Grammatiken

Grammatiken sind regelbasierte Kalküle zur

- Konstruktion von Systemen und Sprachen
- Überprüfung von Systemen und Sprachen

Grammatiken eignen sich besonders zur Modellierung

- beliebig tief geschachtelter, rekursiver Strukturen.

Es existieren verschiedene Typen von Grammatiken

- allgemeine Grammatiken (Typ 0)
- kontextsensitive Grammatiken (Typ 1)
- kontextfreie Grammatiken (Typ 2)
- reguläre Grammatiken (Typ 3)

Betrachten

- **kontextfreie Grammatiken**

Geschachtelte Strukturen - Beispiele

- Tabellen mit Tabellen als Einträgen
- Aufbau von Webseiten: Aufzählungen von Aufzählungen
- Arithmetische und Boolesche Ausdrücke
- Geschachtelte Schleifen in Programmiersprachen
- Klammerausdrücke

Klammerausdrücke

- das leere Wort ϵ ist ein Klammerausdruck
- eine Liste von Klammerausdrücke ist wieder ein Klammerausdruck, z.B. ()()()()
- ist K ein Klammerausdruck, dann auch (K), z.B. $K = ()()$ und $(()())$

Grammatiken

Definition 1

Eine Grammatik G ist ein 4-Tupel (T, N, P, S). Dabei ist

- T eine endliche Menge von Terminalen
- N eine endliche Menge von Nichtterminalen mit $N \cap T = \emptyset$
- P eine endliche Menge von Produktionen
- $S \in N$ das Startsymbol.

Die Elemente aus $V := T \cup N$ heißen Symbole und es gilt

$$P \subseteq (V^+ \setminus T^*) \times V^*.$$

- Für eine Produktion $(A, x) \in P$ schreiben wir auch $A ::= x$.
- Die erste Komponente einer Produktion ist eine nicht-leere Folge von Symbolen, wobei mindestens ein Nichtterminal in der Folge auftauchen muss.

Grammatiken

Definition 1

Eine Grammatik G ist ein 4-Tupel (T, N, P, S). Dabei ist

- T eine endliche Menge von Terminalen
- N eine endliche Menge von Nichtterminalen mit $N \cap T = \emptyset$
- P eine endliche Menge von Produktionen
- $S \in N$ das Startsymbol.

Die Elemente aus $V := T \cup N$ heißen Symbole und es gilt

$$P \subseteq (V^+ \setminus T^*) \times V^*.$$

- Sagen: In der Produktion $A ::= x$ steht A auf der linken Seite und x auf der rechten Seite.
- Geben Produktionen häufig unterschiedliche Namen, wie $p_1 : A ::= x$.

Grammatiken

$\tilde{G} = (T, N, P, S)$ mit

$$T = \{a, b, c\}$$
$$N = \{S, B, C\}$$

$$P = \begin{cases} p1: & S & ::= & aSBC \\ p2: & S & ::= & aBC \\ p3: & CB & ::= & BC \\ p4: & aB & ::= & ab \\ p5: & bB & ::= & bb \\ p6: & bC & ::= & bc \\ p7: & cC & ::= & cc \end{cases}$$

Kontextfreie Grammatiken

Definition 2

Eine Grammatik $G = (T, N, P, S)$ heißt kontextfrei, wenn

$$P \subseteq N \times V^*,$$

also, wenn die linke Seite jeder Produktion aus einem einzelnen Nichtterminal besteht.

Kontextfreie Grammatiken werden angewandt zur Definition von

- Programmen einer Programmiersprache und deren Struktur, z.B. Java, C, Pascal
- Sprachen als Schnittstellen zwischen Software-Werkzeugen und Datenaustauschformaten, z.B. HTML, XML
- Bäumen zur Repräsentation von strukturierten Daten, z.B. XML
- Strukturen von Protokollen beim Austausch von Nachrichten zwischen Prozessen oder Geräten

Kontextfreie Grammatiken

Definition 2

Eine Grammatik $G = (T, N, P, S)$ heißt kontextfrei, wenn

$$P \subseteq N \times V^*,$$

also, wenn die linke Seite jeder Produktion aus einem einzelnen Nichtterminal besteht.

Kontextfreie Grammatiken sind

- komplex genug, um interessante Strukturen mit ihnen zu beschreiben
- einfach genug, um Aussagen über durch sie definierten Strukturen herleiten zu können, z.B. ist ein Programm syntaktisch korrekt

Grammatiken

$G_1 = (T, N, P, S)$ mit

$$
\begin{aligned}
T &= \{\text{MenüName,OperationsName}\}\\
N &= \{\text{Menü,EintragsFolge,Eintrag}\}\\
S &= \ \text{Menü}\\
P &= \{\text{Menü} ::= \text{MenüName EintragsFolge},\\
&\quad\ \text{EintragsFolge} ::= \text{Eintrag},\\
&\quad\ \text{Eintrag} ::= \text{OperationsName},\\
&\quad\ \text{Eintrag} ::= \text{Menü}\\
&\quad \}
\end{aligned}
$$

Die Grammatik G_1 ist kontextfrei.

Klammerausdrücke

$G_2 = (T, N, P, S)$ mit

$$T = \{(\, ,)\}$$
$$N = \{Klammerung, Liste\}$$
$$S = \; Klammerung$$

$$P = \{Klammerung ::= \text{‘(‘} Liste \text{‘)‘},$$
$$Liste ::= Klammerung\ Liste,$$
$$Liste ::= \epsilon$$
$$\}$$

ist kontextfrei.

- ϵ bezeichnet das **leere Wort**
- Um Sonderzeichen, die Terminale einer Grammatik sind, von Zeichen einer Grammatikdefinition zu unterscheiden, setzen wir sie häufig in Apostrophe, z.B. ‘(‘ und ‘)‘ in der Grammatik G_2.

Grammatiken

$\tilde{G} = (T, N, P, S)$ mit

$$T = \{a, b, c\}$$
$$N = \{S, B, C\}$$

$$P = \begin{cases} p1: & S & ::= & aSBC \\ p2: & S & ::= & aBC \\ p3: & CB & ::= & BC \\ p4: & aB & ::= & ab \\ p5: & bB & ::= & bb \\ p6: & bC & ::= & bc \\ p7: & cC & ::= & cc \end{cases}$$

Die Grammatik $\tilde{G}$ ist nicht kontextfrei.

Ableitungen

Definition 3

Sei $G = (T, N, P, S)$ eine kontextfreie Grammatik.

- Sei $w = uAv \in V^+$. Existiert in P eine Produktion $A ::= x$, so ist $w' = uxv$ aus uAv (direkt) ableitbar. Wir schreiben $w \rightarrow w'$.
- Eine solche einmalige Anwendung einer Produktion heißt Ableitungsschritt.
- Seien $w \in V^+, w' \in V^*$. Dann ist w' aus w (indirekt) ableitbar, wenn wir w' aus w durch endlich viele Ableitungsschritte erhalten können, d.h. wenn $w_0 = w, w_1, \ldots, w_n = w'$ existieren mit $w_{i-1} \rightarrow w_i, i = 1, \ldots, n$. Ist w' aus w ableitbar, so schreiben wir $w \rightarrow^* w'$.
- Eine Folge von Ableitungsschritten nennen wir eine Ableitung.

Klammerausdrücke

$G_2 = (T, N, P, S)$ mit

$$T = \{(\ ,)\}$$
$$N = \{Klammerung, Liste\}$$
$$S = \ Klammerung$$

$$\begin{aligned} P = \{ & Klammerung ::= \text{`('}Liste\text{`)'}, \\ & Liste ::= Klammerung\ Liste, \\ & Liste ::= \epsilon \\ \} & \end{aligned}$$

Ableitungsschritte in G_2

$$\begin{aligned} (Liste) &\to (Klammerung\ Liste) \\ ((Liste)Liste) &\to ((\ )Liste) \\ Klammerung &\to (Liste) \end{aligned}$$

Klammerausdrücke

$G_2 = (T, N, P, S)$ mit

$$T = \{(\, ,)\}$$
$$N = \{Klammerung, Liste\}$$
$$S = \ Klammerung$$

$$P = \{Klammerung ::= \text{‘(‘} Liste \text{‘)‘},$$
$$Liste ::= Klammerung \ Liste,$$
$$Liste ::= \epsilon$$
$$\}$$

Eine Ableitung in G_2

$$\begin{aligned} Klammerung &\rightarrow (Liste) \\ &\rightarrow (Klammerung \ Liste) \\ &\rightarrow (Klammerung \ Klammerung \ Liste) \\ &\rightarrow (Klammerung \ (Liste) \ Liste) \\ &\rightarrow ((Liste)(Liste) \ Liste) \end{aligned}$$

Klammerausdrücke

$G_2 = (T, N, P, S)$ mit

$$T = \{(\ ,)\}$$
$$N = \{Klammerung, Liste\}$$
$$S = \ Klammerung$$

$$\begin{aligned} P = \{ & Klammerung ::= \text{`('} Liste \text{`)'}, \\ & Liste ::= Klammerung\ Liste, \\ & Liste ::= \epsilon \\ \} & \end{aligned}$$

Eine Ableitung in G_2

$$\begin{aligned} ((Liste)(Liste)\ Liste) & \rightarrow ((\ )\ (Liste)\ Liste) \\ & \rightarrow ((\ )\ (\ )\ Liste) \\ & \rightarrow ((\ )\ (\ )) \end{aligned}$$

Kurz: $Klammerung \rightarrow^* ((\ )\ (\ ))$

Grammatiken und Sprachen

Definition 4

Sei $G = (T, N, P, S)$ eine kontextfreie Grammatik. Dann heißt

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

die von G erzeugte Sprache. $L(G)$ besteht also aus allen Folgen von Terminalen, die aus dem Startsymbol S der Grammatik G abgeleitet werden können.

$G_3 = (T, N, P, S)$ mit

$$\begin{aligned} T &= \{a\} \\ N &= \{A\} \\ S &= A \\ P &= \{A ::= aA, A ::= a\} \end{aligned}$$

$$L(G_3) = \{a^n \mid n \geq 1\}$$

Grammatiken und Sprachen

Definition 4

Sei $G = (T, N, P, S)$ eine kontextfreie Grammatik. Dann heißt

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

die von G erzeugte Sprache. $L(G)$ besteht also aus allen Folgen von Terminalen, die aus dem Startsymbol S der Grammatik G abgeleitet werden können.

$G_4 = (T, N, P, S)$ mit

$$\begin{aligned} T &= \{a, b\} \\ N &= \{S\} \\ S &= \ S \\ P &= \{S ::= aSb, S ::= \epsilon\} \end{aligned}$$

$$L(G_4) = \{a^n b^n \mid n \geq 0\}$$

Ableitungsbäume

Ableitungen kontextfreier Grammatiken können graphisch durch Ableitungsbäume dargestellt werden.

Ableitungsbäume

Sei $G = (T, N, P, S)$ eine kontextfreie Grammatik. Für eine Ableitung eines Wortes $w \in T^*, w = w_1 \dots w_n$, ist der zugehörige Ableitungsbaum ein Baum mit Wurzel mit Knotenmarkierungen in V. Dabei gilt weiter

- Die Wurzel ist mit dem Startsymbol S markiert.
- Der Baum besitzt n Blätter und nur die Blätter sind mit Terminalen oder dem leeren Wort ϵ markiert. Die Markierungen der Blätter ergeben, von links nach rechts gelesen, das Wort w.
- Innere Knoten des Baums sind mit Nichtterminalen markiert. Für jeden inneren Knoten repräsentieren der Knoten und seine unmittelbaren Nachfolger die Anwendung einer Produktion.

Klammerausdrücke

$G_2 = (T, N, P, S)$ mit

$$T = \{(\ ,)\} \qquad P = \{Klammerung ::= \text{`('}Liste\text{`)'},$$
$$N = \{Klammerung, Liste\} \qquad Liste ::= Klammerung\ Liste,$$
$$S = Klammerung \qquad Liste ::= \epsilon\}$$

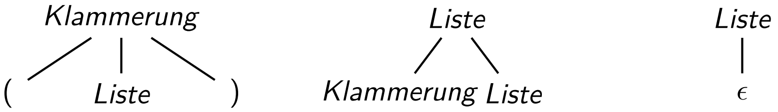
Eine Ableitung in G_2

$$\begin{aligned}
Klammerung &\rightarrow (Liste)\\
&\rightarrow (Klammerung\ Liste)\\
&\rightarrow (Klammerung\ Klammerung\ Liste)\\
&\rightarrow (Klammerung\ (Liste)\ Liste)\\
&\rightarrow ((Liste)(Liste)\ Liste)\\
&\rightarrow ((\ )\ (Liste)\ Liste)\\
&\rightarrow ((\ )\ (\ )\ Liste)\\
&\rightarrow ((\ )\ (\ ))
\end{aligned}$$

Ableitungsbaum für Klammerausdruck

$$P = \{\mathit{Klammerung} ::= \text{‘(‘}\,\mathit{Liste}\,\text{‘)‘},$$
$$\mathit{Liste} ::= \mathit{Klammerung}\ \mathit{Liste},$$
$$\mathit{Liste} ::= \epsilon\}$$

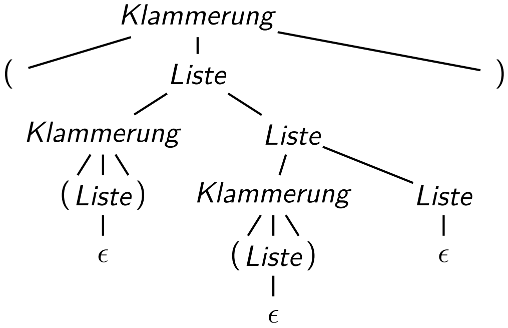
Ableitungen als Teilbäume:



Ableitungsbaum für Klammerausdruck

$$P = \{\mathit{Klammerung} ::= \text{`('}\mathit{Liste}\text{`)'},$$
$$\mathit{Liste} ::= \mathit{Klammerung}\ \mathit{Liste},$$
$$\mathit{Liste} ::= \epsilon\}$$

Ableitungsbaum für *Klammerung* $\to^*$ **(( ) ( )):**



Arithmetische Ausdrücke

Ausdruck	::=	Ausdruck BinOpr Ausdruck
Ausdruck	::=	Zahl
Ausdruck	::=	Bezeichner
Ausdruck	::=	‘(‘Ausdruck‘)‘
Startsymbol	:	Ausdruck

BinOpr	::=	‘+‘
BinOpr	::=	‘-‘
BinOpr	::=	‘*‘
BinOpr	::=	‘/‘

Ableitungsbaum zum Ausdruck $a/(b-1)$:

- Ausdruck
  - Ausdruck
    - Bezeichner (a)
  - BinOpr
    - /
  - Ausdruck
    - (
    - Ausdruck
      - Ausdruck
        - Bezeichner (b)
      - BinOpr
        - -
      - Ausdruck
        - Zahl (1)
    -)

Bezeichner, Zahl als Platzhalter für Zahlen, Variablen

Mehrdeutigkeit

Definition 5

Eine Grammatik $G = (T, N, P, S)$ heißt mehrdeutig, wenn es Elemente in $L(G)$ gibt, für die mehrere Ableitungsbäume existieren.

Mehrdeutigkeit

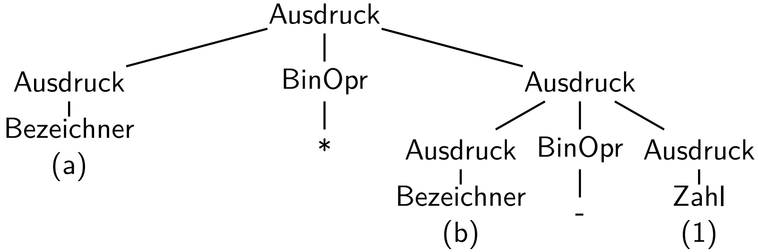
- kann zu Verwirrungen über die Bedeutung von Ausdrücken führen,
- kann den Nachweis von Eigenschaften der von einer Grammatik definierten Sprache erschweren.

Arithmetische Ausdrücke und Mehrdeutigkeit

Ausdruck	::=	Ausdruck BinOpr Ausdruck
Ausdruck	::=	Zahl
Ausdruck	::=	Bezeichner
Ausdruck	::=	‘(‘Ausdruck‘)‘
Startsymbol	:	Ausdruck

BinOpr	::=	‘+‘
BinOpr	::=	‘-‘
BinOpr	::=	‘*‘
BinOpr	::=	‘/‘

Ableitungsbäume zum Ausdruck $a * b - 1$:

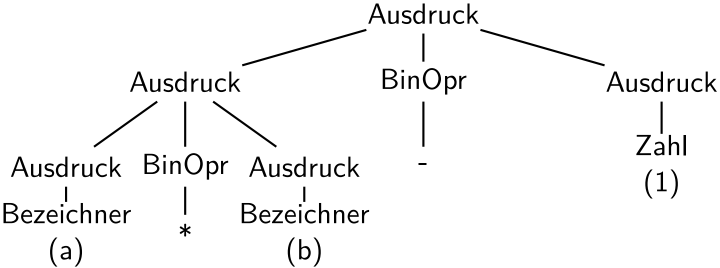


Arithmetische Ausdrücke und Mehrdeutigkeit

Ausdruck	::=	Ausdruck BinOpr Ausdruck
Ausdruck	::=	Zahl
Ausdruck	::=	Bezeichner
Ausdruck	::=	‘(‘Ausdruck‘)‘
Startsymbol	:	Ausdruck

BinOpr	::=	‘+‘
BinOpr	::=	‘-‘
BinOpr	::=	‘*‘
BinOpr	::=	‘/‘

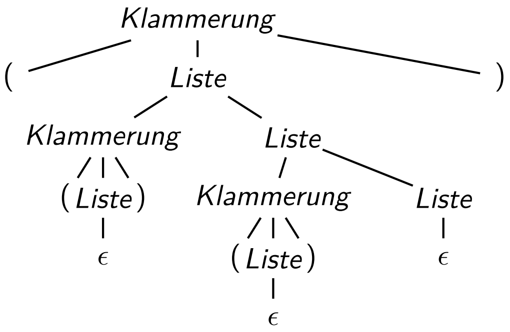
Ableitungsbäume zum Ausdruck $a * b - 1$:



Klammerausdrücke und Mehrdeutigkeit

$$\begin{aligned} P = \{ & \mathit{Klammerung} ::= \text{`('}\,\mathit{Liste}\,\text{`)'}, \\ & \mathit{Liste} ::= \mathit{Klammerung}\ \mathit{Liste}, \\ & \mathit{Liste} ::= \epsilon \} \end{aligned}$$

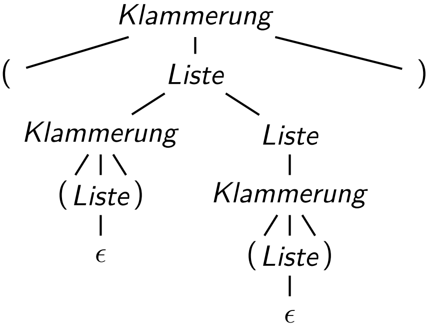
Ableitungsbaum für *Klammerung* $\rightarrow^*$ **(( ) ( )):**



Klammerausdrücke und Mehrdeutigkeit

$$P = \{\mathit{Klammerung} ::= `(`\mathit{Liste}`)`,$$
$$\mathit{Liste} ::= \mathit{Klammerung}\ \mathit{Liste},$$
$$\mathbf{Liste} ::= \mathbf{Klammerung},$$
$$\mathit{Liste} ::= \epsilon\}$$

Ableitungsbaum für $\mathit{Klammerung} \to^* (()\,())$**:**



Klammerausdrücke und Mehrdeutigkeit

$G_2 = (T, N, P, S)$ mit

$$T = \{(\ ,)\}$$
$$N = \{Klammerung, Liste\}$$
$$S = \ Klammerung$$

$$\begin{aligned} P = \{&Klammerung ::= `(`Liste`)`, \\ &Liste ::= Klammerung\ Liste, \\ &Liste ::= \epsilon \\ \}& \end{aligned}$$

Satz 6

Die Grammatik G_2 ist nicht mehrdeutig.

Spezielle Ableitungen

Sei $G = (T, N, P, S)$ eine kontextfreie Grammatik.

Linksableitungen

Eine Linksableitung ist eine Folge von Ableitungsschritten, bei der stets das am weitesten links stehende Nichtterminal durch Anwendung einer Produktion ersetzt wird.

Spezielle Ableitungen

Aritmetische Ausdrücke

Ausdruck	::=	Ausdruck BinOpr Ausdruck	BinOpr	::=	‘+‘
Ausdruck	::=	Zahl	BinOpr	::=	‘-‘
Ausdruck	::=	Bezeichner	BinOpr	::=	‘*‘
Ausdruck	::=	‘(‘Ausdruck‘)‘	BinOpr	::=	‘/‘

Beispiel einer Linksableitung:

Ausdruck → Ausdruck BinOpr Ausdruck
→ Bezeichner (a) BinOpr Ausdruck
→ Bezeichner (a) * Ausdruck
→ Bezeichner (a) * Ausdruck BinOpr Ausdruck
→ Bezeichner (a) * Bezeichner (b) BinOpr Ausdruck
→ Bezeichner (a) * Bezeichner (b) - Ausdruck
→ Bezeichner (a) * Bezeichner (b) - Zahl (1)

$(= a * b - 1)$

Spezielle Ableitungen

Sei $G = (T, N, P, S)$ eine kontextfreie Grammatik.

Rechtsableitungen

Eine Rechtsableitung ist eine Folge von Ableitungsschritten, bei der stets das am weitesten rechts stehende Nichtterminal durch Anwendung einer Produktion ersetzt wird.

Aritmetische Ausdrücke

Ausdruck	::=	Ausdruck BinOpr Ausdruck	BinOpr	::=	'+'
Ausdruck	::=	Zahl	BinOpr	::=	'-'
Ausdruck	::=	Bezeichner	BinOpr	::=	'*'
Ausdruck	::=	'('Ausdruck')'	BinOpr	::=	'/'

Beispiel einer Rechtsableitung:

$$\begin{aligned}\text{Ausdruck} &\to \text{Ausdruck BinOpr Ausdruck}\\ &\to \text{Ausdruck BinOpr Ausdruck BinOpr Ausdruck}\end{aligned}$$

Spezielle Ableitungen und Mehrdeutigkeit

Definition 5

Eine Grammatik $G = (T, N, P, S)$ heißt mehrdeutig, wenn es Elemente in $L(G)$ gibt, für die mehrere Ableitungsbäume existieren.

Satz 7

Sei $G = (T, N, P, S)$ eine kontextfreie Grammatik.

1. *G ist mehrdeutig genau dann, wenn es Elemente in $L(G)$ gibt, für die mehrere Linksableitungen existieren.*
2. *G ist mehrdeutig genau dann, wenn es Elemente in $L(G)$ gibt, für die mehrere Rechtsableitungen existieren.*

Satz 6

Die Grammatik G_2 ist nicht mehrdeutig.

Klammerausdrücke

$G_2 = (T, N, P, S)$ mit

$$T = \{(\ ,)\}$$
$$N = \{Klammerung, Liste\}$$
$$S = \ Klammerung$$

$$\begin{aligned} P = \{ & Klammerung ::= \text{`('} Liste \text{`)'}, \\ & Liste ::= Klammerung\ Liste, \\ & Liste ::= \epsilon \\ \} \end{aligned}$$

Normalformen - Backus-Naur-Form

Backus-Naur-Form (BNF)

- ist eine Kurzschreibweise zur Darstellung einer kontextfreien Grammatik $G = (T, N, P, S)$
- ist $A \in N$ und sind $A ::= x_i, x_i \in V^*, i = 1, \ldots, k$ die Produktionen in P mit linker Seite A, so werden diese zu

$$A ::= x_1 \mid x_2 \mid \ldots \mid x_k$$

zusammengefasst

Arithmetische Ausdrücke

Ausdruck	::=	Ausdruck BinOpr Ausdruck	BinOpr	::=	'+'
Ausdruck	::=	Zahl	BinOpr	::=	'-'
Ausdruck	::=	Bezeichner	BinOpr	::=	'*'
Ausdruck	::=	'('Ausdruck')'	BinOpr	::=	'/'

Normalformen - Backus-Naur-Form

Backus-Naur-Form (BNF)

- ist eine Kurzschreibweise zur Darstellung einer kontextfreien Grammatik $G = (T, N, P, S)$
- ist $A \in N$ und sind $A ::= x_i, x_i \in V^*, i = 1, \ldots, k$ die Produktionen in P mit linker Seite A, so werden diese zu

$$A ::= x_1 \mid x_2 \mid \ldots \mid x_k$$

  zusammengefasst

Arithmetische Ausdrücke in BNF

Ausdruck ::= Ausdruck BinOpr Ausdruck | Zahl | Bezeichner | '('Ausdruck')'

BinOpr ::= '+'| '-'| '*'| '/'

Normalformen - Chomsky-Normalform

Definition 8

Eine kontextfreie Grammatik $G = (T, N, P, S)$ ist in Chomsky-Normalform, wenn jede Regel in P von der Form

$$\begin{array}{lll} & A ::= BC & B, C \in N \\ \text{oder} & A ::= a & a \in T \end{array}$$

ist. Zusätzlich ist die Regel $S ::= \epsilon$ erlaubt. Wenn sie vorkommt, dann darf S nicht auf der rechten Seite einer Produktion auftauchen.

Satz 9

Sei G eine kontextfreie Grammatik. Dann kann aus G eine kontextfreie Grammatik G' in Chomsky-Normalform konstruiert werden, die dieselbe Sprache wie G erzeugt.

Normalformen - Chomsky-Normalform

Definition 8

Eine kontextfreie Grammatik $G = (T, N, P, S)$ ist in Chomsky-Normalform, wenn jede Regel in P von der Form

$$\begin{array}{rll} & A ::= BC & B, C \in N \\ \text{oder} & A ::= a & a \in T \end{array}$$

ist. Zusätzlich ist die Regel $S ::= \epsilon$ erlaubt. Wenn sie vorkommt, dann darf S nicht auf der rechten Seite einer Produktion auftauchen.

$G_3 = (T, N, P, S)$ mit

$$\begin{aligned} T &= \{a\}, N = \{A\} \\ S &= A \\ P &= \{A ::= aA, A ::= a\} \end{aligned}$$

ist nicht in Chomsky-Normalform.

Normalformen - Chomsky-Normalform

Definition 8

Eine kontextfreie Grammatik $G = (T, N, P, S)$ ist in Chomsky-Normalform, wenn jede Regel in P von der Form

$$\begin{array}{lll} & A ::= BC & B, C \in N \\ \text{oder} & A ::= a & a \in T \end{array}$$

ist. Zusätzlich ist die Regel $S ::= \epsilon$ erlaubt. Wenn sie vorkommt, dann darf S nicht auf der rechten Seite einer Produktion auftauchen.

$G_3 = (T, N, P, S)$ mit

$$\begin{aligned} T &= \{a\}, N = \{A\} \\ S &= A \\ P &= \{A ::= aA, A ::= a\} \end{aligned}$$

ist nicht in Chomsky-Normalform.

$G_3' = (T, N', P', S)$ mit

$$\begin{aligned} T &= \{a\}, N' = \{A, A_1\} \\ S &= A \\ P' &= \{A ::= A_1 A \mid a, A_1 ::= a\} \end{aligned}$$

ist in Chomsky-Normalform.
Außerdem ist $L(G_3) = L(G_3')$.

Normalformen - Greibach-Normalform

Definition 10

Eine kontextfreie Grammatik $G = (T, N, P, S)$ ist in Greibach-Normalform, wenn jede Regel in P von der Form

$$A ::= aB_1 \dots B_k$$

mit $a \in T, k \geq 0, B_i \in N$ für $i = 1, \dots k$, ist.
Zusätzlich ist die Regel $S ::= \epsilon$ erlaubt. Wenn sie vorkommt, dann darf S nicht auf der rechten Seite einer Produktion auftauchen.

Satz 11

Sei G eine kontextfreie Grammatik. Dann kann aus G eine kontextfreie Grammatik G' in Greibach-Normalform konstruiert werden, die dieselbe Sprache wie G erzeugt.

Normalformen - Greibach-Normalform

Definition 10

Eine kontextfreie Grammatik $G = (T, N, P, S)$ ist in Greibach-Normalform, wenn jede Regel in P von der Form

$$A ::= aB_1 \dots B_k$$

mit $a \in T, k \geq 0, B_i \in N$ für $i = 1, \dots k$, ist.
Zusätzlich ist die Regel $S ::= \epsilon$ erlaubt. Wenn sie vorkommt, dann darf S nicht auf der rechten Seite einer Produktion auftauchen.

$G_3 = (T, N, P, S)$ mit

$$T = \{a\}$$
$$N = \{A\}$$
$$S = A$$
$$P = \{A ::= aA, A ::= a\}$$

ist in Greibach-Normalform.

HTML

- HTML (Hypertext Markup Language) ist eine Sprache zur Darstellung von verzeigerten Texten
- zusammen mit XML (Extensible Markup Language) verwendet insbesondere im WorldWideWeb (WWW)
- typisch für HTML sind geklammerte Strukturen: $\langle x\rangle \ldots \langle /x\rangle$

Ausschnitt der HTML-Produktionen zur Erzeugung von Tabellen:

$$
\begin{aligned}
\mathit{Table} &::= \langle \mathit{table}\rangle \mathit{Rows} \langle /\mathit{table}\rangle \\
\mathit{Rows} &::= \mathit{Rows}\ \mathit{Row} \\
\mathit{Rows} &::= \mathit{Row} \\
\mathit{Row} &::= \langle \mathit{tr}\rangle \mathit{Cells} \langle /\mathit{tr}\rangle \\
\mathit{Cells} &::= \mathit{Cells}\ \mathit{Cell} \\
\mathit{Cells} &::= \mathit{Cell} \\
\mathit{Cell} &::= \langle \mathit{td}\rangle \mathit{Text} \langle /\mathit{td}\rangle \\
\mathit{Cell} &::= \langle \mathit{td}\rangle \mathit{Table} \langle /\mathit{td}\rangle
\end{aligned}
$$

Achtung: Im Beispiel sind HTML-Strukturklammern nicht aufgeführt.

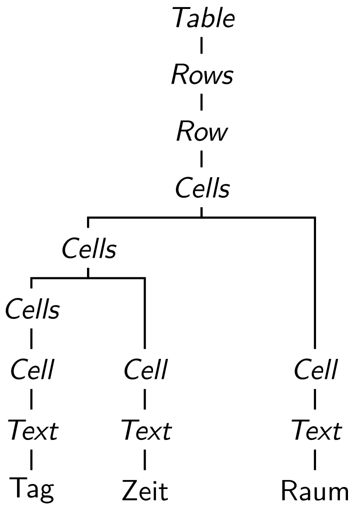
HTML

Ausschnitt der Produktionen von HTML zur Erzeugung von Tabellen:

$$Table ::= \langle table \rangle Rows \langle / table \rangle$$
$$Rows ::= Rows\ Row$$
$$Rows ::= Row$$
$$Row ::= \langle tr \rangle Cells \langle / tr \rangle$$
$$Cells ::= Cells\ Cell$$
$$Cells ::= Cell$$
$$Cell ::= \langle td \rangle Text \langle / td \rangle$$
$$Cell ::= \langle td \rangle Table \langle / td \rangle$$

Text steht als Platzhalter für beliebigen Text.

Ableitungsbaum für Zeile Tag Zeit Raum:



HTML Ableitung

Tag	Zeit	Raum
Mo	18:00 -19:30	AM
Fr	11:00 -13:00	AM