

# Grundbegriffe der Theoretischen Informatik

Sommersemester 2018 - Thomas Schwentick

Teil B: Kontextfreie Sprachen

7: Kontextfreie Grammatiken

Version von: 8. Mai 2018 (11:59)

## ▷ 7.1 Kontextfreie Grammatiken: Beispiele und Definition

7.2 Ableitungen und Ableitungsbäume

7.3 Mehrdeutigkeit

7.4 Konstruktion von Grammatiken

7.5 Die Chomsky-Hierarchie

7.6 Erweiterte kontextfreie Grammatiken

# Motivation

- Ziel: Methode zur Beschreibung der Syntax von Programmiersprachen
- Wir haben im ersten Teil der Vorlesung gesehen, dass sich Bezeichner in Programmtexten durch reguläre Sprachen beschreiben lassen

- Wie ist es mit anderen Konstrukten, die in Programmen vorkommen?

## Beispiel

- In Programmen kommen häufig arithmetische Ausdrücke wie z.B.  $((a + b) \times (a + a)) \times b$  vor
- Die Menge  $M$  der wohlgeformten arithmetischen Ausdrücke über einem Alphabet wie  $\{a, b, +, \times, „(“, „)“\}$  sollte von einer Methode zur Spezifikation der Syntax einer Programmiersprache beschreibbar sein
- Ist  $M$  regulär?

## „Proposition“

- Die Menge  $M$  der wohlgeformten arithmetischen Ausdrücke über  $\{a, b, +, \times, „(“, „)“\}$  ist nicht regulär

## Beweisidee

- Dazu genügt es, die „Zukunftssprachen“  $M/v_n$  der folgenden Strings  $v_n$ , für  $n \geq 1$  zu betrachten:

$$- v_n \stackrel{\text{def}}{=} ({}^n a$$

- Für jedes  $n$  enthält  $M/v_n$  den String  $[+a]^n$ , aber keinen String der Art  $[+a]^m$  für  $m \neq n$

☞ „[“ und „]“ sind Meta-Symbole

- ➡ Die Sprachen  $M/v_n$  sind alle verschieden
- ➡  $M$  hat unendlich viele Nerode-Klassen...
- ➡  $M$  ist nicht regulär

- Wie können wir Sprachen wie  $M$  beschreiben?

# Nicht-reguläre Sprachen

- Die folgenden Sprachen sind ebenfalls nicht regulär:

(a)  $L_{\text{pali}} = \{w \in \{a, b\}^* \mid w^R = w\}$

–  $w^R \stackrel{\text{def}}{=} \text{Umkehrung von } w$ , also  
 $(abdc)^R = cdba$

(b)  $L_{ab} = \{a^n b^n \mid n \geq 0\}$

(c)  $L_{\text{doppel}} = \{ww \mid w \in \{a, b\}^*\}$

(d)  $L_{\text{quad}} = \{a^{n^2} \mid n > 0\}$

(e)  $L_{\text{prim}} = \{a^p \mid p \text{ ist Primzahl}\}$

(f)  $L_{()} = \text{Menge aller wohlgeformten Klammerausdrücke}$

- Für einige dieser Sprachen bieten kontextfreie Grammatiken eine „einfache“ Beschreibungsmöglichkeit

## Beispiel

- Wir betrachten zunächst die Sprache  $L_{\text{pali}}$  der Palindrome über  $\{a, b\}$ :

- Palindrome über  $\{a, b\}$  lassen sich leicht induktiv definieren:

- $\epsilon, a, b$  sind Palindrome
- Ist  $w$  ein Palindrom, so auch  $awa$
- Ist  $w$  ein Palindrom, so auch  $bwb$

- Lässt sich das kompakter aufschreiben?

- Idee: Schreibe  $P \rightarrow w$  statt „ $w$  ist Palindrom“

- Dann lassen sich die genannten Regeln wie folgt zusammenfassen:

$$P \rightarrow \epsilon \quad (1)$$

$$P \rightarrow a \quad (2)$$

$$P \rightarrow b \quad (3)$$

$$P \rightarrow aPa \quad (4)$$

$$P \rightarrow bPb \quad (5)$$

# Palindrome erzeugen

- Die Regeln für Palindrome,

$$P \rightarrow \epsilon \quad (1)$$

$$P \rightarrow a \quad (2)$$

$$P \rightarrow b \quad (3)$$

$$P \rightarrow aPa \quad (4)$$

$$P \rightarrow bPb \quad (5)$$

lassen sich auf verschiedene Weisen interpretieren:

- Wir können Regeln der Art  $P \rightarrow w$  als Rezepte zum Erzeugen von Palindromen „bottom-up“ auffassen:

$b$  ist Palindrom (3)

$\Rightarrow aba$  ist Palindrom (4)

$\Rightarrow babab$  ist Palindrom (5)

$\Rightarrow bbababb$  ist Palindrom (5)

- Wir können Regeln der Art  $P \rightarrow w$  auch als Anleitung zum Erzeugen von Palindromen „top-down“ auffassen:

$$P \xRightarrow{(5)} bPb$$

$$\xRightarrow{(5)} bbPbb$$

$$\xRightarrow{(4)} bbaPabb$$

$$\xRightarrow{(3)} bbababb$$

## Ein weiteres Beispiel

- Wir beschreiben jetzt *arithmetische Ausdrücke mit Bezeichnern*:

- Ein **Bezeichner** sei ein String der Form  $(a + b)(a + b + 0 + 1)^*$

– Zum Beispiel: **bb1**

- Operationssymbole:  $+$ ,  $\times$
- Außerdem: Klammern

- Das Alphabet für unsere arithmetischen Ausdrücke ist also:

$$\Sigma = \{a, b, 0, 1, (, ), +, \times\}$$

- Ein Beispiel-Ausdruck:

$$(a + b0) \times bb1 + a0$$

- Induktive „Definition“ arithmetischer Ausdrücke:

- Bezeichner, oder
- Ausdruck  $+$  Ausdruck, oder
- Ausdruck  $\times$  Ausdruck, oder
- (Ausdruck)

- In „Regel-Schreibweise“:

$$A \rightarrow B$$

$$A \rightarrow A + A$$

$$A \rightarrow A \times A$$

$$A \rightarrow (A)$$

$$B \rightarrow a$$

$$B \rightarrow b$$

$$B \rightarrow Ba$$

$$B \rightarrow Bb$$

$$B \rightarrow B0$$

$$B \rightarrow B1$$

- Den Beispiel-Ausdruck erhalten wir dann so:

$$A \Rightarrow A + A$$

$$\Rightarrow A \times A + A$$

$$\Rightarrow (A) \times A + A$$

$$\Rightarrow (A + A) \times A + A$$

$$\Rightarrow (B + A) \times A + A$$

$$\Rightarrow (a + A) \times A + A$$

$$\Rightarrow (a + B) \times A + A$$

$$\Rightarrow (a + B0) \times A + A$$

$$\Rightarrow (a + b0) \times A + A$$

$$\Rightarrow (a + b0) \times B + A$$

$$\Rightarrow (a + b0) \times B1 + A$$

$$\Rightarrow (a + b0) \times Bb1 + A$$

$$\Rightarrow (a + b0) \times bb1 + A$$

$$\Rightarrow (a + b0) \times bb1 + B$$

$$\Rightarrow (a + b0) \times bb1 + B0$$

$$\Rightarrow (a + b0) \times bb1 + a0$$

# Kontextfreie Grammatiken: Definition

## Definition (Kontextfreie Grammatik)

- Eine **kontextfreie Grammatik**  $G = (V, \Sigma, S, P)$  besteht aus
  - einer endlichen Menge  $V$  von **Variablen**
  - einem Alphabet  $\Sigma$
  - einem **Startsymbol**  $S \in V$ ,
  - einer endlichen Menge  $P$  von **Regeln**:

$$P \subseteq V \times (V \cup \Sigma)^*$$

- Dabei muss gelten:  $\Sigma \cap V = \emptyset$

- Statt  $(A, BC) \in P$  schreiben wir  $A \rightarrow BC$

## Beispiel

- Die Regeln für arithmetische Ausdrücke sind also Regeln einer kontextfreien Grammatik

- Formal lässt sich diese Grammatik wie folgt aufschreiben:

$$(\{A, B\}, \{a, b, 0, 1, (, ), +, \times\}, A, \{(B, a), (B, b), (B, Ba), \dots, (A, (A))\})$$

- Übliche Bezeichnungen:

- Elemente von  $V \cup \Sigma$ : **Symbole**
- Elemente von  $\Sigma$ : **Terminalsymbole**
- Elemente von  $V$ : Variablen oder **Nicht-Terminalsymbole**
- Regeln aus  $P$ : **Produktionen**

- Mit  $|G|$  bezeichnen wir die Größe einer Grammatik:

$$\underline{|G|} \stackrel{\text{def}}{=} |V| + |\Sigma| + \sum_{(X, \alpha) \in P} (|\alpha| + 1)$$

- Die Grammatik für arithmetische Ausdrücke hat die Größe 40

# Kontextfreie Grammatiken: Kompakte Schreibweise

- Alle Regeln mit derselben linken Seite werden üblicherweise zusammengefasst:

– Statt

$$* X \rightarrow \alpha_1$$

$$* X \rightarrow \alpha_2$$

$$* \dots$$

$$* X \rightarrow \alpha_k$$

schreiben wir also:

$$X \rightarrow \alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_k$$

## Beispiel

$$A \rightarrow B \mid A + A \mid A \times A \mid (A)$$
$$B \rightarrow a \mid b \mid Ba \mid Bb \mid B0 \mid B1$$

- Meistens werden Grammatiken einfach durch die Angabe ihrer zusammengefassten Regeln beschrieben
- Dabei gelten folgende Konventionen:
  - Alle Symbole, die links in einer Regel vorkommen, sind Variablen
  - Das Startsymbol ist die Variable der linken Seite der ersten Regel



# Kontextfreie Grammatiken: Semantik

- Informelle Semantik kontextfreier Grammatiken: In einem Ableitungsschritt wird eine Variable  $X$  durch eine rechte Seite  $\gamma$  einer Regel  $X \rightarrow \gamma$  ersetzt, z.B.:

$$- bPb \Rightarrow_G baPab$$

## Definition (Satzform)

- Sei  $G = (V, \Sigma, S, P)$  eine kontextfreie Grammatik
- Eine Satzform ist ein String aus  $(V \cup \Sigma)^*$
- Eine Satzform  $v$  geht aus einer Satzform  $u$  in einem Ableitungsschritt hervor, wenn es
  - Satzformen  $\alpha, \beta, \gamma$ ,
  - eine Variable  $X$  und
  - eine Regel  $X \rightarrow \gamma$  in  $P$gibt, so dass
  - $u = \alpha X \beta$  und
  - $v = \alpha \gamma \beta$
- Schreibweise:  $\underline{u \Rightarrow_G v}$

- Informell: eine Ableitung ist eine Folge von Ableitungsschritten

## Definition (Ableitung, Kontextfreie Sprache)

- Sei  $G = (V, \Sigma, S, P)$  eine kontextfreie Grammatik
- Eine Folge  $u_0, u_1, \dots, u_n$  heißt Ab-  
leitung, falls für jedes  $i \in \{1, \dots, n\}$  gilt:  $u_{i-1} \Rightarrow_G u_i$
- Schreibweise:  $\underline{u_0 \Rightarrow_G^n u_n}$ 
  - oder  $\underline{u_0 \Rightarrow_G^* u_n}$ , wenn es auf die Zahl der Schritte nicht ankommt
- Wir sagen auch:  $u_n$  ist aus  $u_0$  (in  $n$  Schritten) ableitbar
- $\underline{L(G)} \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}$   
ist die von  $G$  erzeugte Sprache
- Eine Sprache  $L$  heißt kontextfrei, falls  $L = L(G)$  für eine kontextfreie Grammatik  $G$

# Ableitung: Beispiel

## Beispiel

$$\begin{aligned} A &\Rightarrow A + A \\ &\Rightarrow A \times A + A \\ &\Rightarrow (A) \times A + A \\ &\Rightarrow (A + A) \times A + A \\ &\Rightarrow (B + A) \times A + A \\ &\Rightarrow (a + A) \times A + A \\ &\Rightarrow (a + B) \times A + A \\ &\Rightarrow (a + B0) \times A + A \\ &\Rightarrow (a + b0) \times A + A \\ &\Rightarrow (a + b0) \times B + A \\ &\Rightarrow (a + b0) \times B1 + A \\ &\Rightarrow (a + b0) \times Bb1 + A \\ &\Rightarrow (a + b0) \times bb1 + A \\ &\Rightarrow (a + b0) \times bb1 + B \\ &\Rightarrow (a + b0) \times bb1 + B0 \\ &\Rightarrow (a + b0) \times bb1 + a0 \end{aligned}$$

# Inhalt

7.1 Kontextfreie Grammatiken: Beispiele und Definition

▷ **7.2 Ableitungen und Ableitungsbäume**

7.3 Mehrdeutigkeit

7.4 Konstruktion von Grammatiken

7.5 Die Chomsky-Hierarchie

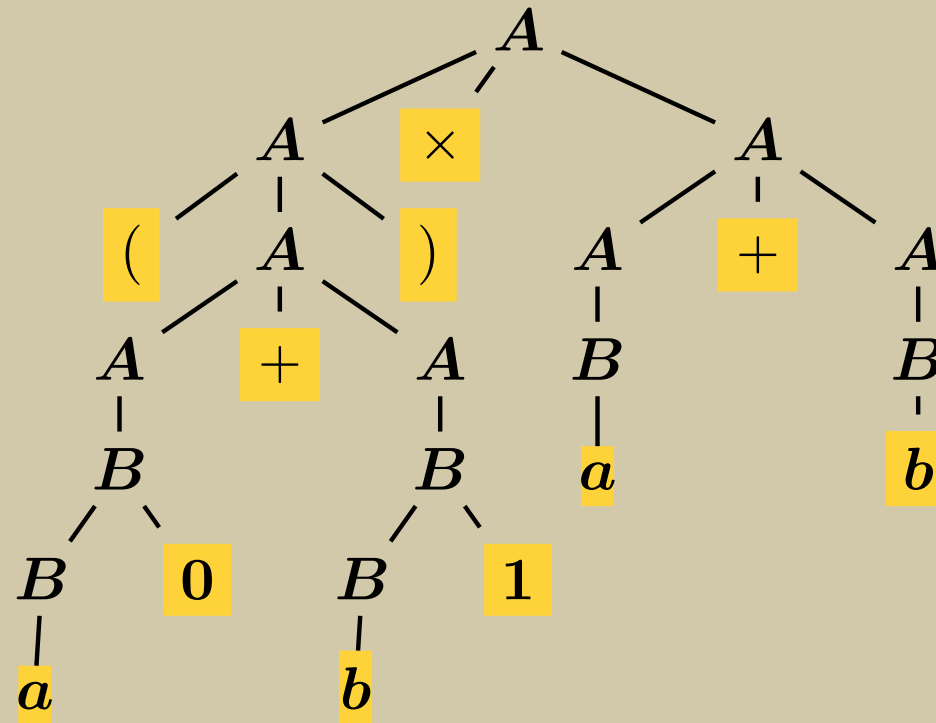
7.6 Erweiterte kontextfreie Grammatiken

# Ableitungsbäume

- Ableitungen lassen sich durch *Ableitungsbäume* visualisieren

## Beispiel

$$\begin{aligned} A &\rightarrow B \mid A + A \mid A \times A \mid (A) \\ B &\rightarrow a \mid b \mid Ba \mid Bb \mid B0 \mid B1 \end{aligned}$$



- Dieser Ableitungsbaum hat den **Blattstring**  $(a0 + b1) \times a + b$

- $S \Rightarrow_G^* w \iff$  es gibt einen Ableitungsbaum  
mit Wurzel  $S$  und Blattstring  $w$

# Ableitungsbäume und Ableitungen: Definitionen (1/2)

## Definition (Ableitungsbaum)

- Ein **Ableitungsbaum** zu einer kontextfreien Grammatik  $G = (V, \Sigma, S, P)$  ist ein geordneter Baum  $T$  mit Wurzel, der die folgenden Eigenschaften hat
  - Die **Blätter** sind mit Terminalsymbolen oder mit  $\epsilon$  markiert
  - Die **inneren Knoten** sind mit Variablen aus  $V$  markiert
  - Die **Wurzel** ist mit  $S$  markiert
  - Für jeden inneren Knoten  $v$  gibt es eine Regel  $X \rightarrow \alpha$  aus  $P$ , so dass
    - $v$  mit  $X$  markiert ist und
    - die Kinder von  $v$  von links nach rechts mit den Zeichen aus  $\alpha$  markiert sind
- Der **Blattstring** eines Ableitungsbaumes besteht aus den Symbolen der Blätter, die nicht mit  $\epsilon$  markiert sind, von links nach rechts gelesen
- Ist  $T$  ein Ableitungsbaum mit Blattstring  $w$ , so nennen wir  $T$  **Ableitungsbaum für  $w$**

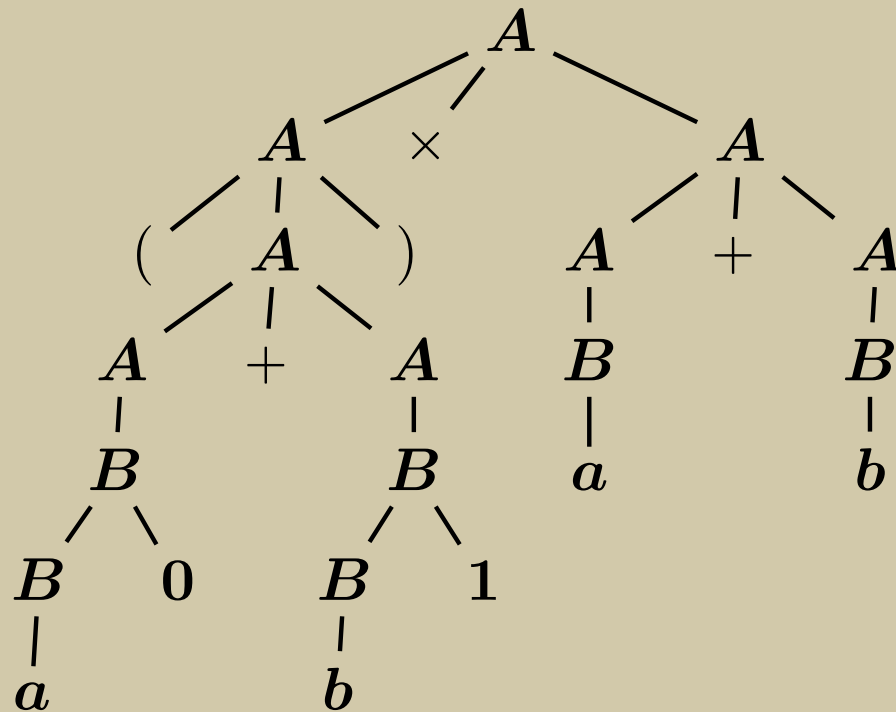
# Ableitungsbäume und Ableitungen: Definitionen (2/2)

## Definition (Linksableitung, Rechtsableitung)

- Zwei spezielle Arten von Ableitungen:
  - Linksableitung: Ableitung, in der in jedem Schritt die am weitesten links stehende Variable ersetzt wird
    - \* Schreibweise:  $S \Rightarrow_l^* w$  bzw.  $S \Rightarrow_{G,l}^* w$
  - Rechtsableitung: analog
    - \* Schreibweise:  $S \Rightarrow_r^* w$  bzw.  $S \Rightarrow_{G,r}^* w$
- Zu jedem Ableitungsbaum gibt es je eine Links- und eine Rechtsableitung

# Eine Linksableitung

## Beispiel

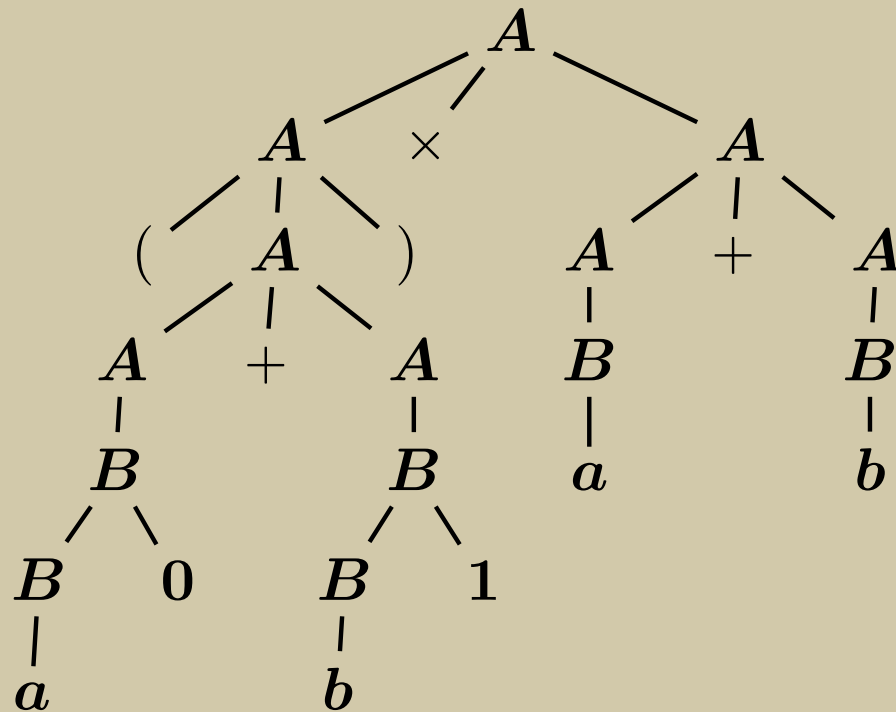


## Beispiel

$$\begin{aligned}
 A &\Rightarrow_l A \times A \\
 &\Rightarrow_l (A) \times A \\
 &\Rightarrow_l (A + A) \times A \\
 &\Rightarrow_l (B + A) \times A \\
 &\Rightarrow_l (B0 + A) \times A \\
 &\Rightarrow_l (a0 + A) \times A \\
 &\Rightarrow_l (a0 + B) \times A \\
 &\Rightarrow_l (a0 + B1) \times A \\
 &\Rightarrow_l (a0 + b1) \times A \\
 &\Rightarrow_l (a0 + b1) \times A + A \\
 &\Rightarrow_l (a0 + b1) \times B + A \\
 &\Rightarrow_l (a0 + b1) \times a + A \\
 &\Rightarrow_l (a0 + b1) \times a + B \\
 &\Rightarrow_l (a0 + b1) \times a + b
 \end{aligned}$$

# Eine Rechtsableitung

## Beispiel



## Beispiel

$$\begin{aligned}
 A &\Rightarrow_r A \times A \\
 &\Rightarrow_r A \times A + A \\
 &\Rightarrow_r A \times A + B \\
 &\Rightarrow_r A \times A + b \\
 &\Rightarrow_r A \times B + b \\
 &\Rightarrow_r A \times a + b \\
 &\Rightarrow_r (A) \times a + b \\
 &\Rightarrow_r (A + A) \times a + b \\
 &\Rightarrow_r (A + B) \times a + b \\
 &\Rightarrow_r (A + B1) \times a + b \\
 &\Rightarrow_r (A + b1) \times a + b \\
 &\Rightarrow_r (B + b1) \times a + b \\
 &\Rightarrow_r (B0 + b1) \times a + b \\
 &\Rightarrow_r (a0 + b1) \times a + b
 \end{aligned}$$



# Inhalt

7.1 Kontextfreie Grammatiken: Beispiele und Definition

7.2 Ableitungen und Ableitungsbäume

▷ **7.3 Mehrdeutigkeit**

7.4 Konstruktion von Grammatiken

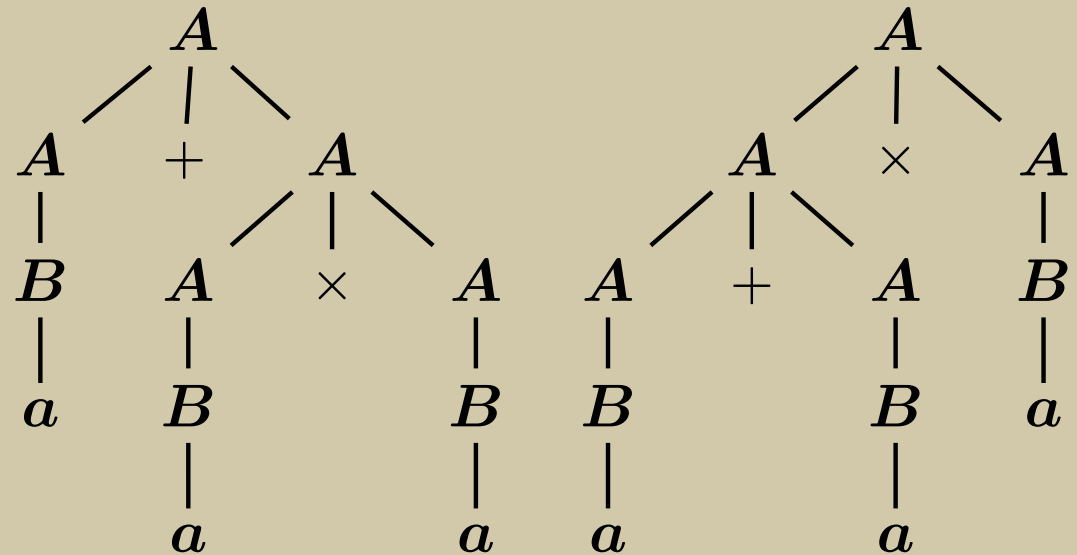
7.5 Die Chomsky-Hierarchie

7.6 Erweiterte kontextfreie Grammatiken

# Eindeutige vs. mehrdeutige Grammatiken (1/2)

- Wir haben gesehen: im Allgemeinen kann es zu einem Ableitungsbaum verschiedene Ableitungen geben
- Also kann es zu einem String mehrere Ableitungen haben
- Kann derselbe String verschiedene Ableitungsbäume haben?

## Beispiel



- Für Compiler ist es ungünstig, wenn der Ableitungsbaum nicht eindeutig ist:
  - Denn der Ableitungsbaum soll die Auswertungsreihenfolge eines Ausdrucks festlegen

## Beispiel

- Der linke Baum entspricht der Auswertung  
$$a + (a \times a)$$
- Der rechte Baum entspricht der Auswertung  
$$(a + a) \times a$$

## Eindeutige vs. mehrdeutige Grammatiken (2/2)

### Definition (Mehrdeutige, eindeutige Grammatiken)

- Eine kontextfreie Grammatik  $G$  heißt **mehrdeutig**, falls es einen String  $w$  gibt, der zwei verschiedene Ableitungsbäume bezüglich  $G$  hat
  - Andernfalls heißt  $G$  **eindeutig**

### Beispiel

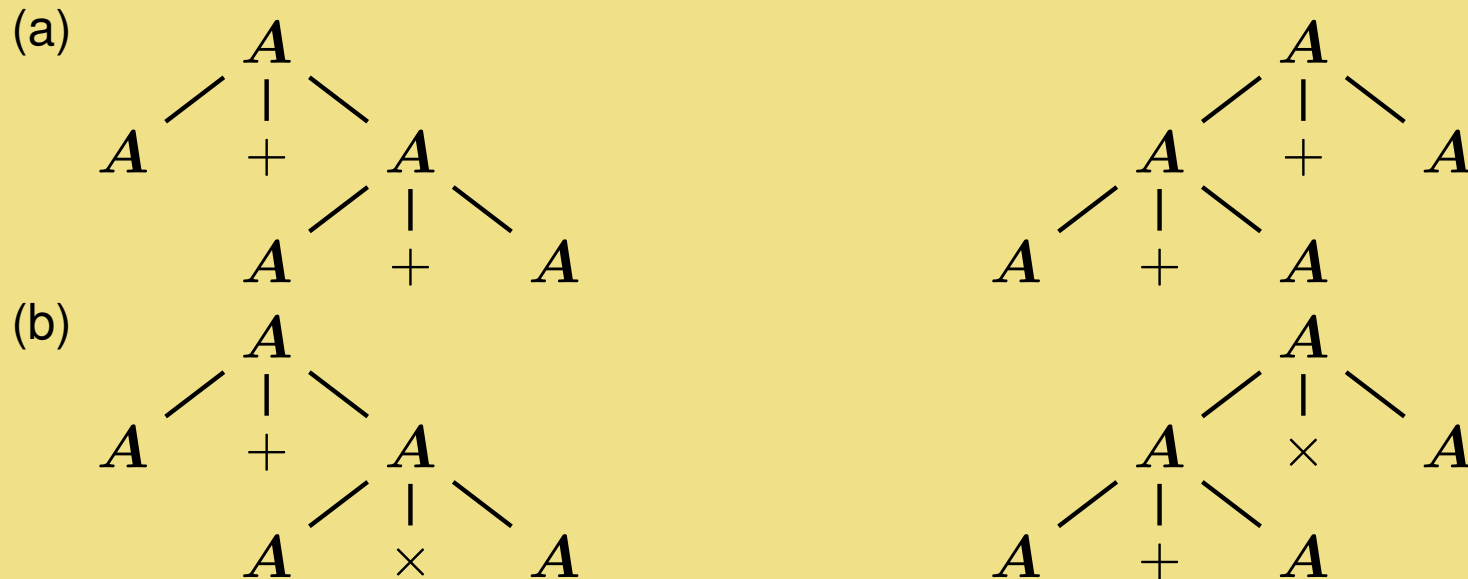
- Die Grammatik für arithmetische Ausdrücke ist mehrdeutig
- Wie sehen gleich: die Sprache der arithmetischen Ausdrücke hat auch eine eindeutige Grammatik


- Wie schwierig ist es zu testen, ob eine Grammatik eindeutig ist?
  - Mehr als schwierig:  
es gibt kein allgemeines Verfahren dafür
- Teil C der Vorlesung

# Arithmetische Ausdrücke

$$A \rightarrow B \mid A + A \mid A \times A \mid (A)$$
$$B \rightarrow a \mid b \mid Ba \mid Bb \mid B0 \mid B1$$

- Die obige Grammatik ist auf zweifache Weise mehrdeutig:



- Die Mehrdeutigkeit (a) lässt sich auf einfache Weise beheben:
  - Da die Operation  $+$  assoziativ ist, genügt es, immer die rechte Struktur zu erzeugen:  $A \rightarrow A + B \mid B$
- Die Mehrdeutigkeit (b) hängt mit der Bindungsstärke der Operatoren zusammen  „Punkt vor Strich“
  - Aber sie lässt sich ebenfalls beheben...

# Eine eindeutige Grammatik für arithmetische Ausdrücke

- **Ziel:** eindeutige Grammatik für arithmetische Ausdrücke
- **Idee:** Operatoren mit geringer Bindung werden später ausgewertet und sollten im Baum deshalb weit oben sein
  - ➔ Die Regeln für  $+$  müssen in der Grammatik in „einer höheren Ebene“ vorkommen als die Regeln für  $\times$

- Modifizierte Grammatik:

$$A \rightarrow A + T \mid T$$

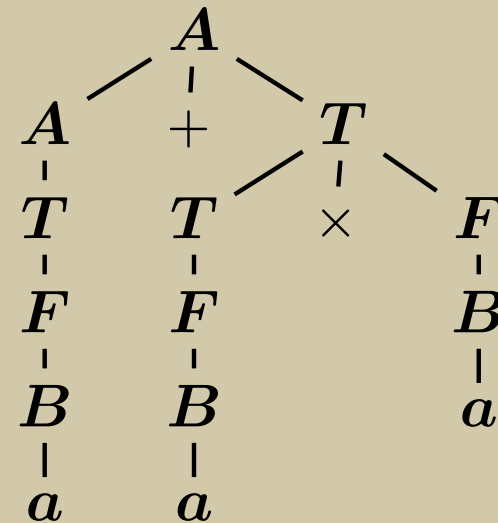
$$T \rightarrow T \times F \mid F$$

$$F \rightarrow (A) \mid B$$

$$B \rightarrow a \mid b \mid Ba \mid Bb \mid B0 \mid B1$$

- Bezüglich dieser Grammatik hat  $a + a \times a$  eine eindeutige Ableitung

## Beispiel



## Fakt

- Die modifizierte Grammatik für arithmetische Ausdrücke ist eindeutig

## Beweisidee

- Zeige für alle Variablen der Grammatik, dass jede aus ihr ableitbare Satzform einen eindeutigen Ableitungsbaum hat
- Dies lässt sich durch Induktion nach der Höhe des minimalen Ableitungsbaums beweisen

# Inhärent mehrdeutige kontextfreie Sprachen

## Definition (Eindeutige Sprache)

- Eine kontextfreie Sprache  $L$  heißt **eindeutig**, falls sie eine eindeutige Grammatik hat
  - Andernfalls heißt  $L$  **inhärent mehrdeutig**

## Beispiel

$$S \rightarrow AX_{bc} \mid X_{ab}C \mid \epsilon$$

$$C \rightarrow Cc \mid \epsilon$$

$$A \rightarrow Aa \mid \epsilon$$

$$X_{ab} \rightarrow aX_{ab}b \mid \epsilon$$

$$X_{bc} \rightarrow bX_{bc}c \mid \epsilon$$

ist eine mehrdeutige kontextfreie Grammatik für die Sprache  $L_{abc} \stackrel{\text{def}}{=} \{a^i b^j c^k \mid i = j \text{ oder } j = k\}$

- $L_{abc}$  ist inhärent mehrdeutig, hat also keine eindeutige Grammatik

- Intuitiver Grund: Strings der Form  $a^n b^n c^n$  erfüllen beide Bedingungen „ $i = j$ “ und „ $j = k$ “ und haben deshalb zwei Ableitungsbäume
  - Aber: der Beweis dafür ist ziemlich kompliziert

- Es gibt auch kein allgemeines Verfahren, das entscheidet, ob die Sprache einer gegebenen kontextfreien Grammatik eindeutig ist

# Inhalt

7.1 Kontextfreie Grammatiken: Beispiele und Definition

7.2 Ableitungen und Ableitungsbäume

7.3 Mehrdeutigkeit

▷ **7.4 Konstruktion von Grammatiken**

7.5 Die Chomsky-Hierarchie

7.6 Erweiterte kontextfreie Grammatiken

# Konstruktion einer kontextfreien Grammatik

## Beispiel

- Wir konstruieren eine kontextfreie Grammatik für  $L_{ab} = \{a^n b^n \mid n \geq 0\}$
- Immer, wenn sie vorne ein  $a$  erzeugt, soll sie hinten ein  $b$  erzeugen:  $S \rightarrow aSb$
- Irgendwann soll sie damit aufhören:  $S \rightarrow \epsilon$
- Insgesamt also:  $S \rightarrow aSb \mid \epsilon$
- Beispielableitung:
$$\begin{aligned} S &\Rightarrow aSb \\ &\Rightarrow aaSbb \\ &\Rightarrow aaaSbbb \\ &\Rightarrow aaabbbb \end{aligned}$$



# Eine etwas kompliziertere kontextfreie Grammatik (1/2)

- Wir betrachten jetzt ein komplizierteres Beispiel einer kontextfreien Grammatik
- Es illustriert, dass eine rekursive Herangehensweise bei der Konstruktion kontextfreier Grammatiken helfen kann

## Beispiel

- Sei  $L_{a=b} \stackrel{\text{def}}{=} \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ 
  - Zur Erinnerung:  $\#_a(w)$  ist die Anzahl der Positionen in  $w$ , an denen  $a$  steht
- Wie lassen sich die Strings dieser Sprache erzeugen?

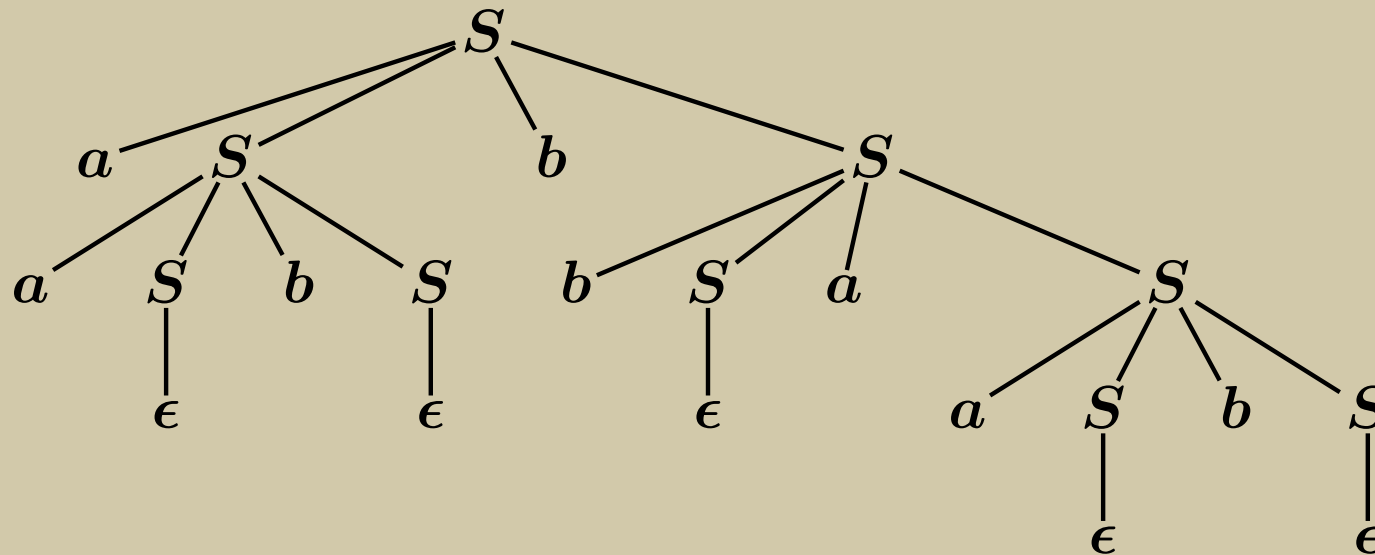
## Beispiel

- **Idee:** Strings aus  $L_{a=b}$  lassen sich schreiben
  - in der Form  $aubv$  oder
  - in der Form  $buav$ ,wobei sowohl in  $u$  als auch in  $v$  gleich viele  $a$ 's wie  $b$ 's vorkommen
- Eine Grammatik für  $L_{a=b}$  könnte also die Regeln  $S \rightarrow aSbS$  und  $S \rightarrow bSaS$  verwenden
- Der Leerstring ist natürlich auch noch in  $L_{a=b}$
- Insgesamt ergibt sich also die folgende Grammatik  $G_{a=b}$ :
$$S \rightarrow aSbS \mid bSaS \mid \epsilon$$

## Eine etwas kompliziertere kontextfreie Grammatik (2/2)

### Beispiel

- Wir betrachten einen Ableitungsbaum für den String  $aabbbaab$
- Zur Erinnerung:  $G_{a=b}$  ist  $S \rightarrow aSbS \mid bSaS \mid \epsilon$



- Dass  $G_{a=b}$  wirklich genau die Strings der Sprache  $L_{a=b}$  erzeugt, zeigen wir in Kapitel 9

# Inhalt

7.1 Kontextfreie Grammatiken: Beispiele und Definition

7.2 Ableitungen und Ableitungsbäume

7.3 Mehrdeutigkeit

7.4 Konstruktion von Grammatiken

▷ **7.5 Die Chomsky-Hierarchie**

7.6 Erweiterte kontextfreie Grammatiken

# Chomsky-Grammatiken: Definition

- Kontextfreie Grammatiken sind der (mit Abstand bedeutendste) Spezialfall eines allgemeineren Konzeptes
- **Chomsky-Grammatiken** wurden in den 50er Jahren von dem Linguisten Noam Chomsky im Zusammenhang der Analyse natürlicher Sprachen eingeführt
- Sie erlauben auf der linken Seite einer Regel nicht nur Variablen sondern Satzformen, z.B.:
  - $aBC \rightarrow De$

## Definition (Chomsky-Grammatik)

- Eine **Chomsky-Grammatik** ist ein 4-Tupel  $(V, \Sigma, P, S)$  mit  $V, \Sigma, S$  wie zuvor und  $P \subseteq (V \cup \Sigma)^* V (V \cup \Sigma)^* \times (V \cup \Sigma)^*$
- Auf der linken Seite jeder Regel ist also immer ein String über  $V \cup \Sigma$  mit mindestens einer Variablen
- Ableitungsschritt:  $\alpha\beta\gamma \Rightarrow \alpha\delta\gamma$ ,  
falls  $\beta \rightarrow \delta$  Regel von  $P$  ist

# Noam Chomsky

## Kurz-Bio: Noam Chomsky

- Geboren: 7.12.1928 in Philadelphia
- Studium der Philosophie und Linguistik an der University of Pennsylvania und in Harvard
- Promotion 1955: University of Pennsylvania
- Er lehrt seit 1955 am MIT
- Grundlegende Studien zur Beschreibung natürlicher Sprachen mit formalen Grammatiken

(Quellen: Wikipedia)

# Chomsky-Grammatiken: Beispiel

## Beispiel-Grammatik

$$\begin{aligned} S &\rightarrow SABC \mid \epsilon \\ AB &\rightarrow BA \\ BA &\rightarrow AB \\ AC &\rightarrow CA \\ CA &\rightarrow AC \\ BC &\rightarrow CB \\ CB &\rightarrow BC \\ A &\rightarrow a \\ B &\rightarrow b \\ C &\rightarrow c \end{aligned}$$

## Beispiel-Ableitung

$$\begin{aligned} S &\Rightarrow SABC \\ &\Rightarrow SABCABC \\ &\Rightarrow ABCABC \\ &\Rightarrow BACABC \\ &\Rightarrow BAACBC \\ &\Rightarrow bAACBC \\ &\vdots \\ &\Rightarrow baacbc \end{aligned}$$

- Diese Grammatik erzeugt die Sprache

$$- L_{abc} \stackrel{\text{def}}{=} \{w \mid \#_a(w) = \#_b(w) = \#_c(w)\}$$


aller Strings über  $\{a, b, c\}$ , bei denen die Anzahl der a, b und c gleich ist

# Die Chomsky-Hierarchie

- Die **Chomsky-Hierarchie** umfasst 4 Klassen von Sprachen

Typ	Name	Regel-Einschränkung $\alpha \rightarrow \beta$
0	Typ 0	keine
1	kontextsensitiv	$ \alpha  \leq  \beta $
2	kontextfrei	$X \rightarrow \beta$
3	regulär	$X \rightarrow \sigma$ oder $X \rightarrow \sigma Y$

- Bei den Typen 1 und 3 ist jeweils auch die Regel  $S \rightarrow \epsilon$  erlaubt, falls  $S$  auf keiner rechten Seite vorkommt

 Bei den Typen 0 und 2 sind  $\epsilon$ -Regeln sowieso erlaubt

- Grammatiken, die nur Regeln der Formen  $X \rightarrow \sigma$  und  $X \rightarrow \sigma Y$  haben, heißen rechtslinear
  - Auch die (analog definierten) linkslinearen Grammatiken erzeugen genau die regulären Sprachen

# Inhalt

7.1 Kontextfreie Grammatiken: Beispiele und Definition

7.2 Ableitungen und Ableitungsbäume

7.3 Mehrdeutigkeit

7.4 Konstruktion von Grammatiken

7.5 Die Chomsky-Hierarchie

▷ **7.6 Erweiterte kontextfreie Grammatiken**



# Erweiterte kontextfreie Grammatiken

- Rechte Seiten der kompakten Notation für kontextfreie Grammatiken erinnern an reguläre Ausdrücke:  $\alpha_1 \mid \cdots \mid \alpha_k$  entspricht  $\alpha_1 + \cdots + \alpha_k$
- Warum nicht reguläre Ausdrücke erlauben?

## Definition (Erweiterte kontextfreie Grammatik)

- Eine **erweiterte kontextfreie Grammatik**  $G = (V, \Sigma, S, P)$  besteht aus
  - einer Menge  $V$  von **Variablen**
  - einem Alphabet  $\Sigma$
  - einem **Startsymbol**  $S \in V$ ,
  - einer Menge  $P$ , die für jede Variable  $X \in V$  genau eine **Regel**  $X \rightarrow \alpha_X$  enthält, wobei  $\alpha_X$  ein regulärer Ausdruck über  $V \cup \Sigma$  ist
- In einem Ableitungsschritt kann dann immer eine Variable  $X$  durch einen String  $\beta \in L(\alpha_X)$  ersetzt werden
- der Knotengrad in Ableitungsbäumen kann beliebig groß werden

## Beispiel

- Zur Erinnerung: Grammatik für arithmetische Ausdrücke:
$$\begin{aligned} A &\rightarrow A + T \mid T \\ T &\rightarrow T \times F \mid F \\ F &\rightarrow (A) \mid B \\ B &\rightarrow a \mid b \mid Ba \mid Bb \mid B0 \mid B1 \end{aligned}$$
- Erweiterte kontextfreie Grammatik für die selbe Sprache:
$$\begin{aligned} A &\rightarrow T[+T]^* \\ T &\rightarrow F[\times F]^* \\ F &\rightarrow (A) \mid B \\ B &\rightarrow [a \mid b][a \mid b \mid 0 \mid 1]^* \end{aligned}$$
- Dabei sind  $[$  und  $]$  Meta-Symbole zum Klammern und  $|$  ist ein Meta-Symbol für die Vereinigung  
☞ anstelle des üblichen „+“

# Erweiterte kontextfreie Grammatiken: Ableitungsbaum

Beispiel: Erweiterte Grammatik

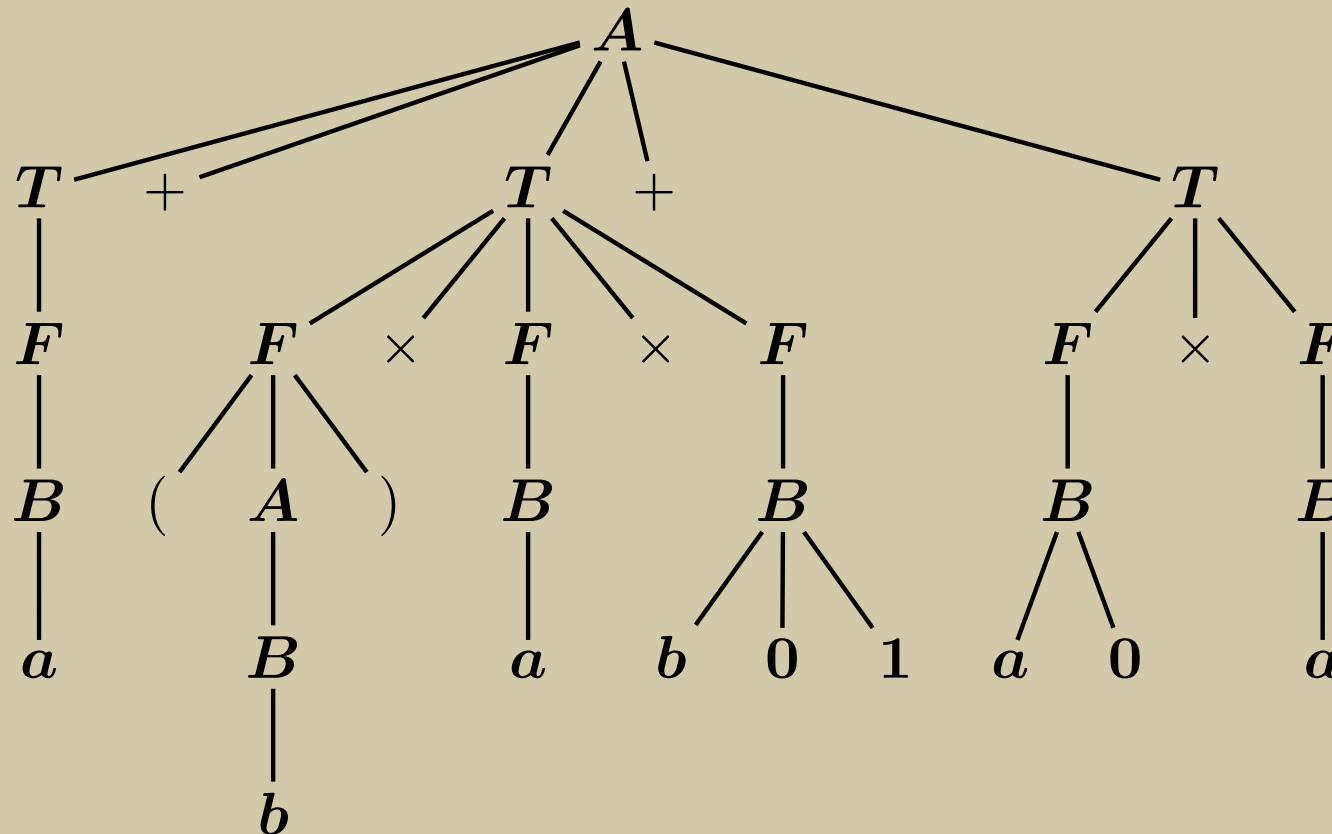
$$A \rightarrow T[+T]^*$$

$$T \rightarrow F[\times F]^*$$

$$F \rightarrow (A) \mid B$$

$$B \rightarrow [a \mid b][a \mid b \mid 0 \mid 1]^*$$

Beispiel: Ableitungsbaum



# Ausdrucksstärke erweiterter kontextfreier Grammatiken

## Satz 7.1

- Sei  $L$  eine Sprache
- Dann sind äquivalent:
  - (a)  $L = L(G)$  für eine kontextfreie Grammatik  $G$
  - (b)  $L = L(G')$  für eine erweiterte kontextfreie Grammatik  $G'$
- Die Beweisidee findet sich im Anhang


# BNF: Backus-Naur Form

- Die **Backus-Naur-Form** ist eine alternative Notation für kontextfreie Grammatiken:

<Programm>	::=	"PROGRAM" <Bezeichner> "BEGIN" <Satzfolge> "END" .
<Bezeichner>	::=	<Buchstabe>   <Buchstabe> <Restbezeichner>
<Restbezeichner>	::=	<Buchstabe oder Ziffer>   <Buchstabe oder Ziffer> <Restbezeichner>
<Buchstabe oder Ziffer>	::=	<Buchstabe>   <Ziffer>
<Buchstabe>	::=	A   B   C   D   ...   Z   a   b   ...   z
<Ziffer>	::=	0   1   2   3   4   5   6   7   8   9
<Satzfolge>	::=	...
...		

 aus: Wikipedia

- Also:
  - statt  $\rightarrow$  wird  $::=$  verwendet
  - Variablen in Klammern <...>
- Außerdem können optionale Elemente in Klammern [...] gesetzt werden

 entsprechend (...) ? in RAs

# EBNF: Erweiterte Backus-Naur Form

- BNF entspricht kontextfreien Grammatiken
- EBNF entspricht erweiterten kontextfreien Grammatiken
- Zusätzliche Möglichkeiten:
  - Konkatenation: durch Komma angedeutet
  - Wiederholung: durch geschweifte Klammern { und }
  - Terminalzeichen werden in Anführungszeichen gesetzt, deshalb für Variablen keine spitzen Klammern mehr nötig
  - ; als Zeilenendesymbol

- Beispiel:

Programm	=	"PROGRAM" Bezeichner "BEGIN" { Zuweisung ["," ] } "END" "." ;
Bezeichner	=	Buchstabe { ( Buchstabe   Ziffer ) } ;
Zahl	=	[ "-" ] Ziffer { Ziffer } ;
String	=	"" { AlleZeichen - "" } "" ;
Zuweisung	=	Bezeichner ":( Zahl   Bezeichner   String ) ;
Buchstabe	=	"A"   "B"   "C"   "D"   "E"   "F"   "G"   "H"   "I"   "J"   "K"   "L"   "M"   "N"   "O"   "P"   "Q"   "R"   "S"   "T"   "U"   "V"   "W"   "X"   "Y"   "Z" ;
Ziffer	=	"0"   "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9" ;
AlleZeichen	=	? alle sichtbaren Zeichen ? ;

 aus: Wikipedia

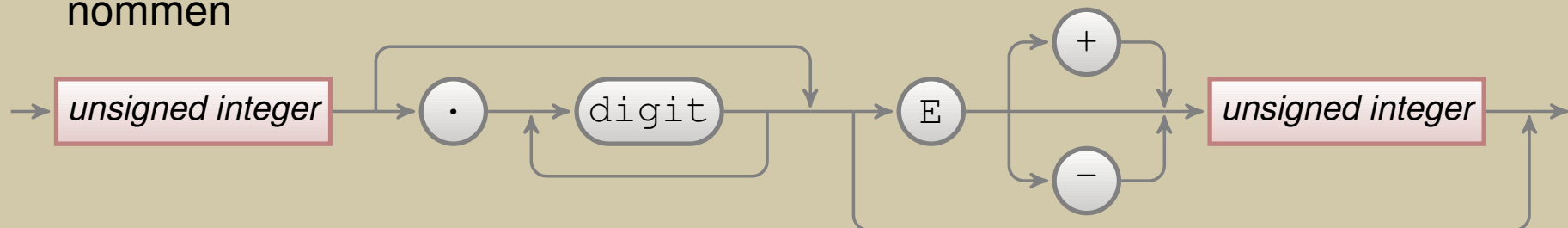
# Syntaxdiagramme

- **Syntaxdiagramme** sind eine weitere, sehr intuitive Notation für erweiterte kontextfreie Grammatiken
- Grammatiken lassen sich wie folgt übersetzen:

Regel	Diagramm
$A \rightarrow BC$	
$A \rightarrow B \mid C$	
$A \rightarrow B^*$	

## Beispiel

- Das folgende Beispiel eines Syntaxdiagramms ist dem TikZ/PGF-Handbuch entnommen



# Zusammenfassung

- Mit Hilfe kontextfreier Grammatiken lassen sich einige nicht reguläre Sprachen wie die Menge aller Palindrome und die Menge (gewisser) arithmetischer Ausdrücke beschreiben
- Ableitungen kontextfreier Grammatiken lassen sich anschaulich durch Ableitungsbäume darstellen
- Für viele Zwecke ist es wünschenswert, dass jeder String der Sprache einen eindeutigen Ableitungsbaum hat, das ist jedoch nicht immer möglich
- Erweiterte kontextfreie Grammatiken sind genauso ausdrucksstark wie kontextfreie Grammatiken
- Syntaxdiagramme und EBNF bieten eine alternative Syntax
- Kontextfreie Grammatiken sind eine eingeschränkte Form von Chomsky-Grammatiken

# Beweisidee für Satz 7.1

## Beweisidee

- „ $(a) \Rightarrow (b)$ “: ✓
- „ $(b) \Rightarrow (a)$ “:
  - Sei  $G'$  eine erweiterte kontextfreie Grammatik für  $L$  mit Startsymbol  $S$
  - Sei  $X \rightarrow \alpha_X$  eine Regel von  $G'$
  - ➡  $\alpha_X$  ist ein regulärer Ausdruck
  - ➡  $L(\alpha_X)$  ist eine reguläre Sprache über  $V \cup \Sigma$
  - ➡  $L(\alpha_X)$  ist kontextfrei ☞ Kapitel 10
  - Sei, für jedes  $X$ ,  $G_X$  eine Grammatik für  $L(\alpha_X)$  mit Startsymbol  $X$
  - Sei  $G$  die Grammatik, die durch Vereinigung der Grammatiken  $G_X$  entsteht, mit Startsymbol  $S$
  - **Behauptung:**  $L(G) = L(G')$
  - Bemerkungen:
    - ✎ Die Grammatiken  $G_X$  haben  $V \cup \Sigma$  als Terminalalphabet und Regeln der Form  $X \rightarrow \sigma$  oder  $X \rightarrow \sigma Y$  ☞ rechtslinear
- \* Ihre Variablenmengen müssen disjunkt gewählt werden