Baumgrammatiken mit multilinearer Interpretation

Diplomarbeit

von

Armin Reichert

nach einem Thema von Prof. Dr. Günter Hotz

Fachbereich 14.1, Informatik, Universität des Saarlandes

Hiermit versichere ich an Eides Statt, daß ich diese Arbeit nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Saarbrücken, im Oktober 1991

Einleitung

In der vorliegenden Arbeit untersuchen wir den Zusammenhang zwischen kontextfreien Ersetzungen auf Bäumen und Ersetzungen auf Wörtern.

Es ist eine bekannte Tatsache, daß eine Korrespondenz zwischen *regulären* Ersetzungen auf Bäumen und kontextfreien Ersetzungen auf Wörten besteht [Br69]. Wir verallgemeinern in natürlicher Weise die Ersetzungen auf Bäumen und zeigen, daß damit eine Verallgemeinerung der kontextfreien Ersetzungen auf Wörtern einhergeht, die wir als *gekoppelt-kontextfreie Ersetzungen* bezeichnen.

Wir erhalten eine Hierarchie von Ersetzungssystemen und Sprachklassen, die alle in der Klasse der kontextsensitiven Sprachen enthalten sind und die Klasse der kontextfreien Sprachen umfassen.

Ersetzungssysteme, die in dieser Hierarchie aufeinanderfolgen, unterscheiden sich in ihrer Mächtigkeit nur wenig. Wir benutzen als Kriterium dafür die Fähigkeit zum „Zählen", d.h. die Sprachen $COUNT_k := \{a_1^n \cdots a_k^n \mid n \in \mathbf{N}_0\},\ k \in \mathbf{N}_0$, zu erzeugen, und die Fähigkeit zum „Kopieren", d.h. die Sprachen $COPY_k := \{w \cdot w^k \mid w \in \{a,b\}^*\},\ k \in \mathbf{N}_0$, zu erzeugen.

Unter diesem Gesichtspunkt können kontextfreie Grammatiken „bis 2 zählen" und „nicht kopieren" und *Tree Adjoining Grammars* [Jo85] können „bis 4 zählen" und „einmal kopieren". Die Sprachklassen dieser beiden Formalismen bilden die ersten zwei Stufen unserer Hierarchie. Die k-te Klasse unserer Hierarchie kann „bis $2k+2$ zählen" und „k-mal kopieren".

Solche Sprachklassen finden in jüngster Zeit Interesse, da sie zur Charakterisierung von natürlicher Sprache geeignet erscheinen. Man interessiert sich in diesem Gebiet für Sprachklassen, die etwas mächtiger als kontextfreie Sprachen sind („mildly-context-sensitive languages"). Es lohnt sich auch von daher, die formalen Eigenschaften solcher Sprachen zu untersuchen.

Ich möchte an dieser Stelle Herrn Professor G. Hotz meinen Dank aussprechen für die Vergabe dieses Themas und für die geduldige Betreuung.

Inhaltsverzeichnis

Kapitel 1

Algebraische Theorie formaler Sprachen

1.1 Begriffe aus der Theorie der $\times$–Kategorien

In diesem Abschnitt werden einige grundlegende Begriffe eingeführt und erläutert.

Definition 1 *Wörtermonoid*
Sei $[1..n] := \{i \in \mathbf{N} \mid 1 \leq i \leq n\}$ für $n \in \mathbf{N}_0$, also $[1..0] = \emptyset$. Für eine Menge B heißt

$$B^* := \{w \mid w : [1..n] \to B \text{ Abbildung}, \ n \in \mathbf{N}_0\}$$

die Menge der *Wörter über B*.

Das (eindeutig bestimmte) Element $\varepsilon_B : [1..0] \to B$ heißt das *leere Wort über B* und $B^+ := B^* \setminus \{\varepsilon_B\}$ heißt Menge der *nichtleeren Wörter über B*.

Bezeichnet „$\cdot$“ die *Konkatenation* von Wörtern aus B^*, dann ist $(B^*, \cdot, \varepsilon_B)$ ein freies Monoid, das *Wörtermonoid über B*.

Für ein Wort $w : [1..n] \to B$ heißt $|w| := n$ die *Länge* von w. Bezeichnet B^n die Menge der Wörter der Länge n über B, so gilt $B^m \cdot B^n := \{u \cdot v \mid u \in B^m, \ v \in B^n\} = B^{m+n}$.
Für $B^0 = \{\varepsilon_B\}$ schreiben wir auch $\{\Box\}$.

Wir definieren jetzt einige kategorientheoretische Begriffe, die wir am Beispiel einer Kategorie von Abbildungen illustrieren.

Sei dazu B eine Menge und M die Menge aller Abbildungen $f : B^m \to B^n, \ m, n \in \mathbf{N}_0$. Die Abbildungen $Q, Z : M \to \mathbf{N}_0$ seien definiert durch $Q(f) := m, \ Z(f) := n$.

Die *Komposition* $f \circ g \in M$ von Abbildungen $f, g \in M$ sei die Abbildung

$$f \circ g : B^{Q(g)} \to B^{Z(f)}, \qquad (f \circ g)(x) := f(\,g(x)\,) \quad \forall x \in B^{Q(g)}.$$

Es ist $Q(f \circ g) = Q(g)$, $Z(f \circ g) = Z(f)$ und $f \circ g$ ist definiert gdw. $Z(g) = Q(f)$.

Mit $\mathbf{1}_n, \ n \in \mathbf{N}_0$, sei die identische Abbildung auf B^n bezeichnet. Es gilt:

$(\mathbf{N}_0, M, Q, Z, \circ)$ erfüllt die Axiome einer *Kategorie*.

Definition 2 *Kategorie*
$K = (O, M, Q, Z, \circ)$ heißt *Kategorie*, wenn gilt:

(K1) O und M sind Mengen, $Q, Z : M \to O$ sind Abbildungen,
$\circ : M \times M \rightsquigarrow M$ ist eine (partielle) Operation auf M.

(K2) Für $f, g \in M$ ist $f \circ g \in M$ genau dann definiert, wenn $Q(f) = Z(g)$, und dann gilt $Q(f \circ g) = Q(g)$ und $Z(f \circ g) = Z(f)$.

(K3) Ist für $f, g, h \in M$ der Ausdruck $(f \circ g) \circ h$ definiert, dann ist auch $f \circ (g \circ h)$ definiert, und es gilt $(f \circ g) \circ h = f \circ (g \circ h)$.

(K4) Zu jedem $u \in O$ existiert ein $\mathbf{1}_u \in M$ mit folgenden Eigenschaften:

1. $Q(\mathbf{1}_u) = u = Z(\mathbf{1}_u)$,
2. $f \circ \mathbf{1}_u = f$ für alle $f \in M$ mit $Q(f) = u$,
3. $\mathbf{1}_u \circ g = g$ für alle $g \in M$ mit $Z(g) = u$.

$Obj\,K := O$ heißt Menge der *Objekte* und $Mor\,K := M$ heißt Menge der *Morphismen* der Kategorie K. Die Elemente $\mathbf{1}_u$ heißen *Einheiten* von K.

Wir schreiben für $f \in Mor\,K$ mit $Q(f) = u$ und $Z(f) = v$ auch $f : u \to v$ oder $u \xrightarrow{f} v$.
Für $U, V \subset Mor\,K$ sei $K(U, V) := \{f \in Mor\,K \mid Q(f) \in U,\ Z(f) \in V\}$.

Wir definieren jetzt einen speziellen Kategorietyp, der für die Beschreibung formaler Sprachen eine zentrale Rolle spielt, vgl. [Ho66], [Ben70] oder [Ben75].

Es handelt sich dabei um eine Kategorie mit je einer zusätzlichen (Monoid-)Operation auf der Morphismen- bzw. Objektmenge.

Wir erweitern unsere Beispielkategorie entsprechend.
Sei dazu $\otimes : M \times M \to M$ die *Parallelausführung* von Abbildungen aus M, d.h. für

$$f : B^m \to B^n, \qquad g : B^r \to B^s, \qquad m, n, r, s \in \mathbf{N}_0,$$

sei

$$f \otimes g : B^{m+r} \to B^{n+s}, \qquad f \otimes g\,(u \cdot v) := f(u) \cdot g(v) \quad \forall\, u \in B^m, v \in B^r.$$

Es gelten folgende Eigenschaften:

1. $(\mathbf{N}_0, +, 0)$ und $(M, \otimes, \mathbf{1}_0)$ sind Monoide.
2. $Q, Z : M \to \mathbf{N}_0$ sind Monoidhomomorphismen.
3. $(f_1 \otimes f_2) \circ (g_1 \otimes g_2) = (f_1 \circ g_1) \otimes (f_2 \circ g_2) \quad \forall f_i,\ g_i \in M,\ Q(f_i) = Z(g_i),\ i = 1, 2.$

Mit diesen Operationen ist $ABB(B, \mathbf{N}_0) := (\,\mathbf{N}_0, M, Q, Z, \circ, (+, \otimes)\,)$ eine $\times$*-Kategorie.*
Es ist üblich, in der Definition der $\times$–Kategorie die Monoidoperationen von O (hier: $+$) und von M (hier: $\otimes$) mit dem gleichen Symbol $\times$ zu bezeichnen.

Definition 3 $\times$*–Kategorie*
$X = (O, M, Q, Z, \circ, \times)$ heißt $\times$*–Kategorie*, falls $(O, M, Q, Z, \circ)$ eine Kategorie ist und zusätzlich gilt:

(X1) $(O, \times)$ und $(M, \times)$ sind Monoide und Q, Z sind Monoidhomomorphismen bzgl. $\times$.

(X2) Für alle Morphismen $f_i, g_i \in M$ mit $Q(f_i) = Z(g_i),\ i = 1,2$, gilt:

$$(f_1 \times f_2) \circ (g_1 \times g_2) = (f_1 \circ g_1) \times (f_2 \circ g_2).$$

Definition 4 *Unterkategorie, Unter–×–Kategorie*
Seien U, K Kategorien (×–Kategorien) und $Obj\,U \subset Obj\,K,\ Mor\,U \subset Mor\,K$. Stimmen die Operationen beider Kategorien (×–Kategorien) auf U überein, so heißt U *Unterkategorie (Unter–×–Kategorie)* von K und wir schreiben dafür $U \subset K$.

Definition 5 *erzeugte Unterkategorie, erzeugte Unter–×–Kategorie*
Sei K eine Kategorie (×–Kategorie), $E \subset Obj\,K,\ A \subset Mor\,K$. Die kleinste E und A umfassende Unterkategorie (Unter–×–Kategorie) von K heißt *die von* (E, A) *in* K *erzeugte Unterkategorie (Unter–×–Kategorie)* und wird mit $\langle E, A\rangle_K$ bezeichnet.

(E, A) heißt *Erzeugendensystem* von $\langle E, A\rangle_K$. Ist $E = Obj\,K$, dann nennt man auch A Erzeugendensystem von $\langle E, A\rangle_K$.

Lemma 1.1.1 ([Ho74])

$$\langle E, A\rangle_K = \left(\widehat{E_K}, \widehat{A_K}, Q, Z, \circ, \times\right).$$

Dabei ist

$$\begin{aligned} \widehat{E_K} &= \bigcap \{Obj\,U \mid U \subset K,\ E \subset Obj\,U,\ A \subset Mor\,U\}, \\ \widehat{A_K} &= \bigcap \{Mor\,U \mid U \subset K,\ E \subset Obj\,U,\ A \subset Mor\,U\}, \end{aligned}$$

und $Q, Z, \circ, \times$ *sind die Einschränkungen der Abbildungen bzw. Operationen von* K.

Die Morphismenmenge von $\langle E, A\rangle_K$ enthält genau die Morphismen von K, die sich aus den Elementen von A und den Einheiten $\mathbf{1}_e,\ e \in E$ mit den Kategorieoperationen gewinnen lassen.

Wir werden im folgenden oft Abbildungen zwischen Kategorien (×–Kategorien) benutzen, die mit den Kategorieoperationen verträglich sind, sog. *Funktoren*. Allerdings ist das Bild einer Kategorie unter einem Funktor i.a. keine Kategorie.

Definition 6 *kovarianter Funktor*
Sind K_1, K_2 Kategorien, $\phi_1 : Obj\,K_1 \to Obj\,K_2$ und $\phi_2 : Mor\,K_1 \to Mor\,K_2$ Abbildungen, dann heißt $\phi = (\phi_1, \phi_2)$ (*kovarianter*) *Funktor* von K_1 in K_2, wenn gilt:

(F1) $Q(\phi_2(f)) = \phi_1(Q(f))$ und $Z(\phi_2(f)) = \phi_1(Z(f)) \qquad \forall f \in Mor\,K_1$,

(F2) $\phi_2(f \circ g) = \phi_2(f) \circ \phi_2(g) \qquad \forall f, g \in Mor\,K_1$ mit $Q(f) = Z(g)$,

(F3) $\phi_2(\mathbf{1}_u) = \mathbf{1}_{\phi_1(u)} \qquad \forall u \in Obj\,K_1$.

Ein Funktor ist also ein Paar von Abbildungen, die mit der Kategorieverknüpfung sowie den Quelle- und Zielabbildungen verträglich sind und die Richtung der Morphismen beibehalten. Daneben betrachtet man auch Funktoren, die die Richtung der Morphismen umkehren.

Definition 7 *kontravarianter Funktor*
Seien K_1, K_2, ϕ_1, ϕ_2 wie oben. $\phi = (\phi_1, \phi_2)$ heißt *kontravarianter Funktor* von K_1 in K_2, wenn gilt:

(F1') $Q(\phi_2(f)) = \phi_1(Z(f))$ und $Z(\phi_2(f)) = \phi_1(Q(f)) \qquad \forall f \in Mor\, K_1$,

(F2') $\phi_2(f \circ g) = \phi_2(g) \circ \phi_2(f) \qquad \forall f, g \in Mor\, K_1$ mit $Q(f) = Z(g)$,

(F3') $\phi_2(\mathbf{1}_u) = \mathbf{1}_{\phi_1(u)} \qquad \forall u \in Obj\, K_1$.

Die Forderungen (F1) und (F1') sind redundant, denn sie lassen sich aus (F2), (F3) und der Eindeutigkeit der Einheiten folgern.

Definition 8 $\times$*–Funktor*
Seien K_1 und K_2 $\times$–Kategorien. Ein Funktor $\phi = (\phi_1, \phi_2)$ von K_1 in K_2 heißt $\times$*–Funktor*, wenn ϕ_1 und ϕ_2 Monoidhomomorphismen bzgl. $\times$ sind.

Eine $\times$–Kategorie heißt *frei* (*erzeugt*), wenn sich ihre Morphismen in eindeutiger Weise (relativ zu den Axiomen der $\times$–Kategorie) durch die Elemente eines ausgezeichneten Erzeugendensystems darstellen lassen.

Definition 9 *freie* $\times$*–Kategorie*
Eine $\times$–Kategorie X heißt *frei* (*erzeugt*), wenn es eine Teilmenge $A \subset MorX$ mit folgender Eigenschaft gibt:

> Ist K eine beliebige $\times$–Kategorie, $\phi_1 : Obj\, X \to Obj\, K$ ein Monoidhomomorphismus und $\phi'_2 : A \to Mor\, K$ eine Abbildung, die mit Quelle und Ziel verträglich ist, d.h. $\forall f \in A$ gilt
> $$Q(\phi'_2(f)) = \phi_1(Q(f)) \text{ und } Z(\phi'_2(f)) = \phi_1(Z(f)),$$
> dann gibt es eine (eindeutige) Fortsetzung $\phi_2 : Mor\, X \to Mor\, K$ von ϕ'_2 derart, daß $\phi = (\phi_1, \phi_2)$ ein $\times$–Funktor von X in K ist.

Die Menge $A \subset MorX$ heißt in diesem Fall *freies Erzeugendensystem* von X.

Sind Mengen A, V und Abbildungen $Q, Z : A \to V^*$ gegeben, so existiert eine freie $\times$–Kategorie mit freiem Erzeugendensystem A und Objektmonoid V^*. Wir bezeichnen diese freie $\times$–Kategorie mit $\mathcal{F}(A, V, Q, Z)$ oder auch kürzer mit $\mathcal{F}(A, Q, Z)$, $\mathcal{F}(A, V)$ oder $\mathcal{F}(A)$, wenn klar ist, was gemeint ist.

Formale Konstruktionen von $\mathcal{F}(A, V, Q, Z)$ findet man in [Ho65], [Ho66] oder [Ben75].

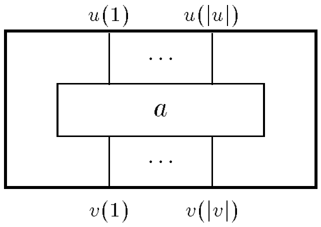
Wir geben einige Eigenschaften von $\mathcal{F}(A, V, Q, Z)$ an (vgl. [HoCl72]):

1. $\mathcal{F}(A, V, Q, Z)$ besitzt genau ein freies Erzeugendensystem, nämlich A.
2. Jeder Morphismus $f \in \mathcal{F}(A, V, Q, Z)$, der keine Einheit ist, besitzt eine *sequentielle Darstellung*
$$f = (\mathbf{1}_{u_m} \times a_m \times \mathbf{1}_{v_m}) \circ \cdots \circ (\mathbf{1}_{u_1} \times a_1 \times \mathbf{1}_{v_1}) \ ,$$
wobei $u_i, v_i \in V^*$ und $a_i \in A$, $1 \leq i \leq m$, $m \in \mathbf{N}$.

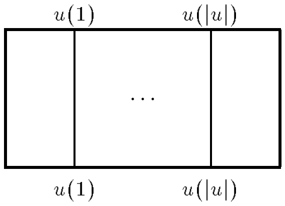
Im allgemeinen ist die sequentielle Darstellung eines Morphismus nicht eindeutig bestimmt. Allerdings kommen in jeder sequentiellen Darstellung die Elemente des freien Erzeugendensystems in der gleichen Anzahl vor, die man als *Länge* des Morphismus bezeichnet.

Die Morphismen einer freien $\times$–Kategorie $\mathcal{F}(A, V, Q, Z)$ lassen sich geometrisch als *Netze* veranschaulichen.

Jedes Element $a \in A$ mit $Q(a) = u$, $Z(a) = v$ wird als „Baustein" mit $|u|$ Eingängen und $|v|$ Ausgängen gedeutet. Die Ein- und Ausgänge sind mit Elementen von V beschriftet. Der i-te Eingang (bzw. Ausgang) trägt die Beschriftung $u(i)$ (bzw. $v(i)$).

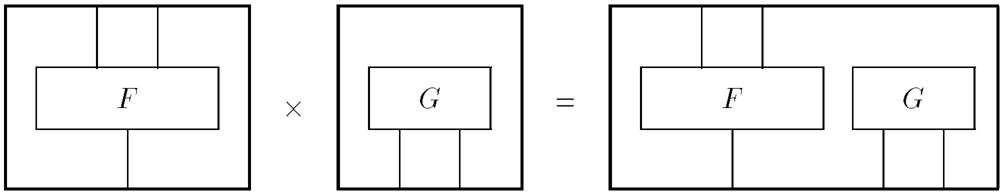


Einer Einheit $\mathbf{1}_u$, $u \in V^*$, entspricht ein „Leitungsbündel“ mit der Ein- und Ausgangsbeschriftung u.



Ausgehend von diesen elementaren Netzen lassen sich mit den Operationen $\circ$ und $\times$ neue Netze zusammensetzen.

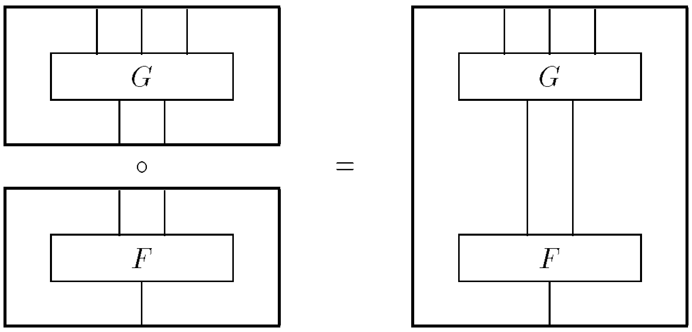
Die Operation $F \times G$ entspricht dem Nebeneinandersetzen der Netze F und G, wobei die Ein- bzw. Ausgangsbeschriftungen konkateniert werden, d.h. $Q(F \times G) = Q(F) \cdot Q(G)$ und $Z(F \times G) = Z(F) \cdot Z(G)$.



$F \times G$

Die partielle Operation $F \circ G$ ist genau dann definiert, wenn die Ausgangsbeschriftung $Z(G)$ gleich der Eingangsbeschriftung $Q(F)$ ist und entspricht dem Übereinandersetzen der Netze F und G.

Die Eingangsbeschriftung $Q(F \circ G)$ des zusammengesetzten Netzes $F \circ G$ ist dann gleich $Q(G)$ und die Ausgangsbeschriftung $Z(F \circ G)$ ist $Z(F)$.



$F \circ G$

Die so konstruierten Netze werden zu Klassen zusammengefaßt, wobei zwei Netze genau dann in der selben Klasse liegen, wenn sie durch Deformationen ineinander überführt werden können, die den Axiomen (X2) bzw. (K4) in der Definition der $\times$–Kategorie entsprechen.

Die Operationen $\circ$ und $\times$ werden auf die Klassen übertragen, die Morphismen der freien $\times$–Kategorie entsprechen genau diesen Klassen von Netzen. Die Netze einer Klasse werden als *unwesentlich verschieden* oder *ähnlich* bezeichnet.

Eine ausführliche Darstellung des geometrischen Aspektes freier $\times$–Kategorien findet man in [Ho65] oder [Ho74].

Wir definieren jetzt einen wichtigen Sonderfall freier $\times$–Kategorien. Er ist dadurch charakterisiert, daß jedes Element des freien Erzeugendensystems genau einen Eingang (bzw. Ausgang) besitzt.

Definition 10 *Baumkategorie, kontextfreie $\times$–Kategorie*

Eine freie $\times$–Kategorie $\mathcal{F}(A, V, Q, Z)$ heißt *Baumkategorie* oder *kontextfreie $\times$–Kategorie*, wenn gilt:

$$|Z(a)| = 1 \qquad \forall a \in A\ .$$

Eine entsprechende Definition gelte für $|Q(a)| = 1 \quad \forall a \in A$.

Ist $\#V = 1$, dann identifizieren wir V^* mit $\mathbf{N}_0$ und fassen Q und Z als Abbildungen von A nach $\mathbf{N}_0$ auf. Für $\mathcal{F}(A, V, Q, Z)$ schreiben wir in diesem Fall $\mathcal{B}(A, Q, Z)$.

Definition 11 *Baum*

Ist $\mathcal{F} := \mathcal{F}(A, V, Q, Z)$ eine Baumkategorie, dann heißt $f \in Mor\,\mathcal{F}$ ein *Baum*, wenn gilt:

$$f = a \circ f', \quad a \in A,\ f' \in Mor\,\mathcal{F}$$

Die Menge der Bäume in $M \subset Mor\,\mathcal{F}$ sei mit $TREES(M)$ bezeichnet. Die Morphismen einer Baumkategorie lassen sich bzgl. der Operation $\times$ in Einheiten und Bäume zerlegen. Es gilt das folgende

Lemma 1.1.2 ([HoCl72]) *Sei $\mathcal{F} := \mathcal{F}(A, V, Q, Z)$ eine Baumkategorie.*

Jedes $f \in Mor\,\mathcal{F},\ Z(f) = w,\ |w| = k$, besitzt eine eindeutige Darstellung

$$f = f_1 \times \cdots \times f_k$$

mit $f_i \in Mor\,\mathcal{F},\ Z(f_i) = w(i),\ 1 \leq i \leq k$.

Jedes f_i in dieser Darstellung ist dann entweder die Einheit $\mathbf{1}_{w(i)}$ oder ein Baum.

Im nächsten Abschnitt zeigen wir, wie sich mit den eingeführten Begriffen formale Sprachen beschreiben lassen. Die Darstellung formaler Sprachen in dieser Theorie wird vielfach als die *Algebraische Theorie formaler Sprachen* bezeichnet, vgl. [Ben75].

1.2 Beschreibung formaler Sprachen durch $\times$–Kategorien

Die Bedeutung der Theorie der $\times$–Kategorien für die Beschreibung formaler Sprachen gründet sich auf den Zusammenhang zwischen *Semi-Thue-Systemen* und freien $\times$-Kategorien (vgl. [Ho66] oder [Ben75]).

Ein Semi-Thue–System ist ein Ersetzungssystem auf Wörtern und besteht aus einer endlichen Menge von *Ersetzungsregeln*. Jede Ersetzungsregel ist ein Paar (u, v) von Wörtern, das einen eindeutigen Namen p trägt, man schreibt dafür $p : u \to v$.

Eine Regel $p : u \to v$ ist auf ein Wort w anwendbar, wenn u Teilwort von w ist. Die Anwendung der Regel p ist die Ersetzung des Teilworts u durch das Wort v. Die *Semi-Thue–Relation* beschreibt, welche Wörter sich durch endlich viele Regelanwendungen ineinander überführen lassen.

Läßt sich ein Wort w durch die Regeln des Semi-Thue–Systems in ein Wort w' überführen, dann heißt w' *ableitbar* aus w. In diesem Fall gibt es eine Folge $w_0, \ldots, w_n$, $n \in \mathbf{N}_0$, von Wörtern, wobei $w = w_0$, $w' = w_n$ und jedes w_{i+1} durch Anwendung einer Regel aus w_i hervorgeht. Eine solche Folge heißt *Ableitung* im Semi-Thue–System.

Die Definition von Ableitungen durch Folgen gibt im allgemeinen keine vollständige Beschreibung des Erzeugungsprozesses. Eine solche Beschreibung ist allerdings bei vielen Fragen wünschenswert, wie z.B. bei Fragen der Eindeutigkeit oder strukturellen Äquivalenz von Semi-Thue–Systemen. Eine vollständige, adäquate Beschreibung der Erzeugungsprozesse in Semi-Thue–Systemen läßt sich in der Theorie der freien $\times$–Kategorien formulieren. Wir wollen die wichtigsten Begriffe und Ergebnisse kurz aufführen.

Definition 12 *Semi-Thue–System*
$S = (P, V, Q, Z, \overset{*}{\Rightarrow}_S)$ oder kürzer $S = (P, V, Q, Z)$ heißt *Semi-Thue–System*, wenn gilt:

1. P ist eine endliche Menge (*Produktionen, Ersetzungsregeln*).
2. V ist eine endliche Menge (*Alphabet*).
3. $Q, Z : P \to V^*$ sind Abbildungen (*Quelle, Ziel*).

Die *Semi-Thue-Relation* $\overset{*}{\Rightarrow}_S$ ist der reflexive, transitive Abschluß der Relation $\Rightarrow_S$, die wie folgt definiert ist:

$$\Rightarrow_S \quad := \quad \{\, (u \cdot Q(p) \cdot v,\ u \cdot Z(p) \cdot v) \mid p \in P,\ u, v \in V^* \,\}.$$

Eine Folge $(w_0, \ldots, w_n)$, $n \in \mathbf{N}_0$, mit den Eigenschaften

1. $w_i \in V^*$, $0 \le i \le n$,
2. $w_i \Rightarrow_S w_{i+1}$, $0 \le i < n$,

heißt *Ableitung der Länge n* von w_0 nach w_n in S. Es gilt $w \overset{*}{\Rightarrow}_S w'$ genau dann, wenn es eine Ableitung von w nach w' in S gibt.

Jedem Semi-Thue–System $S = (P, V, Q, Z, \overset{*}{\Rightarrow}_S)$ läßt sich eindeutig die freie $\times$–Kategorie

$$\mathcal{F}(S) := \mathcal{F}(P, V, Q, Z)$$

zuordnen. Die Objekte von $\mathcal{F}(S)$ sind die Wörter aus V^*, die Morphismen von $\mathcal{F}(S)$ sind Klassen „unwesentlich verschiedener“ Ableitungen in S. Die Morphismen lassen sich als *Netze* deuten, die aus den Produktionen von S aufgebaut sind. Die Strecken der Netze sind mit Symbolen aus V beschriftet. Die Netze sind eine geometrische Repräsentation der Erzeugungsprozesse des Semi-Thue–Systems.

Zwischen dem Semi-Thue–System S und der freien $\times$–Kategorie $\mathcal{F}(S)$ besteht folgender Zusammenhang:

Lemma 1.2.1 *[Ho66] Für alle $w, w' \in V^*$ gilt*

$$w \overset{*}{\Rightarrow}_S w' \iff \mathcal{F}(S)(w, w') \neq \emptyset.$$

Ist eine freie $\times$–Kategorie $\mathcal{F} := \mathcal{F}(P, V, Q, Z)$ gegeben, so ist $S := (P, V, Q, Z)$ ein Semi-Thue–System. S ist durch $\mathcal{F}$ eindeutig bestimmt, weil P das einzige freie Erzeugendensystem von $\mathcal{F}$ ist. Deshalb kann man $S = S(\mathcal{F})$ schreiben und es gilt der

Satz 1 ([Ho66]) *Es besteht eine eindeutige Zuordnung zwischen den Semi-Thue-Systemen $S = (P, V, Q, Z, \overset{*}{\Rightarrow}_S)$ und den freien $\times$–Kategorien $\mathcal{F}(P, V, Q, Z)$.*

Die Ableitungen $w \overset{*}{\Rightarrow}_S w'$ positiver Länge im Semi-Thue–System $S(\mathcal{F})$ entsprechen den sequentiellen Darstellungen der Morphismen $f : w \to w' \in \mathcal{F}(S)$. Die Ableitungen (w) der Länge 0 in $S(\mathcal{F})$ entsprechen den Einheiten $\mathbf{1}_w$ von $\mathcal{F}(S)$.

Auf dem Begriff der Ableitung in Semi-Thue-Systemen beruht die Definition formaler Sprachen durch *Chomsky-Grammatiken.*

Definition 13 *(Chomsky-)Grammatik*
$G = (S, T, s)$ heißt *(Chomsky-)Grammatik*, falls gilt:

1. $S = (V, P, Q, Z)$ ist ein Semi-Thue–System.
2. $T \subset V,\ T \neq \emptyset$. T heißt Menge der *Terminalsymbole* von G.
   $N := V \setminus T,\ N \neq \emptyset$, heißt die Menge der *Nichtterminalsymbole* von G.
3. $s \in N$ heißt *Axiom* der Grammatik.
4. $Q(p) \in N^+$.

Die *Satzmenge* $L(G)$ der Grammatik $G = (S, T, s)$ ist

$$L(G) := \left\{ w \in T^* \mid s \overset{*}{\Rightarrow}_S w \right\}.$$

Ein Semi-Thue–System $S = (P, V, Q, Z)$ bzw. eine Grammatik $G = (S, T, s)$ heißt *kontextfrei*, wenn für alle $p \in P$ gilt: $|Q(p)| = 1$. In diesem Fall ist $\mathcal{F}(S)$ eine Baumkategorie.

Aufgrund der Äquivalenz von Semi-Thue–Systemen und freien $\times$–Kategorien läßt sich eine Chomsky-Grammatik G auch definieren als

$$G = (\mathcal{F}(S), T, s)$$

und die Satzmenge von G ist

$$L(G) = \{ w \in T^* \mid \mathcal{F}(S)(s, w) \neq \emptyset \}.$$

Wir werden wie üblich einfach $G = (V, T, P, s)$ schreiben und P als Menge von (eindeutig benannten) Paaren $p : x \to y$ auffassen.

Im nächsten Abschnitt geben wir weitere Beschreibungen der Satzmenge einer kontextfreien Grammatik. Wir wollen dabei herausarbeiten, daß sich die Satzmenge durch *Interpretation* einer gewissen Menge von *syntaktischen Strukturen* darstellen läßt. Unter syntaktischen Strukturen verstehen wir die Elemente der freien $\times$–Kategorie. Diesen Elementen (Netzen) geben wir eine Interpretation, indem wir jede Produktion der Grammatik als Abbildung interpretieren und die Hintereinander- bzw. Parallelausführung von Ableitungen als Komposition bzw. Parallelausführung von Abbildungen.

1.3 Kontextfreie Chomsky-Sprachen

Wir geben in diesem Abschnitt weitere Darstellungen der Satzmenge einer kontextfreien Grammatik. Wir beschreiben die „Syntax" der kontextfreien Grammatik als freie $\times$–Kategorie und zeigen, daß die Satzmenge durch eine recht einfache Interpretation einer Teilmenge der Syntax (Bäume) beschrieben werden kann. Wir gehen auf diese Darstellung hier etwas ausführlicher ein, weil wir sie im folgenden zur Definition anderer Sprachklassen verwenden werden.

Die im folgenden gegebene Beschreibung ist im wesentlichen die gleiche wie in [Go77]. Dort wird die Erzeugung der Satzmenge einer kontextfreien Grammatik als Beispiel für eine *initial algebra semantics* beschrieben. Dazu werden die Produktionen der Grammatik als Operatoren einer *initial many-sorted algebra* gedeutet. Der Begriff *initial* bedeutet dabei, daß es jeweils einen eindeutigen Homomorphismus von dieser Algebra in jede beliebige Algebra aus der gleichen Klasse gibt. Die Elemente der *initial algebra* kann man als aus den Operatorsymbolen aufgebaute Ausdrücke auffassen, die die Ableitungen in der Grammatik repräsentieren. Es wird gezeigt, daß die Satzmenge sich durch eine einfache Auswertung dieser Ausdrücke beschreiben läßt.

Dazu interpretiert man die Operatorsymbole als Operationen auf dem freien Monoid T^*, d.h. man ordnet jedem Operatorsymbol eine Abbildung zu, die beschreibt, wie eine Produktion ihre Argumente zu einem neuen Wort zusammensetzt. Die Argumente entsprechen den von den Nichtterminalen der rechten Seite abgeleiteten Wörtern.

Eine ähnliche Beschreibung findet sich auch in der Arbeit [Ho67]. Diese Beschreibung ist allerdings kategorientheoretisch formuliert. Dort werden einer kontextfreien Grammatik zwei $\times$–Kategorien zugeordnet:

- die „syntaktische" Kategorie (freie $\times$–Kategorie)
- die „semantische" Kategorie ($\times$–Kategorie von Abbildungen)

Die syntaktische Kategorie entspricht der oben beschriebenen *initial algebra*.

Die Satzmenge der Grammatik wird durch einen $\times$–Funktor von der syntaktischen in die semantische Kategorie definiert.

Ausgehend von dieser Beschreibung wird eine Klasse K formaler Sprachen definiert, die *durch gekoppelte Ersetzungen erzeugten Sprachen*, und es werden spezielle Teilklassen davon bezüglich ihrer Abschlußeigenschaften untersucht.

Sprachen in der Klasse K werden beschrieben durch endlich erzeugte Unter–$\times$–Kategorien kontextfreier $\times$–Kategorien. Ausgehend von einem kontextfreien Produktionensystem P (dargestellt durch terminalfreie Produktionen p mit entsprechender Abbildung f_p) wählt man eine endliche Menge E von Abbildungen, die von den f_p erzeugt sind. Dann betrachtet man die von E erzeugte Unter–$\times$–Kategorie $\mathcal{C}(E)$. Zu dieser $\times$–Kategorie von Abbildungen korrespondiert eine Unter–$\times$–Kategorie $\mathcal{U}$ der von den Produktionen erzeugten freien $\times$–Kategorie $\mathcal{F}(P)$. Nur die Elemente von $\mathcal{U}$ werden als zulässige syntaktische Strukturen gewertet, d.h. hinsichtlich der Ableitungen, daß die Produktionen von P nicht beliebig, sondern nur gekoppelt (wie durch $\mathcal{U}$ beschrieben) angewendet werden dürfen.

Zur Definition einer Satzmenge gibt man zusätzlich eine Menge S von Startsymbolen an und definiert die Satzmenge durch Auswertung der Menge S mit den Abbildungen aus $\mathcal{C}(E)$.

Die Klasse K enthält auch Sprachen (nicht Satzmengen!), die sich nicht durch Chomsky-Grammatiken charakterisieren lassen. Der Grund dafür ist, daß sich mit einer Chomsky-Grammatik die Kopplungen der Ersetzungen nicht beschreiben lassen. Dieser Gesichtspunkt ist auch für die linguistischen Anwendungen von Bedeutung, wo es in erster Linie nicht um die Satzmengen, sondern um die Beschreibung von Abhängigkeiten in Sätzen geht, d.h um die syntaktischen Strukturen.

Wir werden einen ähnlichen Ersetzungsmechanismus erhalten, wenn wir die Korrespondenz von Ersetzungen auf Bäumen und Ersetzungen auf Wörtern untersuchen. Allerdings werden die dabei auftretenden Unter–$\times$–Kategorien i.a. nicht endlich erzeugt sein. Wir geben jetzt eine „triviale Interpretation" der Netze zur Beschreibung der Satzmenge an. Daran anschließend geben wir die formale Beschreibung der oben angesprochenen Darstellung.

Sei $G = (S, T, s)$ eine kontextfreie Grammatik, d.h.

1. $S = (P, V, Q, Z)$ ist ein kontextfreies Semi-Thue–System,
2. $V = N \cup T,\ N \cap T = \emptyset,\ N, T \neq \emptyset$,
3. $Q(p) \in N$ für alle $p \in P$,
4. $s \in N$.

Sei M die Menge aller Abbildungen $\{u\} \xrightarrow{f} \{v\},\ u, v \in V^*$, sei $\circ : M \times M \rightsquigarrow M$ die Komposition und $\times : M \times M \to M$ die Parallelausführung der Abbildungen aus M, d.h.

$$\begin{array}{ccccc} \{v\} \xrightarrow{f} \{w\} & \circ & \{u\} \xrightarrow{g} \{v\} & := & \{u\} \xrightarrow{f \circ g} \{w\} \\ \{u\} \xrightarrow{f} \{v\} & \times & \{u'\} \xrightarrow{f'} \{v'\} & := & \{uu'\} \xrightarrow{f \times f'} \{vv'\} \end{array}$$

Definiert man $d(f) := u$ und $c(f) := v$ für $\{u\} \xrightarrow{f} \{v\}$ aus M, so ist

$$\mathcal{R}(V) := (V^*, M, d, c, \circ, \times)$$

eine $\times$–Kategorie.

Sei $\tau_1 : V^* \to V^*$ die Identität und $\tau_2' : P \to M$ definiert durch

$$\tau_2'(p) \ := \ \{Q(p)\} \xrightarrow{f_p} \{Z(p)\} \qquad \forall p \in P.$$

Dann läßt sich τ_2' eindeutig zu $\tau_2 : \mathcal{F}(S) \to M$ so fortsetzen, daß $\tau = (\tau_1, \tau_2)$ ein $\times$–Funktor von $\mathcal{F}(S)$ in $\mathcal{R}(V)$ ist. Der $\times$–Funktor τ ordnet jeder Ableitung von u nach v in S die Abbildung $\{u\} \xrightarrow{f} \{v\}$ aus M zu und „vergißt" dabei die Struktur der Ableitung.

Die Satzmenge $L(G)$ ist dann

$$L(G) = \{\tau(\Psi)(s) \mid \Psi \in \mathcal{F}(S)(s, T^*)\}.$$

Die Satzmenge der Grammatik G wird beschrieben durch die „triviale" Interpretation τ der Baummenge $\mathcal{F}(S)(s, T^*)$. Die Interpretation ist trivial, weil die $\times$–Kategorie $\mathcal{F}(S)$ bereits die gesamte Information über die erzeugte Satzmenge enthält.

Wir geben jetzt die oben beschriebene Darstellung der Satzmenge durch eine „syntaktische" und eine „semantische" Kategorie.

Die syntaktische Kategorie beschreibt die von der Grammatik erzeugten Strukturen ohne Einbeziehung der Terminalsymbole. Man kann die Produktionen als Bausteine auffassen, aus denen sich mit den Operationen $\circ$ und $\times$ Netze aufbauen lassen. Diese Netze repräsentieren die Ableitungen in G. Will man das produzierte Wort erhalten, so interpretiert man die Produktionen als Abbildungen und wertet die Netze bzw. Bäume unter dieser Interpretation aus. Die semantische Kategorie ist eine $\times$–Kategorie von Abbildungen. Sie wird erzeugt von den Interpretationen der Produktionen, d.h. sie besteht aus allen Abbildungen, die sich durch Komposition und Parallelausführung aus den Interpretationen der Produktionen gewinnen lassen.

Jeder Produktion

$$p : x_0 \to u_1 \cdot x_1 \cdot u_2 \dots u_k \cdot x_k \cdot u_{k+1}, \quad x_i \in N,\ u_j \in T^*,\ k \in \mathbf{N}_0,$$

der kontextfreien Grammatik ordnen wir die terminalfreie Produktion

$$p : x_0 \to x_1 \dots x_k$$

und die Abbildung $f_p : (T^*)^k \to T^*$ mit

$$f_p(w_1, \dots, w_k) := u_1 \cdot w_1 \cdot u_2 \dots u_k \cdot w_k \cdot u_{k+1}$$

zu.

Wir bezeichnen Quelle bzw. Ziel der terminalfreien Produktion p durch $Q'(p)$ bzw. $Z'(p)$ und erhalten das Semi-Thue–System $S' = (P, N, Q', Z')$ bzw. die freie $\times$–Kategorie $\mathcal{F}(S')$.

$\mathcal{F}(S')$ ist die „syntaktische Kategorie“ von G.

$F_P := \{f_p \mid p \in P\}$ erzeugt in $ABB(T^*, \mathbf{N}_0)$ eine Unter–$\times$–Kategorie $\mathcal{C}(F_P, T^*)$. Sie enthält genau die Abbildungen, die sich aus den f_p und den Identitäten auf $(T^*)^n$, $n \in \mathbf{N}_0$, durch Komposition und Parallelausführung gewinnen lassen.

$\mathcal{C}(F_P, T^*)$ ist die „semantische Kategorie“ von G.

Wir definieren jetzt die Interpretation für $\mathcal{F}(S')$. Seien dazu

$$\begin{array}{lll} \eta_1 : N^* \to \mathbf{N}_0, & \eta_1(w) := |w|, & w \in N^*, \\ \eta_2' : P \ \to F_P, & \eta_2'(p) := f_p\ , & p \in P. \end{array}$$

Durch Fortsetzung von η_2' auf $\mathcal{F}(S')$ ist eindeutig der (kontravariante) $\times$–Funktor $\eta = (\eta_1, \eta_2)$ von $\mathcal{F}(S')$ in $\mathcal{C}(F_P, T^*)$ definiert. Dieser $\times$–Funktor „interpretiert“ die Ableitungen in S' (Netze in $\mathcal{F}(S')$) als Funktionen aus $\mathcal{C}(F_P, T^*)$ und stellt die Verbindung zwischen der syntaktischen und der semantischen Kategorie her. Jeder Ableitung $\Psi : s \to \varepsilon$ aus $\mathcal{F}(S')$ wird eine Abbildung $\eta(\Psi) : (T^*)^0 \to T^*$ zugeordnet, die wir mit dem Wort $\eta(\Psi)(\Box) \in T^*$ identifizieren.

Die Satzmenge $L(G) = L(S, T, s)$ läßt sich durch $\mathcal{F}(S')$, $\mathcal{C}(F_P, T^*)$ und η beschreiben. Sei $B(S, x) := \mathcal{F}(S)(x, T^*)$ und $B(S', x) := \mathcal{F}(S')(x, \varepsilon)$ für $x \in N$. Dann gilt folgender

Satz 2

$$L(G) = \{\eta(\Psi)(\Box) \mid \Psi \in B(S', s)\}$$

Zum Beweis des Satzes verwendet man den Zusammenhang zwischen den freien $\times$–Kategorien $\mathcal{F}(S)$ und $\mathcal{F}(S')$. Es zeigt sich, daß durch das Entfernen der Terminale aus den Produktionen die syntaktische Struktur erhalten bleibt oder anders ausgedrückt, daß die Terminale für die Beschreibung der Syntax nicht nötig sind.

Sei $\lambda_1 : V^* \to N^*$ der durch

$$\lambda_1(x) := \begin{cases} x, & x \in N \\ \varepsilon, & x \in T \end{cases}$$

definierte Monoidhomomorphismus („löscht Terminale“) und sei

$$\lambda_2' : P \to \mathcal{F}(S'), \qquad \lambda_2'(p) := p.$$

Die Abbildungen λ_1 und λ_2' definieren eindeutig den $\times$–Funktor $\lambda = (\lambda_1, \lambda_2)$ von $\mathcal{F}(S)$ in $\mathcal{F}(S')$. Dieser Funktor beschreibt das Entfernen der Terminale aus den Produktionen bzw. aus den Ableitungen in S. Die Menge der Bäume, die zur Definition der Satzmenge von G interpretiert wird, ändert sich dadurch nicht wesentlich.

Wir geben drei einfache Hilfssätze ohne Beweis an.

Hilfssatz 1.3.1 *Jeder Baum $\Psi \in B(S, x)$, $x \in N$, besitzt eine eindeutige Darstellung*

$$\Psi = (\mathbf{1}_{u_1} \times \Psi_1 \times \mathbf{1}_{u_2} \times \cdots \times \mathbf{1}_{u_k} \times \Psi_k \times \mathbf{1}_{u_{k+1}}) \circ p$$

mit $p \in P$, $Q(p) = x$, $Z(p) = u_1 \cdot x_1 \cdot u_2 \ldots u_k \cdot x_k \cdot u_{k+1}$, $u_j \in T^*$, $x_i \in N$, $\Psi_i \in B(S, x_i)$, $k \in \mathbf{N}_0$, $1 \leq i \leq k$, $1 \leq j \leq k+1$. □

Hilfssatz 1.3.2 *Jeder Baum $\Psi' \in B(S', x)$, $x \in N$, besitzt eine eindeutige Darstellung*

$$\Psi' = (\Psi_1' \times \cdots \times \Psi_k') \circ p$$

mit $p \in P$, $Q'(p) = x$, $Z'(p) = x_1 \ldots x_k$, $x_i \in N$, $\Psi_i' \in B(S', x_i)$, $k \in \mathbf{N}_0$, $1 \leq i \leq k$. □

Hilfssatz 1.3.3 *Bezeichnet $|\,.\,|$ die Länge der Morphismen in $\mathcal{F}(S)$ bzw. in $\mathcal{F}(S')$ und ist λ der oben definierte $\times$–Funktor, dann gilt für alle $\Psi \in \mathcal{F}(S)$:*

$$|\lambda(\Psi)| = |\Psi| \ .$$

□

Lemma 1.3.1 *λ ist eine Bijektion zwischen $B(S, x)$ und $B(S', x)$.*

Beweis:

Wir zeigen durch Induktion über die Länge, daß jeder Baum $\Psi' \in B(S', x)$ genau ein Urbild unter λ in $B(S, x)$ hat.

Induktionsanfang: $|\Psi'| = 1$

Dann ist $\Psi' = p \in P$, $Z'(p) = \varepsilon$, also $Z(p) \in T^*$ und daher $p \in \lambda^{-1}(\Psi') \cap B(S, x)$. Nach Hilfssatz 1.3.3 hat jedes Element g aus $\lambda^{-1}(\Psi')$ die Länge 1, ist also von der Form $g = \mathbf{1}_{w_1} \times q \times \mathbf{1}_{w_2}$ mit $w_1, w_2 \in T^*$ und $q \in P$. Ist $g \in B(S, x)$, so folgt $w_1 = w_2 = \varepsilon$ und somit $g = q$. Da λ auf P die Identität ist, folgt $g = p$.

Induktionsschritt: $|\Psi'| > 1$

Sei $\Psi' = (\Psi_1' \times \cdots \times \Psi_k') \circ p$ die eindeutige Darstellung von Ψ' aus Hilfssatz 1.3.2.

Aufgrund der Induktionsannahme besitzt jedes Ψ_i' in $B(S, x)$ ein eindeutiges Urbild Ψ_i. Das eindeutige Urbild von p in $B(S, x)$ ist p selbst. Nach Konstruktion von S' ist

$$Z(p) = u_1 \cdot x_1 \cdot u_2 \ldots u_k \cdot x_k \cdot u_{k+1}, \quad x_i \in N,\ u_j \in T^*,$$

und daher ist

$$\Psi := (\mathbf{1}_{u_1} \times \Psi_1 \times \mathbf{1}_{u_2} \times \cdots \times \mathbf{1}_{u_k} \times \Psi_k \times \mathbf{1}_{u_{k+1}}) \circ p \quad \in \lambda^{-1}(\Psi') \cap B(S, x).$$

Sei g ein beliebiges Element von $\lambda^{-1}(\Psi') \cap B(S,x)$ und

$$g = (\mathbf{1}_{w_1} \times g_1 \times \mathbf{1}_{w_2} \times \cdots \times \mathbf{1}_{w_l} \times g_l \times \mathbf{1}_{w_{l+1}}) \circ q, \qquad l \in \mathbf{N}_0,$$

die eindeutige Darstellung von g aus Hilfssatz 1.3.1. Es ist

$$\begin{aligned} \lambda(g) &= (\lambda(g_1) \times \cdots \times \lambda(g_l)) \circ \lambda(q) \\ &= \Psi' \\ &= (\Psi'_1 \times \cdots \times \Psi'_k) \circ p \end{aligned}$$

Wegen $\lambda(q) = q$ und der Eindeutigkeit dieser Darstellung folgt dann $q = p$, $l = k$, $Q'(\lambda(g_i) = x_i$, $\lambda(g_i) = \Psi'_i \in B(S', x_i)$. Damit ist

$$g = (\mathbf{1}_{u_1} \times g_1 \times \mathbf{1}_{u_2} \times \cdots \times \mathbf{1}_{u_k} \times g_k \times \mathbf{1}_{u_{k+1}}),\ g_i \in B(S, x_i).$$

Nach Induktionsvoraussetzung ist das Urbild von Ψ'_i in $B(S, x_i)$ eindeutig bestimmt und aus $\lambda(g_i) = \Psi'_i = \lambda(\Psi_i)$ folgt $g_i = \Psi_i$. Damit ist $g = \Psi$ und das Lemma ist gezeigt. □

Lemma 1.3.2

$$\forall\ \Psi \in B(S,x),\ x \in N\ :\ \eta(\lambda(\Psi))(\Box) = Z(\Psi)$$

Beweis: Wir führen Induktion über die Länge von $\Psi \in B(S,x)$.

Induktionsanfang: $|\Psi| = 1$

Dann ist $\Psi = p : x \to w \in P$, $w \in T^*$ und $f_p : \{\Box\} \to T^*$, $f_p(\Box) = w$. Wegen $\lambda(p) = p$ folgt $\eta(\lambda(\Psi))(\Box) = \eta(p)(\Box) = f_p(\Box) = w = Z(\Psi)$.

Induktionsschritt: $|\Psi| > 1$

Sei

$$\Psi = (\mathbf{1}_{w_1} \times \Psi_1 \times \mathbf{1}_{w_2} \times \cdots \times \mathbf{1}_{w_k} \times \Psi_k \times \mathbf{1}_{w_{k+1}}) \circ p$$

die eindeutige Darstellung von Ψ aus Hilfssatz 1.3.1. Dann ist

$$\begin{aligned} Z(\Psi) &= w_1 \cdot Z(\Psi_1) \cdot w_2 \ldots w_k \cdot Z(\Psi_k) \cdot w_{k+1} \\ &= w_1 \cdot \eta(\lambda(\Psi_1))(\Box) \cdot w_2 \ldots w_k \cdot \eta(\lambda(\Psi_k))(\Box) \cdot w_{k+1} \\ &= f_p(\ \eta(\lambda(\Psi_1))(\Box), \ldots \eta(\lambda(\Psi_k)\ )(\Box) \\ &= \eta(\lambda(\Psi))(\Box) \end{aligned}$$

Damit ist das Lemma gezeigt. □

Aus den beiden Lemmata folgt

$$\begin{aligned} L(G) &= \{Z(\Psi) \mid \Psi \in B(S,x)\} \\ &= \{\eta(\lambda(\Psi))(\Box) \mid \Psi \in B(S,x)\} \\ &= \{\eta(\Psi')(\Box) \mid \Psi' \in B(S',x)\} \end{aligned}$$

womit der Satz bewiesen ist. □

Bemerkung: Aus Hilfssatz 1.3.2 erhält man sofort eine rekursive Definition für die Mengen $B(S',x)$, die man als Grammatik auf $\mathcal{F}(S')$ formulieren kann. Die dabei entstehende Grammatik ist vom Typ der in [Mai74] beschriebenen *regulären Grammatiken* auf einer *many-sorted algebra*. Man betrachte dazu die Produktionen von S' als Terminalsymbole und ordnet jeder Menge $B(S', x)$, $x \in N$, ein Nichtterminalsymbol $[x]$ zu. Die rekursiven Gleichungen für $B(S', x)$ werden zu Produktionen der Grammatik, genauer: ist $p : x \to x_1 \cdots x_k \in P(S')$, dann ist

$$\tilde{p} : [x] \to ([x_1] \times \cdots \times [x_k]) \circ p \quad \in \tilde{P}$$

eine Produktion der Baumgrammatik.

Damit diese Definition konsistent ist, definiert man für die Nichterminalsymbole der Baumgrammatik: $Q'([x]) := x,\ Z'([x]) := \varepsilon$.

Dann gilt nämlich für jede Regel der Baumgrammatik

$$\tilde{p} : [x] \to y \in \tilde{P} \quad \Longrightarrow \quad Q'([x]) = Q'(y),\ Z'([x]) = Z'(y)$$

Diese „Quelle-/Zielbedingung“ für die Produktionen werden wir bei den in einem späteren Abschnitt definierten Grammatiken ebenfalls verlangen. Allerdings werden wir nur Baumkategorien mit Objektmenge $\mathbf{N}_0$ betrachten, d.h. es gibt keine Beschriftungen der Kanten.

Man kann die oben angegebene Baumgrammatik auch für die entsprechenden „unbeschrifteten“ Bäume formulieren, d.h. vom Objektmonoid N^* zu $\mathbf{N}_0$ übergehen. Da die Interpretation die Beschriftung der Kanten ohnehin ignoriert, erhält man trotz dieser Vereinfachung eine Beschreibung der erzeugten Satzmenge.

Die regulären Baumgrammatiken werden wir in natürlicher Weise verallgemeinern, indem wir nicht nur Ersetzungen von „Blättern“, sondern von beliebigen Knoten im Baum zulassen. Allerdings lassen unsere verallgemeinerten Baumgrammatiken keine Löschung oder Vervielfältigung von Teilbäumen zu, wie etwa die in der Literatur beschriebenen kontextfreien Baumgrammatiken. Diese Eigenschaft ist übrigens auch für linguistische Anwendungen wünschenswert.

1.4 Gekoppelt-kontextfreie Grammatiken

Gekoppelt-kontextfreie Grammatiken sind eine Verallgemeinerung kontextfreier Grammatiken. Bei diesem Ersetzungssystem ist es erlaubt, in einem Wort mehrere kontextfreie Ersetzungen parallel vorzunehmen. Diese Ersetzungen sind im allgemeinen jedoch nicht unabhängig voneinander, sondern miteinander *gekoppelt*. Die *Kopplung* der Ersetzungen wird durch die *Klammerung* der aktuellen Satzform gesteuert. Dazu betrachtet man Gruppen von Nichtterminalsymbolen der Grammatik als korrespondierende Klammern und definiert die Grammatikregeln und die Ersetzungsrelation so, daß jeder Ersetzungsschritt die Klammerung der Nichtterminale respektiert. In einem Ersetzungsschritt dürfen nur korrespondierende Klammern gleichzeitig ersetzt werden und das Ergebnis der Ersetzung muß wiederum ein korrekt geklammertes Wort sein. Die Terminalsymbole sind „triviale“ Klammern, d.h. einfache Alphabetsymbole, sie bilden „Füllsel“ zwischen den Nichtterminalklammern. Wir erlauben nicht nur Klammerpaare, sondern Klammern aus beliebig vielen Komponenten.

Um diese informalen Begriffe zu präzisieren, definieren wir zunächst eine Verallgemeinerung der *Semi-Dyck-Sprachen* auf Klammern mit beliebig vielen Komponenten.

Eine Semi-Dyck-Sprache ist die Menge der „korrekt geklammerten“ Wörter über einem endlichen Alphabet von *Klammerpaaren*. Sie läßt sich durch Angabe einer endlichen Menge korrespondierender Klammerpaare (je eine „öffnende“ bzw. „schließende“ Klammer) und einer Menge von „Rechenregeln“ (Gleichungen) formal definieren. Die Rechenregeln drücken aus, daß sich eine öffnende und die zugehörige schließende Klammer gegenseitig aufheben.

Die Gleichungen erzeugen eine Kongruenz auf dem Wörtermonoid über dem Gesamtalphabet aus allen öffnenden und schließenden Klammern. Zwei Wörter sind genau dann kongruent, wenn sie durch endlichfache Anwendung der Gleichungen (d.h. durch korrektes Einfügen oder Löschen von Klammern) ineinander überführt werden können.

Die Menge der korrekt geklammerten Wörter, d.h. die zugehörige *Klammersprache*, besteht aus genau den Wörtern, die sich durch Anwendung der Rechenregeln in das leere Wort überführen lassen. Eine formale Beschreibung von Semi-Dyck-Sprachen findet man z.B. in [Ho90] oder in [Ber79].

Unsere Verallgemeinerung besteht darin, daß wir nicht nur Klammerpaare, sondern *Klammertupel* zulassen. Ein Klammertupel kann nicht nur ein Wort, sondern endlich viele Wörter umklammern.

Dazu geben wir eine endliche Menge von Klammern an und definieren für jede Klammer, wieviele Wörter sie umschließen kann (*Grad* der Klammer). Die Rechenregeln, die Kongruenz und die zugehörige Klammersprache werden wie bei Semi-Dyck-Sprachen definiert.

Definition 14 *Klammeralphabet*
Sei K eine endliche Menge (*Klammern*) und $g : K \to \mathbf{N}_0$ eine Abbildung.

$$[K, g] := \{(k, i) \in K \times \mathbf{N} \mid 1 \leq i \leq g(k) + 1\}$$

heißt *Klammeralphabet* und $g(k)$ heißt *Grad* der Klammer $k \in K$.

Für (k, i) schreiben wir k_i, K_n sei die Menge aller Klammern aus K vom Grad $n \in \mathbf{N}_0$.

Definition 15 *Klammerkongruenz*
Die von den Gleichungen

$$\left\{ \left(\prod_{i=1}^{g(k)+1} k_i \right) = \varepsilon, \quad k \in K \right\}$$

auf $[K, g]^*$ erzeugte Kongruenzrelation „$\equiv_K$" nennen wir die *Klammerkongruenz* zum Klammeralphabet $[K, g]$.

Definition 16 *Klammersprache*
Die *Klammersprache* $D(K, g)$ über dem Klammeralphabet $[K, g]$ ist

$$D(K, g) := \{w \in [K, g]^* \mid w \equiv_K \varepsilon\},$$

d.h. die Menge der durch Anwendung der Klammerungsregeln auf ε kürzbaren Wörter. Wir schreiben auch einfach $D(K)$, wenn g implizit gegeben ist.

Die Klammersprache $D(K)$ läßt sich konstruktiv charakterisieren:
$D(K)$ ist die kleinste Teilmenge von $[K, g]^*$, die folgende Eigenschaften 1. bis 3. erfüllt:

1. $\varepsilon \in D(K)$,
2. $D(K) \cdot D(K) \subseteq D(K)$,
3. $k \in K_n,\ w_1, \ldots, w_n \in D(K) \quad \Longrightarrow \quad k_1 \cdot w_1 \cdot k_2 \ldots k_n \cdot w_n \cdot k_{n+1} \ \in D(K)$.

Wir geben Beispiele für diese Definitionen an:

1. Ist $K = \{x, y\}$ und $g(x) = g(y) = 1$, also $[K, g] = \{x_1, x_2, y_1, y_2\}$, dann sind folgende Wörter Elemente von $D(K, g)$:

   $$\varepsilon, \quad x_1x_2, \quad x_1y_1y_2x_1x_2x_2, \quad x_1x_1x_1x_2x_2x_2$$

   Schreiben wir „[" für x_1, „]" für x_2, „(" für y_1 und „)" für y_2, dann sind die oben angegebenen Wörter:

   $$\varepsilon, \quad [\,], \quad [\,(\,)\,[\,]\,], \quad [\,[\,[\,]\,]\,]$$

   $D(K, g)$ ist in diesem Beispiel die Semi-Dyck-Sprache mit den korrespondierenden Klammerpaaren $x_1 \leftrightarrow x_2$ und $y_1 \leftrightarrow y_2$. Besitzt jede Klammer den Grad 1, dann ist die definierte Klammersprache eine Semi-Dyck-Sprache.

2. Für ein Klammeralphabet $[K, g]$ mit $K = K_0$ ist $D(K, g) \equiv K^*$, dem Wörtermonoid über K, d.h. Klammern vom Grad 0 entsprechen „normalen" Alphabetzeichen. In diesem Fall werden wir gelegentlich $[K, g]$ und K gemäß $(k, 1) \leftrightarrow k$ identifizieren.

In unseren Betrachtungen werden Klammeralphabete $[K, g]$ mit $K = N \dot{\cup} T,\ T = T_0$, vorkommen. Wir bezeichnen die entsprechende Klammersprache meist mit $D(N, T)$ und fassen sie als Teilmenge von $([N, g] \cup T)^*$ auf.

Wir geben ohne Beweis einige Eigenschaften der Klammersprachen $D(K)$ an:

1. Für jede Teilmenge $M \subset K_0$ ist $M^* \subset D(K)$.
2. $D(K)$ ist freies Untermonoid von $[K, g]^*$. Das freie Erzeugendensystem von $D(K)$ ist

   $$E(K) := \bigcup_{k \in K} \{k_1 \cdot d_1 \cdot k_2 \ldots k_n \cdot d_n \cdot k_{n+1} \mid k \in K_n,\ d_1, \ldots, d_n \in D(K)\}.$$

3. Ist $d \in D(K)$, dann gilt für $u, v \in [K, g]^*$
$$udv \in D(K) \quad \Longleftrightarrow \quad uv \in D(K).$$

4. Sind $u, v \in [K, g]^*$, $u \cdot v \in D(K)$, dann gilt
$$u \in D(K) \quad \Longleftrightarrow \quad v \in D(K).$$

Mit diesen Begriffen können wir jetzt die *gekoppelt-kontextfreien Grammatiken* definieren.

Definition 17 *Gekoppelt-kontextfreie Grammatik CCFG*
Sei $[V, g]$ ein Klammeralphabet mit $V = N \dot{\cup} T$, $N, T \neq \emptyset$ und $T = T_0$. Sei $D(N, T)$ die zugehörige Klammersprache.
$$G_{ccf} = \left([V, g], T, P_{ccf}, s, \overset{*}{\Rightarrow}_{ccf}\right)$$
heißt *gekoppelt-kontextfreie Grammatik*, wenn gilt:

1. P_{ccf} ist eine endliche Menge von *gekoppelt-kontextfreien Produktionen*
$$p : (x_1, \ldots, x_{k+1}) \to (y_1, \ldots, y_{k+1})$$
mit $x \in N_k$, $y_1, \ldots, y_{k+1} \in [V, g]^*$, $y_1 \cdots y_{k+1} \in D(N, T)$.
Wir nennen $grad(p) := k$ den *Grad* der Produktion p. Der Grad der Grammatik sei das Maximum über den Graden der Produktionen.

2. Die *Rewriting-Relation* $\overset{*}{\Rightarrow}_{ccf}$ ist der reflexive, transitive Abschluß der Relation
$$\begin{aligned} &\Rightarrow_{ccf} \;:= \\ &\{ \; (u_1 \cdot x_1 \cdot v_1 \cdot x_2 \ldots x_k \cdot v_k \cdot x_{k+1} \cdot u_2, \; u_1 \cdot y_1 \cdot v_1 \cdot y_2 \ldots y_k \cdot v_k \cdot y_{k+1} \cdot u_2 \,) \mid \\ &\quad u_1, u_2 \in [V, g]^*, \; v_1, \ldots, v_k \in D(N, T), \\ &\quad p : (x_1, \ldots, x_{k+1}) \to (y_1, \ldots, y_{k+1}) \in P_{ccf} \quad \}. \end{aligned}$$

3. $s \in N_0$ ist das *Axiom* der Grammatik.

Die von G_{ccf} *erzeugte Satzmenge* ist
$$L(G_{ccf}) := \left\{ w \mid w \in T^*, \; s \overset{*}{\Rightarrow}_{ccf} \; w \right\}.$$

Die gekoppelt-kontextfreien Grammatiken stellen eine Verallgemeinerung der kontextfreien Grammatiken dar. Betrachten wir nämlich den Fall $N = N_0$, d.h. jedes Nichtterminalsymbol ist ein „normales" Alphabetzeichen. Wir können dann jedes $x_1 \in [N, g]$ mit $x \in N$ identifizieren, also auch $[N, g]$ mit N bzw. V mit $[V, g]$.

Die Produktionen von G_{ccf} haben dann die Form $p : (x_1) \to (y_1)$ $x_1 \in [N, g] = N$, $y_1 \in [V, g]^* = V^*$, sind also gewöhnliche kontextfreie Produktionen.

Die Relation $\Rightarrow_{ccf}$ ist in diesem Fall
$$\Rightarrow_{ccf} \; = \{(u_1 \cdot x_1 \cdot u_2, \; u_1 \cdot y_1 \cdot u_2) \mid u_1, u_2 \in V^*, \; p : (x_1) \to (y_1) \in P_{ccf}\},$$

d.h. $\overset{*}{\Rightarrow}_{ccf}$ ist die Semi-Thue-Relation.

Beispiel 1:
Sei $G_{ccf} := (\hat{V}, T, P, s, \overset{*}{\Rightarrow}_{ccf})$, $\hat{V} := \{s,\ x_1, x_2,\ a, b, c, e\}$, $T := \{a, b, c, e\}$,
$P := \{\, p : s \to x_1 e x_2,\quad q : (x_1, x_2) \to (a x_1 b, c x_2),\quad r : (x_1, x_2) \to (ab, c)\,\}$.
Eine Ableitung in G_{ccf} ist

$$s \Rightarrow_{ccf} x_1 e x_2 \overset{n}{\Rightarrow}_{ccf} a^n x_1 b^n e c^n x_2 \Rightarrow_{ccf} a^{n+1} b^{n+1} e c^{n+1},\ n \in \mathbf{N}_0$$

Es ist leicht zu sehen, daß $L(G_{ccf}) = \{a^n b^n e c^n \mid n \in \mathbf{N}\}$. Die Satzmenge dieser gekoppelt-kontextfreien Grammatik ist also eine kontextsensitive Sprache, die nicht kontextfrei ist. Es gilt allerdings $L(G_{ccf}) \in TAL$, d.h. $L(G_{ccf})$ ist Satzmenge einer *Tree-Adjoining-Grammatik* [Jo85].

Beispiel 2:
Sei $G_{ccf} := (\hat{V}, T, P, s, \overset{*}{\Rightarrow}_{ccf})$, $\hat{V} := \{s,\ x_1, x_2,\ a, b\}$, $T := \{a, b\}$,
$P := \{p : s \to x_1 x_2,\ q_a : (x_1, x_2) \to (a x_1, a x_2),\ q_b : (x_1, x_2) \to (b x_1, b x_2),\ q_\varepsilon : (x_1, x_2) \to (\varepsilon, \varepsilon)\}$
Die möglichen Ableitungen in G_{ccf} sind

$$\begin{array}{lll} s & \overset{p}{\Rightarrow}_{ccf} & x_1 \cdot x_2 \\ & \overset{q_{a_1}}{\Rightarrow}_{ccf} & a_1 x_1 \cdot a_1 x_1 \\ & \vdots & \\ & \overset{q_{a_n}}{\Rightarrow}_{ccf} & a_1 \ldots a_n x_1 \cdot a_1 \ldots a_n x_2 \\ & \overset{q_\varepsilon}{\Rightarrow}_{ccf} & a_1 \ldots a_n \cdot a_1 \ldots a_n \end{array}$$

Damit sieht man, daß $L(G_{ccf}) = \{w \cdot w \mid w \in \{a, b\}^*\} = COPY_1$. Diese Menge ist ebenfalls echt kontextsensitiv und liegt in der Klasse der Tree Adjoining Languages.

Das folgende Lemma zeigt, daß die gekoppelt-kontextfreien Ersetzungen die Klammerstruktur respektieren, d.h. daß insbesondere aus korrekt geklammerten Wörtern nur korrekt geklammerte Wörter abgeleitet werden können.

Lemma 1.4.1 *Für $w, w' \in [V, g]^*$ gilt:* $w \overset{*}{\Rightarrow}_{ccf} w' \implies w \equiv_V w'$

Beweis: Induktion über die Länge der Ableitung.
Induktionsanfang: $w \overset{0}{\Rightarrow}_{ccf} w'$. Dann ist $w = w'$ und daher $w \equiv_V w'$.

Induktionsschritt: $w \overset{n+1}{\Rightarrow}_{ccf} w',\ n \in \mathbf{N}_0$.
Es gibt ein Wort $w'' \in [V, g]^*$, so daß $w \overset{n}{\Rightarrow}_{ccf} w'' \Rightarrow_{ccf} w'$. Nach Definition von $\Rightarrow_{ccf}$ gibt es eine Produktion $p : (x_1, \ldots, x_{k+1}) \to (y_1, \ldots, y_{k+1})$ und Wörter $w_1'', w_2'' \in [V, g]^*$, $d_1, \ldots, d_k \in D(N, T)$ mit

$$\begin{aligned} w'' &= w_1'' \cdot x_1 \cdot d_1 \cdot x_2 \ldots x_k \cdot d_k \cdot x_{k+1} \cdot w_2'' \\ w' &= w_1'' \cdot y_1 \cdot d_1 \cdot y_2 \ldots y_k \cdot d_k \cdot y_{k+1} \cdot w_2'' \end{aligned}$$

Wegen $d_i \equiv_V \varepsilon,\ 1 \le i \le k$ und $x_1 \cdots x_{k+1} \equiv_V \varepsilon$ ist dann

$$\begin{aligned} w'' &= w_1'' \cdot x_1 \cdot d_1 \cdot x_2 \ldots x_k \cdot d_k \cdot x_{k+1} \cdot w_2'' \\ &\equiv_V w_1'' \cdot x_1 \cdots x_{k+1} \cdot w_2'' \\ &\equiv_V w_1'' \cdot w_2'' \end{aligned}$$

Nach Definition von P_{ccf} ist $y_1 \cdots y_{k+1} \equiv_V \varepsilon$ und daher

$$\begin{aligned} w' &= w_1'' \cdot y_1 \cdot d_1 \cdot y_2 \dots y_k \cdot d_k \cdot y_{k+1} \cdot w_2'' \\ &\equiv_V w_1'' \cdot y_1 \cdots y_{k+1} \cdot w_2'' \\ &\equiv_V w_1'' \cdot w_2'' \end{aligned}$$

Nach Induktionsvoraussetzung ist $w \equiv_V w''$ und daher $w \equiv w'$. □

Folgerung: Für $w \in [V,g]^*$, $s \overset{*}{\Rightarrow}_{ccf} w$ gilt: $w \in D(N,T)$.
Die „Satzformen“ einer CCFG sind also Elemente der Klammersprache $D(N,T)$.

Wir untersuchen jetzt den Zusammenhang zwischen den Ableitungen in einer CCFG und den Ableitungen in der kontextfreien Grammatik, die durch „Entkoppeln“ der Produktionen entsteht. Diese Grammatik ist dadurch definiert, daß die Komponenten der gekoppelt-kontextfreien Produktionen als unabhängige Produktionen aufgefaßt werden und als Ersetzungsrelation die Semi-Thue-Relation benutzt wird.

Definition 18

Ist $G_{ccf} = ([V,g],T,P_{ccf},s,\overset{*}{\Rightarrow}_{ccf})$ eine CCFG, dann heißt $G_{cf} := ([V,g],T,P_{cf},s,\overset{*}{\Rightarrow})$ mit

$$P_{cf} := \{p_i : x_i \to y_i \mid \exists p : (x_1,\dots,x_{k+1}) \to (y_1,\dots,y_{k+1}) \in P_{ccf},\ 1 \le i \le k+1\}$$

die G_{ccf} *zugrundeliegende* kontextfreie Grammatik.

Es ist klar, daß G_{cf} für jede CCFG eine kontextfreie Grammatik ist. Die Ableitungen in der Grammatik G_{cf} sind die Morphismen der von P_{cf} erzeugten freien $\times$–Kategorie $\mathcal{F}(G_{cf}) := \mathcal{F}(P_{cf},[V,g]^*)$. Wir wollen die Ableitungen in der CCFG G_{ccf} mithilfe dieser $\times$–Kategorie charakterisieren. Es gilt folgender

Satz 3 *Ist $G_{ccf} = ([V,g],T,P_{ccf},s,\overset{*}{\Rightarrow}_{ccf})$ eine CCFG und sind $u,v \in [V,g]^*$, dann gilt:*

$$u \overset{*}{\Rightarrow}_{ccf} v \quad\Longleftrightarrow\quad \mathcal{U}(P_{ccf})(u,v) \neq \emptyset\ .$$

Dabei ist $\mathcal{U}(P_{ccf})$ die Unter–$\times$–Kategorie von $\mathcal{F}(G_{cf})$ mit dem freien Erzeugendensystem

$$\begin{aligned} E(P_{ccf}) := \{\ & p_1 \times \mathbf{1}_{d_1} \times p_2 \times \cdots \times p_k \times \mathbf{1}_{d_k} \times p_{k+1} \mid \\ & p \in P_{ccf},\ grad(p) = k,\ d_i \in D(N,T),\ 1 \le i \le k \quad \} \end{aligned}$$

Beweis:

Sei $f \in \mathcal{U}(P_{ccf})(u,v)$, $u,v \in [V,g]^*$.

1. f ist Einheit, $f = \mathbf{1}_u$, $u = v$, dann gilt $u \overset{*}{\Rightarrow}_{ccf} v$.

2. $f \in E(P_{ccf})$, $f = p_1 \times \mathbf{1}_{d_1} \times p_2 \times \cdots \times p_k \times \mathbf{1}_{d_k} \times p_{k+1}$,
   $p : (x_1,\dots,x_{k+1}) \to (y_1,\dots,y_{k+1}) \in P_{ccf}$, $d_1,\dots,d_k \in D(N,T)$,
   $Q(f) = x_1 \cdot d_1 \cdot x_2 \dots x_k \cdot d_k \cdot x_{k+1}$, $\quad Z(f) = y_1 \cdot d_1 \cdot y_2 \dots y_k \cdot d_k \cdot y_{k+1}$.

   Dann gilt $Q(f) \Rightarrow_{ccf} Z(f)$ nach Definition von $\Rightarrow_{ccf}$.

3. $f = f' \times f''$, $f',f'' \in \mathcal{U}(P_{ccf})$. Ist $Q(f') \overset{*}{\Rightarrow}_{ccf} Z(f')$ und $Q(f'') \overset{*}{\Rightarrow}_{ccf} Z(f'')$, dann gilt nach Definition auch $Q(f) \overset{*}{\Rightarrow}_{ccf} Z(f)$.

4. $f = f'' \circ f',\ f', f'' \in \mathcal{U}(P_{ccf})$. Ist $Q(f') \overset{*}{\Rightarrow}_{ccf} Z(f') = Q(f'')$ und $Q(f'') \overset{*}{\Rightarrow}_{ccf} Z(f'')$, dann gilt nach Definition auch $Q(f') \overset{*}{\Rightarrow}_{ccf} Z(f'')$.

Da sich jeder Morphismus aus $\mathcal{U}(P_{ccf})$ in die Elemente des Erzeugendensystems $E(P_{ccf})$ und Einheiten zerlegen läßt, folgt daraus die eine Richtung des Satzes. Wir zeigen jetzt die Gegenrichtung.

Seien $u, v \in [V,g]^*$ und sei $u \overset{n}{\Rightarrow}_{ccf} v$ eine Ableitung in G_{ccf} von u nach v, $n \in \mathbf{N}_0$.
Wir führen Induktion über die Länge der Ableitung.
Induktionsanfang: $u \overset{0}{\Rightarrow}_{ccf} v$. Dann ist $u = v$ und aus $\mathbf{1}_u \in \mathcal{U}(P_{ccf})(u,v)$ folgt die Behauptung.

Induktionsschritt: $u \overset{n+1}{\Rightarrow}_{ccf} v$. Dann gibt es ein $v' \in [V,g]^*$ mit $u \overset{n}{\Rightarrow}_{ccf} v' \Rightarrow_{ccf} v$ und

$$\begin{aligned} v' &= w_1 \cdot x_1 \cdot d_1 \cdot x_2 \ldots x_k \cdot d_k \cdot x_{k+1} \cdot w_2\ , \\ v &= w_1 \cdot y_1 \cdot d_1 \cdot y_2 \ldots y_k \cdot d_k \cdot y_{k+1} \cdot w_2\ , \end{aligned}$$

wobei $w_1, w_2 \in [V,g]^*$, $d_1, \ldots, d_k \in D(N,T)$, und es gibt eine Produktion $p : (x_1, \ldots, x_{k+1}) \rightarrow (y_1, \ldots, y_{k+1}) \in P_{ccf}$.

Nach Induktionsannahme gibt es einen Morphismus $f' \in \mathcal{U}(P_{ccf})(u, v')$ und nach Definition enthält $E(P_{ccf})$ das Element $p(d_1, \ldots, d_k) := (p_1 \times \mathbf{1}_{d_1} \times p_2 \times \cdots \times p_k \times \mathbf{1}_{d_k} \times p_{k+1})$. Dann ist $f := (\mathbf{1}_{w_1} \times p(d_1, \ldots, d_k) \times \mathbf{1}_{w_2}) \circ f' \in \mathcal{U}(P_{ccf})(u,v)$.

Damit ist der Satz gezeigt. □

Diese Charakterisierung der Ableitungen einer gekoppelt-kontextfreien Grammatik als Elemente einer Unter–×–Kategorie einer kontextfreien ×–Kategorie könnte zur Lösung des Wortproblems „$w \in L(G_{ccf})$?“ hilfreich sein. Ein möglicher Algorithmus würde etwa zuerst eine kontextfreie Analyse für die zugrundeliegende kontextfreie Grammatik durchführen und bei positiver Antwort testen, ob eine der gefundenen Ableitungen in der Unter–×–Kategorie liegt, d.h. das „Wortproblem“ für die Unter–×–Kategorie lösen.

Es besteht die Hoffnung, daß sich für interessante Sprachklassen (z.B. für die *Tree Adjoining Languages*) dadurch effiziente Analysealgorithmen gewinnen lassen. Wir werden nämlich im folgenden zeigen, daß sich die Klasse der Tree Adjoining Languages durch gekoppelt-kontextfreie Grammatiken beschreiben läßt.

Es ist auch möglich, die gekoppelt-kontextfreien Ableitungen als Elemente einer endlich erzeugten Unter–×–Kategorie zu beschreiben. Dazu verwendet man die Konstruktion, die wir im Abschnitt über kontextfreie Sprachen dargestellt haben, d.h. die Terminalsymbole werden aus der Grammatik entfernt und stattdessen gibt man jeder Produktion p die Interpretation $\eta(p) := f_p$, die Abbildung, die beschreibt, wie p aus Teilwörtern ein neues Wort zusammensetzt.

Man erhält dann terminalfreie gekoppelt-kontextfreie Produktionen der Form

$$p : (x_1, \ldots, x_{k+1}) \rightarrow (y_1, \ldots, y_{k+1})$$

mit $y_1 \cdots y_{k+1} \in D(N)$, $y_i \in [N,g]^*$, $1 \leq i \leq k+1$.

Sei $\mathcal{F}'(G_{ccf})$ die von den „entkoppelten“ terminalfreien Produktionen erzeugte freie ×–Kategorie. Zur Beschreibung der Satzmenge braucht man die Morphismen $f : s \rightarrow \varepsilon$ aus $\mathcal{F}'(G_{ccf})$ und zwar genau die, die eine gekoppelte Ableitung beschreiben. Das sind genau die Elemente der Unter–×–Kategorie $\mathcal{U}'(P_{ccf})$, die von

$$E'(P_{ccf}) := \{p_1 \times \mathbf{1}_{d_1} \times p_2 \times \cdots \times p_k \times \mathbf{1}_{d_k} \times p_{k+1} \mid p \in P_{ccf},\ grad(p) = k\}$$

mit $d_i \in D(N),\ 1 \leq i \leq k$, erzeugt wird.

Ein Wort $w \in T^*$ ist genau dann in der Satzmenge $L(G_{ccf})$, wenn es eine Ableitung $f : s \to \varepsilon \in \mathcal{U}'(P_{ccf})$ gibt, so daß $\eta(f)(\Box) = w$.

Jeder solche Morphismus läßt sich erzeugen, indem in jedem Ersetzungsschritt zunächst „direkt nebeneinanderstehende“ Klammern bzw. triviale Klammern ersetzt werden, d.h. es wird immer ein Teilwort $x_1 \cdots x_{k+1},\ x \in N_k$ ersetzt. Da alle Klammern letztendlich zum leeren Wort abgeleitet werden, können auch zunächst auseinanderstehende Klammern schließlich ersetzt werden, nachdem die dazwischen stehenden Klammern zuerst abgeleitet wurden. Solch eine Ableitung „von innen nach außen“ benutzt also nur Ersetzungen der Form $p_1 \times \cdots \times p_{k+1}$ für $p \in P_{ccf}$ vom Grad k. Deshalb lassen sich dann alle Ableitungen aus dem endlichen Erzeugendensystem

$$E''(P_{ccf}) := \{p_1 \times \cdots \times p_{k+1} \mid p \in P_{ccf},\ k = grad(p)\}$$

erhalten.

Das war unter Einbeziehung der Terminalsymbole nicht möglich, weil bereits produzierte Terminalwörter Klammern voneinander trennen konnten und deshalb Erzeugende der Form

$$p_1 \times \mathbf{1}_{w_1} \times p_2 \times \cdots \times p_k \times \mathbf{1}_{w_k} \times p_{k+1}, \quad w_1, \ldots, w_k \in T^*$$

notwendig waren.

Wir werden in den nächsten Abschnitten den Zusammenhang zwischen den gekoppelt-kontextfreien Ersetzungen und Ersetzungen auf Bäumen untersuchen. Im folgenden Abschnitt geben wir die Definition von Grammatiken auf Bäumen und einige Eigenschaften dieser Systeme.

Kapitel 2

Baumgrammatiken mit multilinearer Interpretation

2.1 Baumgrammatiken

Wir definieren jetzt Grammatiken auf Bäumen, die den kontextfreien Grammatiken auf Wörtern nachempfunden sind. Wir beschreiben Bäume als Elemente von Baumkategorien. Wie bei den kontextfreien Grammatiken ist es erlaubt, einzelne Nichtterminalsymbole durch zusammengesetzte Objekte zu ersetzen, allerdings muß man hier darauf achten, daß die „Anschlußbedingungen“ erfüllt sind.

Definition 19 *Baumgrammatik*

Sei $\mathcal{B}_V := \mathcal{B}(V, Q, Z)$ eine Baumkategorie, $Z(v) = 1 \quad \forall v \in V$, sei $V = N \dot{\cup} T$, $N, T \neq \emptyset$, und sei $\mathcal{B}_T$ die von T erzeugte freie Unter$-\times-$Kategorie von $\mathcal{B}_V$. Die Elemente dieser Baumkategorie kann man sich, wie im Einführungskapitel dargestellt, als Netze veranschaulichen, die bezüglich der Operation $\times$ in Bäume und Einheiten zerfallen. Die Strecken sind alle gleich beschriftet, weshalb wir die Beschriftung ignorieren.

$G = (V, T, P, s, \stackrel{*}{\Rightarrow}_P)$ bzw. $G = (V, T, P, s)$ heißt *Baumgrammatik* auf $\mathcal{B}_T$, wenn gilt:

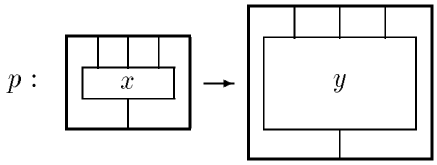
1. P ist eine endliche Menge von *Produktionen* $p : x \to y$ mit
   (a) $x \in N, \; y \in \mathcal{B}_V$.
   (b) $Q(x) = Q(y), \; Z(x) = Z(y) = 1$.
2. Die *Rewriting-Relation* $\stackrel{*}{\Rightarrow}_P$ ist der reflexive, transitive Abschluß der Relation $\Rightarrow_P$ auf $\mathcal{B}_V$. Für $f, \; f' \in \mathcal{B}_V$ ist

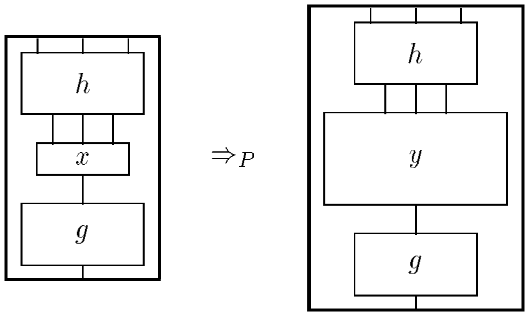
$$f \Rightarrow_P f' :\Longleftrightarrow \left\{ \begin{array}{l} f = g \circ x \circ h, \; f' = g \circ y \circ h \\ p : x \to y \in P, \; g \in \mathcal{B}_V(Z(x), \mathbf{N}_0), \; h \in \mathcal{B}_V(\mathbf{N}_0, Q(x)) \end{array} \right.$$

3. Das *Axiom* von G ist ein Element $s \in N$ mit $Q(s) = 0$.

Die Quelle-/Zielbedingung aus 1. stellt sicher, daß ein Vorkommen einer Variable x in einem Netz immmer durch Anwendung einer Produktion $p : x \to y$ ersetzt werden kann, d.h. daß durch die Ersetzung in jedem Fall ein wohldefiniertes Netz entsteht.

Wir verdeutlichen die Definitionen an folgenden Abbildungen:





Die von G erzeugte *Baumsprache* ist

$$T(G) := \left\{ f \mid s \overset{*}{\Rightarrow}_P f,\ \ f \in \mathcal{B}_T \right\}.$$

Einfache Folgerungen aus diesen Definitionen:

1. $f, g \in \mathcal{B}_V,\ \ f \overset{*}{\Rightarrow}_P g \quad \Longrightarrow \quad Q(f) = Q(g),\ \ Z(f) = Z(g).$
2. $T(G) \subseteq \mathcal{B}_T(0,1)$, d.h. eine Baumgrammatik erzeugt tatsächlich Bäume.
3. Sind $T(G_1)$ und $T(G_2)$ Baumsprachen, dann ist $T(G_1) \cup T(G_2)$ eine Baumsprache.

Wir werden sehen, daß für die Mächtigkeit einer Baumgrammatik die Nichterminale mit der größten Anzahl von Eingängen wesentlich sind und definieren daher:

Definition 20 *Grad einer Baumgrammatik*

$$grad\, G := \max \left\{ Q(x) \mid p : x \to y \in P \right\}.$$

Wir bezeichnen die Klasse der Baumsprachen in $\mathcal{B}_T$, die von einer Baumgrammatik vom Grad $k \in \mathbf{N}_0$ erzeugt werden, mit $CF(\mathcal{B}_T, k)$.

Die Begriffe *überflüssige Symbole, reduzierte Grammatik* seien wie bei Grammatiken auf Wörtern definiert. Genau wie dort zeigt man, daß es zu jeder Baumgrammatik eine äquivalente reduzierte Grammatik gibt. Wir gehen im folgenden immer von reduzierten Grammatiken aus.

Eine *Ableitung* von f_0 nach f_n in G ist eine Folge

$$f_0 \Rightarrow_P f_1 \Rightarrow_P \ldots \Rightarrow_P f_n$$

mit $f_i \in \mathcal{B}_V,\ 0 \le i \le n,\ n \in \mathbf{N}_0$. Die Zahl n heißt *Länge* der Ableitung.

Wir illustrieren unsere Definitionen an zwei Beispielen. An diesen Beispielen wird auch deutlich, welche Rolle der Grad einer Baumgrammatik für die erzeugte Baumsprache spielt.

Beispiel 1:
$V = \{s, f, g\}$, $T = \{f, g\}$, $Q(s) = Q(g) = 0,\ Q(f) = 2$, $Z(x) = 1\ \forall x \in V$.
$P = \{p : s \to g,\ q : s \to f \circ (\mathbf{1}_1 \times g) \circ s\}$.

$G = (V, T, P, s)$ erzeugt die Baumsprache $T(G) = \{(f \circ (\mathbf{1}_1 \times g))^n \circ g \mid n \in \mathbf{N}_0\}$.

Der Grad von G ist $0, T(G) \in CF(\mathcal{B}_T, 0)$. Grammatiken vom Grad 0 entsprechen den *regulären Baumgrammatiken* [GeSt84].

Beispiel 2:
$V = \{s, x, f, g, h\}$, $T = \{f, g, h\}$, $Q(s) = Q(g) = 0,\ Q(x) = Q(h) = 2,\ Q(f) = 1$,
$Z(v) = 1\ \forall v \in V$.
$P = \{p_1 : s \to x \circ (g \times g),\ p_2 : x \to h,\ p_3 : x \to f \circ x \circ (f \times f)\}$.

$G = (V, T, P, s)$ erzeugt die Baumsprache

$$T(G) = \{f^n \circ h \circ (f^n \times f^n) \circ (g \times g) \mid n \in \mathbf{N}_0\}.$$

Der Grad von G ist $2, T(G) \in CF(\mathcal{B}_T, 2)$.

Das folgenden Lemma zeigt, daß sich die Ableitungen in unsere Baumgrammatiken ähnlich den Ableitungen in kontextfreien Grammatiken „faktorisieren" lassen [Ha78].

Lemma 2.1.1 *Sei $G = (V, T, P, s)$ eine Baumgrammatik und seien $\alpha, \beta \in \mathcal{B}_V$.*
Wenn $\alpha \overset{r}{\Rightarrow}_G \beta,\ r \in \mathbf{N}_0$ und $\alpha = \alpha_1 \circ \alpha_2$, dann gibt es $r_1, r_2 \in \mathbf{N}_0,\ \beta_1, \beta_2 \in \mathcal{B}_V$ mit

$$r = r_1 + r_2, \quad \beta = \beta_1 \circ \beta_2, \quad \alpha_i \overset{r_i}{\Rightarrow}_G \beta_i,\ i = 1, 2.$$

Eine analoge Aussage gilt für $\alpha = \alpha_1 \times \alpha_2$.

Beweis: Induktion nach r.

Induktionsanfang: $\alpha = \alpha_1 \circ \alpha_2 \overset{0}{\Rightarrow}_G \beta$. Dann ist $\alpha = \beta$, setzt man $\beta_i := \alpha_i,\ r_i := 0,\ i = 1, 2$, dann gilt die Behauptung.

Induktionsschritt:
Sei $\alpha = \alpha_1 \circ \alpha_2 \overset{r+1}{\Rightarrow}_G \beta$. Dann gibt es ein $\gamma \in \mathcal{B}_V$ mit $\alpha \Rightarrow_G \gamma \overset{r}{\Rightarrow}_G \beta$.
Betrachte $\alpha \Rightarrow_G \gamma$. Es gibt eine Produktion $p : x \to y \in P,\ \alpha_1', \alpha_1'' \in \mathcal{B}_V,\ u, v \in \mathbf{N}_0$, so daß gilt (o.B.d.A. finde die Ersetzung in α_1 statt) :

$$\alpha_1 = \alpha_1' \circ (\mathbf{1}_u \times x \times \mathbf{1}_v) \circ \alpha_1''$$

Setzen wir

$$\begin{aligned} \gamma_1 &:= \alpha_1' \circ (\mathbf{1}_u \times y \times \mathbf{1}_v) \circ \alpha_1'', \\ \gamma_2 &= \alpha_2, \end{aligned}$$

dann gilt: $\alpha_1 \overset{1}{\Rightarrow}_G \gamma_1,\ \alpha_2 \overset{0}{\Rightarrow}_G \gamma_2$ und deshalb

$$\alpha = \alpha_1 \circ \alpha_2 \overset{1}{\Rightarrow}_G \gamma_1 \circ \gamma_2 \overset{r}{\Rightarrow}_G \beta$$

Nach Induktionsannahme gibt es $r_1, r_2, \beta_1, \beta_2$ derart, daß

$$\beta = \beta_1 \circ \beta_2, \quad r = r_1 + r_2, \quad \gamma_i \overset{r_i}{\Rightarrow}_G \beta_i, \ i = 1, 2.$$

Deshalb gilt:

$$\begin{aligned} \alpha_1 \overset{1}{\Rightarrow}_G \gamma_1 \overset{r_1}{\Rightarrow}_G \beta_1, &\qquad r_1' := r_1 + 1 \\ \alpha_2 \overset{0}{\Rightarrow}_G \gamma_2 \overset{r_2}{\Rightarrow}_G \beta_2, &\qquad r_2' := r_2 \end{aligned}$$

und somit gibt es $r_1', r_2', \beta_1, \beta_2$ mit

$$\beta = \beta_1 \circ \beta_2, \ \alpha_i \overset{r_i'}{\Rightarrow}_G \beta_i, \ i = 1, 2, \ r_1' + r_2' = r,$$

was zu zeigen war. □

Der Beweis für das Produkt $\alpha = \alpha_1 \times \alpha_2$ verläuft genauso. Dieses Lemma zeigt, daß unsere Baumgrammatiken sich wie kontextfreie Grammatiken auf Wörtern verhalten, allerdings auf anderen Strukturen arbeiten.

Wir zeigen jetzt, daß es zu jeder Baumgrammatik eine äquivalente Grammatik in Normalform gibt.

Lemma 2.1.2 *Zu jeder Baumgrammatik G gibt es eine äquivalente Baumgrammatik G', deren Produktionen alle von der Form 1. oder 2. sind:*

1. $p : x \to y, \qquad y \in \mathcal{B}_N$
2. $p : x \to y, \qquad y \in T$

Beweis: Wir gehen o.B.d.A. davon aus, daß keine Produktionen $p : x \to \mathbf{1}_1$ vorkommen. Diese können nämlich genau wie bei kontextfreien Grammatiken eliminiert werden. Ist $p : x \to y$ eine Produktion, die nicht von der Form 1. oder 2. ist, dann führt man zu jedem Terminalsymbol t, das in y vorkommt, eine neues Nichtterminal x_t mit $Q(x_t) := Q(t)$, $Z(x_t) := Z(t)$ und eine neue Produktion $p_t : x_t \to t$ ein. In y ersetzt man alle Vorkommen von t durch die Variable x_t. Dies führt man auch für die übrigen Terminalsymbole in y durch. Durch Anwendung dieser Modifikation auf alle Produktionen, die die gewünschte Form noch nicht besitzen, erhält man eine Grammatik in Normalform, die offensichtlich die gleiche Baumsprache erzeugt.

Die angegebene Normalform läßt sich noch so modifizieren, daß die nichtterminalen rechten Seiten Bäume der „Höhe“ 2 sind.

Lemma 2.1.3 *Zu jeder Baumgrammatik G gibt es eine äquivalente Baumgrammatik G', deren Produktionen alle von der Form 1. oder 2. sind:*

1. $p : x \to x_0 \circ (x_1 \times \cdots \times x_k) \qquad x_i \in N, \ 0 \le i \le k, \ Q(x_0) = k,$
2. $p : x \to y, \qquad y \in T.$

Beweis: Sei G bereits in der Normalform des vorigen Lemmas. Dann müssen wir nur noch Produktionen der Form

$$p : x \to x_0 \circ (y_1 \times \cdots \times y_k), \qquad x_0 \in N, \ y_i \in \mathcal{B}_N, \ 1 \le i \le k, \ Q(x_0) = k$$

betrachten, bei denen eines der $y_i \notin N$ und keine Einheit ist.

Wir führen für jedes solche y_i eine neue Variable z_i mit $Q(z_i) := Q(y_i)$, $Z(z_i) := Z(y_i) = 1$ ein und ersetzen in der rechten Seite der Produktion p dieses y_i durch z_i. Zusätzlich nehmen wir neue Produktionen $z_i \to y_i$ auf und wiederholen den Vorgang. Es ist klar, daß dieses Verfahren die gewünschte Normalform liefert und daß die resultierende Baumgrammatik äquivalent zur Ausgangsgrammatik ist.

2.2 Multilineare Interpretation von Baumsprachen

In diesem Abschnitt definieren wir eine „Übersetzung“ der von Baumgrammatiken erzeugten Baumsprachen in Wörtersprachen.

Wir ordnen jedem Element des Erzeugendensystems einer Baumkategorie eine (*multilineare*) Abbildung zu. Dadurch ist für jedes Netz eine *multilineare Interpretation* definiert.

Jeder Baumsprache wird durch diese Interpretation eine Wörtersprache zugeordnet. Wir gehen immer von einer Interpretation der Terminale einer Baumgrammatik aus und erweitern diese in kanonischer Weise auf das Gesamtalphabet der Baumgrammatik. Dadurch ist jeder „Satzform“, d.h. jedem vom Axiom ableitbaren Baum ein Wort spezieller Struktur zugeordnet.

Die Regeln der Baumgrammatik übersetzen sich in *gekoppelt-kontextfreie Regeln* und die Rewriting-Relation der Baumgrammatik übersetzt sich in *gekoppelt-kontextfreies Rewriting* auf Wörtern.

Wir geben zunächst die formale Definition der oben angesprochenen Interpretation. Die unten verwendete $\times$–Kategorie $ABB(B, \mathbf{N}_0)$ wurde im Einführungskapitel definiert.

Definition 21 *multilineare Abbildung*
Eine Abbildung $f : (\Delta^*)^m \to \Delta^*,\ m \in \mathbf{N}_0$, heißt *multilinear*, wenn es Wörter $w_1, \ldots, w_{m+1} \in \Delta^*$ gibt, sodaß

$$f(x_1, \ldots, x_m) = w_1 \cdot x_1 \cdot w_2 \ldots w_m \cdot x_m \cdot w_{m+1} \qquad \forall x_i \in \Delta^*,\ 1 \leq i \leq m.$$

Eine Abbildung $f : (\Delta^*)^m \to (\Delta^*)^n,\ m, n \in \mathbf{N}_0$, heißt *multilinear*, wenn f eine Zerlegung $f = f_1 \times \cdots \times f_n$ in multilineare Abbildungen $f_i : (\Delta^*)^{m_i} \to \Delta^*,\ m_i \in \mathbf{N}_0,\ 1 \leq i \leq n$, besitzt. (Für $n = 0$ ist f genau dann multilinear, wenn $f = \mathbf{1}_0$.)

Lemma 2.2.1 *Ist F eine Menge von multilinearen Abbildungen $f \in ABB(\Delta^*, \mathbf{N}_0)$ dann enthält die von F in $ABB(\Delta^*, \mathbf{N}_0)$ erzeugte Unter–$\times$–Kategorie $\mathcal{C}(F)$ ausschließlich multilineare Abbildungen.*

Beweis:
Es genügt zu zeigen, daß durch Komposition und Parallelausführung multilinearer Abbildungen wieder multilineare Abbildungen entstehen.

Für die Parallelausführung ist die Behauptung nach Definition unmittelbar klar. Wir betrachten jetzt die Komposition.

Seien $f \in \mathcal{C}(F)(m, n),\ g \in \mathcal{C}(F)(l, m),\ l, m, n \in \mathbf{N}_0$, multilinear. Dann gibt es nach Definition multilineare Abbildungen $f_1, \ldots, f_n$ und $g_1, \ldots, g_m$ in $ABB(\Delta^*, \mathbf{N}_0)$ mit $f = f_1 \times \cdots \times f_n,\ g = g_1 \times \cdots \times g_m,\ m_i := Q(f_i),\ Z(f_i) = 1,\ l_j := Q(g_j),\ Z(g_j) = 1.$

Bemerkung: Die Abbildungen f_i und g_j müssen nicht unbedingt Elemente von $\mathcal{C}(F)$ sein. Es ist mit obigen Bezeichnungen $m = \sum_{i=1}^n m_i$ und $l = \sum_{j=1}^m l_j$.

Für die Komposition $f \circ g$ gilt dann:

$$\begin{aligned}(f \circ g)(x_1, \ldots, x_l) &= f(\ g(x_1, \ldots, x_l)\ ) \\ &= f(\ y_1, \ldots, y_m\ )\end{aligned}$$

mit $(\ y_1, \ldots, y_m\ ) := (g_1 \times \cdots \times g_m)(\ x_1, \ldots, x_l\ )$.

$$\begin{aligned}
(\, y_1, \ldots, y_m \,) &= (\, g_1(x_1, \ldots, x_{l_1}),\ \ldots,\ g_m(x_{l-l_m+1}, \ldots, x_l)\,) \\
&= (\, u_{1,1}x_1u_{1,2}\ldots u_{1,l_1}x_{l_1}u_{1,l_1+1},\ \ldots,\ u_{m,1}x_{l-l_m+1}u_{m,2}\ldots u_{m,l_m}x_lu_{m,l_m+1}\,)
\end{aligned}$$

$$\begin{aligned}
f(\, y_1, \ldots, y_m \,) &= (\, f_1(y_1, \ldots, y_{m_1}),\ \ldots,\ f_n(y_{m-m_n+1}, \ldots, y_m)\,) \\
&= (\, v_{1,1}y_1v_{1,2}\ldots v_{1,m_1}y_{m_1}v_{1,m_1+1},\ \ldots,\ v_{n,1}y_{m-m_n+1}v_{n,2}\ldots v_{n,m}y_mv_{n,m+1}\,)
\end{aligned}$$

Die $u_{j,*}$, $v_{i,*}$ in obigen Gleichungen sind Elemente aus Δ^*.

Setzen wir

$$\begin{aligned}
h_1 &:= f_1 \circ (g_1 \times \cdots \times g_{m_1}) \\
&\vdots \\
h_n &:= f_n \circ (g_{m-m_n+1} \times \cdots \times g_m)
\end{aligned}$$

dann sieht man aus den vorherigen Gleichungen, daß jedes h_i eine multilineare Abbildung in $ABB(\Delta^*, \mathbf{N}_0)$ ist und aus

$$\begin{aligned}
(f \circ g) &= (f_1 \times \cdots \times f_n) \circ (g_1 \times \cdots \times g_m) \\
&= f_1 \circ (g_1 \times \cdots \times g_{m_1}) \times \cdots \times f_n \circ (g_{m-m_n+1} \times \cdots \times g_m) \\
&= h_1 \times \cdots \times h_n
\end{aligned}$$

folgt die Aussage des Lemmas. □

Wir wollen jetzt den von einer Baumgrammatik erzeugten Bäumen eine Interpretation dieser Art geben.

Sei $G = (V, T, P, s)$ eine Baumgrammatik, $N := V \setminus T$ das Nichtterminalalphabet von G, und seien $\mathcal{B}_V := \mathcal{B}(V, Q, Z)$, $\mathcal{B}_T := \mathcal{B}(T, Q|_T, Z|_T)$ die von V, T erzeugten Baumkategorien. Δ_T sei ein Alphabet und Δ_T^* das Wörtermonoid über Δ_T.

Jedem Terminalsymbol $t \in T$, $Q(t) = k \in \mathbf{N}_0$, sei eine multilineare Abbildung

$$f_t : (\Delta_T^*)^k \to \Delta_T^* \quad \in ABB(\Delta_T^*, \mathbf{N}_0)$$

zugeordnet, d.h.

$$f_t(w_1, \ldots, w_k) = u_1 \cdot w_1 \cdot u_2 \ldots u_k \cdot w_k \cdot u_{k+1}, \qquad u_1, \ldots, u_{k+1} \in \Delta_T^*.$$

Sei $F_T := \{f_t \mid t \in T\}$ und $\mathcal{C}(F_T)$ die von den Abbildungen aus F_T erzeugte Unter–$\times$–Kategorie von $ABB(\Delta_T^*, \mathbf{N}_0)$. Wie in Lemma 2.2.1 gezeigt wurde, enthält $\mathcal{C}(F_T)$ ausschließlich multilineare Abbildungen.

Sei η_1 die Identität auf $\mathbf{N}_0$ und $\eta_2' : T \to \mathcal{C}(F_T)$, $\eta_2'(t) := f_t$.

Diese beiden Abbildungen induzieren eindeutig den $\times$–Funktor $\eta = (\eta_1, \eta_2)$ von der (freien) $\times$–Kategorie $\mathcal{B}_T$ in die $\times$–Kategorie von Abbildungen $\mathcal{C}(F_T)$.

Wir nennen η eine *multilineare Interpretation für* $\mathcal{B}_T$.

Jedem Element aus $\mathcal{B}_T$ ist durch η eine multilineare Abbildung als Bedeutung zugeordnet. Die Baumsprache $T(G)$ wird durch η in eine Wörtersprache über Δ_T abgebildet, d.h. die Baumsprache wird durch η in eine Wörtersprache übersetzt.

Wir wollen jetzt die Interpretation η kanonisch auf $\mathcal{B}_V$ erweitern und dadurch jeder „Satzform" der Grammatik G eine Interpretation geben.

Dazu sei $\Delta_N := [N, Q|_N]$ das zu N gehörende Klammeralphabet, also

$$\Delta_N = \{x_i \mid x \in N,\ 1 \leq i \leq Q(x) + 1\}.$$

Wir setzen $\Delta := \Delta_N \cup \Delta_T$ und erweitern zunächst die Abbildungen $f_t,\ t \in T$, kanonisch auf Δ^*.

Für jedes Nichtterminalsymbol $x \in N,\ Q(x) = k$, definieren wir die Abbildung

$$f_x : (\Delta^*)^k \to \Delta^*$$

durch

$$f_x(w_1, \ldots, w_k) := x_1 \cdot w_1 \cdot x_2 \ldots x_k \cdot w_k \cdot x_{k+1} \qquad \forall w_1, \ldots, w_k \in \Delta^*.$$

Sei $F_V := \{f_v \mid v \in V\}$ und $\mathcal{C}(F_V)$ die von F_V erzeugte Unter–$\times$–Kategorie von $ABB(\Delta^*, \mathbf{N}_0)$. Wir erweitern $\eta_2' : T \to \mathcal{C}(F_T)$ durch $\eta_2'(x) := f_x,\ x \in N$, auf V. Wie oben definieren η_1 und η_2' eindeutig den $\times$–Funktor $\eta = (\eta_1, \eta_2)$ von $\mathcal{B}_V$ in $\mathcal{C}(F_V)$.

Wir nennen η eine *multilineare Interpretation für die Baumgrammatik G*.

Die kanonische Erweiterung einer multilinearen Interpretation der terminalen Baumkategorie $\mathcal{B}_T$ besteht also darin, jedes Nichtterminalsymbol der Baumgrammatik als eine Abbildung zu interpretieren, die ihre Argumente mit der entsprechenden Klammer umschließt. Diese Klammer markiert Positionen im Wort, an denen noch Ersetzungen möglich sind. In den Elementen der erzeugten Baumsprache $T(G)$ treten keine Klammern mehr auf.

Definition 22 *Baumgrammatik mit multilinearer Interpretation*
Ist $G = (V, T, P, s)$ eine Baumgrammatik und ist $\eta_T : \mathcal{B}_T \to \mathcal{C}(F_T)$ eine multilineare Interpretation für die terminale Baumkategorie $\mathcal{B}_T$, $\eta : \mathcal{B}_V \to \mathcal{C}(F_V)$ die kanonische Erweiterung auf $\mathcal{B}_V$, dann heißt (G, η) eine *Baumgrammatik mit multilinearer Interpretation*, abgekürzt BGmI.
Die von einer BGmI (G, η) erzeugte *Satzmenge* $L(G, \eta)$ ist

$$L(G, \eta) = \{\eta(\Psi)(\Box) \mid \Psi \in T(G)\}.$$

Der *Grad* einer BGmI sei der Grad der zugrundeliegenden Baumgrammatik. Die Klasse der Wörtersprachen, die Satzmenge einer BGmI vom Grad k sind, bezeichnen wir mit BGmI(k).
Wir geben im nächsten Abschnitt einige Eigenschaften der Klassen BGmI(k) und untersuchen dann den Zusammenhang mit gekoppelt-kontextfreien Eretzungen.

2.3 Einige Eigenschaften der Klassen BGmI(k)

Lemma 2.3.1 *BGmI(k) ist abgeschlossen unter Vereinigung.*

Beweis: Seien (G,η), (G',η') BGmIs vom Grad $k \in \mathbf{N}_0$.

$G = (V,T,P,s)$, $\quad \eta_T : \mathcal{B}_T \to \mathcal{C}(F_T, \Delta_T^*)$, $\quad G' = (V',T',P',s')$, $\quad \eta'_T : \mathcal{B}'_T \to \mathcal{C}(F'_T, \Delta_T'^*)$.

Wir nehmen ohne Einschränkung an, daß außer Δ_T, Δ'_T alle Alphabete disjunkt sind und definieren

$$G'' := (V \cup V' \cup \{s''\},\ T \cup T',\ P \cup P' \cup \{p''_1 : s'' \to s,\ p''_2 : s'' \to s'\},\ s'')$$

Die Quelle- und Zielabbildungen seien

$$Q'' := Q \cup Q' \cup \{(s'',0)\},\ Z'' := Z' \cup Z \cup \{(s'',1)\}$$

Die Interpretation $\eta_{T''}$ sei analog definiert. Dann erhalten wir:

1. $T(G'') = T(G) \cup T(G')$.
2. $grad(G'') = \max\{grad(G), grad(G')\} = k$.
3. $\eta''(f) = \begin{cases} \eta(f),\ f \in \mathcal{B}_V \\ \eta'(f),\ f \in \mathcal{B}_{V'} \end{cases}$
4. $L(G'',\eta'') = L(G,\eta) \cup L(G',\eta')$. □

Lemma 2.3.2 *BGmI(k) ist abgeschlossen unter Konkatenation.*

Beweis: Seien (G,η), (G',η') wie oben. Wir setzen

$$G'' := (V \cup V' \cup \{s''\},\ T \cup T' \cup \{[conc_2]\},\ P \cup P' \cup \{p''_1 : s'' \to [conc_2] \circ (s \times s')\},\ s'')$$

Dabei seien Q'', Z'' für s'' wie oben definiert und $Q''([conc_2]) := 2,\ Z([conc_2]) := 1$. $\eta''([conc_2])$ sei die zweistellige Konkatenation, also eine multilineare Abbildung. Dann gilt

1. $grad(G'') = k$.
2. $T(G'') = \{[conc_2] \circ (f \times f') \mid f \in T(G),\ f' \in T(G')\}$.
3. $L(G'',\eta'') = L(G,\eta) \cdot L(G',\eta')$. □

Lemma 2.3.3 *BGmI(k) ist abgeschlossen unter Kleene-Abschluß.*

Beweis: Sei (G,η) wie oben. Definiere

$$G'' := (V \cup \{s''\},\ T \cup \{[conc_2],[eps]\},\ P \cup \{p''_1 : s'' \to [eps],\ p''_2 : s'' \to [conc_2] \circ (s \times s''),\ s'')$$

Es seien s'', $[conc_2]$ wie oben und $Q''([eps]) := 0,\ Z''([eps]) := 1,\ \eta''([eps])(\Box) := \varepsilon$.

Dann gilt:

1. $T(G'') = \{([conc_2] \circ (s \times \mathbf{1}_1))^n \circ [eps] \mid n \in \mathbf{N}_0\}$.
2. $L(G'',\eta'') = \{w_1 \cdots w_n \cdot \varepsilon \mid w_i \in L(G,\eta),\ n \in \mathbf{N}_0\} = L(G,\eta)^*$. □

Lemma 2.3.4 *$BGmI(k)$ ist abgeschlossen unter Homomorphismus.*

Beweis: Sei (G, η) wie oben, und sei $h : \Delta_T^* \to \Sigma_T^*$ ein Homomorphismus. Im ersten Schritt erweitere man den Homomorphismus h kanonisch auf ganz Δ^*.

Ist t Terminalsymbol, $\eta(t) = f_t : (\Delta^*)^k \to \Delta^*$ mit

$$f_t(w_1, \ldots, w_k) = u_1 \cdot w_1 \cdot u_2 \ldots u_k \cdot w_k \cdot u_{k+1}, \quad u_j \in \Delta_T^*,$$

dann definieren wir die neue Interpretation $\eta''(t) := f_t''$ durch

$$f_t''(w_1, \ldots, w_k) := h(u_1) \cdot w_1 \cdot h(u_2) \ldots h(u_k) \cdot w_k \cdot h(u_{k+1})$$

Dann gilt:

$$L(G'', \eta'') = h(L(G, \eta))$$

□

Lemma 2.3.5 *$COPY_k \in BGmI(k) \quad \forall k \in \mathbf{N}_0$.*

Beweis:

Sei $G_k := (V_k, T_k, P_k, s)$ mit $V_k := \{s, x, [\varepsilon, a], [\varepsilon, b], [a, \varepsilon], [b, \varepsilon], [\varepsilon], [conc_k]\}, \quad T_k := V_k \setminus \{s, x\}$.
Sei $Q(s) = Q([\varepsilon]) = 0,\ Q(x) = k,\ Q(v) = 1$ für die übrigen Nichtterminale, $Z(v) = 1\ \forall v \in V_k$.

$$\begin{aligned} P_k := \{ \quad & p_1\colon\ s \to x \circ ([\varepsilon] \times \cdots \times [\varepsilon]) \\ & p_2\colon\ x \to [a, \varepsilon] \circ x \circ ([\varepsilon, a] \times \cdots \times [\varepsilon, a]) \\ & p_3\colon\ x \to [b, \varepsilon] \circ x \circ ([\varepsilon, b] \times \cdots \times [\varepsilon, b]) \\ & p_4\colon\ x \to [conc_k] \quad \} \end{aligned}$$

Die Interpretation der Terminalsymbole sei

$$\begin{aligned} \eta([\varepsilon, a])(w) &:= \varepsilon \cdot w \cdot a \\ \eta([\varepsilon, b])(w) &:= \varepsilon \cdot w \cdot b \\ \eta([a, \varepsilon])(w) &:= a \cdot w \cdot \varepsilon \\ \eta([b, \varepsilon])(w) &:= b \cdot w \cdot \varepsilon \\ \eta([\varepsilon])(\Box) &:= \varepsilon \\ \eta([conc_k])(w_1, \ldots, w_k) &:= w_1 \cdots w_k \end{aligned}$$

Wie man leicht überprüft, gilt

$$\begin{aligned} T(G_k) = \{ \quad & [a_1, \varepsilon] \circ \cdots \circ [a_n, \varepsilon] \circ conc_k \circ \\ & ([\varepsilon, a_n] \times \cdots \times [\varepsilon, a_n]) \circ \cdots \circ ([\varepsilon, a_1] \times \cdots \times [\varepsilon, a_1]) \circ ([\varepsilon] \times \cdots \times [\varepsilon]) \mid \\ & n \in \mathbf{N}_0,\ a_i \in \{a, b\},\ 1 \le i \le n \quad \} \end{aligned}$$

Dann ist

$$L(G_k, \eta) = \left\{ (a_1 \cdots a_n) \cdot (a_1 \cdots a_n)^k \mid a_i \in \{a, b\},\ n \in \mathbf{N}_0 \right\} = COPY_k \Box$$

Lemma 2.3.6 *$COUNT_{2k+2} \in BGmI(k) \quad \forall k \in \mathbf{N}_0$.*

Beweis:

Sei $G_k := (V_k, T_k, P_k, s)$
$V_k := \{s, x, [a_1, a_{2k+2}], [a_2, a_3], \ldots, [a_{2k}, a_{2k+1}], [\varepsilon], [conc_k]\}, \qquad T_k := V_k \setminus \{s, x\}$.

Sei $Q(s) = 0,\ Q(x) = k,\ Q([\varepsilon]) = 0,\ Q([conc_k] = k,\ Q(v) = 1$ für die übrigen Symbole, $Z(v) = 1\ \forall v \in V_k$.

$P_k := \{\quad p_1\colon\ s \to x \circ ([\varepsilon] \times \cdots \times [\varepsilon])$
$p_2\colon\ x \to [a_1, a_{2k+2}] \circ x \circ ([a_2, a_3] \times \cdots \times [a_{2k}, a_{2k+1}])$
$p_3\colon\ x \to conc_k \quad \}$

Die Interpretation der Terminalsymbole sei

$$\begin{aligned} \eta([a_i, a_j])(w) &:= a_i \cdot w \cdot a_j \\ \eta([\varepsilon])(\Box) &:= \varepsilon \\ \eta([conc_k])(w_1, \ldots, w_k) &:= w_1 \cdots w_k \end{aligned}$$

Dann gilt

$$T(G_k) = \{[a_1, a_{2k+2}]^n \circ [conc] \circ ([a_2, a_3] \times \cdots \times [a_{2k}, a_{2k+1}])^n \mid n \in \mathbf{N}_0\}$$

und

$$L(G_k, \eta) = \{a_1^n \cdots a_{2k+2}^n \mid n \in \mathbf{N}_0\} = COUNT_{2k+2} \Box$$

2.4 Baumgrammatiken mit multilinearer Interpretation und gekoppelt-kontextfreie Ersetzungen

Wir untersuchen in diesem Abschnitt den Zusammenhang zwischen Baumgrammatiken mit multilinearer Interpretation und gekoppelt-kontextfreien Grammatiken.

Im ersten Schritt zeigen wir, daß jeder BGmI eine gekoppelt-kontextfreie Grammatik zugeordnet werden kann, die die gleiche Satzmenge erzeugt.

Sei (G, η) eine BGmI, $G = (V, T, P, s)$, $N := V \setminus T$, $\mathcal{B}_V := \mathcal{B}(V, Q, Z)$ die zugehörige Baumkategorie. Das Alphabet für die Interpretation der Terminale von G sei Δ_T und $D(\Delta_N, \Delta_T)$ sei die zu $\Delta_N := [N, Q|_N]$ und Δ_T gehörende Klammersprache, $\Delta := \Delta_N \cup \Delta_T$. Das bedeutet, wir ordnen jedem nichtterminalen Baustein der Baumgrammatik mit k Eingängen eine Klammer vom Grad k zu.

Wir setzen
$$G_{ccf} := (\Delta, \Delta_T, P_{ccf}, \sigma)$$
mit
$$\begin{aligned} P_{ccf} := \{ \quad & p : (x_1, \ldots, x_{k+1}) \to (y_1, \ldots, y_{k+1}) \mid \\ & p : x \to y \in P,\ Q(x) = k,\ \eta(y)(w_1, \ldots, w_k) = y_1 \cdot w_1 \cdot y_2 \ldots y_k \cdot w_k \cdot y_{k+1} \quad \} \end{aligned}$$

Das Axiom von G_{ccf} sei $\sigma := \eta(s)(\Box)$.

Wir fassen also eine Produktion der Baumgrammatik für ein Nichtterminal mit k Eingängen als gekoppelt-kontextfreie Regel aus $k+1$ Komponenten auf. Die rechten Seiten der Komponenten ergeben sich aus der Interpretation der rechten Seite der Produktion der Baumgrammatik. Man erkennt, daß man für eine reguläre Ersetzung im Baum eine einzelne kontextfreie Ersetzung erhält.

Wir zeigen zunächst, daß G_{ccf} tatsächlich eine gekoppelt-kontextfreie Grammatik ist.

Die Alphabete Δ_N, Δ_T erfüllen offensichtlich die Definition und das Axiom σ ist ein Element von N_0. Wir müssen also nur noch nachprüfen, ob die Produktionen von G_{ccf} der Definition genügen. Dazu zeigen wir folgendes

Lemma 2.4.1 *Ist $f \in \mathcal{B}_V(k, 1)$, $k \in \mathbf{N}_0$, und sind $w_1, \ldots, w_k \in D(\Delta_N, \Delta_T)$, dann gilt:*
$$\eta(f)(w_1, \ldots, w_k) \in D(\Delta_N, \Delta_T)$$

Beweis: Induktion über $|f|$
Induktionsanfang: $f \in V$

1. $f \in T$: $\quad w := \eta(f)(w_1, \ldots, w_k) = u_1 \cdot w_1 \cdot u_2 \ldots u_k \cdot w_k \cdot u_{k+1}$, $u_i \in \Delta_T^*$,
   $w \in D(\Delta_N, \Delta_T)$ nach Definition von $D(\Delta_N, \Delta_T)$.

2. $f \in N$: $\quad w := \eta(f)(w_1, \ldots, w_k) = f_1 \cdot w_1 \cdot f_2 \ldots f_k \cdot w_k \cdot f_{k+1}$, $f \in N_k$,
   $w \in D(\Delta_N, \Delta_T)$ nach Definition von $D(\Delta_N, \Delta_T)$.

Induktionsschritt: $f = v \circ (g_1 \times \cdots \times g_l)$, $v \in V$, $g_i \in \mathcal{B}_V(k_i, 1)$, $1 \le i \le l = Q(v)$, $\sum_{i=1}^{l} k_i = k$.
Nach Induktionsvoraussetzung gilt:
$$\begin{aligned} w_1, \ldots, w_{k_1} \in D(\Delta_N, \Delta_T) \quad &\Longrightarrow \quad \eta(g_1)(w_1, \ldots, w_{k_1}) \in D(\Delta_N, \Delta_T) \\ &\vdots \\ w_{k-k_l+1}, \ldots, w_k \in D(\Delta_N, \Delta_T) \quad &\Longrightarrow \quad \eta(g_l)(w_{k-k_l+1}, \ldots, w_k) \in D(\Delta_N, \Delta_T) \end{aligned}$$

Für v gilt entsprechend: $u_1, \ldots, u_l \in D(\Delta_N, \Delta_T) \Longrightarrow \eta(v)(u_1, \ldots, u_l) \in D(\Delta_N, \Delta_T)$.

Damit erhalten wir insgesamt: $w_1, \ldots, w_k \in D(\Delta_N, \Delta_T) \quad \Longrightarrow$

$$\eta(f)(w_1, \ldots, w_k) = \eta(v)\left(\eta(g_1)(w_1, \ldots, w_{k_1}), \ldots, \eta(g_l)(w_{k-k_l+1}, \ldots, w_k)\right) \in D(\Delta_N, \Delta_T)$$

Damit ist das Lemma bewiesen. □

Wir erhalten für jede Produktion $p : x \to y \in P$:

$$\eta(y)(\varepsilon, \ldots, \varepsilon) = y_1 \cdots y_{k+1} \in D(\Delta_N, \Delta_T),$$

und damit erfüllt jede Produktion $p : (x_1, \ldots, x_{k+1}) \to (y_1, \ldots, y_{k+1}) \in P_{ccf}$ die Definition der gekoppelt-kontextfreien Grammatik.

Wir zeigen jetzt, daß die Satzmenge der gekoppelt-kontextfreien Grammatik G_{ccf} die Satzmenge der BGmI (G, η) umfaßt.

Lemma 2.4.2 *Seien (G, η) und G_{ccf} wie oben. Dann gilt für alle $f \in \mathcal{B}_V$:*

$$s \overset{*}{\Rightarrow}_G f \quad \Longrightarrow \quad \sigma \overset{*}{\Rightarrow}_{ccf} \eta(f)(\Box)$$

Beweis: Wir führen Induktion über die Länge der Ableitung $s \overset{*}{\Rightarrow}_G f$.

Induktionsanfang: Dann ist $f = s$, $\eta(f)(\Box) = \sigma$, und damit gilt die Behauptung.

Induktionsschritt: Sei $s \overset{*}{\Rightarrow}_G f$ eine Ableitung positiver Länge. Dann gibt es $g, h \in \mathcal{B}_V$ und $p : x \to y \in P$ derart, daß

$$s \overset{*}{\Rightarrow}_G g \circ x \circ h \Rightarrow_G g \circ y \circ h = f,$$

wobei $Q(g) = Z(g) = 1$, $Q(h) = 0$, $Z(h) = Q(x) = Q(y) = k$.

Nach Lemma 2.2.1 gibt es $u_1, u_2, w_1, \ldots, w_k \in \Delta^*$ mit

$$\begin{aligned} \eta(g)(w) &= u_1 \cdot w \cdot u_2 \quad \forall w \in \Delta^*, \\ \eta(h)(\Box) &= (w_1, \ldots, w_k), \end{aligned}$$

und h besitzt eine Zerlegung $h = (h_1 \times \cdots \times h_k)$ mit $h_i \in \mathcal{B}_V(0,1)$, $\eta(h_i)(\Box) = w_i$, $1 \le i \le k$.

Nach Lemma 2.4.1 sind $w_1, \ldots, w_k \in D(\Delta_N, \Delta_T)$, nach Konstruktion von G_{ccf} enthält P_{ccf} die Ersetzungsregel $p : (x_1, \ldots, x_{k+1}) \to (y_1, \ldots, y_{k+1})$.

Wir erhalten

$$\eta(g \circ x \circ h)(\Box) = u_1 \cdot x_1 \cdot w_1 \cdot x_2 \ldots x_k \cdot w_k \cdot x_{k+1} \cdot u_2$$

und

$$\eta(g \circ y \circ h)(\Box) = u_1 \cdot y_1 \cdot w_1 \cdot y_2 \ldots y_k \cdot w_k \cdot y_{k+1} \cdot u_2$$

Aus der Definition von $\Rightarrow_{ccf}$ und der Induktionsannahme folgt

$$\sigma \overset{*}{\Rightarrow}_{ccf} \eta(g \circ x \circ h)(\Box) \Rightarrow_{ccf} \eta(g \circ y \circ h)(\Box) = \eta(f)(\Box),$$

und damit ist das Lemma bewiesen. □

Wir zeigen jetzt, daß die Satzmengen genau übereinstimmen.

Lemma 2.4.3 *Sei $f \in \mathcal{B}_V(0,1)$.*

Ist $\eta(f)(\Box) = x_1 \cdot w_1 \cdot x_2 \ldots x_k \cdot w_k \cdot x_{k+1},\ x \in N,\ w_i \in D(\Delta_N, \Delta_T),\ 1 \leq i \leq k = Q(x)$, dann gilt:

Es gibt $h_1, \ldots, h_k \in \mathcal{B}_V(0,1),\ g \in \mathcal{B}_V(1,1)$ derart, daß $\eta(h_i)(\Box) = w_i,\ 1 \leq i \leq k,\ \eta(g) = id_{\Delta^*}$ und

$$f = g \circ x \circ (h_1 \times \cdots \times h_k).$$

Beweis: Induktion über $|f|$.

Induktionsanfang: $|f| = 1$

Dann ist $f \in N,\ k = 0$ und für $g := \mathbf{1}_1$ folgt die Behauptung.

Induktionsschritt: $|f| > 1$:

1. $f = y \circ (h'_1 \times \cdots \times h'_l),\ y \in N,\ Q(y) = l,\ h'_i \in \mathcal{B}_V(0,1)$
   $\Longrightarrow \eta(f)(\Box) = y_1 \cdot \eta(h'_1)(\Box) \cdot y_2 \ldots y_l \cdot \eta(h'_l)(\Box) \cdot y_{l+1}$.

   Nach Lemma 2.4.1 gilt $\eta(h'_i)(\Box) \in D(\Delta_N, \Delta_T)$ und nach Voraussetzung ist $\eta(f)(\Box) \in E(\Delta_N, \Delta_T)$, dem freien Erzeugendensystem von $D(\Delta_N, \Delta_T)$. Aus der Eindeutigkeit der Darstellung bzgl. $E(\Delta_N, \Delta_T)$ folgt $y = x,\ l = k$ und mit $g := \mathbf{1}_1$ folgt die Behauptung.

2. $f = t \circ (h'_1 \times \cdots \times h'_l),\ t \in T,\ Q(t) = l,\ h'_i \in \mathcal{B}_V(0,1)$
   $\Longrightarrow \eta(f)(\Box) = u_1 \cdot \eta(h'_1)(\Box) \cdot u_2 \ldots u_l \cdot \eta(h'_l)(\Box) \cdot u_{l+1},\ u_i \in \Delta_T^*$.

   Wegen $\eta(f)(\Box) = x_1 \cdot w_1 \cdot x_2 \ldots x_k \cdot w_k \cdot x_{k+1}$ ist $u_1 = u_{l+1} = \varepsilon$ und es gibt ein $j \in [1..l]$ mit $u_2, \ldots, u_j = \varepsilon,\ \eta(h'_k)(\Box) = \varepsilon,\ k < j,\ \eta(h'_j)(\Box) = x_1 \cdot \alpha,\ \alpha \in \Delta^*$.

   Analog gibt es ein $m \in [1..l]$ mit $u_{m+1}, \ldots, u_l = \varepsilon,\ \eta(h'_k)(\Box) = \varepsilon,\ k > m,\ \eta(h'_m)(\Box) = \beta \cdot x_1,\ \beta \in \Delta^*$.

   Es gilt $h'_j \in \mathcal{B}_V(0,1)$, also $\eta(h'_j)(\Box) \in D(\Delta_N, \Delta_T)$. Wegen $\eta(h'_j)(\Box) = x_1 \cdot \alpha$ muß $\eta(h'_j)(\Box)$ im freien Erz. system $E(\Delta_N, \Delta_T)$ von $D(\Delta_N, \Delta_T)$ liegen. Mit dem gleichen Argument folgt $\eta(h'_m)(\Box) \in E(\Delta_N, \Delta_T)$.

   Damit gilt
   $$\eta(f)(\Box) = \underbrace{\eta(h'_j)(\Box)}_{\in E(\Delta_N,\Delta_T)} \cdot u_{j+1} \cdots u_{m-1} \cdot \underbrace{\eta(h'_m)(\Box)}_{\in E(\Delta_N,\Delta_T)}$$
   und aus $\eta(f)(\Box) \in E$ folgt $j = m$.

   Also ist $\eta(f)(\Box) = \eta(h'_j)(\Box) = \eta(h'_m)(\Box)$ und für
   $$g := t \circ (h'_1 \times \cdots \times h'_{j-1} \times \mathbf{1}_1 \times h'_{j+1} \times \cdots \times h'_l)$$
   ist $\eta(g) = id_{\Delta^*}$ und $f = g \circ h'_j$. Wenden wir auf h'_j die Induktionsannahme an, dann folgt die Behauptung des Lemmas. $\Box$

Lemma 2.4.4 *Seien (G, η) und G_{ccf} wie oben. Dann gilt:*

$$\sigma \Rightarrow_{ccf} w,\ w \in \Delta^* \quad \Longrightarrow \quad \exists f \in \mathcal{B}_V(0,1),\ s \overset{*}{\Rightarrow}_G f,\ w = \eta(f)(\Box)$$

Beweis: Wir führen Induktion über die Länge der Ableitung $\sigma \overset{n}{\Rightarrow}_{ccf} w,\ n \in \mathbf{N}_0$.

Induktionsanfang: $\sigma \overset{0}{\Rightarrow}_{ccf} w$. Dann ist $w = \sigma$, wählen wir $f := s$, dann folgt die Behauptung.

Induktionsschritt: $\sigma \overset{n+1}{\Rightarrow}_{ccf} w,\ n \in \mathbf{N}_0$.

$$\sigma \overset{n}{\Rightarrow}_{ccf} \underbrace{u_1 \cdot x_1 \cdot w_1 \cdot x_2 \ldots x_k \cdot w_k \cdot x_{k+1} \cdot u_2}_{w'} \Rightarrow_{ccf} \underbrace{u_1 \cdot y_1 \cdot w_1 \cdot y_2 \ldots y_k \cdot w_k \cdot y_{k+1} \cdot u_2}_{w}$$

mit $u_1, u_2 \in \Delta^*,\ w_1, \ldots, w_k \in D(\Delta_N, \Delta_T)$, und in P_{ccf} gibt es eine Produktion $p : (x_1, \ldots, x_{k+1}) \to (y_1, \ldots, y_{k+1})$.

Aufgrund der Induktionsannahme gibt es einen Baum $f' \in \mathcal{B}_V(0,1)$ mit der Eigenschaft $s \overset{*}{\Rightarrow}_G f'$ und $\eta(f')(\square) = w'$. Dieser Baum besitzt nach Lemma 2.4.3 eine Darstellung $f' = g \circ x \circ (h_1 \times \cdots \times h_k)$ mit $\eta(g)(v) = u_1 \cdot v \cdot u_2\ \forall v \in \Delta^*$ und $\eta(h_i)(\square) = w_i,\ 1 \le i \le k$. Nach Konstruktion von G_{ccf} enthält P die Produktion $p : x \to y$. Daher gilt

$$s \overset{*}{\Rightarrow}_G \underbrace{g \circ x \circ (h_1 \times \cdots \times h_k)}_{f'} \Rightarrow_G \underbrace{g \circ y \circ (h_1 \times \cdots \times h_k)}_{=:f}$$

und es ist

$$\begin{aligned} \eta(f)(\square) &= \eta(g)\,(\,\eta(y)(\eta(h_1), \ldots, \eta(h_k)\,) \\ &= u_1 \cdot y_1 \cdot w_1 \cdot y_2 \ldots y_k \cdot w_k \cdot y_{k+1} \cdot u_2 \\ &= w \end{aligned}$$

womit das Lemma bewiesen ist. □

Aus der Definition der Satzmengen folgt dann sofort die Äquivalenz von (G, η) und G_{ccf}. Es gilt also

Satz 4 *Zu jeder BGmI (G, η) gibt es eine äquivalente gekoppelt-kontextfreie Grammatik G_{ccf}.*

Wir werden jetzt die Umkehrung zeigen, d.h. wir geben zu jeder gekoppelt-kontextfreien Grammatik eine äquivalente BGmI an.

Lemma 2.4.5 *Sei $G_{ccf} = ([V, g], T, P_{ccf}, s),\ N := V \setminus T,\ D(N,T)$ sei die zugehörige Klammersprache. Es gilt:*

Es gibt ein Terminalalphabet $\mathcal{T}$ und eine multilineare Interpretation η für $\mathcal{B} := \mathcal{B}(N \cup \mathcal{T}, Q, Z)$ derart, daß zu jedem $w \in D(N,T)$ ein $\Psi_w \in \mathcal{B}$ existiert mit

$$Q(\Psi_w) = 0,\ Z(\Psi_w) = 1,\ \eta(\Psi_w)(\square) = w$$

η ist dabei wie immer durch kanonische Erweiterung der Interpretation der Terminale aus $\mathcal{T}$ definiert und für die Variablen aus N gilt $Q(x) := g(x),\ Z(x) := 1$.

Beweis: Sei $E(N,T)$ das freie Erzeugendensystem von $D(N,T)$, also

$$E(N,T) = T \cup \bigcup_{x \in N} \{x_1 \cdot d_1 \cdot x_2 \ldots x_n \cdot d_n \cdot x_{n+1} \mid x \in N_n,\ d_1, \ldots, d_n \in D(N,T)\}$$

w besitzt eine eindeutige Darstellung bezüglich $E(N,T)$, etwa

$$w = w_1 \cdots w_r, \quad w_i \in E(N,T),\ r \in \mathbf{N}_0.$$

Seien o.E. die terminalen Faktoren zu Wörtern $u_j \in T^*$ zusammengefaßt, d.h.

$$w = u_1 \cdot v_1 \cdot u_2 \ldots u_s \cdot v_s \cdot u_{s+1}, \quad s \in \mathbf{N}_0,\ u_j \in T^*,\ v_i \in E(N,T) \setminus T$$

1. $s = 0 \iff w \in T^*$. Definiere ein neues Terminalsymbol $[w] \in \mathcal{T}$ und setze

$$Q([w]) := 0,\ Z([w]) := 1,\ \eta([w]) := w.$$

Dann ist die Behauptung erfüllt.

2. $s = 1,\ u_1 \cdot u_2 = \varepsilon \iff w \in E(N,T) \setminus T$. Ist $w = x_1 \cdot d_1 \cdot x_2 \ldots x_n \cdot d_n \cdot x_{n+1},\ x \in N_n$, dann setzen wir

$$\Psi_w := x \circ (\Psi_{d_1} \times \cdots \times \Psi_{d_n})$$

Wir können induktiv annehmen, daß $\Psi_{d_i} \in \mathcal{B}$ ist und daß $Q(\Psi_{d_i}) = 0,\ Z(\Psi_{d_i}) = 1,\ \eta(\Psi_{d_i})(\Box) = d_i$ gilt. Daher ist Ψ_w definiert und $\eta(\Psi_w)(\Box) = w$.

3. $s = 1,\ u_1 \cdot u_2 \neq \varepsilon$, oder $s > 1$.
Sei $[u_1, \ldots, u_{s+1}]$ eine neues Element von $\mathcal{T}$ mit

$$\begin{aligned} Q([u_1, \ldots, u_{s+1}]) &:= s, \\ Z([u_1, \ldots, u_{s+1}]) &:= 1, \\ \eta([u_1, \ldots, u_{s+1}])(w_1, \ldots, w_k) &:= u_1 \cdot w_1 \cdot u_2 \ldots u_s \cdot w_s \cdot u_{s+1} \end{aligned}$$

Dann setzen wir

$$\Psi_w := [u_1, \ldots, u_{s+1}] \circ (\Psi_{v_1} \times \cdots \times \Psi_{v_s})$$

und erhalten induktiv $\eta(\Psi_w)(\Box) = w$.

Lemma 2.4.6 *Ist mit den Voraussetzungen von Lemma 2.4.5* $w \in D(N,T)$ *und*

$$w = w_1 \cdots w_{k+1}, \quad k \in \mathbf{N}_0,\ w_i \in [V,g]^*,$$

eine Zerlegung von w*, dann gibt es ein Terminalalphabet* $\mathcal{T}$ *und ein* $\Psi_w^k \in \mathcal{B}(N \cup \mathcal{T})(0,1)$ *mit*

$$\eta(\Psi_w^k)(\alpha_1, \ldots, \alpha_k) = w_1 \cdot \alpha_1 \cdot w_2 \ldots w_k \cdot \alpha_k \cdot w_{k+1}$$

Beweis: Sei $[@]$ ein neues Symbol mit $Q([@]) := 0,\ Z([@]) := 1$ und $\eta([@])(\Box) := @$. Wir benutzen diesen Baustein, um die Positionen zu markieren, an denen die Eingänge des gesuchten Baums Ψ_w^k liegen. Sei $w^@ := w_1 @ w_2 \cdots w_k @ w_{k+1}$.

Wir führen die Konstruktion aus Lemma 2.4.5 durch und erhalten einen Baum $\Psi_w^@ \in \mathcal{B}(N \cup \mathcal{T})(0,1)$ mit der Interpretation $\eta(\Psi_w^@)(\Box) = w^@$.

In $\Psi_w^@$ kommt nach Konstruktion der Baustein $[@]$ genau k-mal vor und deshalb gibt es eine Darstellung

$$\Psi_w^@ = \Psi_w \circ (\underbrace{[@] \times \cdots \times [@]}_{k-mal})$$

Dann ist $\Psi_w \in \mathcal{B}(N \cup \mathcal{T})(k,1)$ und $\eta(\Psi_w)(\alpha_1, \ldots, \alpha_k) = w_1 \cdot \alpha_1 \cdot w_2 \ldots w_k \cdot \alpha_k \cdot w_{k+1}$. □

Lemma 2.4.7 *Sei* $G_{ccf} = ([N \cup T, g], T, P_{ccf}, s)$ *eine CCFG und* (G, η) *wie folgt definiert:*

1. $G := (N \cup \mathcal{T}, \mathcal{T}, P, s)$

2. $P := \{p : x \to \Psi_y \mid p : x \to y \in P_{ccf}\}$

3. *$\mathcal{T}$ und η seien wie in den vorigen Lemmata konstruiert*

Dann gilt: (G, η) *ist BGmI,* $grad(G) = grad(G_{ccf})$ *und* $L(G, \eta) = L(G_{ccf})$.

Beweis: Nach Konstruktion erfüllt (G, η) sicher die Definition einer BGmI und der Grad von G ist gleich dem Grad von G_{ccf}. Wir müssen noch zeigen, daß

$$s \overset{*}{\Rightarrow}_{ccf} \alpha, \ \alpha \in [N \cup T, g]^* \iff s \overset{*}{\Rightarrow}_G \Psi, \ \eta(\Psi)(\Box) = \alpha$$

Der Beweis hierfür verläuft genau wie die Beweise der Lemmata 2.4.2 und 2.4.4, weshalb wir ihn nicht wiederholen. $\Box$

Damit gilt der

Satz 5 *Zu jeder gekoppelt-kontextfreien Grammatik gibt es eine äquivalente BGmI gleichen Grades.*

Korollar 1 *Die Klasse BGmI(0) ist gleich der Klasse der kontextfreien Sprachen.*

Dies folgt aus der Tatsache, daß eine gekoppelt-kontextfreie Grammatik vom Grad 0 eine einfache kontextfreie Grammatik und die Ersetzungsrelation in diesem Fall die Semi-Thue-Relation ist. $\Box$

Kapitel 3

Tree Adjoining Grammatiken und BGmI

3.1 Definitionen

Eine *Tree Adjoining Grammar (TAG)* [Jo85] ist ein Ersetzungssystem auf den Ableitungsbäumen einer kontextfreien Grammatik. Sie ist in erster Linie ein Formalismus zur Beschreibung von Baumsprachen. TAGs finden in jüngster Zeit in der Linguistik großes Interesse, da sie zur Beschreibung vieler natürlichsprachlicher Phänomene geeignet erscheinen. In diesem Gebiet interessiert man sich u.a. für Sprachklassen, die mächtiger als kontextfreie Sprachen sind, aber nur in geringem Maße.

Wir interessieren uns hier nicht für diesen Aspekt von TAGs, sondern für die von TAGs erzeugte Sprachklasse, die *Tree Adjoining Languages (TAL)*. Die Klasse der TAL umfaßt die kontextfreien Sprachen echt und liegt in der Klasse der *Indexed Languages*. Wir werden zeigen, daß sie mit der Klasse BGmI(1) übereinstimmt. Wir geben zunächst eine Einführung in den TAG–Formalismus und zeigen dann die Äquivalenz. Die folgenden Definitionen des TAG-Rewriting-Formalismus folgen [Jo85].

Definition 23 *Tree Adjoining Grammar (TAG)*
Eine *Tree Adjoining Grammar* $G = (I, A)$ besteht aus

- einer endlichen Menge I von *Initialbäumen*,
- einer endlichen Menge A von *Hilfsbäumen*.

Diese *elementaren* Bäume sind Ableitungsbäume einer zugrundeliegenden kontextfreien Grammatik $g = (N \cup T, T, P, s)$.

Die initialen Bäume α sind Ableitungsbäume von s nach $w \in T^*$, die Hilfsbäume β sind Ableitungsbäume von x nach $u \cdot x \cdot v,\ x \in N,\ u, v \in T^*$. Das Blatt in β mit der Beschriftung x heißt *Fußknoten* von β.

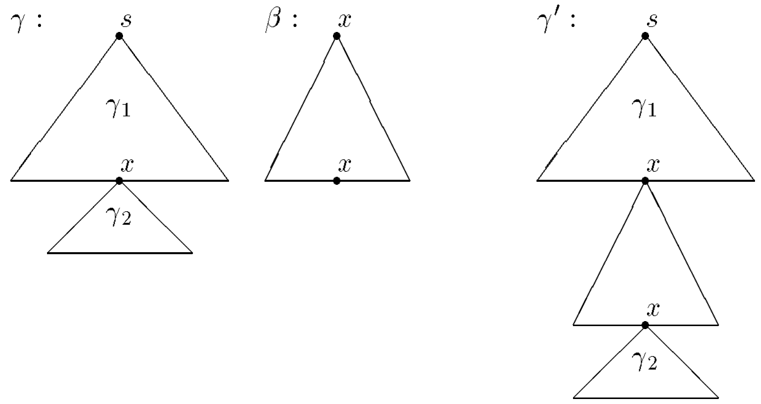
Die Menge der Hilfsbäume mit Wurzelbeschriftung x sei A_x und $A := \bigcup_{x \in N} A_x$.

Die folgende Abbildung zeigt die allgemeine Form eines initialen Baums und eines Hilfsbaums.

Ausgehend von den initialen Bäumen lassen sich durch *Adjunktion* der Hilfsbäume an Knoten neue Bäume gewinnen. Die Adjunktion eines Hilfsbaumes β an einen Knoten k in einem Baum γ ist genau dann definiert, wenn k die Beschriftung $x \in N$ trägt und β in A_x liegt. Die Adjunktion besteht aus den folgenden Schritten:

- Zerlege den Baum γ am Knoten k in einen oberen Baum γ_1 und einen Unterbaum γ_2.
- Setze γ_1 und γ_2 durch Einfügen von β wieder zusammen zum Baum γ'.

Wir geben diese Definition anhand der folgenden Abbildung anschaulich wieder und sparen uns die Einführung der benötigten Terminologie. Eine formale Definition findet man z.B. in [Vi88].



Geht γ' aus γ durch eine Adjunktion hervor, so schreibt man $\gamma \Rightarrow_{adj} \gamma'$. Die von einer TAG G *erzeugte Baumsprache* $T(G)$ ist

$$T(G) := \{\Psi \mid \exists\, \sigma \in I,\ \sigma \overset{*}{\Rightarrow}_{adj} \Psi\}.$$

Es ist klar, daß $T(G)$ eine Teilmenge der Menge aller Ableitungsbäume von s nach T^* der kontextfreien Grammatik g ist.

Die *Satzmenge* der TAG G ist

$$L(G) := \{Blattwort(\Psi) \mid \Psi \in T(G)\}.$$

Blattwort ist die Funktion, die jedem Ableitungsbaum das von den Beschriftungen seiner Blätter von links nach rechts gebildete Wort zuordnet. Damit ist klar, daß $L(G) \subset L(g)$ gilt.

Eine TAG ist also ein Rewriting-System auf den Ableitungsbäumen einer kontextfreien Grammatik. Aus der Theorie der Baumautomaten ist bekannt, daß sich die Menge der Ableitungsbäume einer kontextfreien Grammatik als *von einem endlichen Baumautomaten erkennbare Baumsprache (recognizable tree language)* charakterisieren läßt [Th67].

Die Baumsprache einer TAG dagegen braucht keine erkennbare Menge zu sein. Die Satzmenge einer TAG kann kontextfrei, aber auch echt kontextsensitiv sein. Wir wollen folgende Verallgemeinerung der TAG betrachten [Jo85].

Definition 24 *TAG mit Constraints*
Eine *TAG mit Constraints* (TAGC) $G = (I, A, C)$ ist eine TAG mit folgender Erweiterung:
Jeder innere (also mit einem Nichtterminal $x \in N$ beschriftete) Knoten k eines elementaren Baumes trägt eine der folgenden Informationen $C(k)$ (*Constraints*):

1. *Obligatory–Adjoining Constraint*: $C(k) = (OA, \Gamma),\ \Gamma \subset A_x$
   Bedeutung: An Knoten k *muß* ein Baum adjungiert werden, sonst liegt der k enthaltende Baum nicht in der Baumsprache der TAG. Es können nur Bäume aus Γ an k adjungiert werden.

2. *Selective–Adjoining Constraint*: $C(k) = (SA, \Gamma),\ \Gamma \subset A_x$
   Bedeutung: An Knoten *kann* ein Baum aus Γ adjungiert werden.

3. *Null–Adjoining Constraint*: $C(k) = NA = (SA, \emptyset)$
   Bedeutung: Kein Baum kann an Knoten k adjungiert werden.

Die Adjunktion ist für eine TAG mit Constraints entsprechend definiert. Ist k ein Knoten mit der Beschriftung $x \in N$ in einem Baum γ und $\beta \in A_x$ ein Hilfsbaum, dann läßt sich β an den Knoten k adjungieren, gdw. $C(k) = (SA, \Gamma)$ oder $C(k) = (OA, \Gamma)$ und $\beta \in \Gamma$. Durch die Adjunktion verliert k sein altes Constraint und erbt stattdessen das Constraint des Wurzelknotens von β. Die übrigen Constraints werden alle aus γ bzw. β übernommen.

Die Baumsprache einer TAGC $G = (I, A, C)$ ist

$$T(G) := \{\Psi \mid \exists\, \sigma \in I,\ \sigma \overset{*}{\Rightarrow}_{adj} \Psi,\ C(k) \neq (OA, \Gamma) \text{ für alle inneren Knoten } k \text{ von } \Psi\}.$$

Die Satzmenge $L(G)$ ist wie bei den einfachen TAGs definiert.

Es ist klar, daß jede TAG auch eine TAGC ist (jeder innere Knoten k mit Beschriftung $x \in N$ eines Hilfsbaumes trage das Constraint $C(k) = (SA, A_x)$).
Wir dürfen folgende Annahmen treffen:

1. In jedem elementaren Baum treten nur NA- und OA-Constraints auf.

2. Jede OA-Constraintmenge enthält einen einzigen Baum.

Einen Beweis hierfür findet man in der TAG-Literatur.

Wir wollen den TAG-Rewriting-Mechanismus mit unseren Baumgrammatiken mit multilinearer Interpretation vergleichen und stellen ihn deshalb zunächst in der Theorie der $\times$–Kategorien dar.

3.2 Darstellung des TAG-Formalismus durch Netze

Sei $G = (I, A)$ eine TAG und sei $g = (N \cup T, T, P, s)$ die zugrundeliegende kontextfreie Grammatik. $\mathcal{F}(g) := \mathcal{F}(P, N \cup T, Q, Z)$ sei die zu g gehörende freie $\times$–Kategorie.

Die initialen Bäume aus I sind Elemente $\alpha \in \mathcal{F}(g)$ der Form

$$s \xrightarrow{\alpha} w, \qquad w \in T^*,$$

d.h. I ist eine endliche Teilmenge von $\mathcal{F}(g)(s, T^*)$.

Die Hilfsbäume aus A sind Elemente $\beta \in \mathcal{F}(g)$ der Form

$$x \xrightarrow{\beta} u \cdot x \cdot v, \qquad u, v \in T^*,\ x \in N.$$

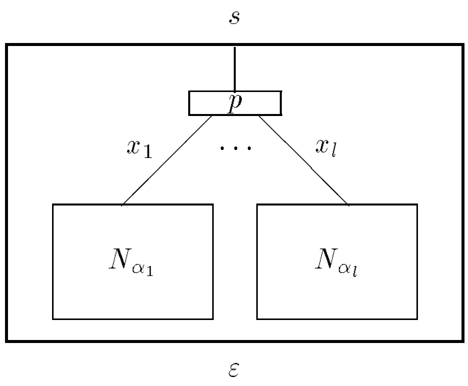
Wir betrachten wie im Abschnitt über kontextfreie Sprachen wieder die terminalfreie Darstellung der Ableitungsbäume, d.h. wir ordnen der kontextfreien Grammatik eine syntaktische und eine semantische Kategorie zu. Wir bezeichnen die syntaktische Kategorie mit $\mathcal{F}'(g) := \mathcal{F}(g, Q', Z')$ und die semantische Kategorie mit $\mathcal{C}(g)$.

Damit erhalten wir folgende Darstellung einer TAG $G = (I, A)$:

1. I ist endliche Teilmenge von $\mathcal{F}'(g)(s, \varepsilon)$.
2. A_x ist endliche Teilmenge von $TREES(\mathcal{F}'(g)(x, x))$, $x \in N$.
3. $A = \bigcup_{x \in N} A_x$.

Wir betrachten die geometrische Darstellung der Elemente von $\mathcal{F}'(g)$ als *Netze* (bzw. Bäume) und veranschaulichen daran die Adjunktion.

Jedem initialen Baum $\alpha \in I$ entspricht ein Netz $N_\alpha \in \mathcal{F}'(g)(s, \varepsilon)$ folgender Form:



Das initiale Netz N_α besitzt eine eindeutige Darstellung

$$N_\alpha = (N_{\alpha_1} \times \cdots \times N_{\alpha_l}) \circ p, \qquad p \in P,\ N_{\alpha_i} \in \mathcal{F}(g)(x_i, \varepsilon),\ l \in \mathbf{N}_0.$$

$p : s \to x_1 \cdots x_l$ ist die terminalfreie Darstellung der unter der Wurzel von α stehenden Produktion $p : s \to u_1 \cdot x_1 \cdot u_2 \dots u_l \cdot x_l \cdot u_{l+1},\ u_1, \dots, u_{l+1} \in T^*$, aus der Grammatik g. Die Netze N_{α_i} entsprechen den Teilbäumen α_i von α, die unter der Produktion p hängen und jeweils die Wurzel x_i haben.

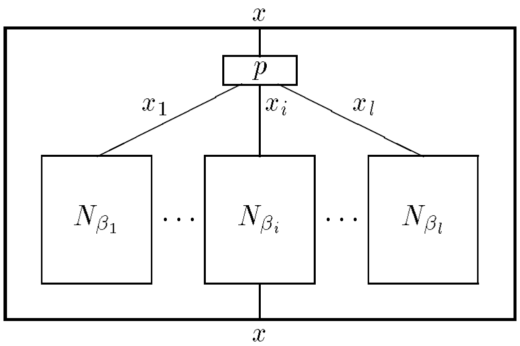
Ist $f_p : (T^*)^k \to T^*$ die Interpretation von p und η die multilineare Interpretation für $\mathcal{F}'(g)$, dann gilt:

$$\begin{aligned} \eta(N_{\alpha_i})(\Box) &= \mathit{Blattwort}(\alpha_i) \\ \eta(N_\alpha)(\Box) &= f_p(\ \eta(N_{\alpha_1}), \ldots, \eta(N_{\alpha_l})\ ) \\ &= \mathit{Blattwort}(\alpha) \end{aligned}$$

Im einfachsten Fall ist $p : s \to w,\ w \in T^*$, eine terminale Produktion der Grammatik g und dann ist $l = 0,\ N_\alpha = p,\ \eta(N_\alpha)(\Box) = f_p(\Box) = w$.

Wir betrachten jetzt die Hilfsbäume $\beta \in A$ der TAG G in dieser Darstellung:

Sei $\beta \in A_x,\ x \in N$, ein Hilfsbaum. Dann hat das korrespondierende Netz $N_\beta \in \mathcal{F}'(g)$ folgendes Aussehen:



N_β besitzt eine eindeutige Darstellung

$$N_\beta = (N_{\beta_1} \times \cdots \times N_{\beta_i} \times \cdots \times N_{\beta_l}) \circ p, \qquad p \in P,\ Q'(p) = x,\ Z'(p) = x_1 \cdots x_l,\ l \in \mathbf{N}.$$

Es ist $Q'(N_{\beta_i}) = x_i,\ Z'(N_{\beta_i}) = x$ und für $j \neq i$ ist $Q'(N_{\beta_j}) = x_j \in N,\ Z'(N_{\beta_j}) = \varepsilon$.

$p : x \to x_1 \cdots x_l$ ist die terminalfreie Darstellung der Produktion $p : x \to u_1 \cdot x_1 \cdot u_2 \ldots u_l \cdot x_l \cdot u_{l+1}$ aus der Grammatik g unter der Wurzel von β.

Es gibt in N_β eine eindeutige mit x beschriftete „Anfangsstrecke“ (sie entspricht der Wurzel von β) und eine eindeutige mit x beschriftete „Ausgangsstrecke“ (entspricht dem Fußknoten von β).

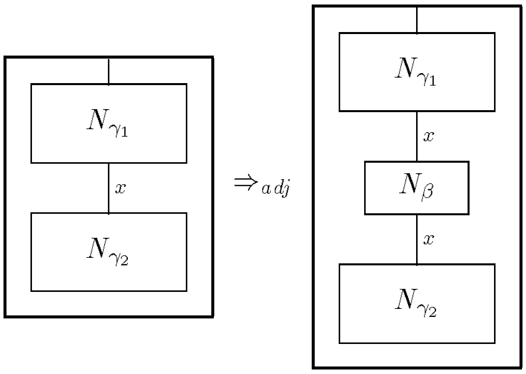
Im einfachsten Fall ist β eine einzelne Produktion $p \in P$ der Form $p : x \to u_1 x u_2,\ u_1, u_2 \in T^*$, d.h. $N_{\beta_i} = \mathbf{1}_x$.

Die Darstellung der Ableitungsbäume von g als Netze in $\mathcal{F}'(g)$ ist in gewisser Weise dual zu der herkömmlichen Darstellung der Ableitungsbäume. Jedem mit einem Nichtterminal beschrifteten inneren Knoten eines Ableitungsbaums entspricht im Netz eine gleichbeschriftete Strecke und den im Ableitungsbaum implizit enthaltenen Produktionen entsprechen im Netz Punkte, die mit dem Namen der jeweiligen Produktion beschriftet sind.

Wir betrachten jetzt die Adjunktionsoperation in der Netzdarstellung.

Sei γ ein Ableitungsbaum von g, der einen mit $x \in N$ markierten Knoten k enthalte. γ besitzt eine Zerlegung in den Unterbaum γ_2 mit Wurzel k und das obere Baumfragement γ_1. Ist k die Wurzel von γ, dann ist γ_1 leer. Sei $w_1 \cdot w_2 \cdot w_3$ das Blattwort von γ, wobei w_2 das Blattwort von γ_2 ist.

Im Netz N_γ gibt es eine zu k korrespondierende mit x markierte Strecke s_k, die N_γ in N_{γ_1} und N_{γ_2} zerlegt, $N_\gamma = N_{\gamma_2} \circ N_{\gamma_1}$.



Der Adjunktion von $\beta \in A_x$ an den Knoten k von γ entspricht in der Netzdarstellung die Ersetzung der Strecke s_k durch das Netz N_β. Schreiben wir N_γ als Ausdruck, dann entspricht die Adjunktion von β einer Ersetzung $\mathbf{1}_x \to \beta$, d.h.

$$N_\gamma = N_{\gamma_2} \circ N_{\gamma_1} = N_{\gamma_2} \circ \mathbf{1}_x \circ N_{\gamma_1} \Rightarrow_{adj} N_{\gamma_2} \circ \beta \circ N_{\gamma_1} = N_{\gamma'}$$

Betrachten wir die Interpretation, so gelten folgende Beziehungen:

$$\begin{aligned} \eta(N_{\gamma_1})(w) &= w_1 \cdot w \cdot w_3 \quad \forall w \in T^*, \\ \eta(N_\beta)(w) &= u \cdot w \cdot v \quad \forall w \in T^*, \\ \eta(N_{\gamma_2})(\Box) &= w_2. \end{aligned}$$

Damit ist

$$\begin{aligned} \eta(N_{\gamma'})(\Box) &= \eta(N_{\gamma_2} \circ N_\beta \circ N_{\gamma_1})(\Box) \\ &= w_1 \cdot u \cdot w_2 \cdot v \cdot w_3 \\ &= Blattwort(\gamma'). \end{aligned}$$

Die Adjoining Operation erzeugt also im Blattwort neue Wörter an zwei Positionen, die durch das Blattwort eines Teilbaums voneinander getrennt sind. Dies ist bei gekoppelten Ersetzungen von Klammerpaaren ähnlich und wir wollen deshalb die Beziehung zwischen diesen Ersetzungssystemen untersuchen.

3.3 Die Äquivalenz von TAGC und BGmI vom Grad 1

Wir zeigen nach diesen Vorbetrachtungen, daß sich TAGCs und BGmIs vom Grad 1 gegenseitig simulieren können und daher die Klassen TAL und BGmI(1) übereinstimmen.

Wir betrachten zunächst die Adjoining-Operation bei TAGs mit Constraints in der Darstellung durch Netze.

In einer TAGC ist jedem inneren Knoten k eines elementaren Baums γ ein Constraint $C(k)$ zugeordnet. Entsprechend ordnen wir der zu k korrespondierenden Strecke s_k dieses Constraint zu.

Trägt die Strecke s_k das Constraint $C(k) = (OA, B)$, so sind alle Ersetzungen $s_k \to N_\beta$, $\beta \in B$, erlaubt. Wegen des Vorkommens eines OA-Constraints ist γ selbst kein Element der Baumsprache der TAGC. Entsprechend gehört ein Netz aus $\mathcal{F}'(g)$ nicht zur Baumsprache, wenn es eine Strecke mit OA-Constraint enthält.

Trägt eine Strecke s_k das NA-Constraint, so läßt sich s_k nicht mehr ersetzen, verhindert aber nicht die Zugehörigkeit zur Baumsprache.

Die Strecken mit OA-Constraint spielen gewissermaßen die Rolle von Nichtterminalen in Grammatiken. Die Constraint-Menge gibt an, welche Bäume für diese Strecke eingesetzt werden dürfen und entspricht der Menge der rechten Seiten der Produktionen für ein Nichtterminal.

Wir wollen TAGs mit Constraints durch Baumgrammatiken mit multilinearer Interpretation simulieren. Nach dem oben gesagten ist die Idee jetzt offensichtlich:

Wir setzen in den elementaren Netzen für die Strecken s_k mit Beschriftung $x \in N$ und Constraint $C(k) = (OA, B)$ einen nichtterminalen „Baustein" $[x, B]$ ein. Damit dieses Einsetzen wohldefiniert ist, müssen wir $Q'([x, B]) = Z'([x, B]) := x$ setzen.

Jedem neuen Baustein $[x, B]$ ordnen wir folgende Menge von Ersetzungsregeln zu:

$$\{[x, B] \to N_\beta \mid \beta \in B\}$$

Dabei bezeichnet N_β das Netz zum modifizierten Hilfsbaum β. Die Strecken s_k mit $C(k) = NA$ bleiben unverändert.

Führen wir jetzt noch für jeden initialen Baum $\alpha \in I$ eine Ersetzungsregel $\sigma \to N_\alpha$ ein, wobei σ ein neuer Baustein mit $Q'(\sigma) := s$, $Z'(\sigma) := \varepsilon$ sei, dann erhalten wir eine Grammatik auf der Baumkategorie $\mathcal{F}'(g)$.

Es ist klar, daß sich die Adjunktionen in der TAGC und die Ersetzungen in dieser Baumgrammatik eineindeutig entsprechen.

Wir haben allerdings Baumgrammatiken nur für Bäume ohne Beschriftungen der Strecken definiert. Es ist allerdings auch mit dieser Vereinfachung möglich, die Ersetzungen einer TAGC zu simulieren.

Betrachten wir die Produktionen $[x, B] \to N_\beta$. Es gilt:

$$Q'([x, B]) = Z'([x, B]) = x = Q'(N_\beta) = Z'(N_\beta).$$

Die Quelle-/Zielbedingung ist also für jede Produktion erfüllt, d.h. jede Regel kann auch tatsächlich angewendet werden. Deshalb können beim Weglassen der Beschriftungen keine vorher verbotenen Ersetzungen zu erlaubten Ersetzungen werden.

Außer den Nichtterminalsymbolen treten im Netz nur noch die (terminalfreien) Produktionen der Grammatik g auf. Sie werden zur Interpretation des erzeugten Baums gebraucht. Unsere Interpretation ist aber unabhängig von den Beschriftungen, denn der Grundbereich ist für jedes Argument der Funktion f_p gleich, nämlich T^*.

Wir erhalten daher durch obige Konstruktion und anschließendes Weglassen der Beschriftungen eine BGmI vom Grad 1, die offensichtlich die gleiche Satzmenge wie die vorgegebene TAGC erzeugt. Es gilt der

Satz 6 *Zu jeder TAGC gibt es eine äquivalente BGmI vom Grad 1.*

Wir zeigen jetzt, wie man umgekehrt aus jeder BGmI vom Grad 1 eine äquivalente TAGC konstruieren kann. Wir konstruieren die TAGC in der Darstellung durch terminalfreie Bäume mit Interpretation, die sich leicht zurück in die herkömmliche TAGC-Darstellung bringen läßt.

Sei (G, η) eine BGmI vom Grad 1, $G = (N \cup T, T, P, s)$ mit Quelle- und Zielabbildungen $Q, Z : N \cup T \to \mathbf{N}_0$. Wir identifizieren für die folgende Konstruktion die Mengen $\mathbf{N}_0$ und $\{I\}^*$.

O.B.d.A. besitze jedes Nichterminalsymbol von G (außer s) genau einen Eingang. Diese Voraussetzung bedeutet keine Einschränkung, da sich jede Produktion $p : x \to y$ mit $Q(x) = 0$ durch eine Produktion $p : x' \to y'$ mit $Q(x') = Q(y') = 1$ so ersetzen läßt, daß sich die erzeugte Satzmenge nicht ändert (Die erzeugte Baumsprache ändert sich natürlich). Dazu geht man wie folgt vor:

1. Führe ein neues Nichtterminalsymbol x', $Q(x') := 1$, und zwei neue Terminalsymbole $[eps]$, $[conc_2]$ mit $Q([eps]) := 0$, $Q([conc_2]) := 2$, mit der offensichtlichen Interpretation ein.

2. Ersetze alle Vorkommen von x in den rechten Seiten von Produktionen durch $x' \circ [eps]$

3. Ersetze jede Produktion $p : x \to y$ durch die Produktion $p : x' \to [conc_2] \circ (\mathbf{1}_1 \times y)$.

Man kann allgemein jedes Nichtterminal vom Grad k durch eines von größerem Grad ersetzen (außer das Axiom) und, wenn man sich nur für die Satzmenge interessiert, davon ausgehen, daß in einer Grammatik vom Grad k alle übrigen Nichtterminale k Eingänge haben.

Wir konstruieren jetzt die zu G äquivalente TAGC. Die Netze in dieser TAGC sind Elemente einer freien $\times$–Kategorie $\mathcal{F}(T \cup T', N \dot\cup \{I\}, Q', Z',)$, d.h. die Punkte der Netze werden den Terminalen von G bzw. neuen Terminalen entsprechen und die Strecken werden mit den Nichtterminalen der BGmI G oder mit dem Symbol I beschriftet sein.

Seien zu jedem $x \in N$ neue Terminale $[I \to x]$ und $[x \to I]$ gegeben mit $Q'([I \to x]) := I$, $Z'([I \to x]) := x$ und entsprechend umgekehrt für $[x \to I]$. Die Interpretation dieser neuen Terminale sei die Identität.

Sei $p : x \to y$ eine Produktion der BGmI. Wir definieren den zu y korrespondierenden Hilfsbaum β_y wie folgt:

- Ersetze in y jedes Vorkommen einer Variable z durch das Netz $[z \to I] \circ [I \to z]$. Dadurch erhält man jeweils eine mit z beschriftete Strecke, die durch die beiden Terminale $[I \to z]$ und $[z \to I]$ begrenzt ist. Ordne der so entstehenden neuen Strecke das Constraint (OA, A_z) zu. Dabei ist A_z die Menge der durch die Konstruktion entstehenden Hilfsbäume für z. Wir erhalten ein Netz β'_y.

- Wir müssen jetzt noch dafür sorgen, daß β'_y ein Hilfsbaum in A_x wird, denn wir gingen ja von der Produktion $p : x \to y$ aus. Dazu setzen wir $\beta_y := [I \to x] \circ \beta'_y \circ [x \to I]$. Dadurch erhält β_y je eine mit x beschriftete Eingangs- und Ausgangsstrecke. Beiden Strecken geben wir das Constraint NA, denn wir wollen sie nur dazu benutzen, β_y als Element von A_x zu kennzeichnen und nur die Ersetzungen im Teilnetz β'_y zulassen. Alle mit I beschrifteten Strecken erhalten das Constraint NA, denn an ihnen war in der Baumgrammatik keine Ersetzung möglich.

Wir erhalten so die Menge A der Hilfsbäume und es ist aus jeder Produktion $p : x \rightarrow y$ genau ein Hilfsbaum $\beta_y \in A_x$ entstanden.

Wir können o.E. annehmen, daß das Axiom s der Baumgrammatik in keiner rechten Seite auftritt und daß die einzige Produktion für s von der Form $p : s \rightarrow x \circ [eps]$ ist, wobei x ein Nichterminal und $[eps]$ das oben beschriebene Terminalsymbol ist. Dann setzen wir $\alpha := [I \rightarrow x] \circ [eps]$ und geben der mit x beschrifteten Strecke das Constraint (OA, A_x). Diese Strecke steht nämlich für das Nichtterminal x.

Durch diese Konstruktion erhalten wir aus der vorgegebenen BGmI vom Grad 1 eine TAGC, wobei die Produktionen der BGmI und die Hilfsbäume der TAGC in einer 1:1-Beziehung stehen.

Durch Induktion über die Länge der Ableitung überzeugt man sich von der Äquivalenz der beiden Ersetzungsysteme.

Damit gilt der

Satz 7 *Zu jeder BGmI vom Grad 1 läßt sich eine TAGC angeben, die die gleiche Satzmenge erzeugt.*

Literaturverzeichnis

[Ben70] Benson D.B. (1970), „Syntax and semantics: a categorical view“, *Information and Control* **17**, 145–160.

[Ben75] Benson D.B. (1975), „The basic algebraic structures in categories of derivations“, *Information and Control* **28**, 1–29.

[Ber79] Berstel, J. (1979), „Transductions and Context-Free Languages“, Teubner, Stuttgart.

[Br69] Brainerd, W.S. (1969), „Tree generating Regular Systems“, *Information and Control* **14**, 217–231.

[GeSt84] Gecseg F., Steinby M. (1984), „Tree Automata“, Akadémiai Kiadó, Budapest.

[Go77] Goguen J.A. , Thatcher J.W., Wagner E.G., Wright J.B. (1977), „Initial Algebra Semantics and Continuous Algebras“, *Journal of the ACM* **24**, 68–95.

[Ha78] Harrison, M. (1978), „Introduction to Formal Language Theory“, Addison-Wesley.

[Hop79] Hopcroft J.E., Ullman J.D. (1979), „Introduction to Automata Theory, Languages, and Computation“, Addison-Wesley.

[Ho65] Hotz G. (1965), „Eine Algebraisierung des Syntheseproblems von Schaltkreisen“, *EIK* **1**, 185–205, 209–231.

[Ho66] Hotz G. (1966), „Eindeutigkeit und Mehrdeutigkeit formaler Sprachen“ , *EIK* **2**, 235–246.

[Ho67] Hotz G. (1967), „Erzeugung formaler Sprachen durch gekoppelte Ersetzungen“, 4.Colloquium über Automatentheorie, Ed. F.L. Bauer, K. Samelson, Math. Institut der TH München, 62–73.

[HoCl72] Hotz G., Claus V. (1972), „Automatentheorie und Formale Sprachen III“ , B.I.-Wissenschaftsverlag, Mannheim.

[Ho74] Hotz G. (1974), „Schaltkreistheorie“, de Gruyter, Berlin.

[Ho90] Hotz G. (1990), „Einführung in die Informatik“, Teubner, Stuttgart.

[Jo85] Joshi A. (1985), „An introduction to Tree Adjoining Grammars“, Technical report, University of Pennsylvania.

[Mai74] Maibaum T.S.E. (1974), „A generalized approach to formal languages“, *Journal of CSS* **8**, 409–439.

[Th67] Thatcher J.W. (1967), „Characterizing derivation trees of context-free grammars through a generalization of Finite Automata Theory“, *Journal of CSS* **1**, 317–322.

[Vi88] Vijay–Shanker K. (1988), „A study of Tree Adjoining Grammars“, Dissertation, University of Pennsylvania.