

Grundlagen der Programmierung

Compiler B: Grammatiken

Prof. Dr. Manfred Schmidt-Schauß

Sommersemester 2017

Programmiersprache: Syntax

Beschreibung der Sprache L_S aller Programme einer Programmiersprache:

- Finde kontextfreie Grammatik (CFG) G mit eindeutigen Syntaxbäumen, so dass $L(G)$ Obermenge von L_S
- Weitere Einschränkung von $L(G)$ (durch Bedingungen) ergibt L_S .

Kontextfreie Grammatik (CFG)

4-Tupel $G = (N, T, P, \sigma)$ mit

- 1 N : endliche Menge von *Hilfszeichen (Nonterminals)*
- 2 T endliche Menge von *Terminalzeichen (Terminals)*, wobei $N \cap T = \emptyset$.
- 3 $P \subseteq N \times (N \cup T)^*$ endliche Menge von *Regeln (Produktionen)*
- 4 $\sigma \in N$ ist *Startzeichen*

Regeln schreibt man auch $A \rightarrow w$ oder $A ::= w$.

Erzeugte Sprache $L(G)$ = erreichbare Worte über T .
= $\{w \in T^* \mid \sigma \xrightarrow{*}_G w\}$

Zwei Beispiel-CFGs für "Ausdruck"

Die **AZ-Grammatik**

$A ::= A + Z \mid A - Z \mid Z$
 $Z ::= 0 \mid \dots \mid 9$

Die **AA-Grammatik**:

$A ::= A + A$
 $A ::= A - A$
 $A ::= 0 \mid \dots \mid 9$

Beide Grammatiken erzeugen dieselbe Sprache.

Diese besteht aus sehr einfachen arithmetische Ausdrücken:

$z_1 op_1 z_2 \dots op_n z_{n+1}$

mit $z_i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$

und $op_i \in \{+, -\}$

z.B. $9 + 2 - 5 + 3$

Rechts- und Linksherleitungen

Linksherleitung: Ersetze stets das linkeste Nichtterminal

Rechtsherleitung: Ersetze stets das rechteste Nichtterminal

Die **AA-Grammatik:**

$A ::= A+A$
 $A ::= A - A$
 $A ::= 0 \mid \dots \mid 9$

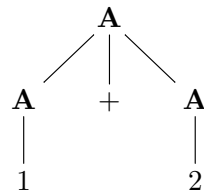
Beispiel „1+2“

Linksherleitung:

$A \rightarrow A+A \rightarrow 1+A \rightarrow 1+2$

Rechtsherleitung:

$A \rightarrow A+A \rightarrow A+2 \rightarrow 1+2$



Herleitungsbaum, Parse-Baum

Zur kontextfreien Grammatik $G = (N, T, P, \sigma)$

Kann man Herleitungen mittels

Herleitungsbäumen, bzw. Parse-Bäumen darstellen.

- pro Herleitungsbaum gibt es **genau eine Rechtsherleitung** und **genau eine Linksherleitung**
- Die Bedeutung eines Programms wird festgelegt **anhand des Herleitungsbaumes**

Eindeutigkeit einer CFG

Definition

Eine kontextfreie Grammatik G heißt **eindeutig**, wenn für alle $w \in L(G)$ **genau ein** Herleitungsbaum existiert. Andernfalls heißt die Grammatik **mehrdeutig**.

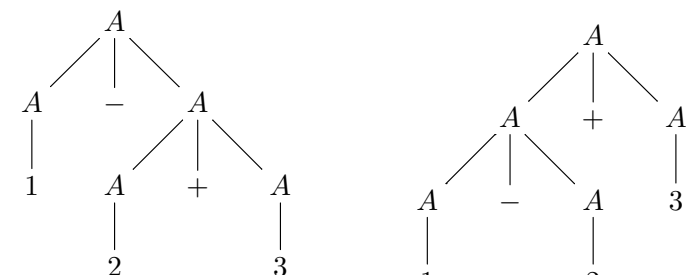
Wenn G eindeutig ist,
 Dann gibt es für **jedes** Wort in $L(G)$
genau einen Herleitungsbaum
 und **genau eine Rechts-Herleitung**
 und **genau eine Links-Herleitung**

Herleitungsbäume

Beispiel zu AA-Grammatik

$A ::= A+A$
 $A ::= A-A$
 $A ::= 0 \mid \dots \mid 9$

Zwei verschiedene Herleitungsbäume für „1 - 2 + 3“ sind:

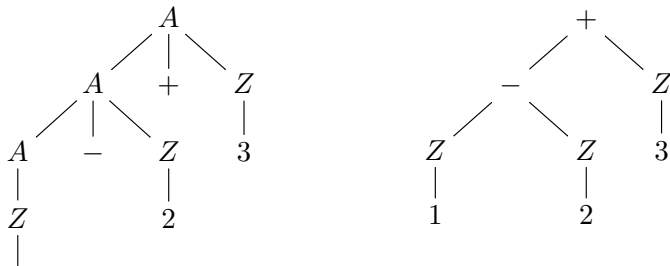


Eindeutige CFG zu **A**

Die AZ-Grammatik ist eindeutig.

A ::= **A**+**Z**
A ::= **A**-**Z**
A ::= **Z**
Z ::= 0 | ... | 9

Der einzige (d.h. eindeutige) AZ-Herleitungsbaum für $1 - 2 + 3$, und ein Syntaxbaum dazu ist:

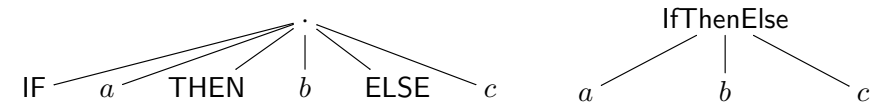


Syntaxbäume statt Herleitungsbäume

Beispiele:

Herleitungsbaum

Syntaxbaum



Semantikfestlegung

Die Struktur eines Ausdrucks/Programms wird erkannt durch den Parsebaum / Syntaxbaum; danach die Semantik anhand des Syntaxbaums

Mehrdeutige Grammatik entspricht i.a. mehrdeutiger (d.h. undefinierter) Semantik

Reparatur: Erzeuge äquivalente eindeutige Grammatik

Beachte Grammatiken, die äquivalent bzgl $L(\cdot)$ sind, könnten verschiedene operationale Semantik für den gleichen String festlegen