Abschnitt 3

Formale Sprachen

3. Formale Sprachen

Grammatiken beschreiben Sprachen. Seit den 1950 er Jahren entwickeln Mathematiker und Linguisten formale Grammatiken zur Beschreibung natürlicher und künstlicher Sprachen. Herausragend ist in diesem Zusammenhang der amerikanische Linguist *NOAM CHOMSKY*.[1] In der Informatik finden Grammatiken Anwendung bei der Sprachanalyse und -synthese und dem Compilerbau. Der erste Compiler wurde 1952 von *HOPPER*[2] entwickelt.

Die Analyse dient zur Erstellung des Syntaxbaums und kann in lexikalische, syntaktische und semantische Analyse unterschieden werden:

1. bei der **lexikalischen Analyse** werden im Text, z.B. dem zu kompilierenden Programmcode, einzelne *Tokens* verschiedener Klassen (Schlüsselwörter, Operatoren, Bezeichner, Konstanten) identifiziert.

2. bei der **syntaktischen Analyse** geht es um den Aufbau des Syntaxbaums (*Ableitungsbaums*)[3]. Diese Phase wird auch als *Parsen* (engl. *parsing*) bezeichnet.[4]

3. bei der **semantischen Analyse** wird die statische Semantik eines Programms überprüft: z.B. ob alle Variablen vor ihrer ersten Benutzung deklariert sind. Die Überprüfung wird durch das Hinzufügen von Attributen (z.B. einer Liste aller deklarierten Variablen) an einen Syntaxbaum geleistet.

Im Folgenden soll ein kurzer Einblick in die grundlegende Verfahrensweise der Syntaxanalyse gegeben werden. Es geht bei der syntaktischen Analyse im Kern um das Wortproblem.

---

1 Chomsky, Noam: Aspekte der Syntaxtheorie, Berlin/ Frankfurt am Main 1969.

2 *(in Vorbereitung)*

3 Vgl. 3.3.2.

4 Werkzeuge wie z.B. *Yacc* helfen bei der Erzeugung eines Parsers zur syntaktischen Analyse von Programmcode.

3.1. Das Wortproblem

Bei dem sogenannten Wortproblem handelt es sich um die Frage, ob ein bestimmter Ausdruck ("Wort") einer Sprache in einer bestimmten Grammatik ableitbar ist.[5] Hierbei sind zwei Aspekte bedeutsam. Das Wortproblem ist

- zum einen eine Frage der Komplexität.

  Das Wortproblem kann durch Bestimmung aller Ableitungen einer bestimmten Länge gelöst werden. Diese Methode ist jedoch ineffizient.

- zum anderen ist es eine Frage der Eindeutigkeit.

In Abschnitt 3.3.2.4.2.1. wird gezeigt, daß Grammatiken mehrdeutig sein können. Das würde aber z.B. im Falle einer Programmiersprache bedeuten, daß einem Codeabschnitt potentiell mehrere Ableitungsbäume und somit mehrere Semantiken zugeordnet werden könnten. Das Interesse besteht daher weiter daran, zu einer gegebenen gegebenenfalls mehrdeutigen Grammatik eine äquivalente Grammatik zu finden, deren Ableitungen eindeutig sind. Es gibt jedoch auch inhärent mehrdeutige Grammatiken. Das sind Grammatiken, die tatsächlich keine äquivalente eindeutige Grammatik besitzen. Will man einen Compiler konstruieren, ist eine eindeutige Grammatik jedoch eine essentielle Voraussetzung.

3.2. Klassen von Grammatiken

Grammatiken lassen sich bezüglich der Art der von ihnen zugelassenen Produktionen klassifizieren.[6] Die Chomsky-Hierachie unterscheidet dabei vier Klassen von Grammatiken. Nach *CHOMKSY* können Grammatiken vom Typ-0, Typ-1, Typ-2 oder Typ-3 sein.[7] Typ-1-Grammatiken sind sog. kontextsensitive Grammatiken. Sie gestatten die kontextabhängige

---

[5] Im Folgenden werden die Begriffe *Wort* und *Satz* zunächst synonym für die Produktionen einfacher kontextfreier Grammatiken verwendet.

[6] Chomsky, Noam: Aspekte der Syntaxtheorie, Berlin/ Frankfurt am Main 1969, S. 84 ff. und S. 259 ff.

[7] Vgl. Schmidt, Ute/ Kindsmüller, Martin C.: Kognitive Modellierung. *Eine Einführung in die logischen und algorithmischen Grundlagen*, Heidelberg/ Berlin/ Oxford 1996, S. 123 f.

Anwendung von Produktionen. Typ-2-Grammatiken sind sog. kontextfreie Grammatiken. Deren Produktionen werden unabhängig vom syntaktischen Kontext angewandt.

3.2.1. Das Wortproblem in kontextsensitiven Sprachen

Gegeben sei folgende Grammatik G_0:[8]

(r_1)	S	→	NP + VP
(r_2)	VP	→	V + NP
(r_3)	NP	→	'Alfred' | 'ein Brötchen' | 'ein Haus'
(r_4)	VP	→	'verschlingt' | 'bewohnt'

Die Grammatik G_0 erzeugt u.a. folgende Teilmenge des Deutschen

(1) *Alfred verschlingt ein Brötchen*
(2) *Alfred bewohnt ein Haus*

Die Grammatik erzeugt auch noch die folgende Teilmenge des Deutschen, deren Sätze von kompetenten Sprechern jedoch als ungrammatikalisch abgelehnt werden:

(3) **Alfred verschlingt einen Stein*
(4) **Alfred bewohnt ein Brötchen*

Die Ausdrücke (3) und (4) gleichen strukturell den Ausdrücken (1) und (2). Produktionen vom Typ (3) und (4) lassen sich im Rahmen einer kontextfreien Grammatik grundsätzlich nicht verhindern. Die Produktionen (3) und (4) sind nur dann auszuschließen, wenn die Regel r_3 dahingehend erweitert und modifiziert wird, daß die Ersetzungen an den Kontext V gebunden werden[9], hier etwa

(r_{3a})	NP	→	'Alfred'	WENN (links_von_V)

und

[8] Die Bezeichnungen bedeuten (in der Reihenfolge): NP Nominalphrase, VP Verbalphrase, V Verb, S Satz (→ *Startsymbol*, vgl. 3.3.1.)

[9] Vgl. dazu das Kapitel "Kontextsensitive Grammatiken" in: Gross, Maurice/ Lentin, André: Mathematische Linguistik, Berlin/ New York 1972, S. 177 ff.

$$(r_{3b}) \begin{cases} \text{NP} \rightarrow \text{'ein Brötchen'} & \text{WENN (V = 'verschlingt')} \\ & \wedge \text{(rechts\_von\_V)} \\ \text{NP} \rightarrow \text{'ein Haus'} & \text{WENN (V = 'bewohnt')} \\ & \wedge \text{(rechts\_von\_V)} \end{cases}$$

Grundsätzlich ist das Wortproblem für die Klasse der kontextsensitiven Sprachen (KS) zwar entscheidbar, nicht bewiesen ist jedoch, ob dieses auch tatsachlich effizient – d.h. in Polynomialzeit – zu erreichen ist.

3.2.2. Das Wortproblem in kontextfreien Sprachen:

Für kontextfreie Grammatiken läßt sich dagegen zeigen, daß das Wortproblem effizient, d.h. in Polynomialzeit, lösbar ist. Da wichtige künstliche Sprachen (wie z.B. die verbreiteten Programmiersprachen) abgesehen von der Typenkonsistenz kontextfrei sind, sind die Klasse (KF) der kontextfreien Sprachen und ihre Teilklassen Gegenstände eingehender Untersuchungen.

Ein wichtiges Hilfsmittel bei der Anwendung kontextfreier Grammatiken ist die Darstellung von Wortableitungen mit Hilfe von Ableitungsbäumen.[10] Diese unterscheiden sich in kontextfreien Sprachen jedoch in der Anzahl und/oder der Reihenfolge der Regelanwendungen.[11] Es läßt sich zeigen, daß jedem Ableitungsbaum genau eine Linksableitung[12] zugeordnet werden kann. Darüber hinaus läßt sich beweisen, daß jedes ableitbare Wort eine Linksableitung besitzt.[13]

3.2.3. Entscheidbarkeit des Wortproblems

Die Lösung des Wortproblems in einer kontextfreien Grammatik G durch die Bestimmung aller Ableitungen einer bestimmten Länge ist ineffizient. Eine effiziente Lösung ist mit Hilfe

---

10 Im Abschnitt 3.3.2.4.1 wird die Ableitung von Wörtern und die Zuordnung von Ableitung und Ableitungsbaum exemplarisch dargestellt.

11 Führen zwei Ableitungen zu demselben Ableitungsbaum, nennt man diese *äquivalent*, s. 3.3.2.3.2.

12 *Vgl. 3.4. (in Vorbereitung)*

13 Vgl. 3.3.2.4.

des Algorithmus von *COCKE, YOUNGER* und *KASAMI* (*CYK-Algorithmus*) möglich.[14] Die Anwendung des *CYK-Algorithmus* ist allerdings an folgende Voraussetzungen gebunden:

- an die Konvertierung der (kontextfreien) Grammatik *G* in die *Chomsky-Normalform* (*CNF*) [15] und
- an den Aufbau des Ableitungsbaums *von unten* (→ *Bottom-Up-Analyse*[16]).

3.3. Kontextfreie Grammatik

Um das Wortproblem und das Problem der Ableitung und (weitergehend dann) der Spracherkennung bearbeiten zu können, definieren wir daher zunächst den Begriff der *Kontextfreien Grammatik*.

3.3.1. Definition

Eine kontextfreie Grammatik $G=(V, T, R, S)$ besteht aus

(1) einer Menge V von nichtterminalen Symbolen

(2) einer Menge T von terminalen Symbolen, mit

$V \cap T = \varnothing$

(3) einem Startsymbol $S \in V$ und

(4) einer endlichen Menge R von Produktionsregeln der Form

$r_i: X \rightarrow v$

mit $X \in V$ und $v \in (V \cup T)^*$.

Das Startsymbol S ist ein Axiom. Die kontextfreie Grammatik kann nicht immanent entscheiden, ob ein Wort korrekt ist oder nicht. Dieses axiomatisch begründete (*Sprach*-)Wissen ist der Grammatik übergeordnet und wird durch das Symbol S repräsentiert. Dabei wird $X \rightarrow v$ in der Regel gelesen als

---

14 Vgl. 3.4.

15 Vgl. 3.3.4.1.

16 *(in Vorbereitung)*

„*X wird durch v ersetzt*“.

Diese formale Sprechweise bedeutet aber tatsächlich, daß das Symbol V selbst strukturiert ist und seinerseits aus Symbolen aus $(V \cup T)^*$ hierarchisch zusammengesetzt ist.[17]

Beispiele

(1) Eine Grammatik für einfache arithmetische Ausdrücke (G_1)

Man kann eine einfache kontextfreie Grammatik, die einfache arithmetische Ausdrücke der Form (a+b)*c beschreibt, wie folgt konstruieren. Die Grammatik heiße G_1. Dazu sollen die Bezeichnungen

<Ausdruck>	für terminale oder nichtterminale Variablen
<id>	Identifizierer

verwendet werden. In der kontextfreien Grammatik G_1 sollen weiter verwendet werden:

(1) ein Startsymbol **<Ausdruck> ,**

(2) als terminales Vokabular T die Menge **{ + , * , (,)** , 3, 5, 4 } und

(3) als nichtterminales Vokabular V die Menge { Ausdruck , id } .

(4) Die Grammatik verfügt über folgende Menge an Produktionsregeln[18]:

---

[17] Die Schreibweise $r_i: X \rightarrow v$ mit $X \in V$ und $v \in (V \cup T)^*$ bedeutet Folgendes: sei speziell $X, Y \in V$ und $t \in T$, dann wären z.B.

$r_1: X \rightarrow t\ Y$ (oder mit '+' als Verkettungsoperator $X \rightarrow t + Y$)

$r_2: X \rightarrow Y\ t$ (oder mit '+' als Verkettungsoperator $X \rightarrow Y + t$)

wohlgeformte Produktionsregeln im Sinne von 3.3.1.(4).

[18] Die Notation der Grammatik G_1 orientiert sich an der Backus-Naur-Notation (→ Schmidt/Kindsmüller, op.cit., S. 121). In dieser leicht nachvollziehbaren Notation bedeutet die Klammerung mit < und > eines Terms, z.B. bei <Ausdruck>, daß dieser noch durch anderen terminale oder nichtterminale Variablen der Grammatik zu ersetzen ist. Diese

(r_1)	<Ausdruck>	→	<Ausdruck>	+	<Ausdruck>	
(r_2)	<Ausdruck>	→	<Ausdruck>	*	<Ausdruck>	
(r_3)	<Ausdruck>	→	(<Ausdruck>)			
(r_4)	<Ausdruck>	→	<id>			
(r_5)	< id >	→	3 | 5 | 4			

Die durch G_1 erzeugten Ausdrücke können Klammern enthalten. Mit G_1 läßt sich z.B. der Ausdruck '(3+5) * 4' ableiten: dabei müssen die Produktionsregeln aus R in der Reihenfolge $(r_2)(r_3)(r_4)(r_1)(r_4)(r_4))(r_5)^3$ angewandt werden:[19]

(r_2)	<Ausdruck>	→	<Ausdruck>	+	<Ausdruck>		
(r_3)	<Ausdruck>	→	(<Ausdruck>)	*	<Ausdruck>		
(r_4)	<Ausdruck>	→	(<Ausdruck>)	*	id		
(r_1)	<Ausdruck>	→	(<Ausdruck>	+	<Ausdruck>)	*	id
(r_4)	<Ausdruck>	→	(<Ausdruck>	+	id)	*	id
(r_4)	<Ausdruck>	→	(id	+	id)	*	id
$(r_5)^3$	< id >	→	(3	+	5)	*	4

Die Grammatik G_1 läßt sich wie folgt stärker formalisieren. Wenn die Variable ***E*** die Variable <Ausdruck> vertritt, dann nehmen die Produktionsregeln von G_1 die Form

(r_1)	E	→	E	+	E
(r_2)	E	→	E	*	E
(r_3)	E	→	(E)		
(r_4)	E	→	id		
(r_5)	id	→	3 | 5 | 4		

an. Stellt man diese Produktionsregeln von E mit Hilfe der Disjunktion dar, notiert mit dem

---

Notation findet üblicherweise bei der syntaktischen Beschreibung prozeduraler Programmiersprachen Anwendung.

[19] Die Notation $(r)^3$ bedeutet die 3-fache Iteration der Produktionsregel r. $(r)^n$ bedeutet entsprechend die n-fache Iteration von r.

Disjunktionsoperator | , lassen sich (r_1), (r_2) , (r_3) und (r_4) wie folgt zusammenfassen:

(r)	E	→	E + E	|	E * E	|	(E)	|	***id***

Die Anwendung der Produktionsregeln auf '(3+5)*4' sieht dann wie folgt aus:

(r_2)	E	→	E * E	
(r_3)	E	→	(E) * E	
(r_4)	E	→	(E) * id	
(r_1)	E	→	(E + E) * id	
(r_4)	E	→	(E + id) * id	
(r_4)	E	→	(id + id) * id	
(r_5)^3	id	→	(3 + 5) * 4	

(2) Die kontextfreie Grammatik G_2

Gegeben sei nun folgende Grammatik G_2 :

G_2 = { {S} , { a, b} , R, S } mit der Menge der Produktionsregeln

R = {

S → a S b (r_1)

S → ab (r_2)

}

Die Grammatik G_2 ist kontextfrei und erzeugt Wörter, die eine bestimmte Anzahl von Terminalen ‘a’ gefolgt von der gleichen Anzahl an Terminalen ‘b’ besitzen. Speziell gilt V = { S } und T = { a, b}. Wir leiten in G_2 exemplarisch die Wörter **ab**, $\mathbf{a^3b^3}$ und $\mathbf{a^nb^n}$ ab. Die Ableitung terminaler Ketten der Länge 2n erfolgt in G_2 durch die (n-1)-malige Anwendung von (r_1) und anschließend durch die 1-malige Anwendung von (r_2). Für die drei oben genannten Wörter gelten folgende Ableitungen:

(a) Ableitung des Wortes a^3b^3

(r_1) $S \rightarrow aSb$
(r_1) $\rightarrow aaSbb$
(r_2) $\rightarrow aaabbb$ [20]

2-malige Anwendung von (r_1) ,
dann anschließend
1-malige Anwendung von (r_2).

(b) Ableitung des Wortes ab

(r_2) $S \rightarrow a\ b$

1-malige Anwendung von (r_2) ,

(c) Ableitung des Wortes a^nb^n

(r_1) $S \rightarrow a\ S\ b$
$(r_1)^{n-2}$ $\rightarrow a^{n-1}\ S\ b^{n-1}$
(r_2) $\rightarrow a^{n-1}\ a\ b\ b^{n-1}$
$= a^n\ b^n$

(n-1)-malige Anwendung von (r_1) ,
dann anschließend
1-malige Anwendung von (r_2).

3.3.2. Ableitung und Ableitungsbaum

3.3.2.1. Direkte Ableitung und Ketten von Ableitungen

3.3.2.1.1. Direkte Ableitung

Ein Wort $v_0 \in T$ heißt in der kontextfreien Grammatik G = { V, T, R, S } aus v_1 *direkt ableitbar*, wenn es eine Produktionsregel $p \in R$ gibt, so daß für $v_1 \in V \cup T$

$$(p)\ v_1 \rightarrow v_0$$

gilt. Dies kann wie folgt schematisch dargestellt werden:

$$v_1 \overset{G}{\Rightarrow} v_0 \quad , v_1 \in V \cup T\ ,\ v_0 \in T$$

[20] Für dieses Wort wird nur zur Abkürzung die Notation a^3b^3 verwendet. Die Grammatik G_2 stellt keine Beschreibung von Potenzen dar.

oder, falls der Beschreibungskontext klar ist, auch durch die verkürzte Darstellung

$$v_1 \overset{}{\underset{G}{\Rightarrow}} v_n.$$

ausgedrückt werden.

3.3.2.1.2. Kette von Ableitungen

Ein Wort v_n kann in der kontextfreien Grammatik G = { V, T, R, S } aus v_1 *abgeleitet* werden, wenn es eine Folge von Produktionsregeln $<p_k>$ aus R gibt, so daß für $v_j \in V \cup T$ ($1 \leq j < n$) , $v_n \in T$ und die Produktionen $p_k \in R$ mit $1 \leq k \leq n-1$

$$(p_1)\ v_1 \rightarrow v_2 \ \wedge (p_2)\ v_2 \rightarrow v_3 \ \wedge \ \ldots \ \wedge (p_{n-1})\ v_{n-1} \rightarrow v_n$$

gilt.[21] Die Ableitungsschritte können vereinfacht wie folgt dargestellt werden:

$$v_1 \underset{p_1}{\longrightarrow} v_2 \underset{p_2}{\longrightarrow} \ldots \longrightarrow v_{n-1} \underset{p_{n-1}}{\rightarrow} v_n$$

Dies kann wie folgt schematisch dargestellt werden:

$$v_1 \overset{*}{\underset{G}{`}} v_n \quad , v_1, \ \ldots, v_{n-1} \in V \cup T \ , v_n \in T$$

oder, falls die Grammatik aus dem Beschreibungskontext klar hervorgeht, auch durch die verkürzte Darstellung

$$v_1 \overset{*}{\Rightarrow} v_n.$$

ausgedrückt werden.

[21] Die Produktionsregeln p_k müssen nicht notwendig verschieden sein.

Die Gesamtheit der Ausdrücke, die in einer bestimmten Grammatik G ableitbar sind, wird als die von der Grammatik erzeugte Sprache L (G) bezeichnet. Dies sagt folgende

3.3.2.2. Definition

Die von der Grammatik G erzeugte Sprache L (G) ist die Menge der in G ableitbaren Wörter, d.h.

$$L(G) = \{ v \mid S \overset{*}{\underset{G}{\Rightarrow}} v \}$$

Beispiel

Für die von der Grammatik G_1 erzeugte Sprache $L(G_1)$ gilt

$$L(G_1) = \{ ab, a^2b^2, ..., a^nb^n \}$$

$$= \{ a^n b^n \mid n \in \mathbb{N} \}$$

3.3.2.3. Kontextfreie Grammatiken und Sprachen

3.3.2.3.1. Kontextfreie Sprachen

Eine Sprache L(G) heißt *kontextfreie Sprache*, wenn die Grammatik G, die L(G) erzeugt, kontextfrei ist.

3.3.2.3.2. Äquivalente kontextfreie Grammatiken

Sei G_n eine kontextfreie Grammatik und $L(G_n)$ die von G_n erzeugte kontextfreie Sprache. Dann heißt die kontextfreie Grammatik G_m *äquivalent* zu G_n, wenn die von G_m erzeugte kontextfreie Sprache $L(G_m)$ gleich $L(G_n)$ ist.

3.3.2.4. Ableitungsbäume

3.3.2.4.1. Aufbau des Ableitungsbaums

Jeder Ableitung läßt sich eindeutig ein Ableitungsbaum zuordnen. Dabei entspricht jeder Ableitungsregel ein Knoten des Ableitungsbaums. Dem Startsymbol des Ableitungsbaums entspricht die Wurzel des Ableitungsbaums. Ein nachgeordneter innerer Knoten entspricht einer Anwendung einer Produktionsregel aus R und wird entsprechend indiziert. Der Produktionsregel r: $X \longrightarrow Y\ Z$ mit $X, Y, Z \in V \cup T$ entspricht der (innere) Teilknoten:

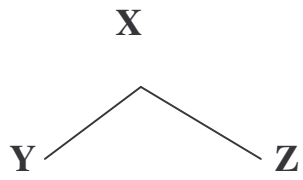

Abb.1: Innerer Knoten eines Ableitungsbaums

Ein Endknoten entspricht einer Ersetzung eines nichtterminalen Symbols durch ein terminales Symbol. Der Ableitungsbaum kann etikettiert werden, um die Ordnung der Anwendung der Produktionsregeln sichtbar zu machen.[22]

[22] Die Etikettierung von Ableitungsbäumen wird in 3.3.2.4.2.1.Beispiel.(Abb.4),(Abb.5) und in 3.3.2.4.2.2.Beispiel.(Abb.6),(Abb.7) angewandt.

Beispiele

(1) Nachfolgende Ableitung des Ausdrucks a^5b^5 in der kontextfreien Grammatik G_2 zeigt diesen Zusammenhang zwischen Ableitung(en) und Ableitungsbaum auf: es gilt dann entsprechend 3.3.2.1.:

$$
\begin{aligned}
S &\xrightarrow{r_1} aSb \\
\wedge \quad aSb &\xrightarrow{r_1} aaSbb \\
\wedge \quad aaSbb &\xrightarrow{r_1} aaaSbbb \\
\wedge \quad aaaSbbb &\xrightarrow{r_1} aaaaSbbbb \\
\wedge \quad aaaaSbbbb &\xrightarrow[r_2]{} aaaaabbbbb
\end{aligned}
$$

Das bedeutet, daß sich das Wort a^5b^5 in der Grammatik G_2 durch die Ableitungskette (r_1) (r_1), (r_1), (r_1), (r_2) ableiten läßt. Wegen

$$S \underset{r_1}{\longrightarrow} aSb \underset{r_1}{\longrightarrow} aaSbb \underset{r_1}{\longrightarrow} aaaSbbb \underset{r_1}{\longrightarrow} aaaaSbbbb \underset{r_2}{\longrightarrow} aaaaabbbbb$$

für a, b, $S \in V \cup T$ und den Produktionen $r_i \in R$ $(1 \leq i \leq 2)$ gilt[23]:

$$S \overset{*}{\underset{G_1}{\Rightarrow}} aaaaabbbbb$$

bzw. in einem geeigneten Beschreibungskontext vereinfachend

[23] $V \cup T$ hat tatsächlich 3 Elemente. Hier gilt: $V \cup T = \{ a, b, S \}$

$S \overset{*}{\hookrightarrow}$ aaaaabbbbb

Dieser Ableitung entspricht dann eineindeutig der folgende nach unten gerichtete offene Graph ("*Ableitungsbaum*"):

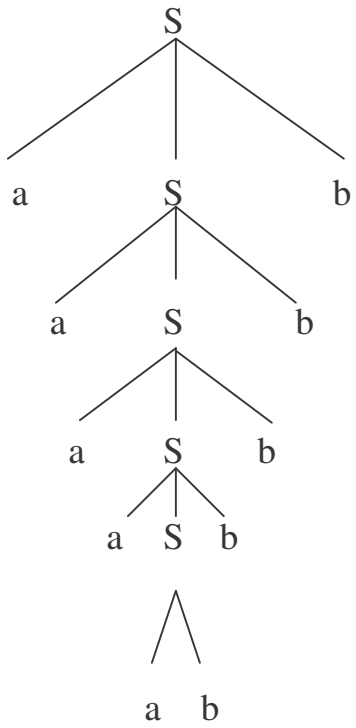

Abb.2: Ableitungsbaum a^3b^3

(2) Die kontextfreie Grammatik G_3

Gegeben sei die folgende kontextfreie Grammatik

$$G_3 = \{ \{ E, id \}, \{ +, *, 3, 5, 4 \}, R , S \}$$

mit den Produktionsregeln

$$\begin{aligned} R = \{ \; & S \rightarrow \; E + E \\ & S \rightarrow \; E * E \\ & S \rightarrow \; id \\ & id \rightarrow \; 3 \mid 5 \mid 4 \\ \} & \end{aligned}$$

Die Grammatik G_3 erzeugt einfache arithmetische Ausdrücke. Die Grammatik G_3 unterscheidet sich von G_1 dadurch, daß in ihr keine geklammerten Ausdrücke erzeugt werden können. Die Grammatik G_3 kennt daher keine Prioritäten der arithmetischen Operatoren. Nachfolgende Abbildung zeigt den Ableitungsbaum, der zu der Ableitung

$$E \rightarrow \quad E * E \rightarrow \quad E \ * \ E + E \rightarrow \quad id * id + id \rightarrow \quad 3 * 5 + 4$$

des Ausdrucks '3*5+4' in G_3 gehört:

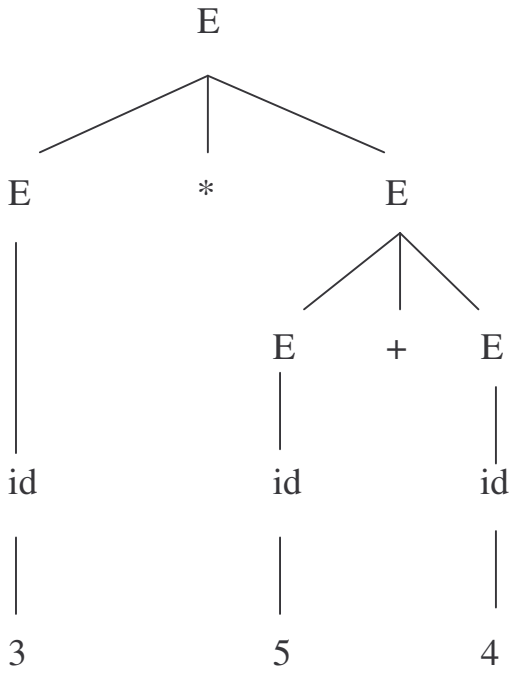

Abb.3: Ableitungsbaum von '3*5+4' in G_3

Grundsätzlich kann ein Wort mehrere Ableitungen besitzen. Insbesondere besitzt '3+5*4' weitere Ableitungen.[24] Diese können auf denselben Ableitungsbaum oder auf verschiedene Ableitungsbäume führen. Dies gibt Anlaß, den Begriff der *Äquivalenz* von Ableitungen zu definieren.

[24] Dies möge der Leser in einer Übungsaufgabe nachweisen!

3.3.2.4.2 Äquivalenz von Ableitungen

Sei G = { V, T , R, S } eine kontextfreie Grammatik. Seien ferner $<p_k>$ und $<q_i>$ zwei Folgen von Produktionsregeln aus R mit

$$(1) \quad v_1 \underset{p_1}{\longrightarrow} v_2 \underset{p_2}{\longrightarrow} \ldots \longrightarrow v_{n-1} \underset{p_{n-1}}{\longrightarrow} v_n$$

und

$$(2) \quad v_1 \underset{q_1}{\longrightarrow} v'_2 \underset{q_2}{\longrightarrow} \ldots \longrightarrow v'_{n-1} \underset{q_{n-1}}{\longrightarrow} v_n$$

zwei Ableitungen von v_n in G.[25] Die Ableitungen (1) und (2) heißen genau dann *äquivalent*, wenn sie denselben Ableitungsbaum erzeugen.

3.3.2.4.3 Rechts- und Linksableitungen

Die Anforderungen an die bisher vorgenommen Ableitungen können präzisiert werden. Anders als in den bisher vorgenommenen Ableitungen muß nicht zugelassen werden, daß ein beliebiges Symbol einer nichtterminalen Kette ersetzt werden darf. Stattdessen kann gefordert werden, daß in jedem Ableitungsschritt nur ein einziges nichtterminales Symbol zu ersetzen ist. Ist dies jeweils das am weitesten links- oder rechts stehende nichtterminale Symbol, so spricht man von Links- bzw. Rechtsableitungen. Alle Wörter, die sich überhaupt ableiten lassen, müssen solche Ableitungen besitzen.[26] In diesem Rahmen ergibt sich dann auch ein Ansatz zur Normierung von Ableitungen und es wird möglich sein, die Länge von Ableitungen, d.h. die Anzahl der Ableitungsschritte, zu messen.[27]

---

[25] Zum Wertebereich von V, T und dem der Zähler i und k siehe 3.3.2.1.

[26] *(in Vorbereitung)*

[27] *(in Vorbereitung)*

3.3.2.4.3.1 Rechtsableitung

Der Begriff *Rechtsableitung* läßt sich dadurch definieren, daß gefordert wird, daß in jedem Ableitungsschritt nur ein einziges nichtterminales Symbol, und zwar das am weitesten rechts stehende nichtterminale Symbol, ersetzt werden darf. Eine derartige Ableitung heißt *Rechtsableitung*.

3.3.2.4.3.2 Linksableitung

Man kann von einer Ableitung verlangen, daß die Produktionsregeln jeweils immer auf die am weitesten links stehende Variable angewendet werden. Man kann dann zeigen, daß jedem Ableitungsbaum genau eine solche Linksableitung zugeordnet werden kann, da durch die Forderung, jeweils immer die erste Variable zu ersetzen, die Reihenfolge der Regelanwendungen durch den Baum festgelegt ist. Wie dies geschieht, wird im Folgenden gezeigt.

3.3.2.4.3.2.1. Definition

Eine Ableitung, in der bei jedem Ableitungsschritt die am weitesten links stehende nichtterminale Variable ersetzt wird, heißt *Linksableitung*.

Bemerkung

Es läßt sich zeigen, daß insbesondere jedes ableitbare Wort eine Linksableitung besitzt.[28]

Beispiel

Gegeben sei die Grammatik G_4 mit dem Startsymbol S, dem nichtterminalen Vokabular V = { S }, dem nichtterminalen Vokabular T = { a, b } und R mit den Produktionsregeln

(r_1) $S \rightarrow SS$

(r_2) $S \rightarrow bS$

(r_3) $S \rightarrow a$

[28] `(in Vorbereitung)`

Es gilt ***bbaba*** $\in L(G_4)$, denn es läßt sich zeigen, daß es eine Ableitung von ***bbaba*** in G_4 gibt. Es gibt sogar mehrere derartige Ableitungen, wie die folgenden Überlegungen zeigen. Zunächst stellt die Ableitung

$$(R) \quad \left\{ \begin{array}{llll} (1) & (r_1) & S & \rightarrow SS \\ (2) & (r_2) & & \rightarrow SbS \\ (3) & (r_3) & & \rightarrow Sba \\ (4) & (r_2) & & \rightarrow bSba \\ (5) & (r_2) & & \rightarrow bbSba \\ (6) & (r_3) & & \rightarrow bbaba \end{array} \right.$$

eine Rechtsableitung von ***bbaba*** in G_4 dar. [29]

Dieser Rechtsableitung (r_1) (r_2) (r_3) (r_2) (r_2) (r_3) ist folgender Ableitungsbaum eindeutig zugeordnet:

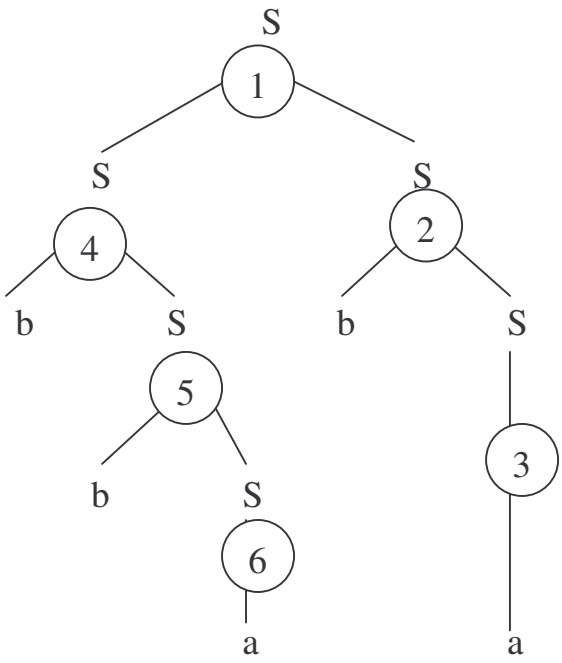

Abb.4: Ableitungsbaum zu (R) in G_4

[29] Zusätzlich zur Markierung der zu ersetzenden am weitesten rechts stehenden nichtterminalen Symbole sind die Ableitungsschritte und die anzuwendenden Produktionsregeln angegeben.

Darüber hinaus stellt die Ableitung

$$
(L)\quad
\begin{cases}
(1) \quad (r_1) \quad S \quad \rightarrow SS \\
(2) \quad (r_2) \quad \quad \rightarrow bSS \\
(3) \quad (r_2) \quad \quad \rightarrow bbSS \\
(4) \quad (r_3) \quad \quad \rightarrow bbaS \\
(5) \quad (r_2) \quad \quad \rightarrow bbabS \\
(6) \quad (r_3) \quad \quad \rightarrow bbaba
\end{cases}
$$

eine Linksableitung von ***bbaba*** in G_4 dar.[30] Dieser Linksableitung $(r_1)\ (r_2)\ (r_2)\ (r_3)\ (r_2)\ (r_3)$ ist folgender Ableitungsbaum eindeutig zugeordnet:

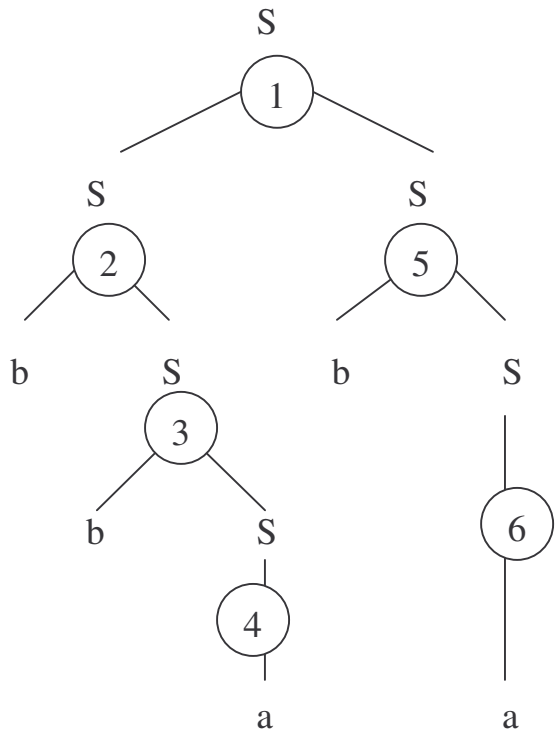

Abb.5: Ableitungsbaum zu (L) in G_4

Die Ableitungen (R) und (L) sind äquivalent. Allgemein gilt:

[30] Zusätzlich zur Markierung der zu ersetzenden am weitesten rechts stehenden nichtterminalen Symbole sind die Ableitungsschritte und die anzuwendenden Produktionsregeln angegeben.

3.3.2.4.3.2.2. Satz

Zu jeder Ableitung

$$S \overset{*}{\underset{G}{\Rightarrow}} v$$

von $v \in T$ in der kontextfreien Grammatik G existiert genau eine äquivalente Linksableitung.[31]

Bemerkung

Wir können im vorangegangenen Beispiel der Ableitung (R) tatsächlich eine äquivalente Linksableitung (L) zuordnen. Durch die Forderung, jeweils immer die am weitesten links bzw. rechts stehende Variable zu ersetzen, wird auch die Ordnung der Regelanwendungen durch den Baum festgelegt. Es können dennoch mehrere Linksableitungen zu einem Wort gehören. Dies zeigt folgendes

Beispiel

Gegeben sei wieder die Grammatik $G_3 = \{ \{ E, id \} , \{ +, *, 3, 5, 4 \}, R, \{ E \} \}$ mit den Produktionsregeln

,

$$R = \{$$

$$E \rightarrow E + E \qquad (r_1)$$

$$E \rightarrow E * E \qquad (r_2)$$

$$E \rightarrow id \qquad (r_3)$$

$$id \rightarrow 3 \mid 5 \mid 4 \qquad (r_4) \}$$

Es ist mindestens eine Linksableitung zu dem Wort

$$v = 3 + 5 * 4$$

in G_3 anzugeben.

[31] Beweisidee: *(in Vorbereitung)*

Man findet leicht die folgende Ableitung von v in G_3 , in der jeweils das am weitesten links stehende nichtterminale Symbol ersetzt wird:

(L1)

(1)	(r_1)	E	→	E + E
(2)	(r_3)		→	id + E
(3)	(r_4)		→	3 + E
(4)	(r_2)		→	3 + E * E
(5)	(r_3)		→	3 + id * E
(6)	(r_4)		→	3 + 5 * E
(7)	(r_3)		→	3 + 5 * id
(8)	(r_4)		→	3 + 5 * 4

Also ist die Ableitungskette

$$E \rightarrow E + E \rightarrow id + E \rightarrow 3 + E \rightarrow 3 + E * E \rightarrow 3 + id * E \rightarrow 3 + 5 * E \rightarrow 3 + 5 * id \rightarrow 3 + 5 * 4$$

eine Linksableitung von '3+5*4' in $G3_4$. (L1) wird folgender Ableitungsbaum zugeordnet:

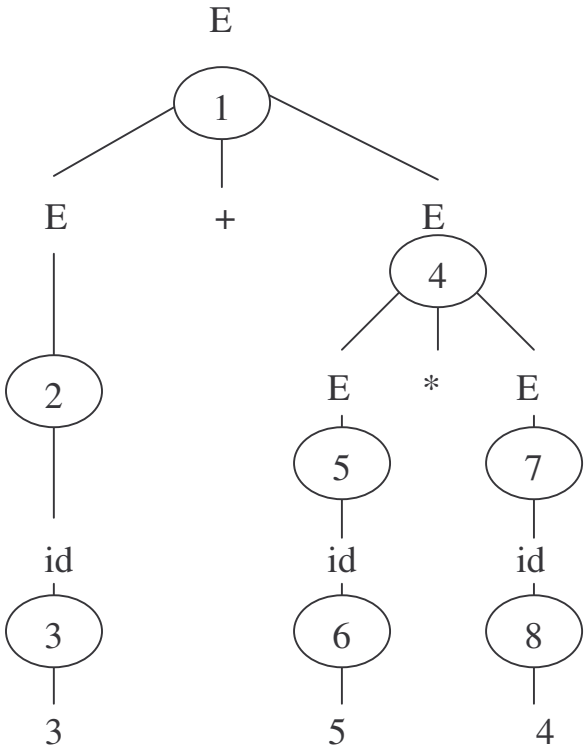

Abb.6: Ableitungsbaum zu (L1) in G_3

Neben (L1) existiert in der Grammatik G_3 aber noch eine weitere Linksableitung von '3+5*4' Dies zeigt die folgende Ableitungskette:

(L2)

(1)	(r_2)	E	→	E * E
(2)	(r_1)		→	E + E * E
(3)	(r_3)		→	id + E * E
(4)	(r_4)		→	3 + E * E
(5)	(r_3)		→	3 + id * E
(6)	(r_4)		→	3 + 5 * E
(7)	(r_3)		→	3 + 5 * id
(8)	(r_4)		→	3 + 5 * 4

Folglich ist auch die Ableitungskette

$$E \rightarrow E * E \rightarrow E + E * E \rightarrow id + E * E \rightarrow 3 + E * E \rightarrow 3 + id * E \rightarrow 3 + 5 * E \rightarrow 3 + 5 * id \rightarrow 3 + 5 * 4$$

eine Linksableitung des Ausdrucks '3+5*4' in der Grammatik G_3. Der Linksableitung (L2) wird folgender Ableitungsbaum eindeutig zugeordnet:

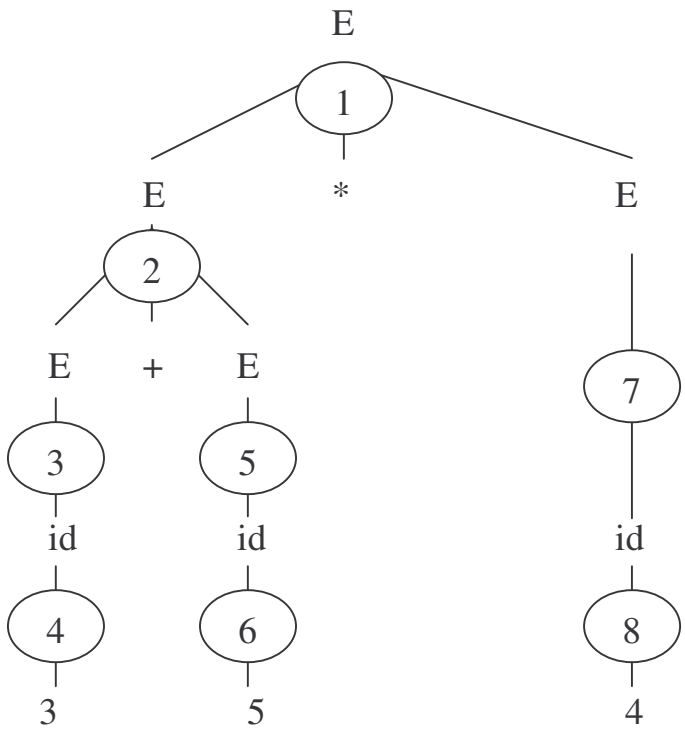

Abb.7: Ableitungsbaum zu (L2) in G_3

Beide Ableitungen sind nicht äquivalent. Der Satz v = **3 + 5 * 4** und damit die Grammatik G_3 sind mehrdeutig.

3.3.2.5. Mehrdeutigkeit in kontextfreien Grammatiken

Aus dem vorangegangen Beispiel ergibt sich also folgendes Kriterium zur Unterscheidung von eindeutigen bzw. mehrdeutigen kontextfreien Grammatiken.

3.3.2.5.1. Definition

Eine Grammatik G heißt genau dann *mehrdeutig*, wenn es (mindestens) ein $v \in L(G)$ gibt, zu dem mindestens zwei Linksableitungen existieren.

3.3.2.5.2. Definition

Eine Grammatik G heißt genau dann *eindeutig*, wenn für alle $v \in L(G)$ genau eine Linksableitung existiert.

3.3.4. Normalformen kontextfreier Grammatiken

Zur effizienten Behandlung des Spracherkennungsproblems müssen Grammatiken zu Normalformen vereinfacht werden. Jede kontextfreie Grammatik kann durch geeignete Operationen schrittweise z.B. in die sog. *Chomsky-Normalform* gebracht werden, die dann die Anwendung von Algorithmen zur effizienten Lösung des Wortproblems ermöglicht. [32]

3.3.4.1. Chomsky-Normalform

Eine kontextfreie Grammatik ist in der *Chomsky-Normalform*, wenn alle Regeln die Form

$X \rightarrow YZ$ oder

$X \rightarrow t$

haben, wobei $X, Y, Z \in V$ und $t \in T$ gilt. Dies läßt sich verkürzt durch folgende Produktionsregel darstellen:

$X \rightarrow YZ \mid t$

Beispiel

Wir betrachten die Grammatik $G_2 = \{ \{S\} , \{ a, b\} , R, S \}$ von 3.3.1.Beispiele(2). Die Grammatik G_2 ist zu folgender kontextfreien Grammatik $G'_2 = \{\{S, A, B\}, \{ a, b\} , R', S\}$ mit den Produktionsregeln

$R' = \{$

$S \rightarrow ASB$	(r_1)
$S \rightarrow AB$	(r_2)
$A \rightarrow a$	(r_3)
$B \rightarrow b$	(r_4)

$\}$

[32] *(in Vorbereitung)*

äquivalent. Die Grammatik G'_2 ist eine kontextfreie Grammatik im Sinne von (3.3.1.). Die Grammatik G_2' befindet zudem sich in der Chomsky-Normalform (CNF). Wir leiten in G'_2 exemplarisch die Wörter $\mathbf{a^3b^3}$, **ab** und $\mathbf{a^nb^n}$ ab.[33] Die Ableitung terminaler Ketten der Länge $2n$ erfolgt in G'_2 durch (n-1)-malige Anwendung von (r_1) , anschließend durch die 1-malige Anwendung von (r_2), anschließend durch die n-malige Anwendung von (r_3) und zuletzt die n-malige Anwendung von (r_4). Für die drei oben genannten Wörter gelten folgende Ableitungen.

(a) Ableitung des Wortes a^3b^3

(r_1) $S \rightarrow ASB$
(r_1) $\rightarrow AASBB$
(r_2) $\rightarrow AAABBB$
$(r_3)^3$ $\rightarrow aaaBBB$
$(r_4)^3$ $\rightarrow aaabbb = a^3b^3$

2-malige Anwendung von (r_1) ,

dann anschließend

1-malige Anwendung von (r_2),
3-malige Anwendung von (r_3) und
3-malige Anwendung von (r_4).

(b) Ableitung des Wortes a^nb^n

(r_1) $S \rightarrow ASB$
…..
$(r_1)^{n-2}$ $\rightarrow A^{n-1} S B^{n-1}$
(r_2) $\rightarrow A^{n-1} AB B^{n-1} = A^nB^n$
$(r_3)^n$ $\rightarrow a^nB^n$
$(r_4)^n$ $\rightarrow a^nb^n$

(n-1)-malige Anwendung von (r_1) ,

dann anschließend

1-malige Anwendung von (r_2),
n-malige Anwendung von (r_3) und
n-malige Anwendung von (r_4)

(c) Ableitung des Wortes ab

(r_2) $S \rightarrow AB$
(r_3) $\rightarrow aB$
(r_4) $\rightarrow ab$

1-malige Anwendung von (r_2) ,
1-malige Anwendung von (r_3) und
1-malige Anwendung von (r_4).

[33] Zur Notation vgl. 3.3.1.Beispiel(2).

Die Grammatik G'_2 erzeugt dieselbe Sprache wie G_2. Es gilt $L(G'_2) = L(G_2)$.[34]

3.4. Der CYK-Algorithmus

(in Vorbereitung)

[34] Vgl. 3.3.2.2.Beispiel