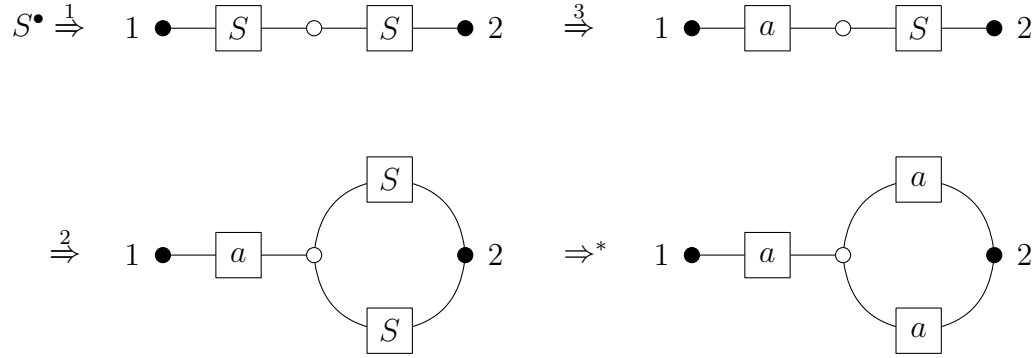
Definition (Ableitungsbäume). Wie schon die intuitive Beschreibung ist auch die formale Definition der Menge $TREE(P)$ der Ableitungsbäume über einer Menge P von Produktionen rekursiv aufgebaut:

- $N \subseteq TREE(P)$ mit $\text{root}(A) = A$ und $\text{result}(A) = A^\bullet$ für alle $A \in N$
- Für jede Produktion $(A, R) \in P$ und jede Abbildung $\text{branch} : E_R^N \rightarrow TREE(P)$ $N \subseteq TREE(P)$ mit $\text{type}(e) = \text{type}(\text{root}(\text{branch}(e)))$ für alle $e \in E_R^N$, ist das Tripel $t = (A, R, \text{branch}) \in TREE(P)$. Hierbei sei $\text{root}(t) = A$, und $\text{result}(t) = R[\text{repl}]$, wobei für alle $e \in E_R^N$, $\text{repl}(e) = \text{result}(\text{branch}(e))$ sei.

Beispiel. Um das Prinzip von Ableitungen in einer Hyperkanten-Ersetzungs-Grammatik und der dazugehörigen Ableitungsbäume zu veranschaulichen, sei ein einfaches Beispiel betrachtet: Abbildung 6 zeigt die drei Produktionen einer Hyperkanten-Ersetzungs-Grammatik, die alle Kombinationen von seriellen und parallelen 2-Hyperkanten erzeugen kann. Dazu zeigt Abbildung 7 eine mögliche Folge von Ableitungen. Stelle man sich die Produktionen von links nach rechts numeriert mit 1 bis 3 vor, dann wird mit der Zahl über den Ableitungspfeilen die im betreffenden Schritt verwendete Produktion vermerkt. Beim letzten Schritt wurden zwei Ableitungsschritte getätigt, die jeweils die dritte Produktion verwendeten, um die beiden nichtterminalen mit S beschrifteten Hyperkanten durch terminale zu ersetzen. Ordnet man die verwendeten Produktionsnummern in einer Baumstruktur an, so ergibt sich der in Abbildung 8 gezeigte Ableitungsbaum.



Abbildungung 6: Produktionsregeln zum Ableitungsbaum-Beispiel

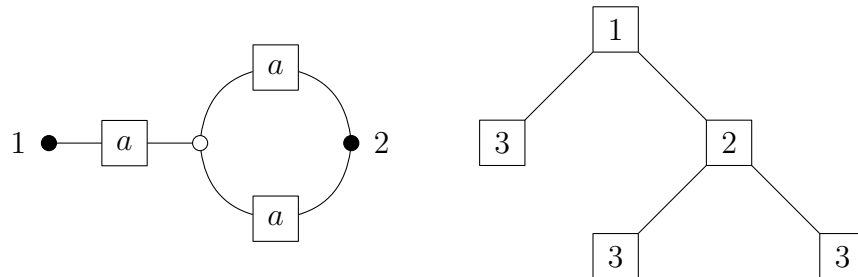


Abbildungung 7: Ableitung mit den Produktionsregeln aus Abbildung 6

Theorem 3.1.2 (Zusammenhang zwischen Ableitbarkeit und Ableitungsbäumen). Sei P eine Menge von Produktionsregeln über N , sei $A \in N$ und $H \in \mathcal{H}_C$. Dann existiert eine Ableitung $A^\bullet \Rightarrow^* H$ genau dann, wenn ein Ableitungsbaum $t \in \text{TREE}(P)$ mit $\text{root}(t) = A$ und $\text{result}(t) = H$ existiert.

Beweis. durch Induktion. □

Folgerung: Indem man zu jedem $t \in \text{TREE}(P)$ die Kombination $(\text{root}(t), \text{result}(t))$ als neue Produktion betrachtet, d.h. die Menge aller indirekt ableitbaren Wörter *direkt* ableitbar macht, kann man die transitive Hülle von P



Abbildungung 8: Der Beispiel-Hypergraph aus 7 und der zugehörige Ableitungsbaum

erzeugen: $P^* = \{(root(t), result(t)) \mid t \in TREE(P)\}$. Dann gilt: $A \Rightarrow_P^* H$ g.d.w. $A \Rightarrow_{P^*} H$.

3.2 Fixpunkt-Theorem

Das für Zeichenketten-Grammatiken bekannte Fixpunkt-Theorem von Ginsburg und Rice lässt sich für die Verwendung mit Hyperkanten-Ersetzungs-Sprachen verallgemeinern. Betrachtet man die Menge der Produktionen, die eine Hyperkanten-Ersetzungs-Sprache erzeugen, als ein Gleichungssystem, so stellt die erzeugte Sprache seinen kleinsten Fixpunkt dar. Dies wird im folgenden gezeigt. Dabei bezeichnet $\mathcal{HL}_C$ die Menge aller Hypergraph-Sprachen über C . Es gilt also $\mathcal{HL}_C = \mathcal{P}(\mathcal{H}_C)$.

Definition. Sind nun $N, T \in C$ disjunkte Mengen von Beschriftungen, dann ist die Abbildung $EQ : N \rightarrow \mathcal{HL}_{N \cup T}$ ein *Gleichungssystem über N* .

Die Abbildung $L : N \rightarrow \mathcal{HL}_T$ ist ein *Fixpunkt von EQ* wenn für alle $A \in N$ gilt:

$$L(A) = \bigcup_{R \in EQ(A)} \{R[repl : E_R^N \rightarrow \mathcal{H}_T] \mid repl(e) \in L(lab_R(e)) \forall e \in E_R^N\}.$$

Ein Fixpunkt L heißt *kleinster Fixpunkt von EQ* wenn für alle Fixpunkte L' von EQ und alle $A \in N$ gilt: $L(A) \subseteq L'(A)$.

Anmerkungen:

1. Um den Fixpunkt-Charakter hervorzuheben, kann ein Gleichungssystem $EQ : N \rightarrow \mathcal{HL}_{N \cup T}$ über N auch mit $A = EQ(A)$ für $A \in N$ bezeichnet werden.
2. Ein Gleichungssystem ist eine Repräsentation einer Menge von Produktionen über N .
3. Das mit einer Menge von Produktionen P assoziierte Gleichungssystem $EQ_P : N \rightarrow \mathcal{HL}_{N \cup T}$ über N ist definiert als $EQ_P(A) = \{R \mid (A, R) \in P\}$.
4. Umgekehrt liefert ein Gleichungssystem $EQ : N \rightarrow \mathcal{HL}_{N \cup T}$ über N eine Menge von Produktionen $P(EQ) = \{(A, R) \mid R \in EQ(A)\}$.
5. Die von einer Hyperkanten-Ersetzungs-Grammatik HRG erzeugte Sprachfamilie $L : N \rightarrow \mathcal{HL}_T$ ist gegeben durch $L(A) = L_A(HRG)$ für $A \in N$.

Theorem 3.2.1 (Fixpunkt-Theorem). Sei $HRG = (N, T, P, S)$ eine Hyperkanten-Ersetzungs-Grammatik, $EQ_P : N \rightarrow \mathcal{HL}_{N \cup T}$ das mit P assoziierte Gleichungssystem und $L : N \rightarrow \mathcal{HL}_T$ die von HRG erzeugte Sprachfamilie. Dann ist L der kleinste Fixpunkt von EQ_P .

Beweis. Sei $L' : N \rightarrow \mathcal{HL}_T$ ein Fixpunkt von EQ . Zu zeigen ist, dass für alle $A \in N$ gilt $L(A) \subseteq L'(A)$, also $H \in L'(A)$ für alle $H \in \mathcal{H}_T$ mit $A^\bullet \Rightarrow_P^* H$. Dies wird per Induktion über die Länge der Ableitung k gezeigt.

Ist $k = 1$, also $A^\bullet \Rightarrow_P^1 H$, so existiert eine Produktion $(A, R) \in P$ mit $H \in \mathcal{H}_T$, so dass die leere Abbildung $empty : \emptyset \rightarrow \mathcal{H}_T$ definiert ist als $H[empty] = H$. Da L' Fixpunkt von EQ ist, ist $H = H[empty] \in L'(A)$.

Betrachten wir nun eine Ableitung der Länge $k + 1$, also $A^\bullet \Rightarrow_P^{k+1} H$. Nach dem Kontext-Freiheits-Lemma gibt es Produktionen $(A, R) \in P$ und Ableitungen $lab_R(e)^\bullet \Rightarrow^{k(e)} repl(e)$ für $e \in E_R^N$ mit $k(e) \leq k$ und $H = R[repl]$. Nach der Induktionsvoraussetzung ist $repl(e) \in L'(lab_R(e))$. Da L' Fixpunkt von EQ ist, ist deshalb auch $H = R[repl] \in L'(A)$. $\square$

3.3 Pumping Lemma

Im folgenden wird ein Pumping Lemma für Hyperkanten-Ersetzungs-Sprachen vorgestellt. Demnach lässt sich jeder ausreichend große Hypergraph einer Hyperkanten-Ersetzungs-Sprache in drei Teil-Hypergraphen zerlegen, und zwar so, dass durch Vervielfältigung ("Pumpen") des mittleren Teil-Hypergraphen der neu entstandene Hypergraph auch der Sprache zugehörig ist. Das Lemma kann benutzt werden, um zu zeigen, dass bestimmte Sprachen keine Hyperkanten-Ersetzungs-Sprachen sind. Diese Anwendung ist analog zum Gebrauch des Pumping Lemmas im Bereich der kontextfreien Zeichenketten-Sprachen.

Ein Hypergraph H ist *substantiell* wenn $V_H \neq [ext_H]$ oder $|E_H| > 1$, d.h. er enthält interne Knoten oder mehr als eine Hyperkante. Eine Hyperkanten-Ersetzungs-Grammatik heißt *wachsend* wenn jede ihrer Produktionen eine substantielle rechte Seite hat. Eine wachsende Hyperkanten-Ersetzungs-Grammatik erzeugt ausschliesslich substantielle Hypergraphen.

Bemerkung: Alle bis auf endlich viele Hypergraphen in einer Hyperkanten-Ersetzungs-Sprache L sind substantiell, weil $\{|ext_H| \mid H \in L\}$ endlich ist.

Lemma 3.3.1. Zu jeder Hyperkanten-Ersetzungs-Grammatik HRG kann effizient eine wachsende Hyperkanten-Ersetzungs-Grammatik HRG' der gleichen Ordnung konstruiert werden, die $L(HRG') = \{H \in L(HRG) \mid H \text{ substantiell}\}$ erfüllt. (Ohne Beweis)

Definition. Sei $X \in C$. Ein Hypergraph H heißt *X-henkelig* wenn genau eine Kante $e \in E_H$ mit X beschriftet ist. e heißt dann *X-Henkel von H*. Der Hypergraph ohne den X -Henkel wird mit $H^\ominus = H - \{e\}$ bezeichnet.

Wird der X -Henkel e eines Hypergraphen H durch einen anderen Hypergraphen H' ersetzt, spricht man von der *Komposition von H und H'*, geschrieben als $H \otimes H' = H[e/H']$.

Man schreibt H^k für die k -malige Ersetzung des X -Henkels von H durch H selbst, wobei bei jeder Ersetzung die X -henkelig-Eigenschaft erhalten bleibt. Formal ist H^k rekursiv definiert durch $H^0 = X^\bullet$ und $H^{i+1} = H \otimes H^i$ für $i \geq 0$.

Theorem 3.3.1 (Pumping Lemma). Sei L eine Hyperkanten-Ersetzungs-Sprache r -ter Ordnung. Dann existieren Konstanten p und q , so dass folgendes gilt: Zu jedem Hypergraph H in L mit $|H| > p$ gibt es für ein $X \in C$:

- einen X -henkeligen Hypergraphen $FIRST$,
- einen substantiellen X -henkeligen $type(X)$ -Hypergraphen $LINK$ und
- einen $type(X)$ -Hypergraphen $LAST$,

mit folgenden Eigenschaften:

- $H = FIRST \otimes LINK \otimes LAST$,
- $|LINK \otimes LAST| \leq q$ und
- $type(LINK) \leq r$,

so dass gilt:

- $FIRST \otimes LINK^k \otimes LAST \in L \forall k \in \mathbb{N}$ und
- für jeden Hypergraph H in L mit $|V_H| > p$ kann man $LINK$ so wählen, dass $V_{LINK} \setminus [ext_{LINK}] \neq \emptyset$ (d.h. $LINK$ hat mindestens einen internen Knoten).

Beweis. Sei $HRG = (N, T, P, S) \in \mathcal{HRG}_r$ mit $L(HRG) = L$ und $|N| = n$. Da es nur endlich viele nicht-substantielle Hypergraphen in $L(HRG)$ gibt, ist HRG wachsend. Sei max die Größe der größten rechten Seite der Produktionen von HRG .

Der Beweis benutzt nun den Ableitungsbaum eines beliebigen Hypergraphens aus L , um eine korrekte Zerlegung nach dem Pumping-Lemma zu finden: Sei $t \in TREE(P)$ der Ableitungsbaum eines Hypergraphen H mit $root(t) = S$ und $H = result(t) \in \mathcal{H}_T$. Wenn $|H| > max^n$, dann enthält t einen Pfad länger als n , d.h. auf diesem Pfad kommt ein Nichtterminal-Symbol X zweimal vor. t hat demzufolge einen Unterbaum t' mit der Wurzel X , und dieser hat wiederum einen Unterbaum t'' mit der Wurzel X .

Sei nun $t - t'$ (analog $t' - t''$) der Baum, den man erhält wenn man den Unterbaum entfernt. Dann wähle man $LAST = result(t'')$, $LINK = result(t' - t'')$ und $FIRST = result(t - t')$. Dadurch, dass X nach dieser Zerlegung das einzige in den Teilgraphen vorkommende Nichtterminal-Symbol ist, sind $FIRST$ und $LINK$ X -henkelig. $LINK$ und $LAST$ sind $type(X)$ -Hypergraphen, da sie gerade durch eine Produktion der Art $X \rightarrow R$ innerhalb von H erzeugt worden sind. Aus der Definition von Ableitungsbäumen ist klar, dass $H = FIRST \otimes LINK \otimes LAST$ gilt. Weil HRG wachsend ist, muss $LINK$ ein substantieller Hypergraph sein.

Dass $FIRST \otimes LINK^k \otimes LAST \in L \forall k \in \mathbb{N}$ gilt, soll an dieser Stelle nicht formal bewiesen werden, aber man kann sich leicht überlegen, dass man aus t weitere Ableitungsbäume von HRG erzeugen kann, indem man an den ursprünglichen Einhängpunkt von t'' (innerhalb von t') noch einmal komplett t' einhängen kann.

Die im Lemma geforderten Konstante p kann man demnach wie gesehen auf max^n festlegen. Es ist leicht ersichtlich, dass man auch eine Konstante q finden kann, die man als Schranke für die Größe der Komposition aus $LINK$ und $LAST$ festlegen kann.

Nun muss noch die zweite Behauptung des Lemmas bewiesen werden. Sei also $|V_H| > p$. Dann können wir $t' - t''$ so wählen, dass $result(t' - t'')$ mindestens einen internen Knoten hat. In diesem Fall kann man den Ableitungsbaum reduziert um die Knoten, deren rechte Seiten keine internen Knoten besitzen, betrachten, und man wird eine Zerlegung nach dem obigen Schema finden können, bei der $LINK$ nicht ausschliesslich externe Knoten enthält.

□

Beispiel. Zur Veranschaulichung des Lemmas wollen wir die bereits im Beispiel verwendete Sprache $L_3 = \{(a^n b^n c^n)^\bullet \mid n \geq 1\}$ noch einmal betrachten. Wir haben gesehen, dass es eine Hyperkanten-Ersetzungs-Grammatik gibt, die L_3 erzeugt. Demnach lassen sich nach dem Pumping-Lemma ausreichend große Hypergraphen aus L_3 in drei Teil-Hypergraphen zerlegen, so dass durch Pumpen des mittleren Teils neue Hypergraphen aus L_3 entstehen.

Eine derartige Zerlegung ist in Abbildung 9 dargestellt. Den Teil-Hypergraphen $LINK$ stelle man sich als X -henkelig ähnlich dem Hypergraphen auf der rechten Seite der dritten Produktion aus Abbildung 4 vor. Dabei müsste anstelle der A -Beschriftung natürlich das entsprechende X auftreten. Pumpen von $LINK$ auf $LINK^k$ bedeutet somit in diesem speziellen Falle nichts anderes als wiederholtes Anwenden dieser Produktion. Das Ergebnis also $FIRST \otimes LINK^k \otimes LAST$ ist in Abbildung 10 skizziert.

Wie bei kontextfreien Zeichenketten-Grammatiken, kann man mit Hilfe des Pumping-Lemmas zeigen, dass das Endlichkeitsproblem für Hyperkanten-Ersetzungs-Sprache entscheidbar ist. Ein weiteres nützliches Ergebnis des Pumping-Lemmas ist das lineare Wachstum unendlicher Hyperkanten-Ersetzungs-Sprachen.

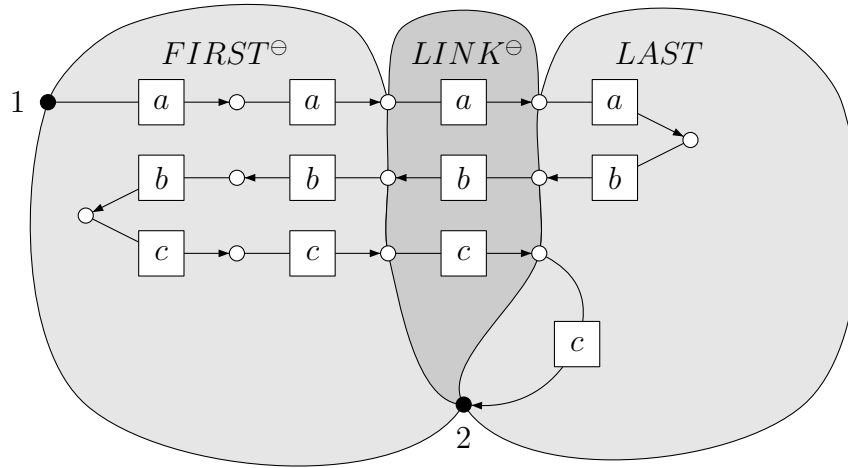
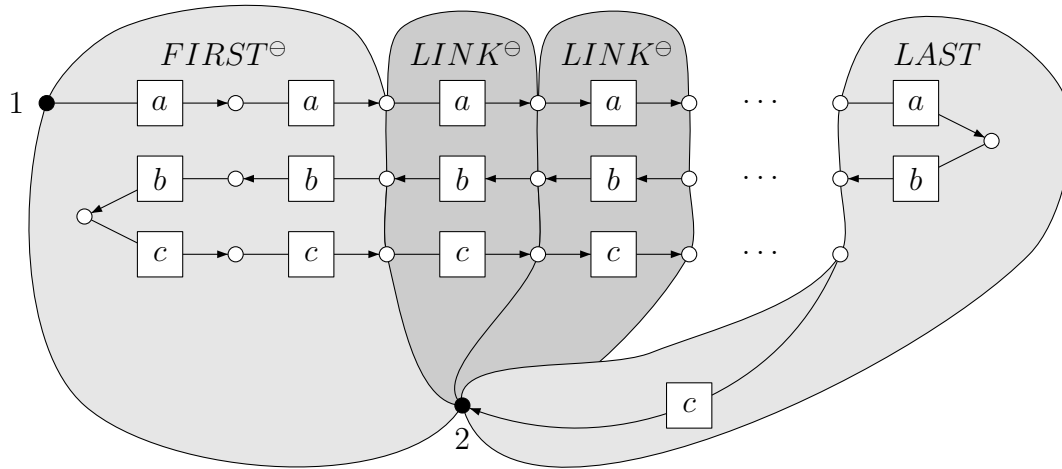
Abbildung 9: Zerlegung von $(a^4b^4c^4)^\bullet$ in $FIRST \otimes LINK \otimes LAST$ 

Abbildung 10: Beispiel: Hypergraph nach 'Pumpen' des mittleren Teil-Hypergraphens

Dies folgt daraus, dass beim Pumpen eines Hypergraphs der Sprache bei jedem Pump-Schritt die Anzahl von Knoten und Hyperkanten um einen konstanten Wert steigt.

Mit Hilfe dieser Eigenschaft lässt sich zum Beispiel leicht erkennen, dass die Zeichenketten-Graph-Sprache $\{(a^{n^2})^\bullet \mid n \geq 1\}$ nicht von einer Hyperkanten-Ersetzungs-Grammatik erzeugt werden kann. Die Anzahl der Kanten in den Graphen dieser Sprache wächst nicht linear beim Übergang von n nach $n + 1$.

Zum Schluss soll noch exemplarisch gezeigt werden, dass es keine Hyperkanten-Ersetzungs-Sprache mit unbeschränkter Konnektivität gibt. Die Konnektivität eines Hypergraphen ist definiert als die minimale Anzahl von Knoten, die entfernt werden müssen, um zwei nicht-verbundene Teil-Hypergraphen zu erzeugen.

Angenommen es gäbe eine Grammatik der Ordnung r , die die geforderte Sprache mit unbeschränkter Konnektivität erzeugt. Dann würde ein ausreichend großer Graph mit Konnektivität größer als r nach dem Pumping-Lemma zerlegt werden können zu $FIRST \otimes LINK \otimes LAST$ mit $|LINK \otimes LAST| \leq q$ für ein bestimmtes q und $type(LINK) \leq r$, wobei $LINK$ mindestens einen internen Knoten besitzt. Da q aber fest ist, können wir annehmen, dass $FIRST$ mehr als r Knoten besitzt. Also hat $FIRST \otimes LINK \otimes LAST$ höchstens die Konnektivität r , denn durch Entfernen der externen Knoten von $LINK$ aus dem Gesamt-Hypergraphen würden auf jeden Fall mindestens zwei Teil-Hypergraphen entstehen.

3.4 Parikhs Theorem

Parikhs Theorem stellt eine Beziehung zwischen Hyperkanten-Ersetzungs-Sprachen und semilinearen Mengen her. Im folgenden betrachten wir n -Tupel $x = (x_1, \dots, x_n)$, $y = (y_1, \dots, y_n)$ in $\mathbb{N}^n$ und definieren:

- $x + y = (x_1 + y_1, \dots, x_n + y_n)$
- $x - y = (x_1 - y_1, \dots, x_n - y_n)$
- $cx = (cx_1, \dots, cx_n)$ für ein $c \in \mathbb{N}$

Definition (Parikh-Abbildung und semilineare Menge). Sei $T = \{a_1, \dots, a_n\}$ ein Alphabet und $\psi : \mathcal{H}_T \rightarrow \mathbb{N}^n$ eine Abbildung mit $\psi(H) = (\#_{a_1}(H), \dots, \#_{a_n}(H))$, wobei $\#_{a_i}(H)$ die Anzahl der mit a_i -bezeichneten Hyperkanten in $H \in \mathcal{H}_T$ ist. ψ heißt dann *Parikh-Abbildung*. Für eine Sprache $L \subseteq \mathcal{H}_T$ bezeichnet $\psi(L)$ die Menge $\psi(L) = \{\psi(H) \mid H \in L\}$.

Eine Menge $S \subseteq \mathbb{N}^n$ ist *linear*, wenn S von der Form

$$S = \{x_0 + \sum_{i=1}^k c_i x_i \mid c_1, \dots, c_k \in \mathbb{N}\}$$

ist, mit $k \geq 1$ und $x_1, \dots, x_k \in \mathbb{N}^n$.

S heißt *semilinear*, wenn S die Vereinigung einer endlichen Anzahl linearer Mengen ist.

Theorem 3.4.1 (Parikhs Theorem). Sei L eine Hyperkanten-Ersetzungs-Sprache und ψ eine Parikh-Abbildung. Dann ist die Menge $\psi(L)$ semilinear.

Der Beweis soll hier nicht durchgeführt werden. Er verwendet Parikhs Theorem für kontextfreie Zeichenketten-Sprachen, indem er Hyperkanten-Ersetzungs-Grammatiken auf kontextfreie Zeichenketten-Grammatiken abbildet.

Anmerkungen:

1. Eine interessante Feststellung ist, dass man eine 1-Kante mit einer bestimmten Beschriftung an jeden internen Knoten der rechten Seite einer Produktion anhängen kann, so dass ihre Anzahl mit der der internen Knoten übereinstimmt. Also liefert das Zählen der internen Knoten eines Hypergraphen einer Hyperkanten-Ersetzungs-Sprache ebenfalls eine semilineare Menge.
2. Es gibt einige Sprachen L , die die Semilinearitäts-Eigenschaft besitzen aber keine Hyperkanten-Ersetzungs-Sprachen sind. Beispiel: Die Menge aller Graphen.

4 Generative Mächtigkeit, Hierarchie-Theoreme

Da nach Definition ein Typ- k -Hypergraph nur mit Hilfe einer Hyperkanten-Ersetzungs-Grammatik vom Typ $k' \geq k$ erzeugt werden kann, ist ersichtlich, dass

$\mathcal{HRL}_0 \subsetneq \mathcal{HRL}_1 \subsetneq \mathcal{HRL}_2 \subsetneq \dots$ gilt. Dieses Kapitel beschäftigt sich mit Graphen und Zeichenketten-Graphen, die von Hyperkanten-Ersetzungs-Grammatiken erzeugt werden können.

4.1 Graphen

Sei $\mathcal{L}_{GRAPH}$ die Klasse aller Graph-Sprachen, und sei $\mathcal{GL}_k = \mathcal{HRL}_k \cap \mathcal{L}_{GRAPH}$. Das bedeutet, $\mathcal{GL}_k$ ist die Menge aller Graph-Sprachen die durch Hyperkanten-Ersetzungs-Grammatiken der Ordnung k erzeugt werden.

Theorem 4.1.1 (Hierarchie-Theorem 1). $\mathcal{GL}_k \subsetneq \mathcal{GL}_{k+1}$ für alle $k \in \mathbb{N}$.

Um die durch dieses Theorem beschriebene Klassen-Hierarchie zu veranschaulichen, sei das Beispiel der k -Bäume und partiellen k -Bäume betrachtet. Zu diesem Zweck werden zunächst einige Begriffe aus der Graphentheorie eingeführt:

- Zwei Knoten heissen *adjazent* wenn sie durch eine Kante verbunden sind.
- Eine k -Clique eines Graphen ist eine Teilmenge von k paarweise adjazenten Knoten des Graphen.
- Für $k \geq 1$ ist die Menge $kTREE$ der k -Bäume rekursiv definiert durch:
 - Jeder vollständige, ungerichtete Graph mit k Knoten ist ein k -Baum mit k Knoten.
 - Sei H ein k -Baum mit n Knoten und $G \subseteq H$ eine k -Clique. Wird nun ein zusätzlicher Knoten mit jedem Knoten aus G verbunden, so erhält man einen neuen k -Baum mit $n + 1$ Knoten.
- Entfernt man von einem k -Baum eine beliebige Anzahl von Kanten, erhält man einen *partiellen k -Baum*. Die Menge aller sich so ergebender Graphen wird durch $kTREE^P$ bezeichnet.

Für beliebige $k \in \mathbb{N}$ kann man sehr einfach eine entsprechende Hyperkanten-Ersetzungs-Grammatik HRG_{kTREE} der Ordnung k konstruieren, die die Menge aller k -Bäume erzeugt. Die Konstruktion, für die man unabhängig von k immer zwei Produktionen benötigt, orientiert sich direkt an der rekursiven Definition der k -Bäume. Dabei repräsentiert eine Hyperkante jeweils eine k -Clique im Graphen. Die erste Produktion verbindet somit einfach die k Verbindungsknoten der Hyperkante untereinander durch gewöhnliche Kanten (bzw. 2-Hyperkanten). Die zweite Produktion fügt in einen k -Baum einen weiteren Knoten ein, indem sie zum einen die alte k -Clique durch eine Hyperkante bestehen lässt und zusätzlich k weitere Hyperkanten einfügt, die die zukünftigen k -Cliquen mit dem neuen Knoten repräsentieren.

Beispiel. Für den Fall $k = 3$ sind die gerade beschriebenen Produktionen in Abbildung 11 dargestellt. Der interne Knoten von Produktion 1 ist der durch diese Produktion neu zum 3-Baum hinzugefügte Knoten. Die alte 3-Clique wird durch die als oberstes dargestellte Hyperkante dargestellt. Sie ist mit den externen Knoten 1, 2 und 3 verbunden.

Folglich kann man die Sprache $kTREE$ mit Hilfe einer Hyperkanten-Ersetzungs-Grammatik k -ter Ordnung erzeugen. Dass dies nicht mit einer Hyperkanten-Ersetzungs-Grammatik kleinerer Ordnung geht, zeigt man wie folgt: Angenommen es gäbe eine Hyperkanten-Ersetzungs-Grammatik $(k - 1)$ -ter Ordnung, die $kTREE$ erzeugt, dann hätten nach dem Pumping Lemma alle bis auf endlich viele Elemente in $kTREE$ eine Konnektivität kleiner als $k - 1$. Dies ist ein Widerspruch zu der Tatsache, dass jeder k -Baum mit mehr als k Knoten die Konnektivität k besitzt, und hiervon gibt es unendlich viele.

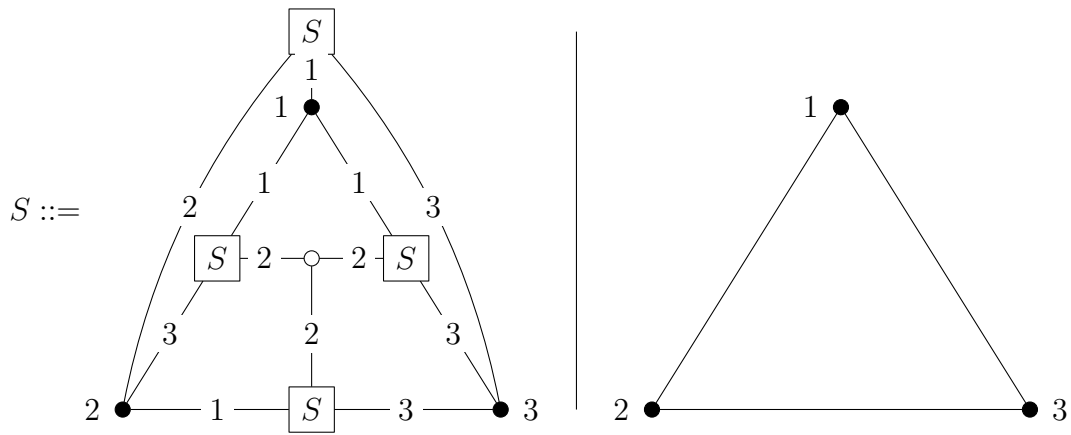


Abbildung 11: Produktionsregeln einer 3-Bäume-erzeugenden Grammatik

Auf gleiche Weise zeigt man auch, dass die Menge $kTREE^P$ der partiellen k -Bäume durch eine Hyperkanten-Ersetzungs-Grammatik k -ter Ordnung aber nicht durch eine der Ordnung $k - 1$ erzeugt werden kann. Hierzu stelle man sich die oben skizzierte Hyperkanten-Ersetzungs-Grammatik HRG_{kTREE} ergänzt durch eine Produktion, die eine Kante entfernt, vor.

4.2 Zeichenketten-Graphen

Analog zur Menge $\mathcal{GL}_k$, die im vorhergehenden Kapitel eingeführt wurde, definieren wir nun $\mathcal{SL}_k$ als die Menge aller von Hyperkanten-Ersetzungs-Grammatiken erzeugten Sprachen, die ausschließlich aus Zeichenketten-Graphen bestehen. Für diese Klassen von Hyperkanten-Ersetzungs-Sprachen gilt jedoch ein etwas anderes Hierarchie-Theorem:

Theorem 4.2.1 (Hierarchie-Theorem 2). $\mathcal{SL}_{2k} \subsetneq \mathcal{SL}_{2(k+1)}$ für alle $k \in \mathbb{N}$.

Ziel dieses Kapitels, ist es diese Hierarchie zu beweisen, und gleichzeitig zu zeigen, dass für die dabei nicht betrachteten Mengen $\mathcal{SL}_{2k+1}$ gilt $\mathcal{SL}_{2k+1} = \mathcal{SL}_{2k}$.

Einführend verdeutliche man sich den Fall $k = 2$. Dabei ist zu einer gegebenen Zeichenketten-Sprache L ihre entsprechende Zeichenketten-Graph-Sprache $L^\bullet$ genau dann in $\mathcal{SL}_2$, wenn L kontextfrei ist. Um dies zu zeigen, mache man sich klar, dass man jede Produktion (A, w) der kontextfreien Grammatik, die L erzeugt, einfach in eine entsprechende Produktion $(A, w^\bullet)$ verwandeln kann, in dem man zu w den entsprechenden Zeichenketten-Graph $w^\bullet$ konstruiert. Dies geschieht auf dem bereits in Kapitel 2 beschriebenen Weg.

Für die andere Richtung des Beweises entfernt man aus der $L^\bullet$ -erzeugenden Hyperkanten-Ersetzungs-Grammatik alle Produktionen, die Kanten löschen. Danach muss jede rechte Seite einer Produktion, die in einer Ableitung angewendet wird, die als Endergebnis einen Zeichenketten-Graphen liefert, selbst auch ein Zeichenketten-Graph sein. Solche Produktionen können wiederum leicht in Zeichenketten-Produktionen umgewandelt werden, was dann die L erzeugende kontextfreie Grammatik liefert.

Anhand des Beispiels der Zeichenketten-Graph-Sprache $L_k = \{(a_1^n \dots a_{2k}^n)^\bullet \mid n \in \mathbb{N}\}$ soll die Klassenhierarchie der Zeichenketten-Graph-Sprachen bewiesen werden, indem gezeigt wird, dass L_k zwar durch eine Hyperkanten-Ersetzungs-Grammatik der Ordnung $2k$, nicht jedoch durch eine der Ordnung $(2k - 1)$ erzeugt werden kann. Man beachte, dass sich die hier definierten Sprachen L_k von den in vorhergehenden Beispielen verwendeten Sprachen L_m dahingehend unterscheiden, dass hier nur Sprachen mit gerader Anzahl an aufeinanderfolgenden a_i -Ketten betrachtet werden. Es gilt also $L_k = L_{2m}$. Zur Veranschaulichung betrachte man das Beispiel im Kapitel zum Pumping Lemma, in dem die Sprache $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ mit Hilfe einer Hyperkanten-Ersetzungs-Grammatik 4-ter Ordnung erzeugt wurde. Indem man die Produktionen dieser Grammatik wie in Abbildung 12 gezeigt so modifiziert, dass sie eine vierte Reihe mit d -beschrifteten Kanten erzeugen, erhält man eine Grammatik, welche ebenfalls die Ordnung 4 besitzt und $\{(a^n b^n c^n d^n)^\bullet \mid n \in \mathbb{N}\}$ also L_2 erzeugt.

Ebenso lassen sich die Sprachen $\{(a^n b^n c^n d^n e^n)^\bullet \mid n \in \mathbb{N}\}$ und $\{(a^n b^n c^n d^n e^n f^n)^\bullet \mid n \in \mathbb{N}\}$

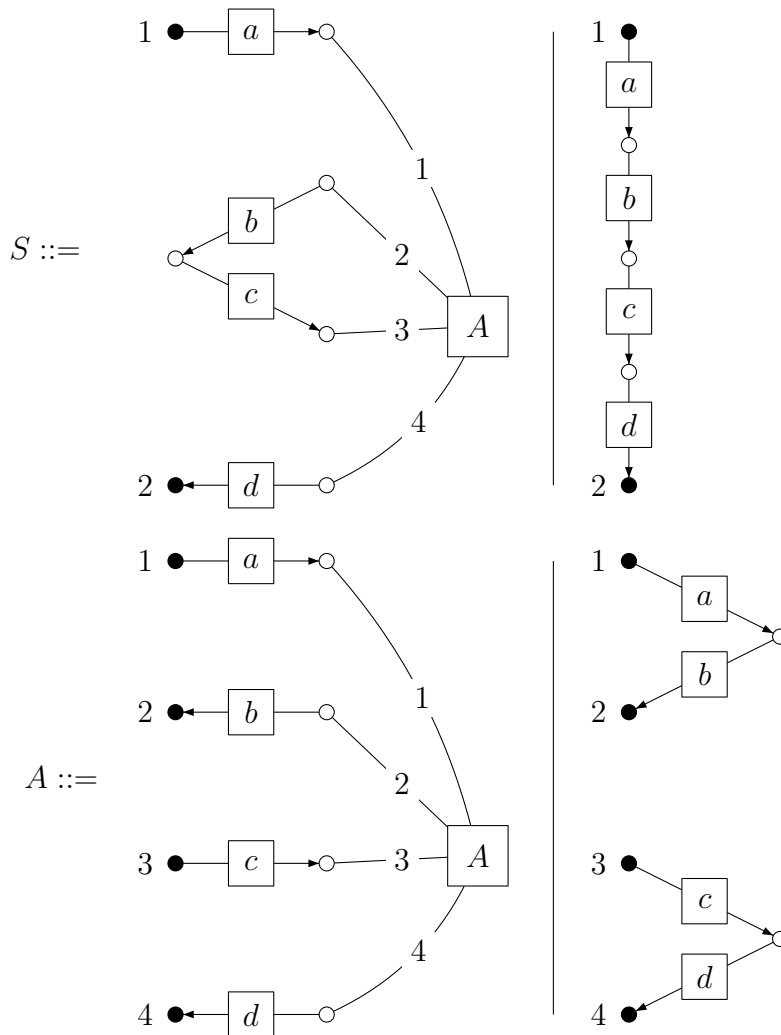


Abbildung 12: Produktionen zur Erzeugung von $(a^n b^n c^n d^n)^\bullet$

$n \in \mathbb{N}\} = L_3$ beide nur durch Hyperkanten-Ersetzungs-Grammatiken mindestens 5-ter Ordnung erzeugen, es ist also überraschenderweise nur jeder zweite Schritt mit einer Erhöhung der Ordnung verbunden.

Allgemein ist L_k also eine Zeichenketten-Graph-Sprache $2k$ -ter Ordnung.

Beweis. Gäbe es eine Hyperkanten-Ersetzungs-Grammatik $(2k - 1)$ -ter Ordnung, die L_k erzeugt, dann müsste ein ausreichend großer Zeichenketten-Graph in L_k nach dem Pumping Lemma zerlegt werden können in die Teil-Graphen *FIRST*, *LINK* und *LAST*. Hierbei wäre *LINK* vom Typ kleiner als $2k$ und substantiell, was hier bedeutet, dass $E_{LINK} \neq \emptyset$, denn in Zeichenketten-Graphen ist die Anzahl der Knoten immer um eins größer als die Anzahl der Kanten.

$FIRST \otimes LINK \otimes LAST$ ist ebenfalls ein Zeichenketten-Graph. Betrachtet man *LINK* ohne seinen X -Henkel e , also $LINK^\ominus$, so bleiben einzelne Pfade übrig, die nicht miteinander verbunden sind, und deren Enden in $[ext_{LINK}] \cup [att_{LINK}(e)]$ liegen. Die Anzahl dieser einzelnen Pfade ist also höchstens $k - 1$ und zudem kann keiner von ihnen Kanten mit unterschiedlichen Beschriftungen enthalten, da sonst das Pumpen von *LINK* Graphen erzeugen würde, die nicht aus L_k sind. Folglich kommt eine Kantenbeschriftung in $LINK^\ominus$ nicht vor. Da $E_{LINK} \neq \emptyset$ gilt, folgt aus $FIRST \otimes LINK \otimes LAST \in L_k$ also $FIRST \otimes LAST \notin L_k$, was ein Widerspruch zum Pumping Lemma ist. $\square$

Zur Beantwortung der Frage, warum nicht gilt $\mathcal{SL}_k \subsetneq \mathcal{SL}_{k+1}$, werden wir im folgenden *deterministische baumtraversierende Umsetzer* (*deterministic tree-walking transducer* = DTWT) zu Hilfe ziehen. Diese von Aho und Ullman eingeführte Klasse endlicher Automaten arbeitet auf regulären Mengen von Termen, die hier als Mengen von geordneten Bäumen über einem Rang-Alphabet betrachtet werden.

Baum-Grammatiken und DTWTs Bevor wir DTWTs einführen können, müssen zunächst reguläre Baum-Grammatiken erklärt werden: Eine *reguläre Baum-Grammatik* $R = (\Sigma, \Gamma, \Psi, \gamma_0)$ erzeugt eine reguläre Termmenge. Hierbei ist Σ das Rang-Alphabet, Γ eine Menge von Nicht-Terminalsymbolen, $\gamma_0 \in \Gamma$ das Startsymbol und Ψ eine endliche Menge von Produktionen der Form $\gamma \rightarrow f(\gamma_1, \dots, \gamma_n)$ für $\gamma, \gamma_1, \dots, \gamma_n \in \Gamma$ und $f \in \Sigma$ mit dem Rang n . Die von R erzeugte Sprache $L(R)$ enthält alle Terme über Σ , die von γ_0 abgeleitet werden können, indem man die Produktionen als Termersetzungsregeln auffasst. Sei t ein Term, dann ist die Menge der Vorkommen von t über Σ wie folgt definiert: $occ(t) = \{\lambda\} \cup \{io \mid 1 \leq i \leq n, o \in occ(t_i)\}$. $o(t)$ bezeichnet das Symbol an der Stelle des Vorkommens $o \in occ(t)$ und t/o den Teilterm von t , dessen Wurzel das Vorkommen o in t ist. Das Stammvorkommen von $oi(i \in \mathbb{N})$ ist o und das von λ ist $\perp$.

Definition. Ein *deterministischer baumtraversierender Umsetzer* (DTWT) ist ein Tupel $M = (Q, R, \Delta, \delta, q_0, F)$, wobei Q eine Zustandsmenge mit Startzustand $q_0 \in Q$ und Endzustandsmenge $F \subseteq Q$ ist. $R = (\Sigma, \Gamma, \Psi, \gamma_0)$ ist eine reguläre Baum-Grammatik, Δ das Ausgabe-Alphabet und $\delta : Q \times \Sigma \rightarrow Q \times D \times \Delta^*$ die Übergangsfunktion mit $D = \{stay, up\} \cup \{down(i) \mid i \geq 1\}$.

Eine *Konfiguration* von M ist ein Tupel (q, t, o, w) mit $q \in Q, t \in L(R), o \in occ(t) \cup \{\perp\}$ und $w \in \Delta^*$. Ein *Schritt* von M lässt den DTWT von der Konfiguration (q, t, o, w) nach (q', t, o', ww') übergehen, wenn $\delta(q, o(t)) = (q', d, w')$ und eine der folgenden Bedingungen gilt:

- $d = up$ und o' ist das Stammvorkommen von o , oder
- $d = stay$ und $o' = o$, oder
- $o(t)$ ist vom Rang $\geq i$, $d = down(i)$ und $o' = oi$.

Startet M mit einer Konfiguration $(q_0, t, \lambda, \lambda)$ und erreicht schließlich eine Endkonfiguration $(q, t, \perp, w)$ mit $q \in F$, so ist die Berechnung erfolgreich, und w wird als Ausgabe geliefert. $L(M)$ ist die Sprache aller Ausgaben von M und $\mathcal{DTWT}^+$ die Klasse aller DTWTs M mit $\lambda \notin L(M)$.

Eine Eigenschaft von DTWTs, die wir für unseren Zweck heranziehen werden ist die sogenannte *Kreuzungszahl*. Wenn o' das Stammvorkommen von o ist, dann ist eine Kreuzung die Überquerung der Kante zwischen o und o' während einem erfolgreichen Lauf von M auf t an. Wählt man die Kreuzungszahl k so, dass bei allen möglichen erfolgreichen Läufen eine Kante höchstens $2k$ -mal gekreuzt wird, so nennt man M *k-kreuzend*. Man beachte, dass man für jeden DTWT solch ein k bestimmen kann, denn zumindest die Anzahl der Zustände $|Q|$ stellt eine Schranke für $2k$ dar, denn wird eine Kante mehr als $|Q|$ -mal gekreuzt, bedeutet dies, dass die laufende Berechnung zweimal im gleichen Zustand war, und deshalb nie enden wird, also keinen erfolgreichen Lauf darstellt. Die Klasse der k -kreuzenden DTWTs wird im folgenden mit $\mathcal{DTWT}_k^+$ bezeichnet. Mit den eingeführten DTWTs kann man folgendes Lemma beweisen.

Lemma 4.2.1. $\mathcal{SL}_{2k+1} \subseteq L(\mathcal{DTWT}_k^+)$ für alle $k \in \mathbb{N}$.

Beweis. Der Beweis vollzieht sich auf folgendem, hier nur in groben Zügen dargelegtem Wege. Zunächst wird gezeigt, dass eine Zeichenketten-Graphen erzeugende Hyperkanten-Ersetzungs-Grammatik $HRG = (N, T, P, S)$ ohne ihre Ordnung zu verändern so modifiziert werden kann, dass sie folgende beiden Eigenschaften aufweist:

1. Für jedes Nichtterminalsymbol $A \in N$ kann man eine 0,1-Folge $out_A \in \{0, 1\}^{type(A)}$ bestimmen, die angibt, wie groß der Aussengrad der externen Knoten jedes aus A ableitbaren Hypergraphens ist. Der Aussengrad eines Knotens ist definiert als die Anzahl von ihm auslaufender Kanten. Für $i = 1, \dots, type(A)$ ist der Aussengrad des i -ten externen Knotens $ext_H(i)$ eines aus A abgeleiteten Hypergraphens H dann $out_A(i)$, also entweder 0 oder 1.
2. In keinem Hypergraphen, der auf einer rechten Seite einer Produktion aus P auftritt, kommen zwei gleich beschriftete Hyperkanten vor.

Die entsprechenden Modifikationen werden im folgenden verwendet, um einen DTWT $M = (Q, R, T, \delta, q_0, F)$ mit $M \in \mathcal{DTWT}_k^+$ zu konstruieren, der die von HRG erzeugte Sprache akzeptiert, also $L(M) = L(HRG)$.

Dabei wird M folgendermassen definiert: Die Baumgrammatik R wird, vereinfacht dargestellt, so konstruiert, dass mit ihr die Ableitungsbäume von HRG aufgebaut werden können. Dazu werden die Nichtterminalsymbole N inklusive Startsymbol S wie in HRG gewählt. Die Menge der Produktionen P wird zum Rangalphabet, indem jeder Produktion $p = (A, H) \in P$ der Rang $|E_H^N|$, also die Anzahl nicht-terminaler Hyperkanten im Hypergraphen der rechten Seite, zugeordnet wird. Zudem werden eben diese Hyperkanten auf beliebige Weise als geordnet festgelegt, so dass man die neuen Produktionen von R als Termersetzungsregel $A \rightarrow p(lab_H(e_1), \dots, lab_H(e_n))$ betrachten kann. Damit ist R vollständig bestimmt. Nun soll der DTWT M auf den von R erzeugten Ableitungsbäumen t laufen. Jeder

Knoten der in einer rechten Seite einer Produktion aus P vorkommt, ist ein Zustand von M . Befindet sich M am Vorkommen o von $t \in L(R)$ und $o(t) = (A, H)$, dann ist der aktuelle Zustand ein Knoten v innerhalb von H . M arbeitet so, dass das Vorkommen in t gesucht wird, das die terminale Kante e generiert, deren erster angeknüpfter Knoten im generierten Zeichenketten-Graph das Bild von v ist. Dabei gibt es drei Möglichkeiten:

- Die Suche ist erfolgreich, d.h. eine Kante $e \in E_H^T$ existiert mit $att_H(e, 1) = v$. M geht dann in den Zustand über, der dem zweiten Anknüpfungs-Knoten von e , also $att_H(e, 2)$ entspricht, produziert die Ausgabe $lab_H(e)$ und verbleibt am gleichen Vorkommen in t .
- Eine solche gesuchte Kante existiert nicht. Stattdessen wird eine nichtterminale Kante e_i gefunden, deren j -ter Verknüpfungs-Knoten v ist. Gilt zudem $out_{lab_H(e_i)}(j) = 1$, dann läuft M mit einer $down(i)$ -Bewegung in den i -ten Teilbaum und befindet sich dann im Zustand ext_H , wobei H die rechte Seite des neuen Vorkommens ist.
- Wenn keine der beiden anderen Möglichkeiten zutrifft, bedeutet dies, dass die gesuchte Kante nicht in diesem Teilbaum erzeugt wurde. Da ein Zeichenketten-Graph erzeugt wurde, bedeutet dies, dass v ein externer Knoten $ext_H(i)$ von H sein muss. M vollzieht nun eine up -Bewegung. Ist H das dabei erreichte Stammvorkommen, geht M in den Zustand $att_H(e, i)$ über, wobei e die einzige Hyperkante in H ist mit der Beschriftung A .

Um die Aufwärts- und Abwärts-Bewegungen korrekt ausführen zu können, benötigen wir einige Hilfs-Zustände, die sich i , bzw. (A, i) merken können, da wir H zunächst nicht kennen.

Da die Zustände, mit denen M die Vorkommen eines Eingabe-Baums betritt und wieder verlässt, externen Knoten von rechten Seiten sind und keiner dieser Knoten zweimal benutzt wird, muss M k -kreuzend sein. □

Folgendes Lemma hat ebenfalls Gültigkeit, was hier jedoch nicht bewiesen werden soll.

Lemma 4.2.2. $L(DTWT_k^+) \subseteq \mathcal{SL}_{2k}$ für alle $k \in \mathbb{N}$.

Beweis. Siehe [rozenberg97]. □

Somit haben wir zusammenfassend bislang die folgenden Erkenntnisse erzielt:

- $\mathcal{SL}_{2k+1} \subseteq L(DTWT_k^+) \subseteq \mathcal{SL}_{2k}$ und
- $\mathcal{SL}_{2k} \subsetneq \mathcal{SL}_{2(k+1)}$.

Es ergibt sich also, dass die generative Mächtigkeit von HRGs tatsächlich nur in jedem zweiten Schritt steigt. Mit HRGs $(2k + 1)$ -ter Ordnung kann man keine Sprache erzeugen, die man nicht auch mit einer HRG $2k$ -ter Ordnung erzeugen kann, was in folgendem Theorem zum Ausdruck gebracht ist.

Theorem 4.2.2. $\mathcal{SL}_{2k} = L(DTWT_k^+) = \mathcal{SL}_{2k+1}$ für alle $k \in \mathbb{N}$.

Da wie oben schon bemerkt die k -kreuzend-Eigenschaft auf jeden DTWT für ein bestimmtes k zutrifft, ist folgende Generalisierung gültig, die die Klasse der von HRGs erzeugten Zeichenketten-Sprachen charakterisiert als genau die Klasse von Sprachen, die durch DTWTs erzeugt werden können.

Theorem 4.2.3. $\mathcal{SL} = L(DTWT^+)$.

5 Entscheidungsprobleme

Durch eine Hyperkanten-Ersetzungs-Grammatik wird, wie bereits gesehen, eine Hypergraphen-Sprache spezifiziert. Selbstverständlich ist man daran interessiert, Aussagen über eine solche Sprache treffen zu können, bzw. gewisse Eigenschaften zu beweisen. Abhängig von der gegebenen Grammatik kann jedoch eine Aufzählung aller Wörter (Hypergraphen) dieser Sprache durch deren Erzeugung mittels Ableitung sehr aufwändig oder gar unmöglich sein, da die Menge der Wörter einer Hyperkanten-Ersetzungs-Sprache nicht endlich sein muss. Daher ist man bemüht, Beweise auf Basis der (laut Definition endlichen!) Menge von Produktionsregeln P einer Hyperkanten-Ersetzungs-Grammatik zu führen.

Die Grundlage hierfür stellt das Kontext-Freiheits-Lemma und die dort gezeigte Zerlegung einer Ableitung dar: Sei eine Grammatik HRG gegeben, und $H \in L(HRG)$ ein terminaler Hypergraph der von HRG erzeugten Sprache, ableitbar mittels $A^\bullet \Rightarrow R \Rightarrow^* H$, so erhalten wir eine Zerlegung von H in kleinere Komponenten, welche ihrerseits von den Henkeln der Hyperkanten in R ableitbar sein müssen. Mit diesem Ansatz kann man sowohl graph-theoretische Eigenschaften (*Kompatible Eigenschaften*), als auch numerische Funktionen auf Hypergraphen (*Kompatible Funktionen*) beweisen.

5.1 Kompatible Eigenschaften

Wenn eine graph-theoretische Eigenschaft, wie oben angedeutet, für jedes H getestet werden kann, indem man diese Eigenschaft (bzw. zugehörige Eigenschaften, die sich im Rahmen eines Beweises entsprechend kombinieren lassen) für alle Komponenten von H testet, so nennt man diese Eigenschaft *kompatibel*. Für derartige Eigenschaften sind folgende Typen von Fragestellungen entscheidbar:

- Existiert ein Hypergraph mit dieser Eigenschaft in $L(HRG)$?
- Haben alle (evtl. bis auf endliche viele) Hypergraphen diese Eigenschaft?
- Hat nur eine endliche Menge von Hypergraphen diese Eigenschaft?

Diese Fragestellungen lassen sich mit Hilfe des sogenannten Filter-Theorems, welches beweist, dass der Schnitt einer Hyperkanten-Ersetzungs-Sprache mit einer Menge von Hypergraphen, die eine kompatible Eigenschaft erfüllen, wiederum eine Hyperkanten-Ersetzungs-Sprache darstellt, auf das Leerheits- und das Endlichkeits-Problem reduzieren.

Beispiel. Als einführendes Beispiel in kompatible Eigenschaften untersuchen wir die Eigenschaft, dass zwei externe Knoten eines Hypergraphen durch einen Pfad verbunden sind. Sei $H \in \mathcal{H}_C$ ein Hypergraph und $v, v' \in V_H$ zwei Knoten dieses Hypergraphen H . Ein vv' -Pfad (in H) ist definiert als eine alternierende Folge von Knoten und Kanten, $v_0 e_1 v_1 \dots e_n v_n$ mit $v_0 \dots v_n \in V_H$, $e_1 \dots e_n \in E_H$ sowie $v = v_0$ und $v' = v_n$, so dass v_{i-1} und v_i Anknüpfungs-Knoten der Hyperkante e_i (mit

$i = 1, \dots, n$) sind. Wenn ein solcher Pfad existiert, bezeichnet man H als vv' -verbunden.

Sei $HRG = (N, T, P, S) \in \mathcal{HRG}$, und $A^\bullet \Rightarrow R \Rightarrow^* H$ mit $(A, R) \in P$ und $H \in \mathcal{H}_T$ ebenso wie $lab_R(e)^\bullet \Rightarrow^* H(e)$ für $e \in E_R^N$ durch das Kontext-Freiheits-Lemma gegeben. Wir können also annehmen, dass $H = R[repl]$.

Falls H nun vv' -verbunden ist, muss ein Pfad $p = v_0 e_1 v_1 \dots e_n v_n$ existieren, der v mit v' verbindet. Dann ist entweder $e_i \in E_R^T$ oder $e_i \in E_{repl(e)}$ für ein $e \in E_R^N$ ($i = 1, \dots, n$). Ersetzen wir nun für alle $e \in E_R^N$ jeden längsten Unterpfad $v_{i(e)} e_{i(e)+1} \dots v_{j(e)} e_{j(e)}$ von p in $repl(e)$ mit $i(e) < j(e)$ durch $v_{i(e)} e v_{j(e)}$.

Dies ergibt einen vv' -Pfad in R , so dass $repl(e)$ für alle $e \in E_R^N$ auf diesem Pfad $v_{i(e)} v_{j(e)}$ -verbunden ist. Umgekehrt gilt, dass, wenn solche Pfade in R und den Komponenten $repl(e)$ existieren, offensichtlich auch ein vv' -Pfad in H existiert.

Obiges Ergebnis bedeutet, dass sich die vv' -Verbundenheit von H überprüfen lässt, indem man einen vv' -Pfad in R sucht und die zugehörigen Verbundenheits-Eigenschaften für Komponenten von H untersucht. Sind nun v und v' externe Knoten von H , so ergibt dies einen terminierenden rekursiven Algorithmus, da die Ableitungen der Komponenten kürzer sind als die Ableitung von H . Darüber hinaus muss die Verbundenheit nur für eine endliche Anzahl von Paaren natürlicher Zahlen (bzw. dadurch identifizierter Knoten) geprüft werden, denn wenn ein Hypergraph wie hier aus einem Henkel abgeleitet ist, so ist die Anzahl seiner externen Knoten durch die Ordnung der Grammatik nach oben hin beschränkt.

Definition (kompatible Prädikate). Sei $\mathcal{C} \subseteq \mathcal{HRG}$ und I eine (möglicherweise unendliche) Menge von Indizes, so dass eine effektive Prozedur existiert, um zu jedem $HRG = (N, T, P, S) \in \mathcal{C}$ eine endliche Teilmenge I_{HRG} von I zu konstruieren, mit einem ausgezeichneten Index $i(type(S)) \in I_{HRG}$. Sei $PROP'$ ein entscheidbares Prädikat, definiert auf Tripeln (R, ass, i) mit $R \in \mathcal{C}$, der Abbildung $ass : E_R^N \rightarrow I$ und $i \in I$, und sei $PROP(H, i) = PROP'(H, empty, i)$ für $H \in \mathcal{H}_T$ und $i \in I$. Dann ist $PROP(\mathcal{C}, PROP')$ -kompatibel, wenn für jede $HRG = (N, T, P, S) \in \mathcal{C}$, alle Ableitungen $A^\bullet \Rightarrow R \Rightarrow^* H$ mit $H \in \mathcal{H}_T$ und alle $i \in I_{HRG}$ gilt, dass $PROP(H, i)$ genau dann gilt, wenn eine Abbildung $ass : E_R^N \rightarrow I$ existiert, so dass $PROP'(R, ass, i)$ gilt und $PROP(H(e), ass(e))$ für alle $e \in E_R^N$ gilt. Das unäre Prädikat $PROP_0$, welches auf $H \in \mathcal{H}_T$ mit $type(H) = K$ genau dann gilt, wenn $PROP(H, i(K))$ gilt, heißt $\mathcal{C}$ -kompatibel.

Kompatible Prädikate sind unter Booleschen Operationen abgeschlossen.

Wir kommen nun zu dem bereits angekündigten Filter-Theorem, welches besagt, dass die Menge aller Hypergraphen einer Hyperkanten-Ersetzungs-Sprache, welche eine kompatible Eigenschaft erfüllen, wiederum eine Hyperkanten-Ersetzungs-Sprache darstellt.

Theorem 5.1.1 (Filter-Theorem). Sei $PROP_0$ ein $\mathcal{C}$ -kompatibles Prädikat für ein beliebiges $\mathcal{C} \in \mathcal{HRG}$. Zu jeder $HRG \in \mathcal{C}$ existiert eine HRG_{PROP_0} , so dass $L(HRG_{PROP_0}) = \{H \in L(HRG) \mid PROP_0(H)\}$.

Theorem 5.1.2 (Entscheidbarkeit von kompatiblen Eigenschaften). Sei $PROP_0$ in Bezug auf eine beliebige Klasse $\mathcal{C}$ von HRGs $\mathcal{C}$ -kompatibel. Dann sind für alle $HRG \in \mathcal{C}$ folgende Fragestellungen entscheidbar:

1. Gilt $PROP_0$ für mindestens ein $H \in L(HRG)$?
2. Gilt $PROP_0$ für alle $H \in L(HRG)$?

3. Gilt $PROP_0$ für nur endlich viele $H \in L(HRG)$?
4. Gilt $PROP_0$ für fast alle $H \in L(HRG)$?

Beweis. Mit Hilfe des Filter-Theorems können wir zu jeder $HRG \in \mathcal{C}$ sowohl HRG_{PROP_0} als auch $HRG_{\neg PROP_0}$ erzeugen, und somit auch die Mengen $L(HRG_{PROP_0}) = \{H \in L(HRG) \mid PROP_0(H)\}$ und $L(HRG_{\neg PROP_0}) = \{H \in L(HRG) \mid \neg PROP_0(H)\}$ definieren.

Nun gelten die oben aufgeführten Fälle genau dann, wenn folgende Bedingungen erfüllt sind:

1. $L(HRG_{PROP_0})$ ist nicht leer.
2. $L(HRG_{\neg PROP_0})$ ist leer.
3. $L(HRG_{PROP_0})$ ist leer bzw. endlich.
4. $L(HRG_{\neg PROP_0})$ ist leer bzw. endlich.

Das Endlichkeits-Problem einer Hyperkanten-Ersetzungs-Sprache war, wie als Konsequenz aus dem Pumping-Lemma gesehen, entscheidbar. Das Leerheits-Problem ist mit der selben Argumentation wie im Fall von Zeichenketten zu lösen. $\square$

5.2 Kompatible Funktionen

Kompatible Funktionen stellen eine Verallgemeinerung von kompatiblen Prädikaten dar: Eine Funktion auf Hypergraphen heißt kompatibel, wenn sie in der selben Weise berechnet werden kann, wie kompatible Prädikate getestet werden. Die Definition in [rozenberg97] beschränkt sich auf Funktionen, die aus Minima, Maxima, Summen und Produkten natürlicher Zahlen konstruiert sind. Dies induziert kompatible Prädikate der Form: *Der Funktionswert eines Graphen überschreitet (nicht) einen bestimmten konstanten Wert.* Mögliche Betrachtungsaspekte sind z.B. die Größe, der minimale/maximale Grad, die Anzahl von Komponenten, das Verhältnis von Knoten- zu Kantenanzahl oder die Länge des kürzesten Pfades zwischen externen Knoten eines Hypergraphen.

Aufgrund des Umfangs und der vorher nötigen Hilfs-Definitionen soll auf die formale Beschreibung hier verzichtet werden. Doch schon nach der intuitiven Einführung, welche auf kompatible Prädikate zurückverweist, sollte es wenig überraschend sein, dass sich ähnliche Ergebnisse ableiten lassen.

Für eine Hyperkanten-Ersetzungs-Grammatik HRG und eine dazu kompatible Funktion f_0 sind folgende Fragen entscheidbar:

1. Gilt $f_0(H) \bowtie n$ für mindestens ein $H \in L(HRG)$?
2. Gilt $f_0(H) \bowtie n$ für alle $H \in L(HRG)$?
3. Gilt $f_0(H) \bowtie n$ für nur endlich viele $H \in L(HRG)$?
4. Gilt $f_0(H) \bowtie n$ für fast alle $H \in L(HRG)$?

Das Symbol $\bowtie$ steht hierbei für eine der binären Relationen $<, \leq, =, \geq, >$.

Es gilt darüber hinaus sogar folgendes Theorem, das die Grundlage für Existenzbeweise oberer Schranken liefert:

Theorem 5.2.1 (Meta-Theorem für Beschränktheits-Probleme). Sei f_0 eine $(\mathcal{C}, \max, +, \cdot)$ -kompatible Funktion für eine Klasse $\mathcal{C}$ von Hyperkanten-Ersetzungs-Grammatiken. Dann ist für alle $HRG \in \mathcal{C}$ entscheidbar, ob eine natürliche Zahl $n \in \mathbb{N}$ existiert, so dass $f_0(H) \leq n$ für alle $H \in L(HRG)$ erfüllt ist.

Beweis. Siehe [rozenberg97]. $\square$

6 Das Wort-Problem

Das Wort-Problem beschäftigt sich mit einer Frage, welche sich stets im Zusammenhang mit Sprachen und Grammatiken stellt: Ist ein gegebener Hypergraph H in der Sprache $L(HRG)$, die durch eine ebenfalls gegebene Hyperkanten-Ersetzungs-Grammatik HRG erzeugt wird? Eng damit verwandt ist das Parsing-Problem, dessen Ausgabe, sofern $H \in HRG$, ein Ableitungsbaum für H sein soll. Im Folgenden werden wir uns der Übersichtlichkeit wegen auf das Wort-Problem beschränken; eine Erweiterung der Algorithmen um eine Abspeicherung der verfolgten Ableitungen ist jedoch relativ einfach möglich, so dass sich damit auch das Parsing-Problem lösen lässt.

6.1 NP-Vollständigkeit

Ein einfacher Ansatz zur Lösung des Wort-Problems in NP ergibt sich aus der Feststellung, dass zu jeder Hyperkanten-Ersetzungs-Grammatik HRG eine lineare Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ existiert, so dass für alle Hypergraphen H der Größe n gilt: Wenn $H \in HRG$, existiert eine Ableitung von H in HRG mit maximal $f(n)$ Schritten.

Dies lässt sich leicht einsehen, indem man eine Partitionierung der Produktionen von HRG vornimmt: Produktionen mit substantieller rechter Seite enthalten per Definition immer mindestens einen internen Knoten oder eine Hyperkante mehr als der Henkel, den sie ersetzen, somit kann man maximal n solcher Produktionen ausführen, bevor man einen Hypergraphen mit Größe $> n$ erhält. Produktionen, deren rechte Seite einen Henkel darstellt, ändern gewissermaßen nur die Beschriftung einer Hyperkante; man kann maximal jedes Element so oft umbenennen, wie derartige Produktionen vorhanden sind, somit ist auch die Anzahl derartiger Ableitungsschritte linear beschränkt. Produktionen der letzten Gruppe, deren rechte Seite nur externe Knoten enthält, können abgesehen vom Startsymbol, welches durch Addition von 1 kompensiert werden kann, nur im Zusammenhang mit Produktionen der ersten Gruppen auftreten; durch Betrachtung der möglichen Kombinationen nach Ausschluss von Schleifen, die isomorphe Graphen produzieren und hier nicht von Relevanz sind, erhält man einen konstanten Faktor, welcher mit obigem Ergebnis zu multiplizieren ist, und somit insgesamt die geforderte lineare Funktion.

Wir können also mittels eines nichtdeterministischen Algorithmus' eine Ableitung raten und das Resultat dieser Ableitung auf Isomorphie zu H testen.

Offensichtlich wäre ein effizienterer und vor allem deterministischer Algorithmus wünschenswert. Hier jedoch zeigt sich eines der wenigen grundsätzlich unterschiedlichen Ergebnisse der Hyperkanten-Ersetzungs-Grammatiken im Vergleich zu den kontext-freien Chomsky-Grammatiken: Während für letztere ein kubischer Algorithmus zur Entscheidung des Wort-Problems existiert (Cocke, Kasami, Younger), lassen sich mit Hyperkanten-Ersetzungs-Grammatiken auch NP-vollständige Graph-Sprachen erzeugen. Unter der Annahme $P \subsetneq NP$ kann man also davon ausgehen, dass kein in polynomieller Zeit lösbarer deterministischer Algorithmus für das Wortproblem in HRG existiert.

Theorem 6.1.1 (NP-Vollständigkeit 1). Es existiert eine Hyperkanten-Ersetzungs-Grammatik, welche eine NP-vollständige Graph-Sprache vom Grad 2, also

eine Sprache, in der jeder Knoten mit maximal 2 Kanten inzident ist, erzeugen kann.

Beweis. Der in [rozenberg97] durchgeführte Beweis beruht auf einer Reduktion des bekannten NP-vollständigen *Hamilton-Pfad-Problems* auf das Wort-Problem; ebenso ist dies mit dem *3-PARTITION-Problem* gezeigt worden. Wir verweisen an dieser Stelle auf die Literatur. $\square$

Theorem 6.1.2 (NP-Vollständigkeit 2). Es existiert eine Hyperkanten-Ersetzungs-Grammatik, welche eine NP-vollständige Graph-Sprache von verbundenen Graphen erzeugen kann.

Beweis. Siehe ebenfalls [rozenberg97]. $\square$

6.2 Einschränkungen und Polynomielle Algorithmen

Die Beweise der oben aufgeführten Theoreme legen nahe, dass bestimmte strukturelle Eigenschaften von Hypergraphen für die exponentielle Laufzeit verantwortlich sind. Durch zusätzliche Bedingungen an die in einer Grammatik erlaubten Produktionen lassen sich je nach Wahl der Einschränkungen polynomielle Algorithmen variablen Grades oder sogar solche mit garantiert höchstens kubischer Laufzeit aufstellen. Aufgrund der Spezialität und des Umfangs dieser Beweise verweisen wir an dieser Stelle wiederum auf [rozenberg97].

7 Zusammenfassung und Ausblick

Wir verfügen nun über eine formale Definition von Hypergraphen und Hyperkanten-Ersetzungs-Grammatiken und haben einige ihrer grundlegende Eigenschaften untersucht, wobei immer wieder deutliche Parallelen zu kontext-freien Zeichenketten-Grammatiken gefunden wurden; viele wichtige Lemma und Theoreme lassen sich übertragen. Die Überlegungen zur generativen Mächtigkeit bei der Erzeugung von Hypergraphen, gewöhnlichen Graphen und Zeichenketten(-Graphen) geben uns einen Eindruck von Möglichkeiten und Grenzen in der Anwendung. Mit Hilfe der im vorletzten Kapitel definierten kompatiblen Prädikate und Funktionen können wir allein auf Grundlage einer Hyperkanten-Ersetzungs-Grammatik Aussagen über die von ihr erzeugte Sprache treffen, und in der Praxis gewünschte Eigenschaften beweisen. Beim zuletzt untersuchte Wort-Problem stoßen wir mit der NP-Vollständigkeit auf einen wichtigen Unterschied zu kontext-freien Zeichenketten-Grammatiken.

Von dieser Basis ausgehend sind weitere Überlegungen möglich und durchgeführt worden. So wurden andere Formen von Grammatiken auf Hypergraphen (*handle-rewriting*) ebenso definiert wie Anwendungen der Hyperkanten-Ersetzung auf (fraktalen) Bildern, auch verschiedene Erweiterungen zu kontext-sensitiven Formen wurden in entsprechenden Arbeiten vorgestellt. Zur Vermeidung der NP-Komplexität werden auch in [rozenberg97] verschiedene Einschränkungen von Hyperkanten-Ersetzungs-Grammatiken präsentiert. Insgesamt sind also verschiedenste sinnvolle Modifikationen, Erweiterungen und Einschränkungen denkbar, welche jedoch immer auch zu einer aufwändigeren und somit schwerer zu handhabenden Theorie führen. In der praktischen Anwendung ist deshalb sorgfältiges Abwägen angeraten.

Literatur

[rozenberg97] *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 2*; H. Ehrig, H.-J. Kreowski, U. Montanari, G. Rozenberg; World Scientific; ISBN 981-02-4021-X;